

DEDICACE

A LA FAMILLE ASSAME

Rédigé par **ASSAME NNAMA Orlane Sandy**, élève ingénieur en Deuxième année, option Génie Logiciel,

IAI-CAMEROUN, Centre d'Excellence Technologique PAUL BIYA

Année académique 2023-2024

REMERCIEMENTS

Nous tenons à remercier toutes les personnes qui ont participé de près ou de loin au bon déroulement de ce stage, à la réalisation de ce rapport jusqu'à la présentation de notre travail. Nos sincères remerciements s'adressent :

- Au Représentant Résident de l'Institut Africain d'Informatique représentation du Cameroun, **M. Armand Claude ABANDA** ;
- Au fondateur de **MERCY FOR CAMEROON**, **M. Thierry FOZING** ;
- A notre encadrant académique, Madame **Estelle Grayce BELINGA epse BONONO** pour son encadrement, ses conseils, son suivi notamment sur l'analyse mais aussi sur la rédaction de notre rapport de stage et sa grande disponibilité pour le bon déroulement de ce dernier ;
- A notre encadrant professionnel, Monsieur **Maximilien APOBA** pour ses suggestions, ses corrections rudes et conseils avisés qui nous ont aidé à mener à bien ce travail ;
- A tout le corps administratif et professoral de l'IAI-Cameroun ;
- A mon père **M. Jean-Claude ASSAME MVONDO** pour son amour inconditionnel, ses conseils et a prise en charges des dépenses ;
- A ma mère **Mme Prisca Pauline MENGUE VOUNDI** pour son amour inconditionnel, son soutien et la prise en charge dans les dépenses.
- A mes frères et sœurs pour leurs conseils et leurs soutien.
- A mon ami **Brice Alexis MOTAZE** pour ses encouragement son aide dans la réalisation du projet.
- A **M. Sergio Junior CHEBEU** pour son grand apport dans le projet.
- A mes amis pour leurs encouragements.
- A toutes les personnes qui m'ont prodigué des conseils, leurs attentons et leurs encouragements.

SOMMAIRE

DEDICACE.....	i
REMERCIEMENTS	ii
SOMMAIRE	iii
LISTE DES TABLEAUX	iv
LISTE DES FIGURES	v
GLOSSAIRE.....	vii
RESUME.....	viii
ABSTRACT	ix
INTRODUCTION GENERALE.....	1
PARTIE I : PHASE D'INSERTION	2
PARTIE II : PHASE TECHNIQUE	13
DOSSIER 1 : L'EXISTANT	14
DOSSIER 2 : CAHIER DES CHARGES	22
DOSSIER 3 : LE DOSSIER D'ANALYSE	38
DOSSIER 4 : LE DOSSIER DE CONCEPTION	81
DOSSIER 5 : LE DOSSIER DE DEPLOIEMENT.....	97
DOSSIER 6 : TEST DE FONCTIONNALITES	111
DOSSIER 7 : GUIDE D'INSTALLATION ET GUIDE D'UTILISATION.....	121
CONCLUSION GENERALE	131
TABLE DE MATIERES	xv
ANNEXE	xx

LISTE DES TABLEAUX

Tableau 1: Ressources matérielles.....	9
Tableau 2: Ressources Logicielles	10
Tableau 3: Critique de l'existant	19
Tableau 4 : Tableau des besoins fonctionnels de notre application.....	27
Tableau 5 : Liste des intervenants du projet	31
Tableau 6: Ressources logicielles.....	33
Tableau 7: Ressources matérielles.....	34
Tableau 8: Ressources humaines	34
Tableau 9: Tableau récapitulatif	35
Tableau 10: Etude comparative entre MERISE et UML.....	41
Tableau 11: Etude comparative entre les processus unifiés	43
Tableau 12: Eléments de modélisation du diagramme de cas d'utilisation	49
Tableau 13: Description du cas "S'AUTHTENTIFIER"	56
Tableau 14: Description textuelle du cas 'EFFECTUER DEMANDE CNI'	57
Tableau 15: Description textuelle du cas "EFFECTUER PAIEMENT"	59
Tableau 16: Description textuelle du cas "ETABLIR PLANNING"	61
Tableau 17: Elément de modélisation d'un diagramme de communication	62
Tableau 18: Tableau des éléments du diagramme de sequence.....	65
Tableau 19: Eléments de modélisation du diagramme d'activité	73
Tableau 20: Eléments de modélisation du diagramme de classe.....	84
Tableau 21: Elément d'un diagramme d'objet	89
Tableau 22: Elément de modélisation d'un diagramme d'état transition	91
Tableau 23: Elément de modélisation d'un diagramme de paquetage.....	93
Tableau 24: Présentation des logiciels utilisés	100
Tableau 25: Présentation des langages utilisés.....	101
Tableau 26 : Elément de modélisation d'un diagramme de déploiement	106
Tableau 27: Elément de modélisation du diagramme de composant	108

LISTE DES FIGURES

Figure 1: Plan de localisation de la startup Mercy Innovation Lab	7
Figure 2: Organigramme de la Startup Mercy Innovation Lab	7
Figure 3: Logo de notre application	30
Figure 4: Police Roboto	31
Figure 5: Diagramme de Gantt de notre projet	32
Figure 6: processus 2TUP (source : cours UML 2)	48
Figure 7: Exemple de diagramme de cas d'utilisation	51
Figure 8: Diagramme de cas d'utilisation global du système	52
Figure 9: Diagramme spécifique du cas ' GERER SON PROFIL '	53
Figure 10: Diagramme spécifique du cas ' GÉRER COMPTES '	54
Figure 11: Diagramme spécifique du cas ' GERER PLANNING '	54
Figure 12: Exemple de diagramme de communication	63
Figure 15: diagramme de communication du cas ' EFFECTUER PAIEMENT '	63
Figure 13: diagramme de communication du cas ' S'AUTHENTIFIER '	63
Figure 14: diagramme de communication du cas ' EFFECTUER DEMANDE CNI '	64
Figure 16: diagramme de communication du cas ' ETABLIR PLANNING '	64
Figure 17: Exemple de diagramme de sequence	68
Figure 18: Diagramme de cas d'utilisation ' S'AUTHENTIFIER '	69
Figure 19: Diagramme de sequence du cas ' EFFECTUER DEMANDE CNI '	70
Figure 20 : Diagramme de sequence du cas ' EFFECTUER PAIEMENT '	71
Figure 21: Diagramme de sequence du cas ' ETABLIR PLANNING '	72
Figure 22: exemple de diagramme de sequence	75
Figure 23: Diagramme d'activité du cas ' S'AUTHENTIFIER '	76
Figure 24: Diagramme d'activité du cas ' EFFECTUER DEMANDE CNI '	77
Figure 25: Diagramme d'activité du cas ' EFFECTUER PAIEMENT '	78
Figure 26: Diagramme d'activité du cas ' ETABLIR PLANNING '	79
Figure 27: Exemple de diagramme de classe	85
Figure 28: Diagramme de classe du système	86

Figure 29: Exemple de diagramme d'objet.....	89
Figure 30: Diagramme d'objet du système.....	90
Figure 31: Exemple de diagramme d'état transition.....	92
Figure 32: Diagramme d'état transition de "COMPTE".....	92
Figure 33: Diagramme d'état transition de "PLANNING"	93
Figure 34 : Exemple de diagramme de paquetage	94
Figure 35: Diagramme de paquetage du système.....	95
Figure 36: Notre architecture logicielle	104
Figure 37: Exemple de diagramme de déploiement.....	107
Figure 38 : Diagramme de déploiement du système	107
Figure 39: Exemple de diagramme de déploiement.....	108
Figure 40: Diagramme de composant du système	109
Figure 41 : Fonction d'authentification.....	114
Figure 42: Fonction télécharger fiche PDF	115
Figure 43: Test de la fonction d'authentification	116
Figure 44: Test de la fonction télécharger fiche PDF.....	118
Figure 45: interface d'identification par téléphone.....	125
Figure 46: Interface de choix de demande	126
Figure 47: Interface de remplissage de formulaire.....	127
Figure 48: Interface de paiement.....	128
Figure 49: interface de téléchargement de fiche PDF	129

GLOSSAIRE

- **CNI** : désigne l'appellation en abrégé de la carte nationale d'identité qui est une pièce officielle permettant de justifier sa nationalité.
- **API (Application Programming Interface)** : est une interface de programmation qui définit les interactions possibles entre plusieurs logiciels. Elle permet à des applications de communiquer et d'échanger des données.
- **Application mobile** : est un logiciel conçu pour fonctionner sur des appareils mobiles comme les smartphones et les tablettes.
- **Application multiplateforme** : est une application qui peut fonctionner sur différents systèmes d'exploitation ou environnements matériels, sans nécessiter de modification majeure du code source.
- **Dart** : est un langage de programmation open-source développé par Google. Il est utilisé notamment pour le développement d'applications avec le framework Flutter.
- **Flutter** : est un framework open-source développé par Google pour la création d'applications mobiles multiplateformes. Il permet de développer des applications natives pour iOS et Android à partir d'un seul code source en utilisant le langage de programmation Dart.
- **SGBD (Système de Gestion de Base de Données)** : Est un logiciel de haut niveau qui permet la collecte, le stockage, le traitement et la diffusion des données de manière organisée et cohérente.
- **Framework** : est une structure logicielle servant de base de développement pour des applications informatiques. Il fournit un ensemble d'outils, de bibliothèques et de conventions permettant d'accélérer le développement d'applications.
- **UML (Unified Modeling Language)** : est un langage de modélisation standardisé permettant de visualiser, spécifier, construire et documenter les éléments d'un système logiciel.

RESUME

Le secteur de l'informatique joue un rôle crucial dans de nombreux domaines de la vie quotidienne. En tant qu'étudiants en génie logiciel, il est de notre responsabilité de concevoir des systèmes, en nous basant sur une analyse approfondie et une conception précise des applications de traitement des données. Dans le cadre de notre stage académique, il a été question de mettre sur pied une application mobile de pre-enrolment et de suivi de statut de CNI. Ce projet vise à répondre aux défis liés à l'obtention de cette pièce officielle en fournissant un service plus rapide aux utilisateurs. Dans ce projet ont été utilisées une approche basée sur une architecture n-tiers et une méthodologie de développement adaptée pour réaliser ce projet. Pour ce faire, concevoir une application basée sur une architecture n-tiers avec l'approche MVC en addition avec le framework multiplateforme Flutter nous a semblé approprié. Ainsi, nous avons réalisé une étude conceptuelle par le langage de modélisation UML associé à la méthode 2TUP. Le présent rapport décrit les différentes étapes de réalisation de notre projet.

Mots clés :

- ❖ CNI
- ❖ Architecture n-tiers
- ❖ Plateforme mobile
- ❖ Gain de temps

ABSTRACT

The IT industry plays a crucial role in many areas of daily life. As software engineering students, it is our responsibility to design systems, based on in-depth analysis and accurate design of data processing applications. As part of our academic internship, we discussed setting up a mobile application for pre-enrolment and monitoring of the status of CNI. This project aims to address the challenges of obtaining this official document by providing faster service to users. In this project, an approach based on an n-tier architecture and an adapted development methodology have been used to carry out this project. To do this, we thought it was appropriate to design an application based on an n-tier architecture with the MVC approach in addition to the Flutter cross-platform framework. Thus, we carried out a conceptual study using the UML modeling language associated with the 2TUP method. This report describes the different stages of our project.

Keywords :

- National Identity Card (CNI)
- N-tier Architecture
- Mobile Platform
- Time Savings

INTRODUCTION GENERALE

Dans notre société moderne, l'utilisation des technologies de l'information et de la communication est devenue essentielle pour assurer une gestion efficace et faciliter la croissance d'entreprises de tous secteurs (agriculture, commerce, finances, transport, etc.). Le développement d'applications est ainsi devenu incontournable pour automatiser de nombreuses tâches et faciliter le traitement des données. Dans le cadre de notre stage académique en génie logiciel, il a été question de mettre en place une plateforme mobile de pre-enrolment et de suivi de statut de CNI. Cette application vise à améliorer l'expérience utilisateur en permettant aux individus d'initier le processus d'enrôlement en ligne, réduisant ainsi le besoin de visites physiques dans les bureaux administratifs. Les utilisateurs peuvent remplir les formulaires nécessaires, télécharger les documents requis et recevoir des mises à jour en temps réel sur l'état de leur demande. En intégrant une interface conviviale avec une architecture backend robuste, l'application non seulement fait gagner du temps, mais minimise également les charges administratives pesant sur les services publics. De plus, la mise en œuvre d'une architecture n-tiers garantit évolutivité et flexibilité, permettant à l'application de gérer une demande croissante tout en maintenant la sécurité et l'intégrité des données. En conséquence, cette solution innovante améliore non seulement l'accessibilité aux services d'identité, mais favorise également un système plus transparent et efficace pour la gestion des cartes nationales d'identité.

PARTIE I : PHASE D'INSERTION

RESUME

La phase d'insertion décrit le déroulement de l'accueil et de l'intégration des stagiaires dans la structure d'accueil, présente l'ensemble des informations recueillies pendant la période d'insertion aussi bien sur l'entreprise que sur la tâche à accomplir pendant la période de stage.

APERCU

INTRODUCTION

- I. ACCEUIL ET INTEGRATION
- II. PRESENTATION DE LA STRUCTURE
- III. PRESENTATION DU THEME

CONCLUSION

INTRODUCTION

Le rapport d'insertion est ce document qui décrit de manière explicite le processus d'intégration du stagiaire au sein de la structure d'accueil. Dans notre cas, à savoir celui de l'IAI Cameroun, cette phase est d'une durée limite de deux (2) semaines. Au cours de cette insertion, nous stagiaires, prenons contact avec les membres de la structure parmi lesquels notre encadreur professionnel, nous nous informons sur le fonctionnement de l'entreprise et en particulier celui de la cellule informatique. Ce document contient en son sein tout d'abord le thème qui a été affecté au dit stagiaire, ensuite les différents objectifs à atteindre et enfin les tâches effectuées durant le temps imparti dans l'entreprise. Il sera question pour nous dans ce dit rapport de mettre en exergue d'une part l'accueil au sein de l'entreprise et d'autre part, faire une présentation complète de l'entreprise MERCY INNOVATION LAB.

I – ACCUEIL ET INTÉGRATION EN ENTREPRISE

I.1 – Accueil

À notre arrivée au sein de l'entreprise MERCY INNOVATION LAB le 08 juillet 2024 à 8 h, nous avons été chaleureusement accueillis par M. KENNE Aurèle (responsable du service de développement) et M. WAMGBARA Thierry qui ont pris le soin de nous briefer sur le cheminement des activités à mener dans la structure, mais aussi sur certaines règles à respecter. Après cet entretien, le directeur des lieux M. APOBA Maximilien nous a passer un mot de bienvenu puis nous avons été pris à part par M. KENNE Aurèle pour une brève leçon sur la programmation web ensuite par M. CHEBEU DE CHEBEU Sergio (développeur) sur la programmation orienté objet. Chaque matin, nous étions fixés sur les notions de programmation web puis en fin de journée sur la programmation mobile.

I.2 – Intégration

À la suite de cet accueil chaleureux, un encadrant professionnel nous a été attribué en vue de nous accompagner tout au long de notre stage, celui-ci nous a fait passer quelques tests de performance pour juger nos différents niveaux afin de combler les vides en connaissances technologiques. L'intégration s'est faite très facilement, et la familiarité assez rapidement car étant tous jeunes, ils amenèrent une atmosphère très détendue au sein de la startup. Mais en ce qui concernait le travail effectué, tout restait professionnel et nous étions toujours guidé par M. APOBA Maximilien.

II – PRÉSENTATION DE L'ENTREPRISE

II.1 – Historique

MERCY INNOVATION LAB est la branche informatique de l'association caritative **Mercy for Cameroon**, fondée en 2023 par Monsieur **Thierry FOZING**. Notre mission est d'utiliser les technologies numériques pour améliorer les conditions de vie des populations les plus vulnérables au Cameroun et participer au développement par le numérique du continent africain.

Sous la direction de Monsieur **Maximilien APOBA**, notre directeur général, nous intervenons dans les domaines suivants :

- Développement d'applications mobiles et web pour faciliter l'accès aux services essentiels (santé, éducation) ;
- Formation aux compétences numériques pour les jeunes et les communautés.

Nous travaillons en étroite collaboration avec les communautés locales, les autorités gouvernementales et d'autres organisations à but non lucratif pour concevoir des solutions adaptées aux besoins réels du terrain. Notre équipe pluridisciplinaire est composée d'ingénieurs, de designers, de développeurs et de spécialistes des réseaux informatique. Ensemble, nous mettons l'innovation technologique au service du progrès social au Cameroun.

II.2 – Situation géographique

La Startup Mercy Innovation Lab est située à Yaoundé plus précisément à la base aérienne mais un peu plus bas que celle-ci. Elle est de couleur blanche et pour y aller vous pouvez passer par deux chemins en venant par le carrefour Mvan vous allez prendre le premier couloir en face de la base elle-même et descendre jusqu'à la deuxième entrée et puis vous y êtes, en passant par les brasseries vous allez vous arrêter au centre médical BA101 vous longez la descente juste en face avant de prendre à droite, descendre tout droite puis vous y êtes.

L'image ci-dessous est une esquisse du plan de localisation de MERCY INNOVATION LAB :

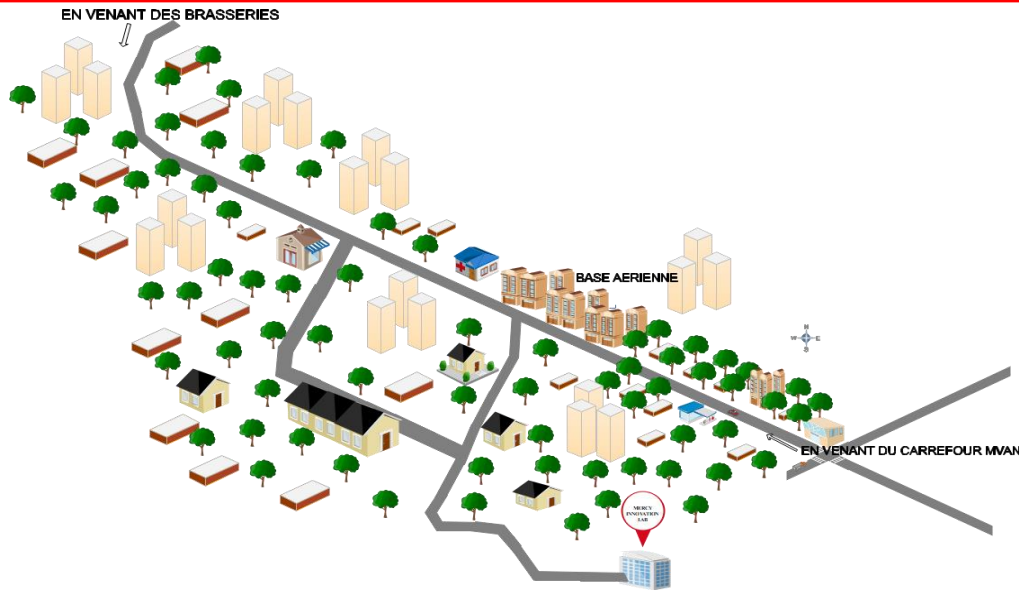


Figure 1: Plan de localisation de la startup Mercy Innovation Lab

II.3 – Organigramme

Le schéma suivant illustre l'organigramme de l'entreprise.

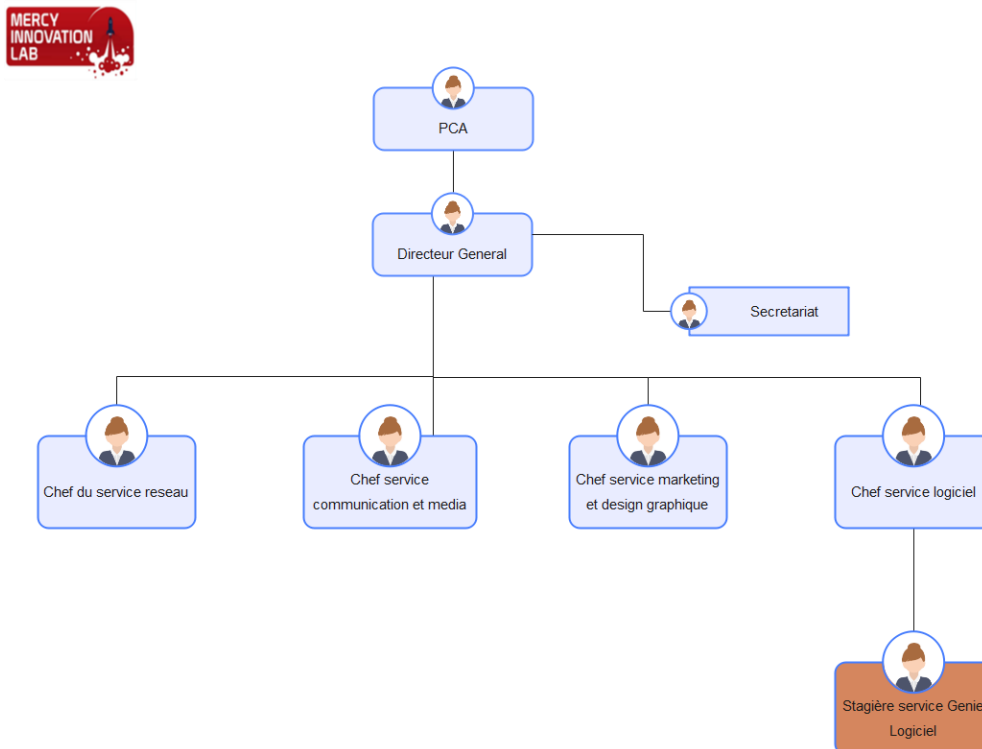


Figure 2: Organigramme de la Startup Mercy Innovation Lab

II.4 – Missions

L'objectif de Mercy Innovation Lab est de développer des solutions innovantes permettant de répondre aux besoins spécifiques des populations africaines. Nous croyons que la connaissance technologique n'a pas de frontières et que le développement de l'expertise locale peut avoir un impact positif à l'échelle mondiale.

II.5 – Activités

En tant qu'incubateur, nous proposons :

- Des espaces de travail modulables et équipés ;
- Un accompagnement personnalisé par des mentors expérimentés ;
- Des formations et ateliers sur des sujets clés pour les startups ;
- Un soutien administratif et logistique pour faciliter le lancement des projets ;
- Du développement mobile et web ;
- Du réseau informatique ;
- Du design graphique et du marketing Digitale.

II.6 – Ressources matérielles et logicielles

II.6.1 – Ressources matérielles de Mercy Innovation Lab

Dans notre cas, nous nous référerons à l'ensemble des équipements informatiques utilisés pour le traitement automatique, d'envoi, de réception, et/ou de restitution des informations. Il s'agit du parc informatique de MERCY INNOVATION LAB. Le tableau ci-dessus démontre bien le fait que MERCY INNOVATION LAB est encore à ses débuts et ne possède pas encore assez de ressources matérielles pour déployer ses services de livraison sur toute l'étendue nationale du Cameroun.

Tableau 1: Ressources matérielles

Nom	Quantité	Rôle
Projecteurs NEC	2	Equiperment utiliser pour présenter les notions d'apprentissage
Equiperment solaire	1	Pour une continuité de service même après coupure d'électricité
Imprimante SAMSUNG	1	Est un équipement matériel informatique qui se charge de l'impression des document
Mac Book pro air 2017 Desktop	1	Terminal de travail
Gameur MSI	1	Terminal de travail
Ecran IIYAMA	1	Terminal de travail
Modem (2 fybox, 1 fibre optique Camtel)	3	Pour la fourniture continue de connexion internet
Serveur Synology	1	Appareil utilisé pour stocker les fichiers et gérer la base de données de l'entreprise
Switch	1	Il permet de relier en réseau différents équipements informatiques entre eux, et contribue aussi à la sécurité du réseau d'entreprise.
Routeur	1	Est un équipement matériel informatique qui se charge de diriger et d'orienter le transit des données à travers un réseau.
Camera canon eds700d	1	Capturer
Camescope IR night vision	1	
Laptop Samsung	2	Terminal de travail
Laptop HP	3	Terminal de travail

Nom	Quantité	Rôle
Laptop Aser	1	Terminal de travail
Laptop Toshiba	1	Terminal de travail
Laptop Lenovo	1	Terminal de travail

II.6.2 – Ressources Logicielles de Mercy Innovation Lab

Tableau 2: Ressources Logicielles

NOM	TYPE	ROLE
Licence Windows 11	Licence	Mettre à niveau un appareil Windows 10 Professionnel vers Windows 11 Professionnel
300 giga sur Google Cloud	Stockage	Storage utilise des serveurs distants pour enregistrer des données telles que des fichiers, des données d'entreprise, des vidéos ou des images.
UBUNTU version 21, 22, 23	Système d'exploitation	Distribuer et modifier le code qui compose ce système d'exploitation
ODOO version 16	Application	Créer ou de dupliquer des articles et leur hiérarchie pour partager les connaissances de l'entreprise et même les rendre personnalisables par chacun en interne.
MANAGE GLI 1.0	Application	Gestion de glycémie
SAVE CAB 2.0	Application	Aide en cas de danger
MON PHARMACIEN 1.0	Application	Suivi de patient
GERMAN LANGUAGE 1.0	Application	Apprendre l'allemand a des fins médicales

III – PRÉSENTATION DU THÈME

Le développement de solutions technologiques pour répondre aux défis sociaux et environnementaux étant une priorité, nous avons décidé de mettre sur pied une application mobile pour le pre-enrolment et le suivi de statut de cni. L'application permettra aux utilisateurs de s'enregistrer et de prendre rendez-vous afin de faciliter le processus d'établissement de cni. Les autorités compétentes pourront alors accéder aux informations entrées par les utilisateurs pour les visualiser et ainsi accélérer cette démarche. L'application permettra également aux utilisateurs de signaler des cas de pertes et de péremptions de leur cni pour un traitement plus optimal. Enfin, l'application permettra de centraliser les informations sur les utilisateurs, ce qui peut aider à accélérer les opérations de recherche lors des cas d'incrimination d'un utilisateur.

CONCLUSION

Parvenus au terme de cette phase d'insertion, il était question pour nous de prendre connaissance de l'entreprise MERCY INNOVATION LAB et de l'ensemble de l'équipe avec laquelle nous serons appelés à coopérer tout au long de notre stage académique. Cette prise de contact avec de nouveaux logiciels et langages appris ont fait de notre insertion une phase très passionnante, pleine de curiosité et bien sûr ardeur dans le travail. Nous pouvons dire avec certitude que nous nous sommes intégrés et avons pu adhérer aux idées de cette entreprise de la plus belle des manières tout en maintenant une atmosphère conviviale et collaborative avec le personnel. N'étant qu'une mise en jeu d'introduction aux autres phases, il nous revient maintenant d'élaborer un document d'existant qui fera l'objet de la prochaine partie.

PARTIE II : PHASE TECHNIQUE

DOSSIER 1 : L'EXISTANT

RÉSUMÉ

Le dossier sur l'existant est l'une des parties les plus importantes dans la réalisation d'un projet. En effet, il présente de manière exhaustive le thème, présente clairement l'étude de l'existant, la critique de cet l'existant, la problématique et enfin une proposition de solution.

APERCU

INTRODUCTION

I. PRÉSENTATION DU THÈME

II. ETUDE DE L'EXISTANT

III. CRITIQUE DE L'EXISTANT

IV. PROBLEMATIQUE

V. PROPOSITION DE LA SOLUTION

CONCLUSION

INTRODUCTION

L'une des parties les plus importantes dans la réalisation d'un projet réside dans la collecte et l'étude de l'information dans notre système d'étude. Il sera donc question pour nous ici de faire une quête minutieuse de l'information qui constituera notre existant, dans le but de la critiquer afin de proposer des solutions.

I – PRESENTATION DU THEME

Notre thème qui porte sur la réalisation d'une application mobile de pre-enrolment et de suivi de statut de cni, est mis sur pied pour faciliter le processus d'établissement de carte nationale d'identité (CNI) au Cameroun, dans l'application, vous aurez des formulaires à remplir avec vos données personnelles, vous aurez aussi la possibilité de payer en un clic grâce au système de paiement intégré, et télécharger votre fiche de renseignement contenant vos données personnelles qui vous permettra ainsi de finaliser la production de votre cni après la prise de rendez-vous. L'application permettra non seulement aux utilisateurs de pouvoir effectuer des demandes de cni, mais aussi signaler des pertes/vols et péremptions de celle-ci. Elle facilitera ainsi le travail aux autorités compétentes qui pourront selon leur disponibilité prendre une marge de personnes par jours afin d'éviter des déplacements inutiles de certains. L'administrateur quant à lui se charge de la gestion des comptes des autorités compétentes de leurs créations à leurs destructions.

II –ETUDE DE L'EXISTANT

II.1 – Cas 1 étude du site web PassCam

Il existe un site web sur lequel nous avons pris exemple pour la réalisation de notre application. Son nom PASSCAM. PassCam est un site web de pre-enrolment en ligne du passeport biométrique Camerounais, en effet, tout citoyen peut demander, renouveler ou remplacer un passeport biométrique depuis leur site web www.passcam.cm ou www.dgsn.cm . Les étapes de pre-enrolment se font comme suit, vous allez sur leur site et vous serez accueillis par la première page de bienvenu qui vous incitera également à fournir soit votre adresse email soit notre numéro de téléphone, passé cette étape il vous sera envoyé un numéro d'identification qui vous permettra de continuer la procédure, après cette étape il vous sera affiché une page dans laquelle on vous demandera de remplir un formulaire avec des informations telles que le type de demandes, la destination du voyage et le motif de celui-ci. Par la suite un autre formulaire vous sera envoyé avec plus d'information sur vous telles que :

- Données du demandeur : nom, prénom, date de naissance, sexe, pays de naissance, région, département, lieu de naissance,
- Informations sur le demandeur : numéro de téléphone et adresse électronique,
- Lieu de résidence : pays, région, département, lieu, rue, profession.
- Information sur les parents : nom et prénom du père et de la mère, et des informations du tuteur légal,
- Informations sur la cni : format de cni, numéro de cni, date de délivrance de cni, date d'expiration

Par la suite on vous permettra de visualiser les informations que vous avez remplis pour dernière vérification. Ensuite il vous sera affiché une page de paiement où vous aurez la possibilité de choisir le moyen de paiement (visa, MasterCard, mtn, orange, yoomee...).

Après paiement une fiche contenant ses informations avec un code barre lui sera envoyé permettant ainsi de l'identifier de manière unique, il aura donc la possibilité de télécharger cette fiche sous format PDF, et enfin vous pourriez prendre rendez-vous pour la finalisation de l'obtention du passeport.

II.2 – Cas 2 étude sur l'établissement de la cni (Carte Nationale d'Identité) au Cameroun

Obtenir une carte d'identité est une démarche importante pour tout citoyen. Cette pièce officielle d'identification est nécessaire pour de nombreuses formalités administratives, comme l'ouverture d'un compte en banque, l'inscription à la Sécurité sociale ou encore pour voyager dans certains pays. La carte d'identité atteste de votre identité, de votre nationalité et de votre domicile. Elle est donc un document essentiel à posséder pour affirmer votre identité légale et accéder à vos droits. La procédure pour l'obtenir est relativement simple, au Cameroun elle se fait généralement dans des commissariats. Vous devriez tout d'abord être de nationalité camerounaise, vous devez établir un certificat de nationalité, faire une copie de votre acte de naissance, avoir un document justificatif de votre emploi actuel et vous munir des frais d'établissement qui s'élève à la somme de 2800FCFA. Dès la possession de tous les

documents illustrés vous vous rendez dans un commissariat de votre choix et vous soumettez votre demande d'établissement de cni, après une phase de question il vous sera remis un récépissé de cni valable trois mois temps d'attente pour la délivrance de la cni proprement dite. Pour ce qui est du cas de cni égarée ou volée, vous devez vous rendre au commissariat puis acheter l'imprimé de l'attestation de déclaration de perte/vol (100FCFA) le remplir y mettre un timbre fiscal (1500FCFA) et enfin le faire légaliser par un officier de police ou avec la photocopie de sa cni, ensuite il sera question pour les autorités compétentes d'utiliser ce document pour refaire la pièce égarée.

Informations prises auprès de Madame BENDJA Marie Liliane inspecteur de police principal du groupement régional de la voie publique et de la circulation du centre basé à l'école de police.

III – CRITIQUE DE L'EXISTANT

Par la présentation du fonctionnement actuel, nous avons relevé les limites présentées dans le tableau qui suit :

Tableau 3: Critique de l'existant

CRITIQUES	LIMITES	SOLUTIONS
<i>L'établissement de cni passe par un entretien individuel</i>	Le service est lent et la plupart des citoyens est renvoyés au lendemain pour l'établissement de celle-ci.	Un service de qualité plus rapide.
<i>Pour savoir si sa cni est délivré, il faut se rendre au commissariat et attendre que les employés fouillent les cartes et parfois même en sachant que la vôtre n'est pas disponible</i>	Déplacement inutile, temps perdus.	Suivi du statut de sa cni jusqu'à délivrance pour le retrait de celle-ci.
<i>Lorsqu'on va au commissariat il est possible que l'on oublie un document pour l'établissement de la cni</i>	Gaspillage de temps	L'application vous permet d'entrer vos informations étant chez vous ce qui permettra de fournir tous les documents sans oublier un.

CRITIQUES	LIMITES	SOLUTIONS
<i>Il peut arriver que le personnel ne soit pas disponible</i>	Mouvement inutile,	L'application est disponible 24h/24 et 7jours/7.

IV – PROBLEMATIQUE

Pour donner suite à l'existant présenté ci haut ainsi qu'aux différentes limites exposées, on peut se poser la question de savoir comment faciliter le processus d'établissement de la carte nationale d'identité ?

V – PROPOSITION DE LA SOLUTION

Face aux différentes limites rencontrées, il sera donc question de mettre sur pied une plateforme qui permettra tout d'abord aux utilisateurs de pouvoir gagner en temps dans le processus d'établissement de cni, signaler des cas de perte ainsi que de péremption, permettre aux administrateurs de gérer les comptes pour une meilleure sécurité des données.

CONCLUSION

Au terme de cette étape, il a été question de faire une descente sur le terrain dans le but de recueillir le maximum d'informations possibles en rapport avec notre système d'étude. Par la suite, il a été question d'en faire une critique afin de ressortir les limites, les conséquences engendrées et enfin de proposer des solutions qui permettront de résoudre efficacement le problème général.

DOSSIER 2 : CAHIER DES CHARGES

RESUME

Le cahier de charges est document projet établi de commun accord entre le maitre d'ouvrage et le maitre d'œuvre, qui permet de définir de manière exhaustive les différentes fonctionnalités et exigences à fournir par la future solution. Ce document présente aussi une estimation du coût des différentes ressources à mettre en place, ainsi que du délai pour réaliser le projet.

APERCU

INTRODUCTION

- I. CONTEXTE ET JUSTIFICATION
- II. BENEFICIAIRES ET CIBLES
- III. OBJECTIFS
- IV. EXPRESSIONS DES BESOIN
- V. CHARTE GRAPHIQUE
- VI. PLANIFICATION DU PROJET
- VII. ESTIMATION DU COUT DU PROJET
- VIII. CONTRAINTES DU PROJETS
- IX. LIVRABLES

CONCLUSION

INTRODUCTION

L'une des informations les plus capitales retenues lors de la période d'insertion a été le thème du stage académique, ce thème qui était : « Analyse Conception et Réalisation d'une application mobile de pre-enrolment et de suivi de statut de cni ». Il sera question pour nous dans cette partie de définir les objectifs, les caractéristiques fonctionnelles et non fonctionnelles du système à mettre en place et faire une estimation financière du projet ainsi que son coût.

I – CONTEXTE ET JUSTIFICATION

I.1 – Contexte

Pays situé en Afrique centrale et s'étendant sur une superficie de 475 442 km², le Cameroun regorge environ 25 millions d'habitants en son sein et ainsi, pour jouir de ses droits et affirmer sa nationalité il faudrait détenir un document officiel (carte nationale d'identité) qui attesterait bien évidemment votre statut de camerounais. Cependant, la méthode employée par les commissariats s'avère ne pas convenir à toute la population camerounaise puisqu'il n'est pas toujours facile de s'en procurer au Cameroun. Le processus d'obtention de sa carte nationale d'identité passe par une procédure assez fatigante puisqu'il faudrait se rendre très tôt dans un commissariat muni de votre acte de naissance, de votre certificat de nationalité ainsi qu'un justificatif de votre profession actuelle et d'une somme de 2800 FCFA, ce processus peut sembler parfois pénible car il peut arriver que la marge de personne ait déjà été atteinte lors de votre venue au commissariat et que l'on vous renvoie chez vous dans certains cas, d'autre situation peuvent se produire, par exemple il peut arriver que vous ayez oublié certains documents ce qui vous fera perdre en temps et si vous vivez loin du commissariat en argent aussi, par la suite vous serez appelé à faire preuve d'une grande patience jusqu'à la délivrance de celle-ci car vous n'êtes pas informés de ladite date et vous devrez de ce fait vous rendre le commissariat de départ vérifier que votre carte est bel et bien été délivrée. Ces conditions d'obtention de ce document officiel aussi pénible qu'elles soient dans certains cas peut occasionner un désintéressement de la population à établir leur carte nationale d'identité.

I.2 – Justification

Dans le souci de faciliter et d'accélérer le cheminement pour l'obtention de sa carte nationale d'identité ainsi que la gestion des cas de perte et de péremption, la mise sur pied d'une application mobile de pre-enrolment et de suivi de statut de cni devient tout à fait légitime et utile à tous.

II – BÉNÉFICIAIRES ET CIBLES

II.1 – Bénéficiaires

Les bénéficiaires de ce projet sont **les commissariats** ;

II.2 – Cibles

Le projet de mise en place d'une application mobile de pre-enrolment et de suivi de statut de cni est destiné à toute la population vivant sur le territoire Camerounais et pouvant justifier son statut camerounais grâce au certificat de nationalité.

III – OBJECTIFS

III.1 – Objectif général

L'objectif général consiste à exprimer le sens du projet, le but à atteindre, une visée à long terme. Ainsi, l'objectif général de ce projet est de faciliter le processus d'obtention de sa carte nationale d'identité.

III.2 – Objectifs spécifiques

Les objectifs spécifiques précisent les chemins qu'il faut prendre pour atteindre l'objectif général. Notre objectif global répond aux objectifs spécifiques ci-après :

- Facilité le processus de traitement des cas de perte et de péremption des cartes d'identités,
- Gain de temps,
- Facilité le travail des autorités compétentes chargés d'établir les cartes d'identité.
- Accélérer le service

IV – EXPRESSION DES BESOINS

Les besoins du système représentent les attentes en termes de fonctionnalités ou non des utilisateurs vis-à-vis du système.

IV.1 – Besoins fonctionnels

Les besoins fonctionnels sont les exigences qui précisent ce que le système doit faire. En d'autres termes, spécifient une fonction, un comportement ou une action que le système doit exécuter. Il s'agit pour nous ici des différentes fonctionnalités attendues.

Les exigences fonctionnelles qui concernent notre projet sont présentées dans le tableau ci-contre :

Tableau 4 : Tableau des besoins fonctionnels de notre application

ACTEURS	TITRES	OBJECTIFS
Autorités compétentes Administrateur	Créer compte	Avant de pouvoir effectuer la moindre opération, chaque acteur doit avoir un compte.
Autorités compétentes (policier)	Gérer profil	Il est possible de consulter son profil, de le modifier ainsi que de le supprimer.
	Gérer planning	Il est possible de consulter, d'établir un planning afin d'informer les utilisateurs sur la disponibilité du service, mais aussi de le modifier en cas d'erreur et de le supprimer.
	Voir historique de paiement	Il est possible de voir l'historique de paiement pour les confirmations de paiement lors de la venue des utilisateurs pour la finalisation du processus.

ACTEURS	TITRES	OBJECTIFS
Administrateur	Gérer comptes	Il est possible d'ajouter, supprimer, bloquer, modifier, consulter et rechercher n comptes dans l'application.
Citoyens Camerounais		Il est possible de pouvoir effectuer une demande d'établissement de cni, de signaler un cas de perte/vol, ainsi que signaler un cas de péremption.
	Télécharge fiche de renseignement en PDF	Il est possible de télécharger une fiche de renseignement contenant toute informations de l'utilisateurs rempli au préalable par lui
	Effectuer paiement	Il est possible d'effectuer un paiement via orange money ou même MTN Money
	Suivre état cni	Il est possible de suivre d'état de sa cni en recevant une notification qui informera sur le statut de la cni.
	Prendre rendez-vous	Il est possible de prendre rendez-vous dans un commissariat donné pour la finalisation du traitement.

IV.2 – Besoins non fonctionnels.

Les besoins non fonctionnels représentent les aspects d'un système qui ne sont pas directement liés à sa fonctionnalité principale. Elles sont tout aussi nécessaires à la mise en œuvre de l'application que les exigences ou besoins fonctionnels.

Les exigences non fonctionnelles qui concernent notre projet sont :

- **La sécurité informatique** : il s'agit d'un ensemble de piliers, de dispositifs, vastes et multiformes visant à protéger un système informatique et ses données contre toute violation, fuite, publication d'informations privées ou attaque. De manière plus détaillée, cette sécurité informatique se repartira en ses cinq grands piliers à savoir :
 - **L'authentification** : procédure permettant pour un système informatique de vérifier l'identité d'une personne ou d'un terminal et d'autoriser l'accès de cette entité aux ressources. Il faudra donc au préalable s'authentifier dans la plateforme avant d'effectuer la moindre opération.
 - **La disponibilité des données** : chaque utilisateur doit en effet avoir accès aux données en temps réel. Les données doivent être disponibles pour n'importe qui et à n'importe quel moment
 - **La confidentialité** : cela signifie que les données doivent être accessibles uniquement aux personnes concernées
 - **La non-répudiation** : principe qui oblige un utilisateur à ne réfuter ou nier la moindre opération effectuée dans la plateforme
 - **L'intégrité des données** : aptitude de la plateforme à protéger son code et ses différentes données contre des accès non-autorisés. Il s'agit de l'état des données qui, lors de chaque traitement, conservation ou transmission, ne subissent aucune altération, aucune destruction volontaire ou involontaire.
- **La maintenabilité** : c'est la facilité à apporter des corrections ou à ajouter des nouvelles fonctionnalités à l'application
- **La robustesse** : aptitude de l'application à pouvoir fonctionner quel que soit les conditions d'exploitation
- **La réutilisabilité** : aptitude du logiciel à pouvoir être utilisé en tout ou en partie
- **L'ergonomie** : capacité de la plateforme à être facilement utilisée par une personne pour réaliser la tâche pour laquelle elle a été conçue. On doit en effet mettre en place une plateforme attrayante.

V – CHARTE GRAPHIQUE

La charte graphique est un document de travail qui contient l'ensemble des règles fondamentales d'utilisation des signes graphiques qui constituent l'identité graphique d'une marque, d'un projet, d'une application... elle rassemble donc tous les éléments visuels essentiels à la marque, tels que le logo, les polices, etc. Elle explique comment et dans quelles conditions ces éléments peuvent être utilisés, et permet de communiquer en gardant une cohérence graphique.

Nous pouvons présenter la charte graphique de notre projet ainsi qu'il suit :

V.1 – Logo

Notre logo présentera clairement le nom de l'application, à savoir **CamCard** :

- « Cam » représente ici le nom du Cameroun (les trois premières lettres).
- « Card » représente le mot carte en anglais ce qui spécifie le thème proprement dit.

Nous avons ainsi l'illustration suivante :

The logo for 'CamCard' is displayed in a large, bold, sans-serif font. The word 'Cam' is colored green and 'Card' is colored orange.

Figure 3: Logo de notre application

V.2 – Typographie

La typographie regroupe les différents procédés de composition et d'impression utilisant des caractères et des formes en relief, ainsi que l'art d'utiliser les différents types de caractères dans un but esthétique et pratique. Elle consiste à designer la manière dont les mots et le texte sont mis en forme en utilisant des caractères individuels : les lettres, accents, ponctuations, symboles, chiffres et autres caractères spéciaux. Elle peut se caractériser par la sélection de polices d'écriture, de types de caractères, comme les lettres (minuscules, majuscules), les chiffres, les signes de ponctuation, les caractères spéciaux, ainsi que les règles de mise en page.

- La police principale utilisée pour l'application est la police **Roboto** comme le montre l'illustration suivante :

Whereas disregard and contempt for human rights have resulted

Figure 4: Police Roboto

V.3 – Palette de couleurs

Plusieurs couleurs sont utilisées dans les différentes pages de notre application :

- L'indigo (#4b0082)
- Le noir (#000000)
- Le rouge (#FF0000)
- Le vert (#00FF00)
- Le bleu (#0000FF)
- Le blanc (FFFFFF)

VI – PLANIFICATION DU PROJET

VI.1 – Intervenants

Les différentes personnes qui interviennent de manière active dans la réalisation du projet sont citées dans le tableau ci-dessous :

Tableau 5 : Liste des intervenants du projet

Noms et prénoms	Fonctions	Rôles
ASSAME NNAMA ORLANE SANDY	Etudiante en deuxième année à l'IAI Cameroun	Concepteur et réalisateur du projet
BELINGA ESTELLE GRAYCE epse BONONO	Enseignante à l'IAI Cameroun	Encadrant académique
APOBA ELAH MAXIMILIEN	Directeur général de Mercy Innovation Lab	Encadrant professionnel

VI.2 – Diagramme de Gantt

Afin de mettre sur pied notre projet, l'ensemble du travail a été subdivisé en plusieurs tâches et chacune d'elle représente une phase d'avancement et de réalisation de notre projet. Le diagramme ci-dessous nous décrit l'agencement de ces différentes phases avec les différents jalons, les durées ainsi que les ressources impliquées.

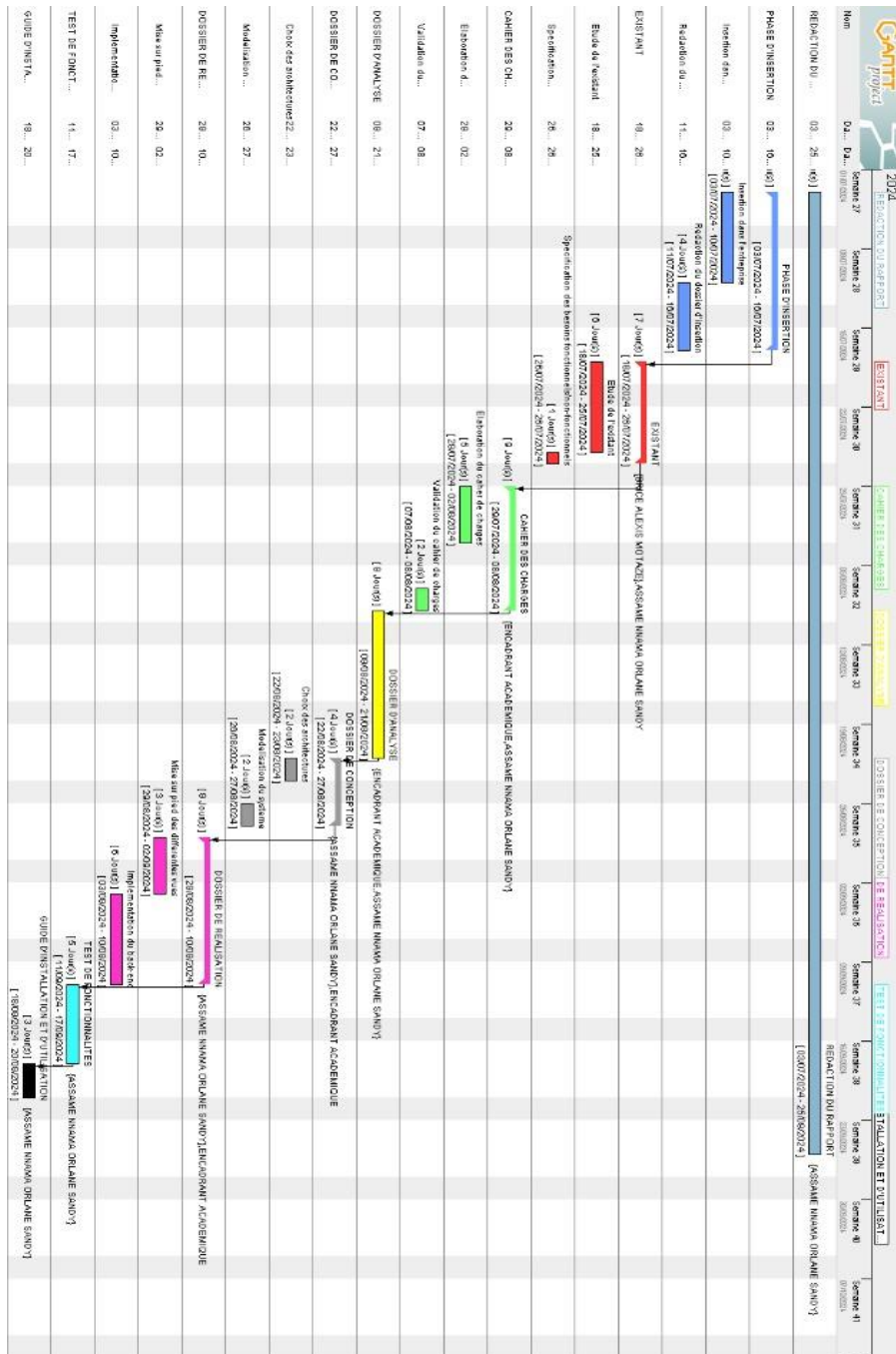


Figure 5: Diagramme de Gantt de notre projet

VII – ESTIMATION DU COÛT DU PROJET

Le coût est une variable fortement liée à tout projet. Pour réaliser notre application de pre-enrolment et de suivi de statut de cni nous avons évalué les dépenses liées à la rémunération des intervenants, des achats du matériel et des logiciels adéquats pour le bon déroulement du projet. A cet effet, il est donc nécessaire de disposer de la ressource humaine qualifiée, d'une configuration matérielle minimale, d'une ressource logicielle et financière conséquente.

VII.1 – Ressources Logicielles

Le tableau ci-dessous résume les logiciels indispensables au développement et déploiement de l'application.

Tableau 6: Ressources logicielles

Logiciel	Version	Rôle	Prix (FCFA)
Power AMC	15.1	Atelier de Génie Logiciel	50 000
Microsoft Office	365	Suite bureautique	Gratuit
Microsoft Edge		Navigateur	Gratuit
Microsoft Windows	11	Système d'exploitation	150 000
Gantt Project	2.8	Outil de planification	209 600
Edraw Max	12.0	Outil de planification	36 200
WampServer	3.3	Serveur local	Gratuit
MySQL	8.0	Système de gestion de base de données	Gratuit
TOTAL		445 800	

Source : les prix consacrés aux ressources logicielles sont tirés de la mercuriale des prix et tarifs de références 2024.

VII.2 – Ressources matérielles

Le tableau ci-dessous résume le matériel indispensable au déploiement de l'application.

Tableau 7: Ressources matérielles

Matériel	Caractéristiques	Quantité	P.U (FCFA)	Prix total (FCFA)
Ordinateur portable HP	Core i7 RAM : 8,0Go	01	380 000	380 000
CD-ROM	CD-RW	01	550	550
Rame de papier	A4 80G Blanc	01	5000	5000
Router WIFI	Huawei 4G	01	35000	35000
TOTAL			420 550	

Source : les prix consacrés aux ressources matérielles sont tirés de la mercuriale des prix et tarifs de références 2024.

VII.3 – Ressources humaines

Le tableau ci-dessous récence les ressources humaines nécessaires :

Tableau 8: Ressources humaines

Main d'œuvre	Durée (en jours)	Quantité	P.U/jour (FCFA)	Prix total (FCFA)
Analyste	30	01	150000	4 500 000
Concepteur				
Programmeur	20	01	150000	3 000 000
Front-end designer	10	01	150000	1 500 000
Testeur	05	01	90000	450 000
TOTAL			9 450 000	

VII.4 – Tableau récapitulatif

Le tableau suivant résume le cout total indispensable au déploiement de l'application.

Tableau 9: Tableau récapitulatif

RESSOURCES	MONTANT
LOGICIELLES	445 800
MATERIELLES	420 550
HUMAINES	9 450 000
IMPREVUS (10%)	1 031 635
TOTAL	11 347 985 FCFA

Soit en lettre **onze millions trois-cent quarante-sept mille neuf cent quatre-vingt-cinq Francs CFA.**

VIII – CONTRAINTES DU PROJET

- **Le coût** : la réalisation d'un projet est toujours précédée par son évaluation tant sur le plan physique que sur le plan matériel. En effet, l'estimation du coût du projet passe par l'évaluation des dépenses indispensables pour l'achat des logiciels de développement, du matériel et surtout de la main d'œuvre qui la mettra sur pied. Dans le cadre de notre projet, le budget devant être alloué est de 11 347 985 FCFA (onze millions trois-cent quarante-sept mille neuf cent quatre-vingt-cinq Francs CFA.)
- **La qualité du produit** : la contrainte de qualité constitue un aspect très important. Etant donné qu'une application ne peut être totalement parfaite, elle devrait se rapprocher du zéro défaut. Pour plus de sécurité, la plateforme devra respecter les conditions suivantes :
 - **L'exactitude** : aptitude d'un logiciel à fournir les résultats attendus dans les conditions normales d'utilisation.
 - **La robustesse** : aptitude de l'application à pouvoir fonctionner quel que soit les conditions d'exploitation
 - **L'ergonomie** : capacité de la plateforme à être facilement utilisée par une personne pour réaliser la tâche pour laquelle elle a été conçue. La plateforme doit en effet proposer des interfaces de qualité

- **La maintenabilité** : il s'agira de la capacité à apporter des corrections ou à ajouter des nouvelles fonctionnalités à la plateforme
- **La réutilisabilité** : il s'agira ici de l'aptitude du logiciel à pouvoir être utilisé en tout ou en partie
- **Le délai** : notre projet est valide uniquement si le livrable suivant est terminé avant la limite de nos trois mois de stage, il s'agit d'un rapport du stage avec les parties suivantes : un dossier d'insertion, un cahier des charges, un dossier d'analyse, un dossier de conception, un dossier de réalisation et un guide d'utilisateur. Plus concrètement, le projet devra être mis sur pied durant nos trois mois de stage académique, allant du 08 Juillet 2024 au 30 Septembre 2024.

IX – LES LIVRABLES

Le projet ne sera jugé acceptable que si et seulement si au terme de notre réalisation les livrables ci-dessous sont achevés :

- ✚ Un prototype de l'application gravé sur un DVD ;
- ✚ Un fichier Powerpoint pour la présentation ;
- ✚ Un rapport de stage composé de :
 - Un rapport d'insertion ;
 - Un dossier sur l'existant ;
 - Un cahier des charges ;
 - Un dossier d'analyse ;
 - Un dossier de conception ;
 - Un dossier de réalisation ;
 - Les tests des fonctionnalités ;
 - Un guide d'utilisateur ;

CONCLUSION

Le cahier des charges est un récapitulatif des besoins du client en termes de réalisation d'un projet et des ressources nécessaires à la réalisation favorable dudit projet. En effet, pour répondre aux exigences du client, il nous faut concilier les ressources humaines, les ressources matérielles, et les mêmes techniques favorables à la satisfaction du résultat attendu. Après la réalisation du cahier des charges il est temps pour nous maintenant de nous lancer dans la phase d'analyse.

DOSSIER 3 : LE DOSSIER D'ANALYSE

RESUME

Le dossier d'analyse a pour but d'affiner le choix de la méthode à utiliser tout au long du projet. Il permet dans la même lancée de faire comme son nom l'indique une analyse détaillée du projet à mettre en œuvre.

APERCU

INTRODUCTION

I. DEMARCHE D'ANALYSE

II. DIAGRAMME INTERVENANTS

CONCLUSION

INTRODUCTION

La réalisation d'un bon projet repose principalement sur son analyse. Ainsi, un projet mal analysé ne pourrait donner de bons résultats. Les résultats de l'analyse ne dépendent d'aucune technologie particulière. Le but principal de cette partie est d'indiquer comment le travail sera organisé et mené. Pour ce faire, le choix d'une démarche d'analyse permettant un suivi aisé du développement de notre plateforme est primordial. A cet effet, le choix s'est posé sur le processus 2TUP du langage UML. Il est donc question pour nous dans cette partie, de présenter de façon détaillée la démarche d'analyse, de justifier le choix de cette démarche par rapport à notre champ d'étude, de choisir le processus unifié et enfin de les appliquer pour la modélisation des premiers diagrammes.

I – DEMARCHE D'ANALYSE

I.1 – Étude comparative MERISE et UML

Merise est une méthode d'analyse, de conception et de gestion de projet informatique. Cette méthode reste adaptée pour la gestion des projets internes aux organisations, se limitant à un domaine précis. UML, en français, le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet. L'UML est une synthèse de langages de modélisation objets antérieurs. Outre des différences évidentes de natures, buts et d'utilisations (une méthode d'une part et un langage d'autre part), l'aspect graphique de l'UML est une caractéristique qui le distingue de Merise.

Tableau 10: Etude comparative entre MERISE et UML

MERISE	UML
Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprises	Unified Modeling Language
<i>Est une méthode systémique d'analyse et de conception de systèmes d'information. C'est-à-dire qu'elle utilise une approche systémique.</i>	N'est cependant pas une méthode, mais plutôt un langage de modélisation objet à qui il faut associer une démarche pour en faire une méthode. C'est le cas de la méthode 2TUP ; RUT et XP.
<i>Propose de considérer le système réel selon deux points de vue :</i> <ul style="list-style-type: none"> - Une vue statique (données) - Une vue dynamique (traitements). 	Propose une approche différente de celle de MERISE en ce sens qu'il associe les données et les traitements.
<i>C'est-à-dire qu'avec la méthode MERISE, nous avons une étude séparée des données et des traitements</i>	Car avec UML, centraliser les données d'un type et les traitements associés permet de limiter les points de maintenance dans le code et facilite l'accès à l'information en cas d'évolution du logiciel. De plus, UML décrit la dynamique du système d'information comme un ensemble d'opérations attachées aux objets du système.

I.2 – Choix Et Présentation De La Démarche D'analyse

Après cette étude comparative entre le langage UML et la méthode MERISE, notre choix sera porté sur **le langage de modélisation UML**.

Sur le plan historique, trois développeurs, JAMES RUMBAUCH, GRADY BOOCH et NARY JACOBSON ont décidé de fusionner plusieurs méthodes (la méthode BOOCH, OMT et la méthode OOSE) en un langage commun et standardisé appelé UML. Il s'agit d'un langage de modélisation graphique à base de schémas appelés diagrammes conçus pour fournir une norme de visualisation lors de la conception d'un logiciel.

C'est en 1997 que l'OMG met UML à sa version 1.0. Non satisfait de cette version, les développeurs ont mis en place un groupe de travail pour améliorer le langage sur plusieurs versions, c'est ainsi qu'on est quitte de la version 1.0 à la version 1.5 qui comporte neuf diagrammes et dont la dernière version validée par l'OMG en 2017 est la version UML 2.5.1 qui comporte quatorze diagrammes. Elle s'articule autour de l'approche objet qui s'organise autour de trois principales fonctions :

- Elle est itérative et incrémentale
- Elle est guidée par les besoins de l'utilisateur
- Elle est centrée sur l'architecture du logiciel.

Elle se subdivise en deux vues :

- La vue statique composée des diagrammes UML structurels qui représente le système physiquement et comporte sept (07) diagrammes :
 - ✓ **Diagramme de classe** : illustre les classes avec les attributs, les méthodes et les relations entre les classes,
 - ✓ **Diagramme de composants** : illustre les éléments logiciels (exécutable, librairie...) et leurs dépendances.
 - ✓ **Diagramme de structures composite** : décrit les relations entre les composants d'une classe.
 - ✓ **Diagramme de déploiement** : illustre la disposition physique du matériel et la répartition des composants sur ce matériel.

- ✓ **Diagramme d'objets** : est une instance spécifique d'un diagramme de classe à un moment précis.
 - ✓ **Diagramme de packages** : schématise les dépendances entre les packages (ensemble d'élément de modélisation),
 - ✓ **Diagramme de profil** : depuis la version 2.2, permet de spécialiser, personnaliser pour un domaine particulier un métamodèle de référence d'UML.
- La vue dynamique constituée des diagrammes UML comportementaux qui représente les interactions effectuées dans le système, elle comporte sept (07) diagrammes :
- ✓ **Diagramme d'activité** : décrit le comportement du système sous forme de flux ou d'enchaînement d'activités.
 - ✓ **Diagramme de communication** : représente à l'image du diagramme de sequence les échanges des messages entre les différents objets.
 - ✓ **Diagramme global d'interaction** : modélise la façon dont les groupes d'objets collaborent pour réaliser un comportement donné.
 - ✓ **Diagramme de séquence** : représente de façon séquentielle la manière dont se font les traitements et indiquent les interactions entre les éléments du système.
 - ✓ **Diagramme d'états-transition** : montre les différents états que peut prendre un objet lors de son cycle de vie.
 - ✓ **Diagramme de temps** : décrit les interactions entre les comportements de différents objets en indiquant la chronologie d'exécution des opérations sur ces objets.
 - ✓ **Diagramme de cas d'utilisation** : montre les fonctionnalités du système.

I.3 – Étude Comparative Des Processus Unifiés

L'étude comparative des processus unifiés peut se résumer dans le tableau ci-dessous :

Tableau 11: Etude comparative entre les processus unifiés

Critères de comparaison	UP	2TUP	XUP
Cycle de vie	Il s'agit d'un processus itératif et incrémental, divise en quatre phases	C'est une autre variante d'UP, qui propose un double flux de	C'est une variante agile d'UP, qui se

Critères de comparaison	UP	2TUP	XUP
	principales (Inception, élaboration, construction et transition) et qui s'adapte aux besoins de chaque projet.	développement : l'un pour les fonctionnalités stables et l'autre pour les changements évolutifs et incrémentaux. Il s'agit donc d'un processus hybride adapte aux projets avec des exigences changeantes. Il utilise le cycle en « Y »	concentre sur la livraison rapide de fonctionnalités utiles et prioritaires, en s'appuyant sur des pratiques comme le développement pilote par les tests et l'intégration continue
Collaboration étroite avec le client	Le client est impliqué dès le début du processus UP, dans la phase d'Inception en tant que partenaire de l'équipe de développement. Les exigences sont établies et validées en collaboration avec le client qui peut donner son avis tout au long du projet	Il implique également le client dans la phase d'Inception en établissant en accord sur les exigences stables. Les changements incrémentaux sont valides en continue avec le client qui peut donner son approbation à chaque itération.	L'approche XUP place le client au centre du processus, impliquant régulièrement le client dans les revus de code et les tests de validation. Le client est encouragé à donner un feedback permanent sur les fonctionnalités livrées qui permet d'ajuster

Critères de comparaison	UP	2TUP	XUP
Documentation	Il propose une documentation complète et détaillée qui couvre toutes les phases du cycle de vie y compris les exigences, les spécifications, les plans de tests, les modèles et conception... cependant cette documentation peut être assez lourde et difficile à maintenir	Il propose une documentation adaptée à chaque flux de développement, en fonction de leur stabilité. Les fonctionnalités stables sont documentées de manière plus exhaustive tandis que les changements incrémentaux nécessitent moins de documentation	rapidement les priorités Il préconise une documentation légère et juste suffisante pour répondre aux besoins du projet en accordant une grande importance aux tests automatisés et au code bien écrit. Les descriptions des fonctionnalités sont généralement exprimées en termes de tests plutôt qu'en documents formels
Application fonctionnelle	Il est bien adapté aux projets de grandes envergures et complexes en offrant une vision globale du cycle de vie et en favorisant une architecture solide et évolutive. Cependant le processus peut sembler trop lourd pour les projets	Il convient aux projets avec des exigences stables mais également avec des changements incrémentaux qui peuvent être gérés en parallèle. Cette approche offre une grande flexibilité tout en conservant une	Il convient aux projets avec les exigences changeantes qui nécessitent une livraison de fonctionnalités utiles. Cependant cette approche peut

Critères de comparaison	UP	2TUP	XUP
	de petites tailles	architecture solide et qualité de code élevée	nécessiter une forte discipline de la part de développeurs pour maintenir une qualité de code élevée

I.4 – Choix Et Présentation Du Processus Unifié

Après cette étude comparative, notre choix sera porté sur 2TUP (Two tracks unified process).

Dans le processus 2TUP, les activités de développement sont organisées suivant 5 workflows qui décrivent :

- La capture des besoins,
- L'analyse,
- La conception,
- L'implémentation
- Et le test.

Le processus 2TUP est une trame des meilleures pratiques de développement, il doit être utilisé comme un guide pour réaliser un projet et non comme l'arme ultime et universelle de développement.

2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système. En produire le

cahier des charges et à modéliser le contexte. Le processus s'articule autour de 3 phases essentielles :

✚ **Une branche technique** : capitalise un savoir-faire technique et/ou des contraintes techniques. Cette branche droite comporte :

- La capture des besoins techniques, qui recense toutes les contraintes et les choix dimensionnant la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte de contraintes d'intégration avec l'existant conditionnent généralement des prérequis d'architecture technique ;
- La conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle a pour objectif d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système informatique et écarte la plupart des risques de niveau technique. L'importance de sa réussite est telle qu'il est conseillé de réaliser un prototype pour assurer sa validité.

✚ **Une branche fonctionnelle** : capitalise un savoir-faire technique et/ou des contraintes techniques. Cette branche comporte :

- La capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs. De son côté, la maîtrise d'œuvre consolide les spécifications et en vérifie la cohérence et l'exhaustivité.
- L'analyse, qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier. Les résultats de l'analyse ne dépendent d'aucune technologie particulière.

✚ **Une phase de réalisation** : consiste à réunir les deux branches, permettant de mener une conception applicative et enfin la livraison d'une solution adaptée aux besoins. La branche du milieu comporte :

- La conception préliminaire, qui représente une étape délicate, car elle intègre le modèle d'analyse dans l'architecture technique de manière à tracer la cartographie des composants du système à développer,
- La conception détaillée, qui étudie ensuite comment réaliser chaque composant ;
- L'étape de codage, qui produit ces composants et teste au fur et à mesure les unités de code réalisées,
- L'étape de recette, qui consiste enfin à valider les fonctions du système développé.

Représentation :

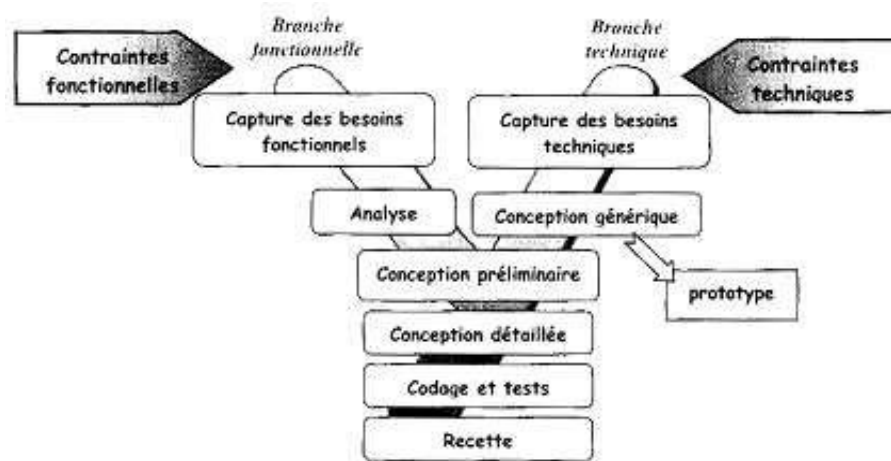


Figure 6: processus 2TUP (source : cours UML 2)

II – DIAGRAMMES INTERVENANTS

II.1 – Diagramme des Cas d'Utilisation

Un diagramme de cas d'utilisation permet de représenter les fonctions du système du point de vue utilisateur. Les cas d'utilisation permettent de structurer les besoins et les objectifs correspondants d'un système en définissant le contour ou le périmètre du système. Les objectifs principaux des diagrammes des cas d'utilisation sont :

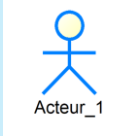
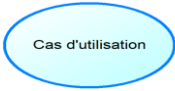

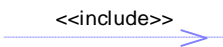
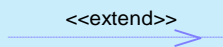
- Fournir une vue de haut niveau de ce que fait le système ;
- Identifier les utilisateurs (acteurs) du système ;


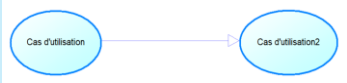

- Déterminer des secteurs nécessitant des interfaces homme-machine.

II.1.1 – Formalisme

Les cas d'utilisation sont représentés par une ellipse contenant le nom dudit cas. Un acteur (représenté par un bonhomme) et un cas d'utilisation sont mis en relation par une association représentée par une ligne. Les éléments de modélisations du diagramme de cas d'utilisation sont présentés dans le tableau ci-contre :

Tableau 12: Eléments de modélisation du diagramme de cas d'utilisation

COMPOSANT DU DIAGRAMME	DESCRIPTIONS	REPRESENTATION GRAPHIQUE
Acteur	Il désigne le rôle joué par toute entité qui interagit avec le système	
Cas d'utilisation	Il correspond à un objectif du système motivé par un besoin d'un ou de plusieurs acteurs	
Association	Elle exprime l'interaction existant entre un acteur et un cas d'utilisation et est représentée par un trait	
Inclusion	Un cas 1 est inclus dans un cas 2 si l'exécution du cas 1 passe obligatoirement par l'exécution du cas 2.	
Extension	On dit qu'un cas 1 étend un autre cas 2 si la réalisation du cas 1 ajoute un ou plusieurs	

COMPOSANT DU DIAGRAMME	DESCRIPTIONS	REPRESENTATION GRAPHIQUE
<i>Généralisation (héritage)</i>	comportements à la réalisation du cas 2.	
	On cas 1 hérite d'un cas 2 si le cas 1 en plus de réaliser complètement les actions du cas 2 réalise ou non d'autres nouvelles actions.	
<i>Le système</i>	On cas 1 hérite d'un cas 2 si le cas 1 en plus de réaliser complètement les actions du cas 2 réalise ou non d'autres nouvelles actions.	
	Il représente le périmètre auquel se rapportent les cas d'utilisation	

II.1.2 – Exemple De Diagramme De Cas D'utilisation

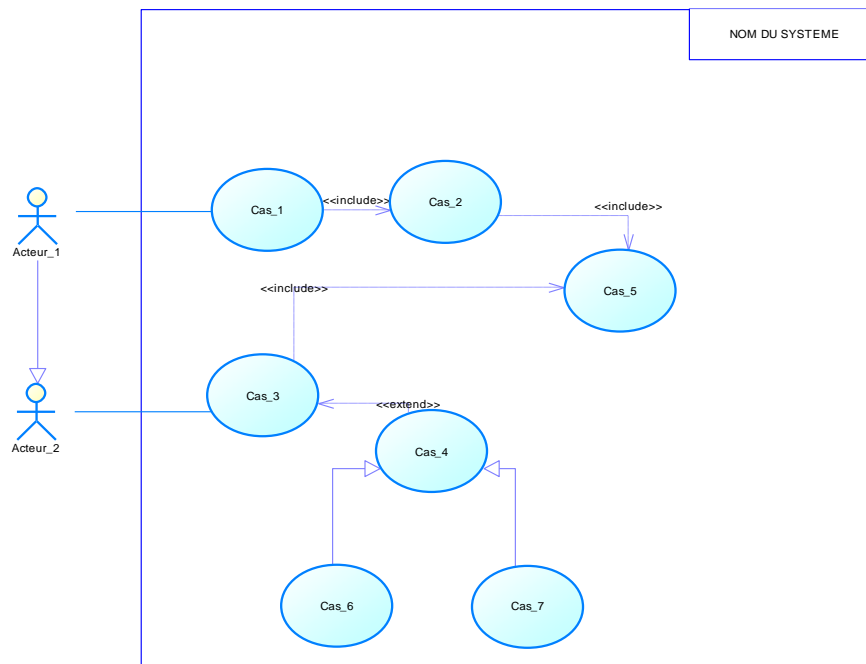


Figure 7: Exemple de diagramme de cas d'utilisation

II.1.3 – DIAGRAMME DE CAS D'UTILISATION DU SYSTEME

Pour une meilleure compréhension, nous présenterons le diagramme de cas d'utilisation principal, qui sera ensuite subdivisé en sous-diagrammes.

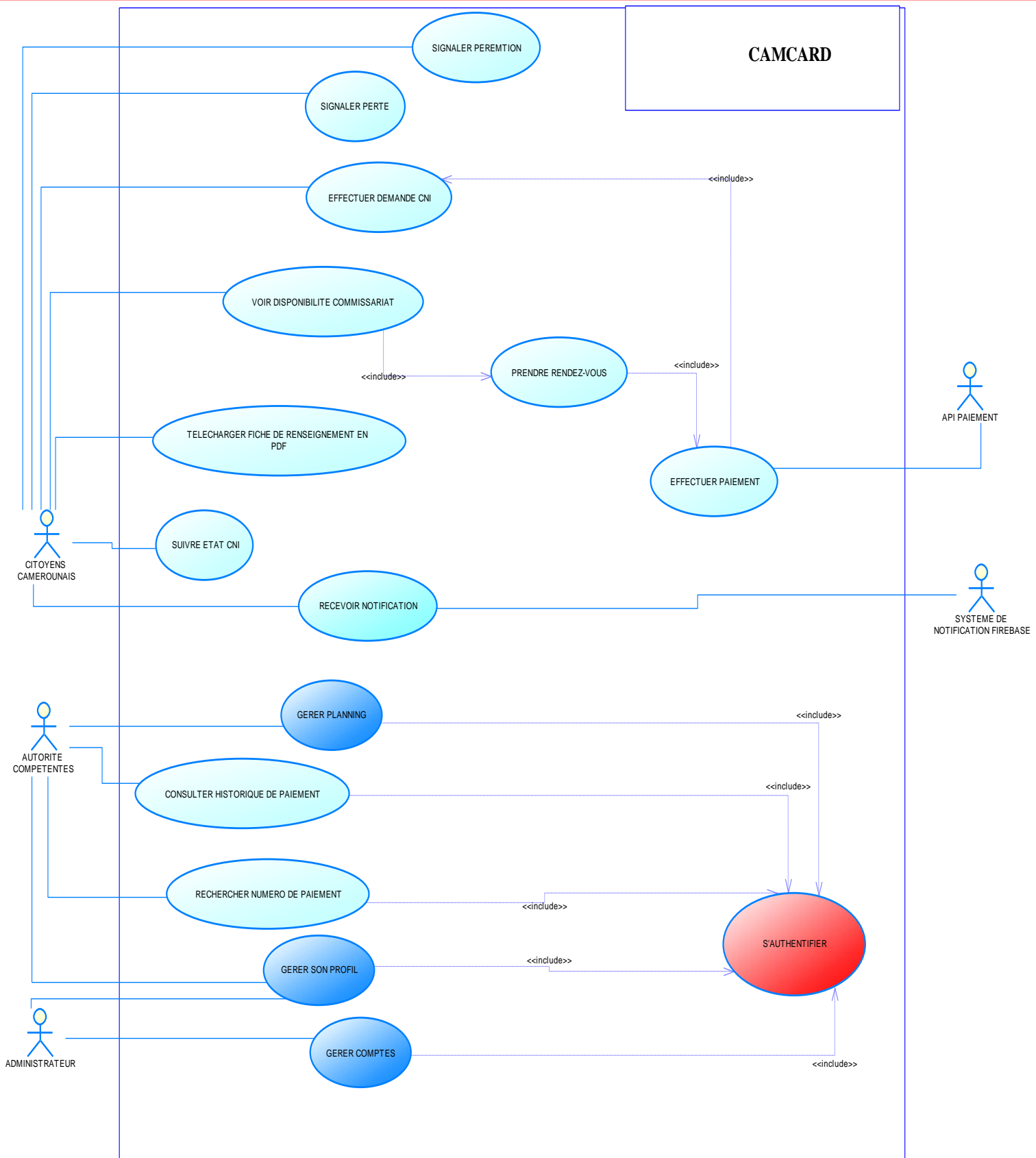


Figure 8:Diagramme de cas d'utilisation global du système

Légende :

- Cas d'utilisation englobant d'autres cas d'utilisation
- Cas d'utilisation élémentaire
- Cas d'utilisation principal

Par la suite nous allons présenter en détails les cas d'utilisation qui englobent d'autre cas d'utilisation.

❖ Diagramme spécifique du cas 'GERER SON PROFIL' :

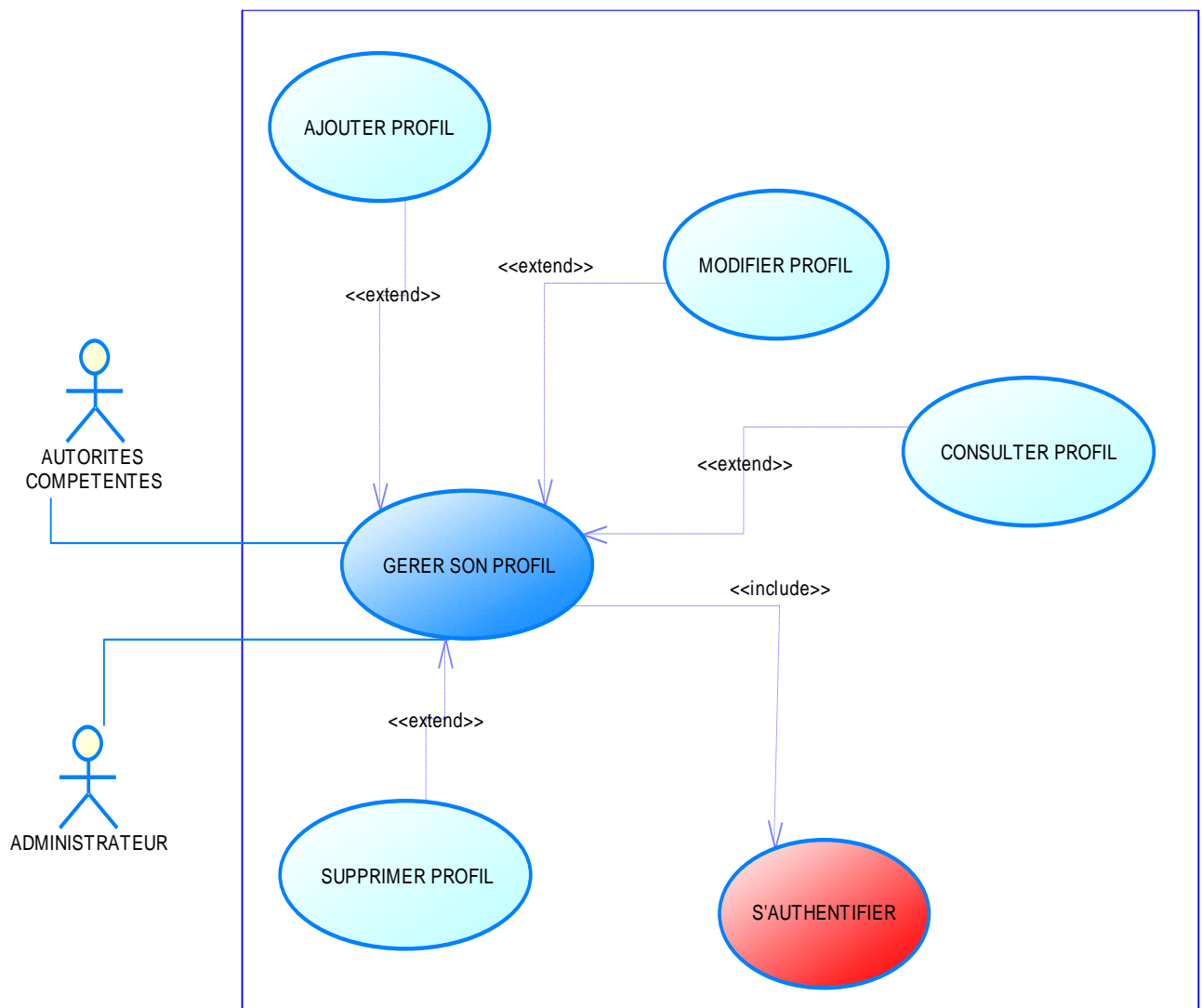


Figure 9: Diagramme spécifique du cas ' GERER SON PROFIL '

❖ Diagramme spécifique du cas 'GERER COMPTES'

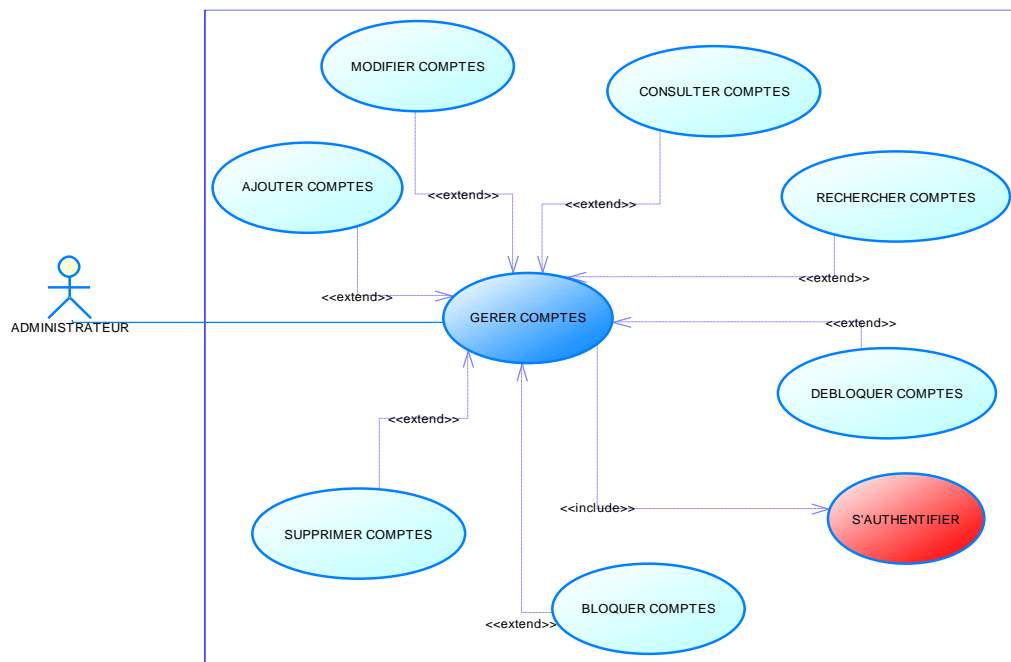


Figure 10: Diagramme spécifique du cas 'GÉRER COMPTES'

❖ Diagramme spécifique du cas 'GERER PLANNING'

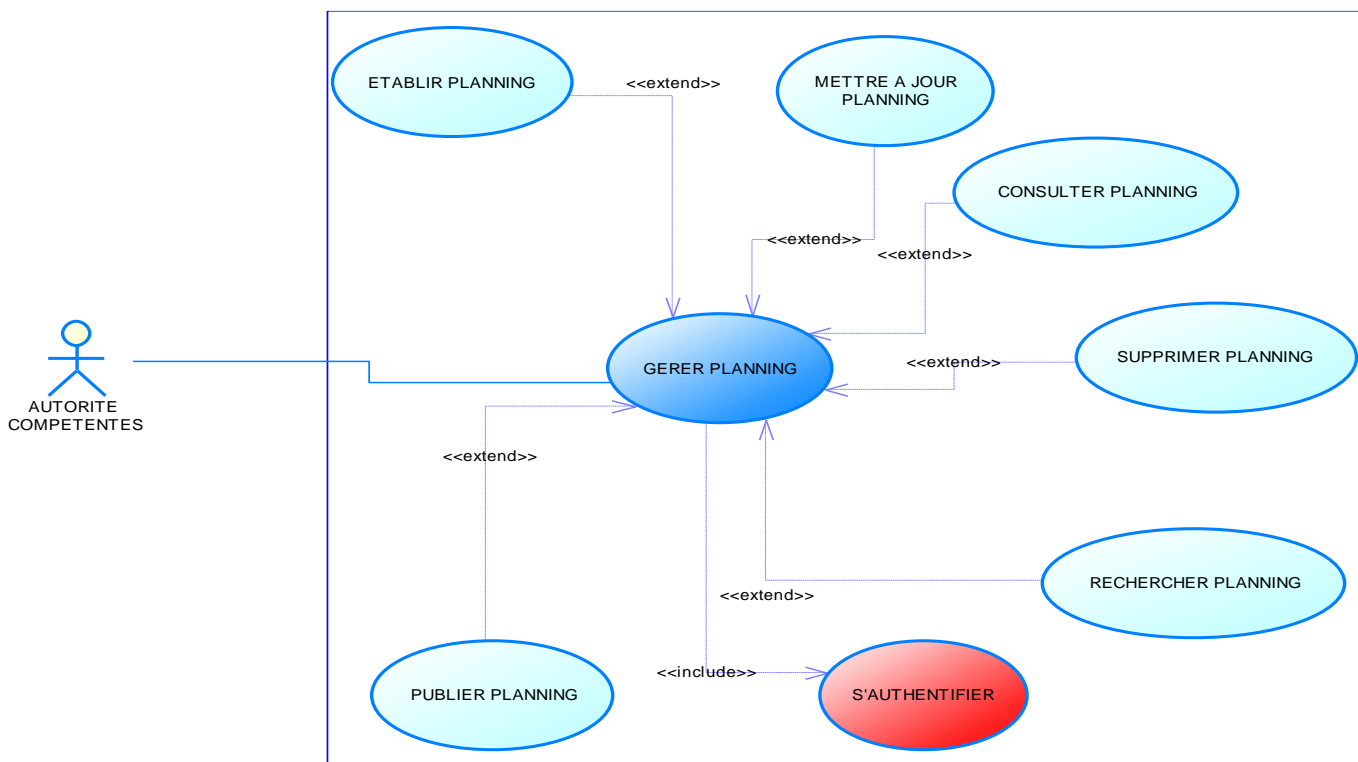


Figure 11: Diagramme spécifique du cas 'GERER PLANNING'

II.1.4 – DESCRIPTION TEXTUELLE DE QUELQUES CAS D'UTILISATION

La description textuelle d'un cas d'utilisation se fait par des scénarii qui définissent la suite logique des actions qui constituent ce cas d'utilisation. Cette description précise ce que fait l'acteur et ce que fait le système, elle clarifie ainsi le déroulement de la fonctionnalité en décrivant la chronologie des actions qui devront être respectées. Celle-ci présente trois principales parties :

Partie I : Identification

- Titre : Nom du cas d'utilisation
- Résumé : description du cas d'utilisation
- Acteur : description des acteurs principaux et secondaires
- Date : date de création et date de mise à jour
- Responsable(s) : Nom du/des responsable(s)
- Version : numéro de la version

Partie II : Description des scénarii

- Les préconditions : état du système avant que le cas d'utilisation ne puisse être déclenché
- Les scénarii : il y a plusieurs types de scénarii à savoir :
 - Le scénario nominal : qui correspond à un déroulement normal d'un cas d'utilisation
 - Les scénarii alternatifs : qui sont des variantes du scénario nominal
 - Les scénarii d'exception : lorsqu'une étape du scénario nominal pourrait être perturbée à cause d'un évènement anormal

- Les postconditions : elles décrivent l'état du système après l'issue de chaque scenario

✚ Partie III : Exigences non fonctionnelles (performance)

- Description textuelle du cas d'utilisation 'S'AUTHENTIFIER'

Tableau 13: Description du cas "S'AUTHENTIFIER"

TITRE	S'AUTHENTIFIER
Résumé	Permet au système de vérifier que chaque utilisateur ait accès à l'interface qui est le sien
Acteurs	Administrateur, autorités compétentes
Date de création	08 Aout 2024
Responsable	ASSAME NNAMA Orlane Sandy
Version	1.0
Description des scenariii	
Présupposé	Démarrer l'application
Précondition	Avoir un compte
Scenario nominal	<ol style="list-style-type: none"> 1. Clique sur l'icône de l'application 2. Le système lui affiche la page d'authentification 3. L'utilisateur saisi son courriel et son mot de passe 4. Le système vérifie la conformité des champs 5. Le système envoi la requête a la base de données 6. La base de données analyse la requête 7. La base de données envoi le résultat de l'analyse 8. Le système envoie la page de travail
Scenarii alternatifs	4.a. A l'étape 4 du scenario nominal,

<i>TITRE</i>	<i>S'AUTHTENTIFIER</i>
	l'utilisateur a entré des informations non correspondantes ou manquantes 4.b. Le système renvoie un message d'erreur puis retourne à l'étape 3 du scenario nominal 7.a. A l'épate 7 du scenario nominal la base données envoie le résultat de l'analyse 'compte inexistant' 7.b. Le système renvoie un message d'erreur puis retourne à l'étape 2 du scenario nominal
<i>Post condition de succès</i>	Il a réussi à s'authentifier
<i>Post condition d'échec</i>	L'utilisateur n'a pas réussi à se connecter à son compte

- Description textuelle du cas 'EFFECTUER DEMANDE CNI'

Tableau 14: Description textuelle du cas 'EFFECTUER DEMANDE CNI'

<i>TITRE</i>	<i>EFFECTUER DEMANDE CNI</i>
<i>Résumé</i>	Permet à l'utilisateur d'effectuer une demande de son choix
<i>Acteurs</i>	Citoyens Camerounais
<i>Date de création</i>	08 Aout 2024
<i>Responsable</i>	ASSAME NNAMA Orlane Sandy
<i>Version</i>	1.0
<i>Description des scenariii</i>	
<i>Présupposé</i>	Démarrer l'application
<i>Précondition</i>	L'utilisateur possède un numéro de téléphone
<i>Scenario nominal</i>	<ol style="list-style-type: none"> 1. L'utilisateur clique sur l'icône de l'application 2. Le système affiche la page d'identification 3. L'utilisateur entre son numéro de

TITRE

EFFECTUER DEMANDE CNI

	<p>téléphone</p> <ol style="list-style-type: none"> Le système vérifie la conformité du champ. Le système envoie un code de vérification à l'utilisateur L'utilisateur remplit le champ avec le code reçu Le système vérifie le code entré par l'utilisateur Le système lui affiche la page de demande L'utilisateur clique sur le bouton 'effectuer demande de cni' Le système lui affiche des formulaires à remplir L'utilisateur remplit les champs des formulaires Le système vérifie la conformité des champs Le système lui affiche ses informations remplies L'utilisateur clique sur 'VALIDER' Le système envoie une requête à la base de données La base de données enregistre le formulaire La base de données envoie un résultat au système Le système affiche la page de paiement
<i>Scenarii alternatifs</i>	<ol style="list-style-type: none"> A l'étape 4 du scénario nominal, l'utilisateur a entré des informations non correspondantes ou manquantes. Le système renvoie un message d'erreur puis retourne à l'étape 3 du scénario nominal

TITRE	EFFECTUER DEMANDE CNI
	<p>5.a A l'étape 5 du scenario nominal, l'utilisateur n'a pas reçu de numéro,</p> <p>5.b L'utilisateur retourne à l'étape 3 du scenario nominal.</p> <p>6.a A l'étape 6 du scenario nominal, l'utilisateur a entré un code non correspondant.</p> <p>6.b Le système renvoie un message d'erreur puis retourne à l'étape 3 du scenario nominal</p> <p>12.a A l'étape 12 du scenario nominal, l'utilisateur a entré des informations non correspondantes ou manquantes</p> <p>12.b. Le système renvoie un message d'erreur puis retourne à l'étape 11 du scenario nominal</p>
Post condition de succès	L'utilisateur réussi à effectuer sa demande
Post condition d'échec	L'utilisateur n'a pas réussi à effectuer sa demande

- Description textuelle du cas 'EFFECTUER PAIEMENT'

Tableau 15: Description textuelle du cas "EFFECTUER PAIEMENT"

TITRE	EFFECTUER PAIEMENT
Résumé	Permet à l'utilisateur de créer sa cni
Acteurs	Citoyens camerounais
Date de création	08 Aout 2024
Responsable	ASSAME NNAMA Orlane Sandy
Version	1.0
Description des scenari	
Présupposé	Démarrer l'application
Précondition	Avoir effectuer une demande de cni

TITRE	EFFECTUER PAIEMENT
<i>Scenario nominal</i>	<ol style="list-style-type: none"> 1. L'utilisateur clique sur le moyen de paiement. 2. Le système affiche le formulaire à remplir 3. L'utilisateur entre son numéro de paiement 4. Le système vérifie la conformité des champs 5. Le système envoie le numéro de paiement à l'API 6. L'API de paiement vérifie le solde de l'utilisateur 7. L'API effectue le traitement 8. L'API envoie un message système 9. Le système envoie une requête à la base de données 10. La base de données enregistre le paiement 11. Le système affiche la page de prise de rendez-vous à l'utilisateur.
<i>Scenario alternatif</i>	<p>4.a A l'étape 4 du scenario nominal, l'utilisateur a entré des informations non correspondantes ou manquantes.</p> <p>4.b. Le système renvoie un message d'erreur puis retourne à l'étape 3 du scenario nominal</p> <p>6.a A l'étape 6 du scenario nominal, l'API de paiement signale le solde de l'utilisateur insuffisant.</p> <p>6.b Le système renvoie l'utilisateur à l'étape 3.</p>
<i>Post condition de succès</i>	L'utilisateur réussi à effectuer un paiement
<i>Post condition d'échec</i>	L'utilisateur n'a pas réussi à effectuer un paiement

- Description textuelle du cas 'ETABLIR PLANNING'

Tableau 16: Description textuelle du cas "ETABLIR PLANNING"

<i>TITRE</i>	<i>ETABLIR PLANNING</i>
<i>Résumé</i>	Permet aux autorités compétentes de publier leur planning pour permettre aux utilisateurs de voir leur disponibilité
<i>Acteurs</i>	Autorités compétentes
<i>Date de création</i>	08 Aout 2024
<i>Responsable</i>	ASSAME NNAMA Orlane Sandy
<i>Version</i>	1.0
<i>Description des scenarii</i>	
<i>Présupposé</i>	Démarrer l'application
<i>Précondition</i>	L'utilisateur s'est authentifié
<i>Scenario nominal</i>	<ol style="list-style-type: none"> 1. L'utilisateur clique sur 'établir planning' 2. Le système lui affiche un tableau avec des champs à remplir 3. L'utilisateur remplit les champs avec les jours et les horaires de leur disponibilité 4. L'utilisateur clique sur 'publier planning' 5. La base de données enregistre le planning 6. Le système affiche la page d'accueil.
<i>Scenario alternatif</i>	<p>3.a A l'étape 3 du scenario nominal l'utilisateur a entré des informations non correspondantes ou manquantes</p> <p>3.b. Le système renvoie un message d'erreur puis retourne à l'étape 2 du scenario nominal</p>
<i>Post condition de succès</i>	L'utilisateur réussi à établir un planning

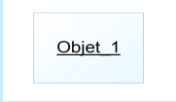

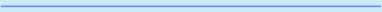
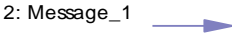
II.2 – Diagramme de Communication

Un diagramme de communication dans le langage de modélisation unifié fait référence à un graphique qui représente le flux de messages dans un système. En un mot, il montre comment les parties d'un système interagissent ou, dans ce cas, communiquent entre elles.

II.2.1 – Formalisme

Comme tout autre diagramme, le diagramme de communication UML comporte également plusieurs composants qui constituent son intégralité. La bonne chose à propos de ce schéma est qu'il peut être réalisé avec seulement quelques composants.

Tableau 17: Elément de modélisation d'un diagramme de communication

Composants	Descriptions	Représentations
<i>Objet</i>	Il est représenté par un rectangle contenant le nom et le type de l'instance.	
<i>Acteur</i>	Entité externe qui interagit avec le système.	
<i>Connecteur</i>	Les relations entre les lignes de vie sont appelées connecteurs et se représentent par un trait plein reliant deux lignes de vie.	
<i>Message</i>	Désignent une communication particulière entre les lignes de vie, et sont généralement ordonnés selon un ordre de numéro croissant.	

II.2.2 – Exemple de diagramme de communication



Figure 12: Exemple de diagramme de communication

II.2.2 – Quelques diagrammes de communication du système

❖ Diagramme de Communication du cas 'EFFECTUER PAIEMENT'

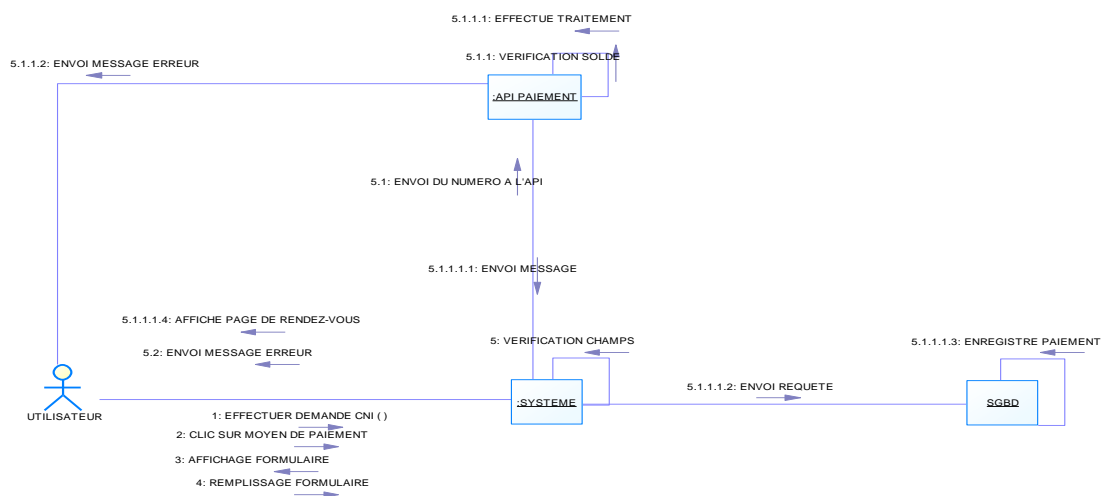


Figure 13: diagramme de communication du cas 'EFFECTUER PAIEMENT'

❖ Diagramme De Communication Du Cas 'S'AUTHTENTIFIER'

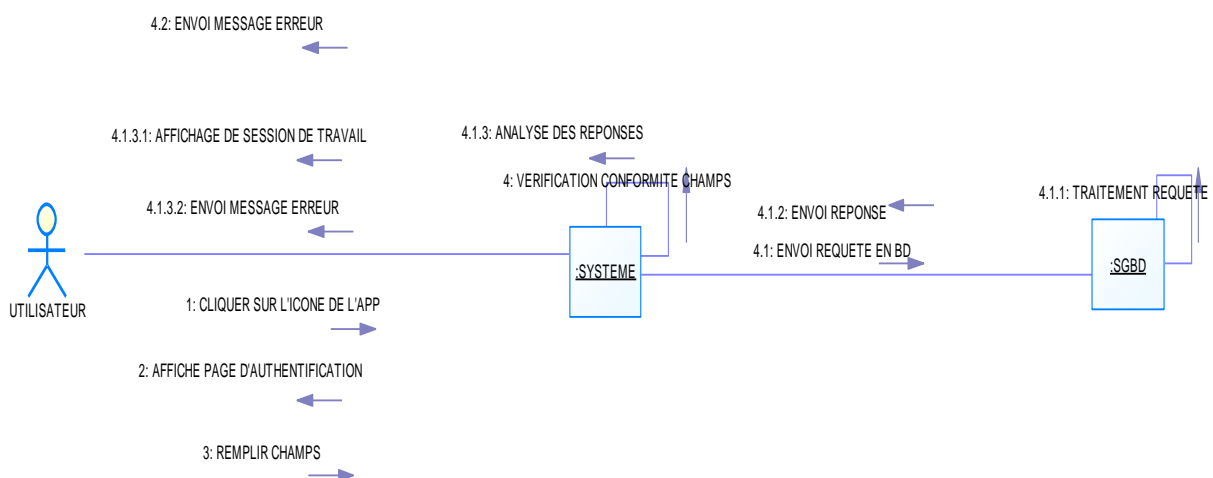


Figure 14: diagramme de communication du cas 'S'AUTHTENTIFIER'

❖ Diagramme De Communication Du Cas 'EFFECTUER DEMANDE CNI'

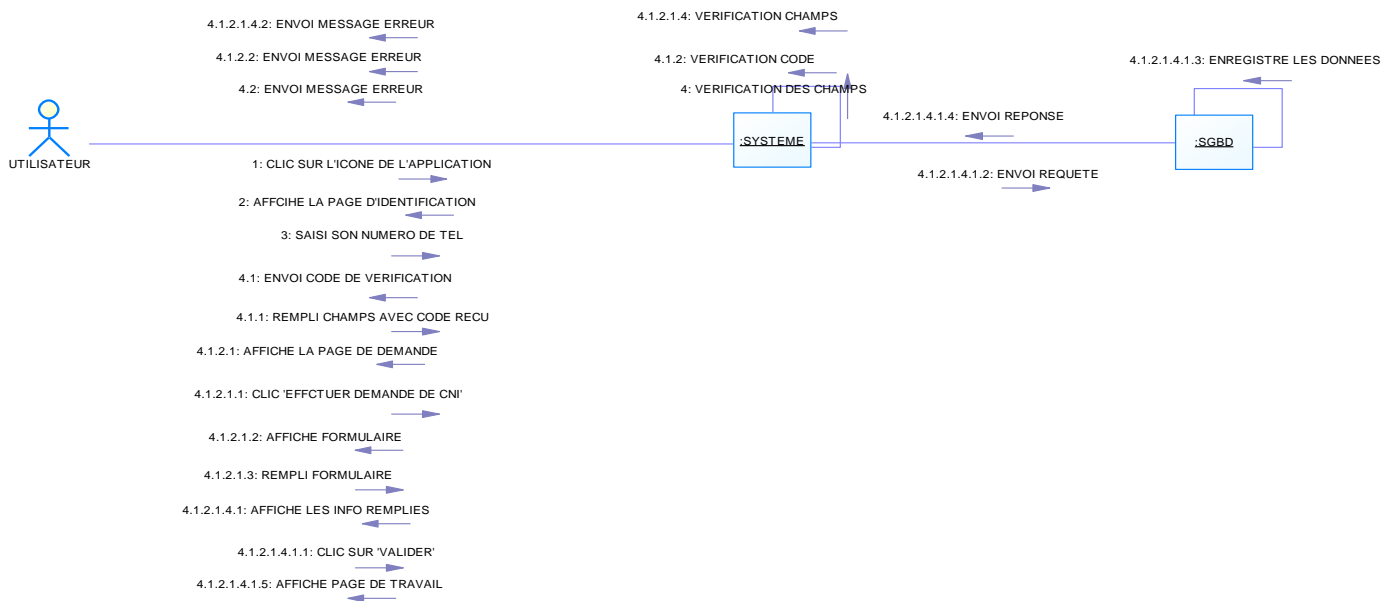


Figure 15: diagramme de communication du cas 'EFFECTUER DEMANDE CNI'

❖ Diagramme de Communication du cas 'ETABLIR PLANNING'

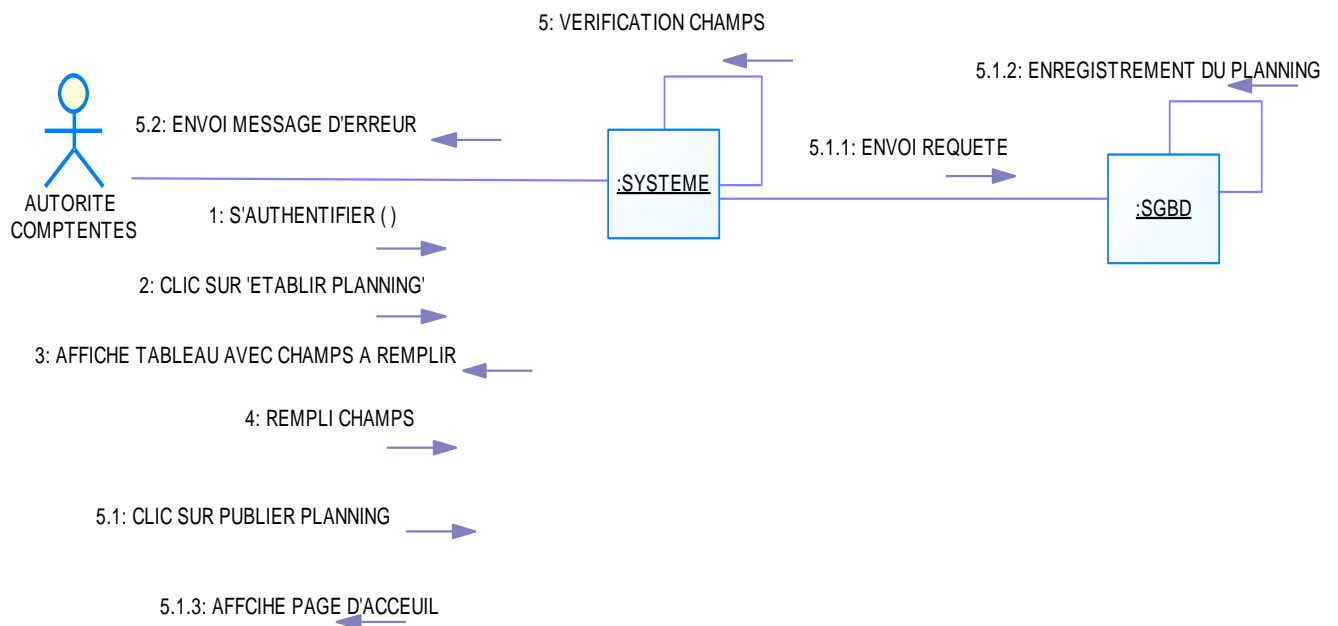


Figure 16: diagramme de communication du cas 'ETABLIR PLANNING'

II.3 – Diagramme de Séquence

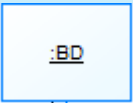

Les diagrammes de séquences sont des représentations graphiques des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. Il a pour buts de :



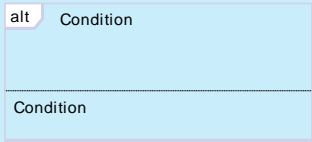
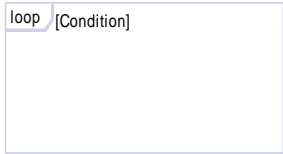
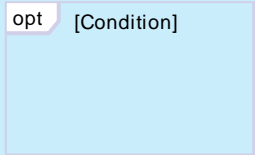
- Représenter les détails d'un cas d'utilisation ;
- Modéliser le déroulement logique d'une procédure, fonctions ou opérations complexes ;
- Voir comment les objets et les composants interagissent entre eux pour effectuer un processus ;
- Schématiser et comprendre le fonctionnement détaillé d'un scenario

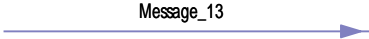
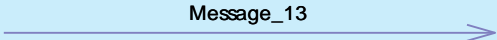
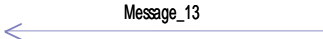

II.3.1 – Formalisme

Le diagramme de séquence énumère les objets horizontalement et le temps verticalement. Il modélise l'exécution des différents modèles en fonction du temps. Dans ce diagramme, les objets et les acteurs sont énumérés en colonnes avec leur ligne de vie verticale indiquant la durée de vie de l'objet. Les éléments modélisant un diagramme de séquence sont indiqués dans le tableau ci-contre :

Tableau 18: Tableau des éléments du diagramme de sequence

Composant	Description	Représentation
Objet	Se décrit comme étant l'instance d'une classe	
Acteur	Il s'agit d'une personne qui interagit et communique avec le système.	

Composant	Description	Représentation
<i>Ligne de vie</i>	La ligne de vie, représente le déroulement temporel d'un processus.	
<i>Activation</i>	Il s'agit ici du temps nécessaire pour qu'un objet ou un acteur accomplisse une tâche, elle indique quand l'objet effectue une action.	
<i>Fragment d'interaction</i>	Il représente une interaction sous forme de condition dans laquelle tout se passe bien ou dans le cas où il y'a des erreurs	
	Il représente une interaction sous forme de boucle	
	Il représente une articulation d'interaction et est défini par un operateur et des opérande	
<i>Message (indique les communications entre</i>	<p>Les messages</p> <p>Synchrones : la réception d'un message synchrone</p>	

Composant	Description	Représentation
objets)	doit provoquer chez le destinataire le lancement d'une de ses méthodes ou opérations. L'expéditeur du message reste bloqué pendant toute l'exécution de la méthode et attend donc la fin de celle-ci avant de pouvoir lancer un nouveau message.	
	Les messages asynchrones : ils ne nécessitent pas de réponse du destinataire, avant que l'expéditeur ne continue	
	Les messages de retour : ils représentent la réponse à un message synchrone.	
	Les messages réflexifs : ils représentent les traitements que les objets effectuent sur eux même.	

II.3.2 – Exemple de diagramme de sequence

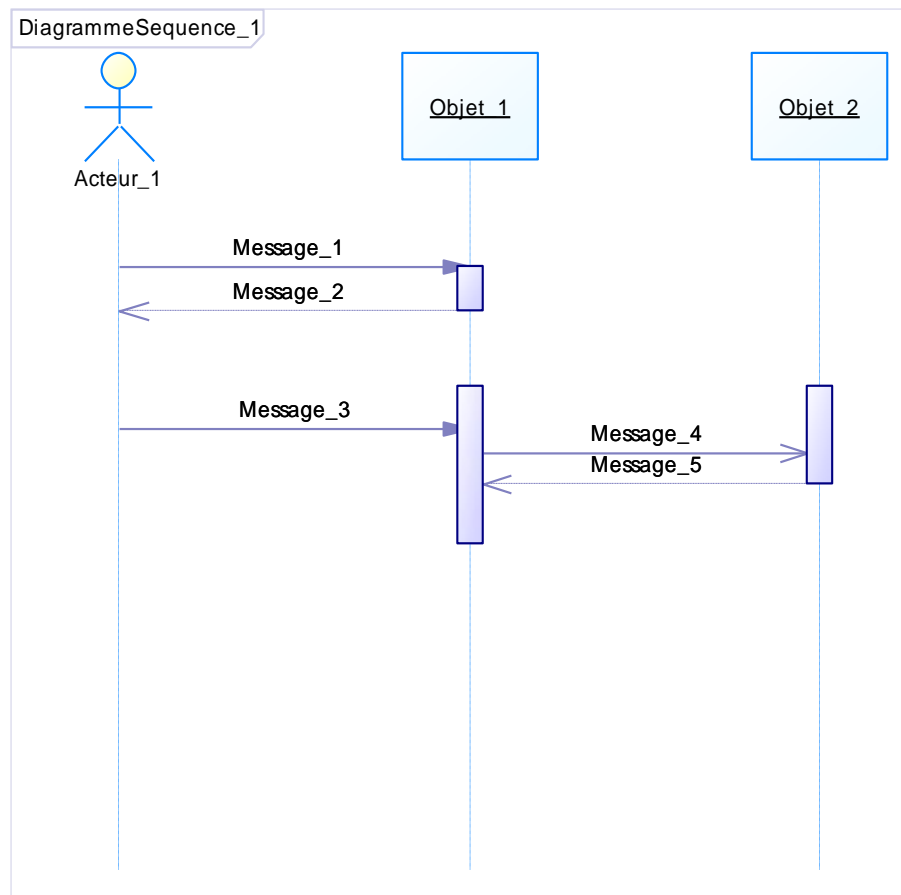


Figure 17: Exemple de diagramme de sequence

II.3.3 – Quelques diagrammes de sequence du système

❖ Diagramme de sequence du cas « S'AUTHTENTIFIER »

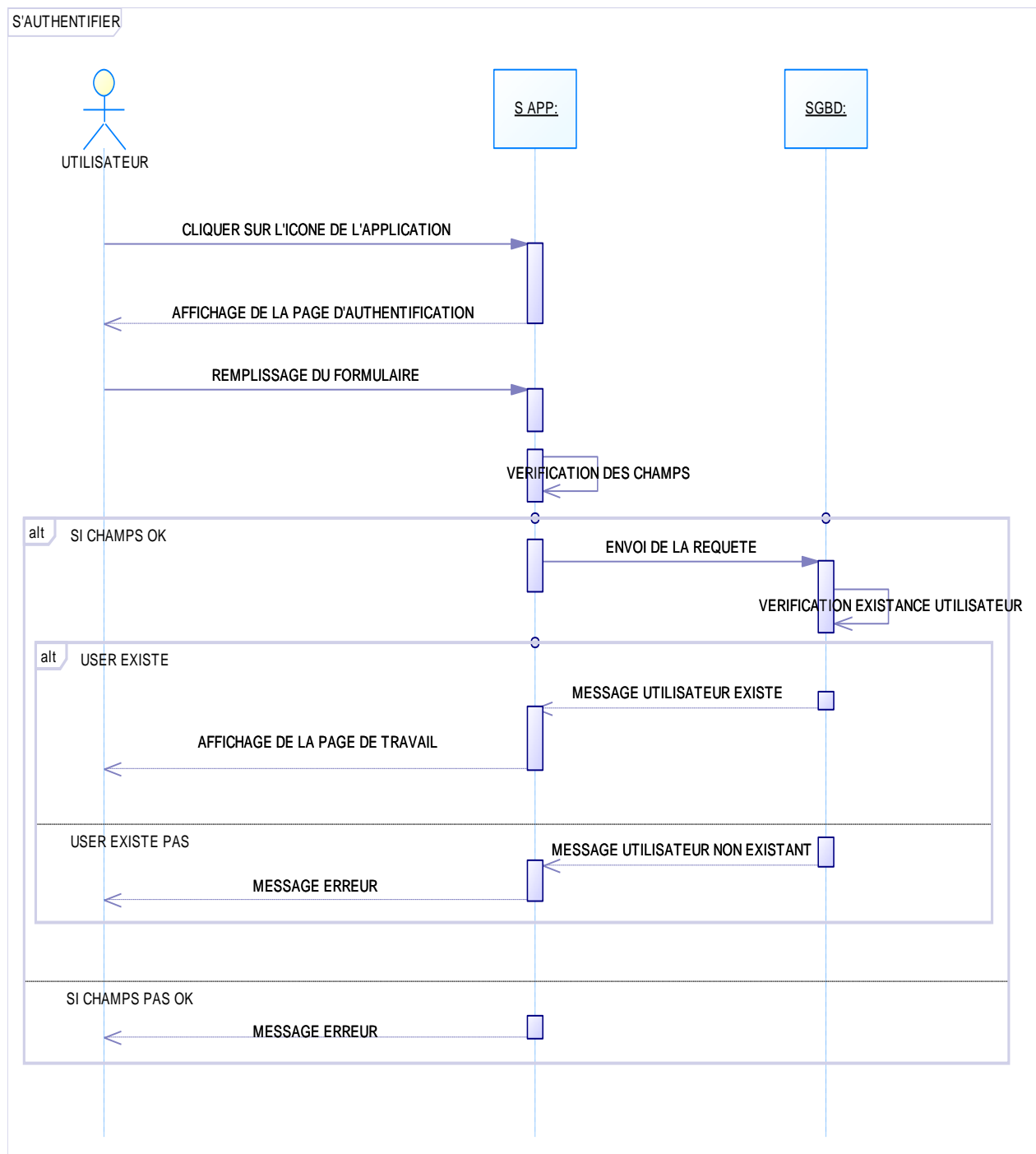


Figure 18: Diagramme de cas d'utilisation 'S'AUTHTENTIFIER'

❖ Diagramme de sequence du cas « EFFECTUER DEMANDE CNI »

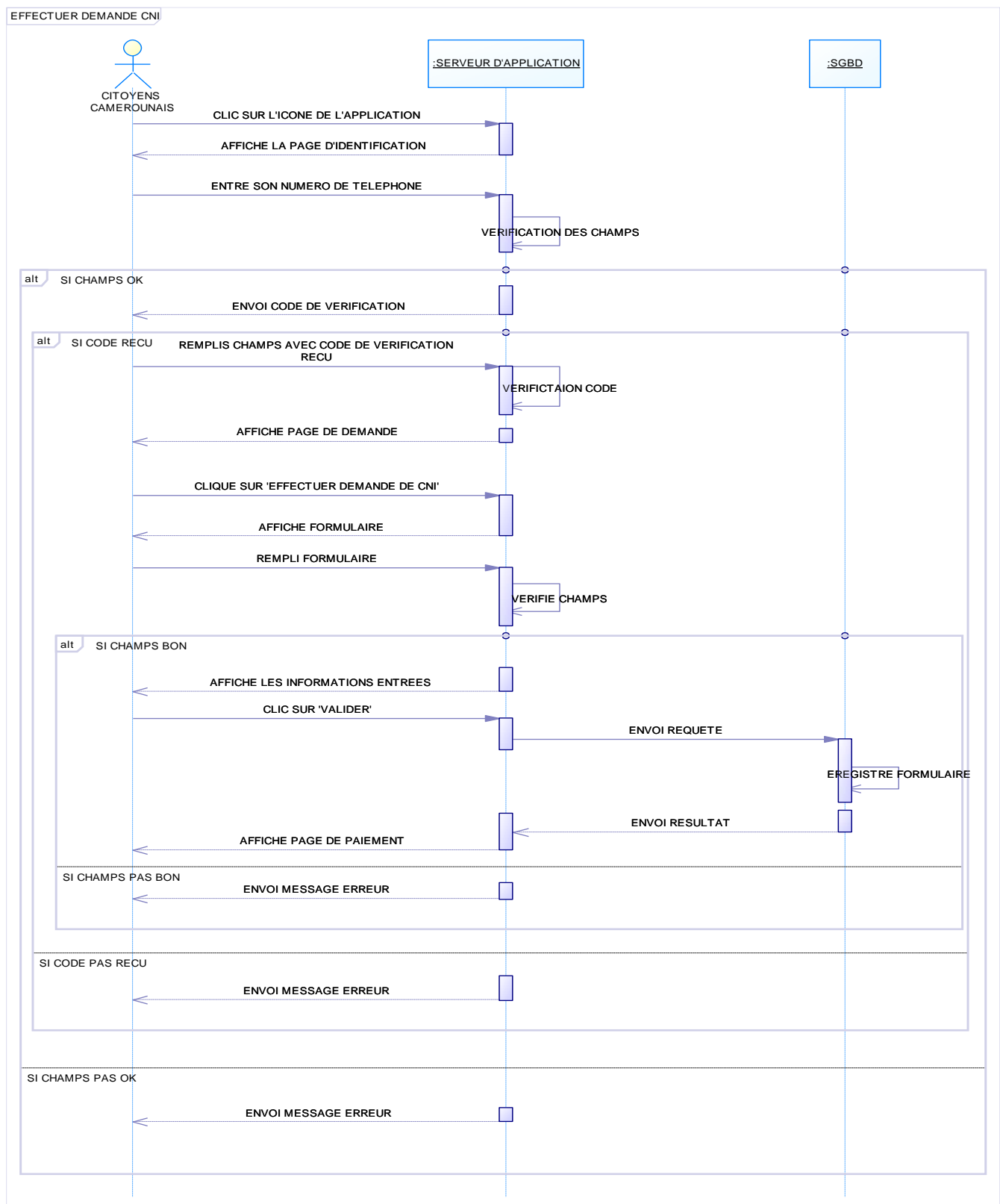


Figure 19: Diagramme de sequence du cas 'EFFECTUER DEMANDE CNI'

❖ Diagramme de sequence du cas « EFFECTUER PAIEMENT »

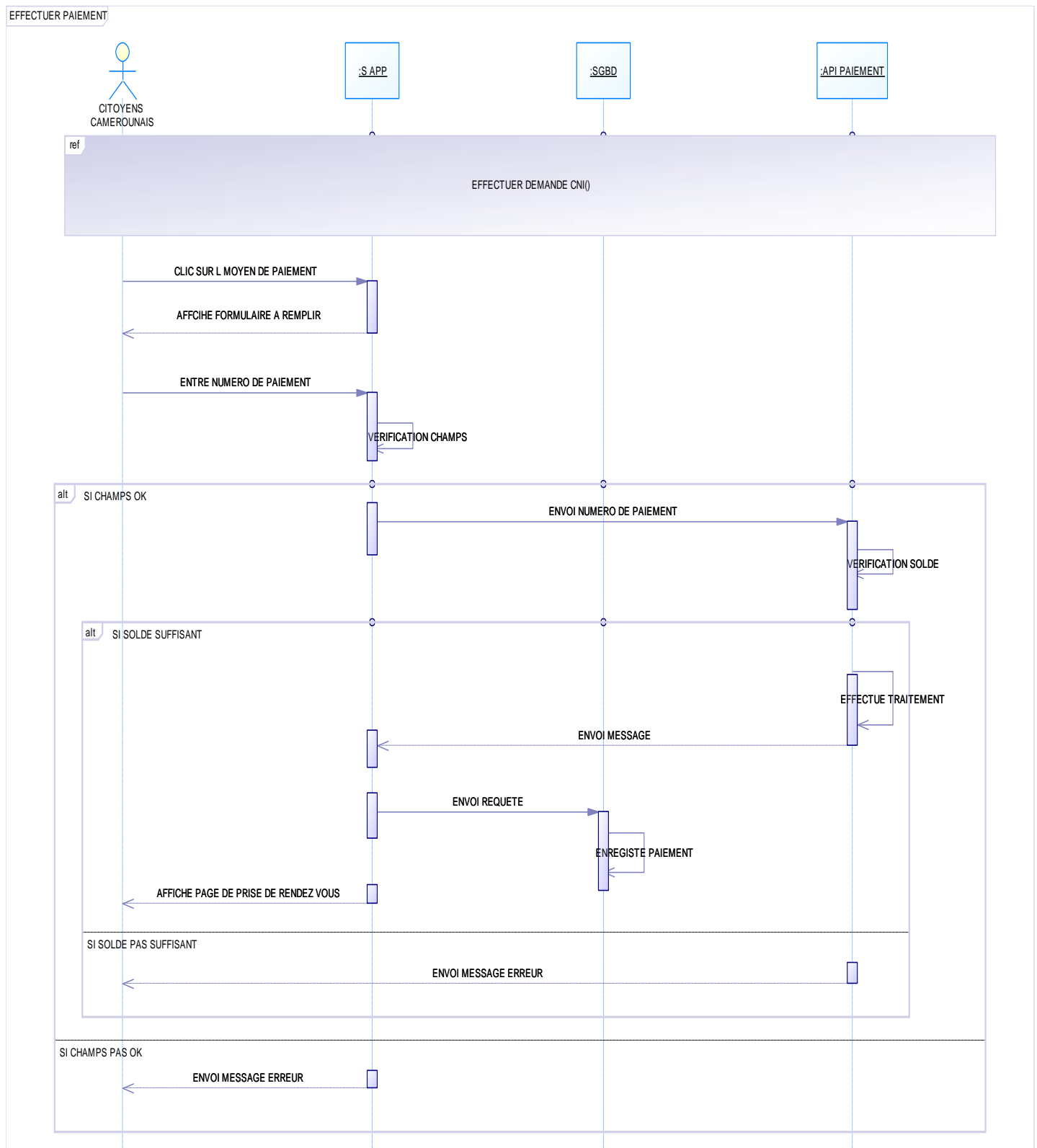


Figure 20 : Diagramme de sequence du cas 'EFFECTUER PAIEMENT'

❖ Diagramme de sequence du cas 'ETABLIR PLANNING'

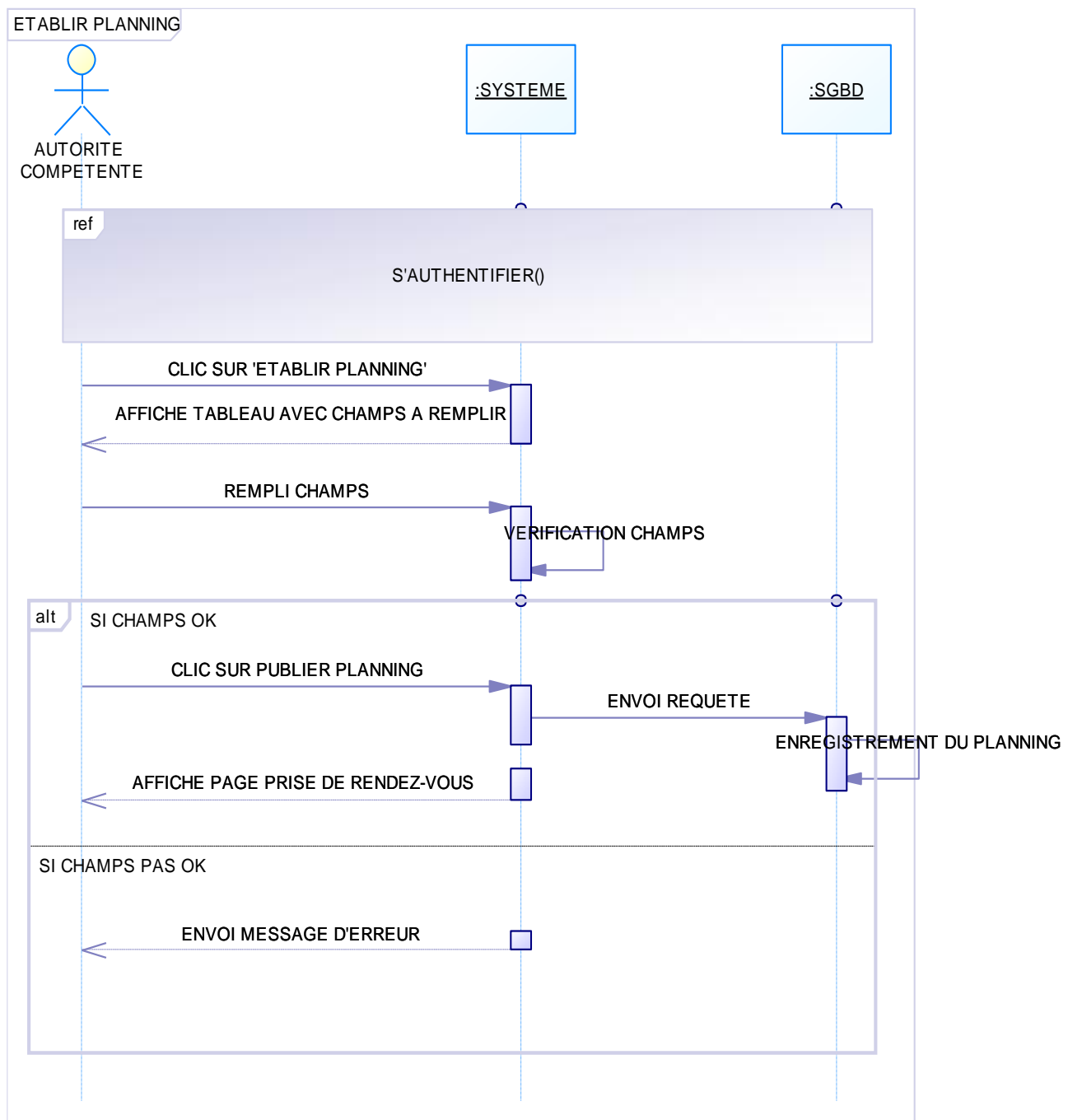


Figure 21: Diagramme de sequence du cas 'ETABLIR PLANNING'

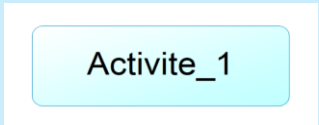

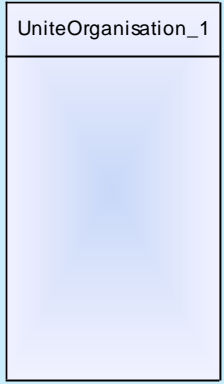
II.4 – DIAGRAMME D'ACTIVITE





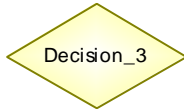
Les diagrammes d'activités sont particulièrement adaptés à la description des cas d'utilisation. Plus précisément, ils viennent illustrer et consolider la description textuelle des cas d'utilisation.

II.4.1 – Formalisme

L'usage général du diagramme d'activité permet de mettre l'accent sur les traitements et de formaliser graphiquement la séquence d'actions réalisées dans un cas d'utilisation.

Tableau 19: Eléments de modélisation du diagramme d'activité

Composant	Description	Représentation
Activité	Une activité marque une action faite par un objet. Elle est représentée par un rectangle arrondi.	
Transitions ou Flux	Quand un état d'activité est accompli, le traitement passe à un autre état d'activité. Les transitions sont utilisées pour marquer ce passage. Les transitions sont modélisées par des flèches.	
Couloir	Le diagramme d'activités fait intervenir les acteurs de chaque activité. Chaque activité sera placée dans une colonne (couloir) qui correspond à l'acteur. Les objets sont énumérés au-dessus de la colonne, et les barres verticales séparent les colonnes pour former les couloirs d'activités.	

Composant	Description	Représentation
Etat	L'état initial marque le point d'entrée la première activité. Il est représenté, comme dans le diagramme d'état, par un cercle plein. Il ne peut y avoir qu'un seul état initial sur un diagramme.	
	L'état final marque la fin du déroulement des opérations modélisées. Il peut y avoir des états finaux multiples sur un diagramme. Ils sont représentés par un cercle plein entouré d'un autre cercle.	
	Nœud de fin d'activité est le nœud marquant la fin de l'exécution de toute l'activité	
	Souvent, certaines activités peuvent être faites en parallèle. Pour dédoubler le traitement "Fork", ou le reprendre quand des activités multiples ont été accomplies ("join"), des barres de synchronisation sont utilisées. Celles-ci sont modélisées par des rectangles pleins, avec des transitions multiples entrantes ou sortantes.	
Barre de synchronisation		
Décision	Indique le flux de contrôle d'activité	

II.4.2 – exemple de diagramme d'activité

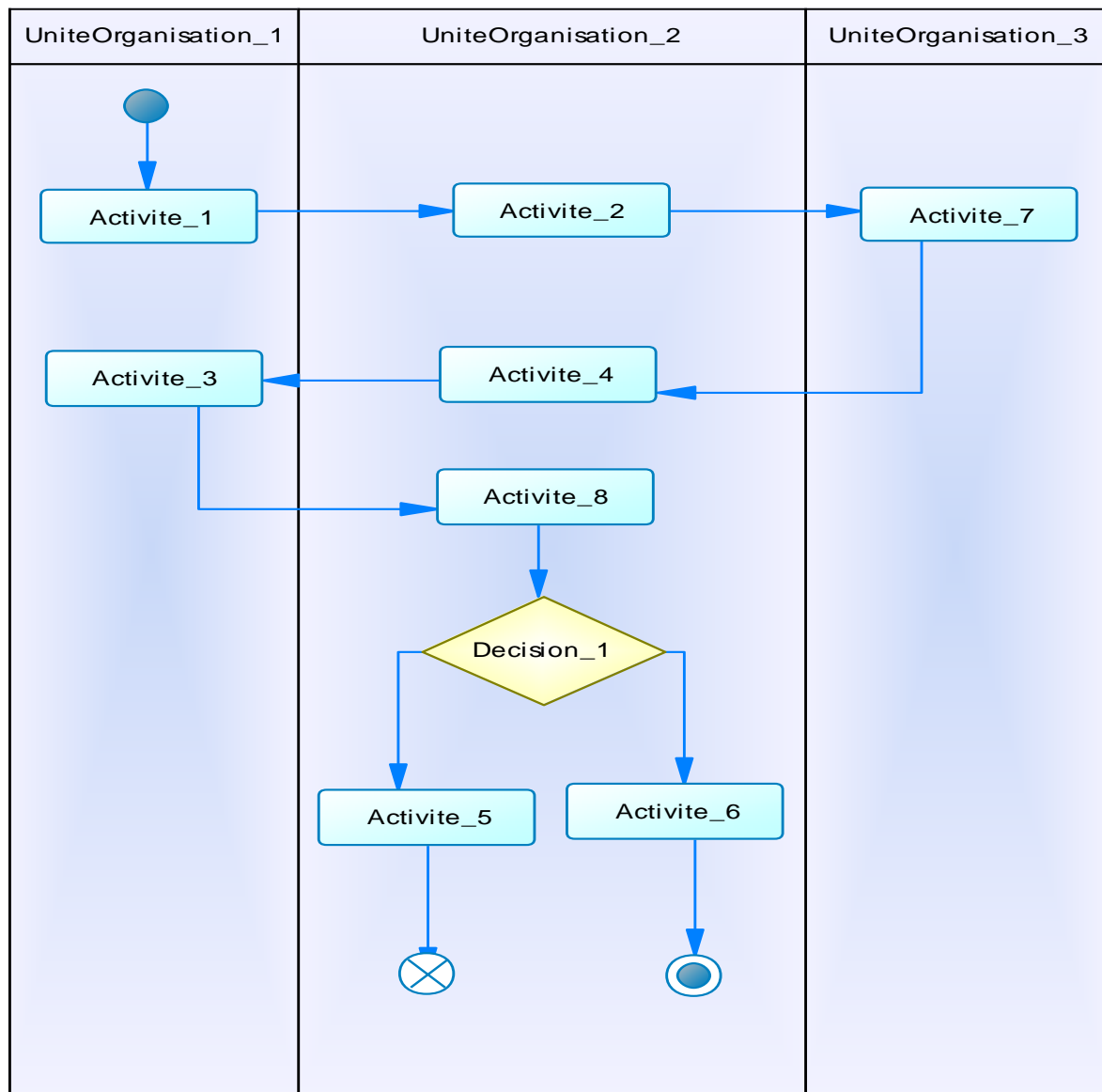


Figure 22: exemple de diagramme de sequence

II.4.3 – Quelques diagrammes d'activité du système

❖ Diagramme d'activité du cas d'utilisation « S'AUTHTENTIFIER »

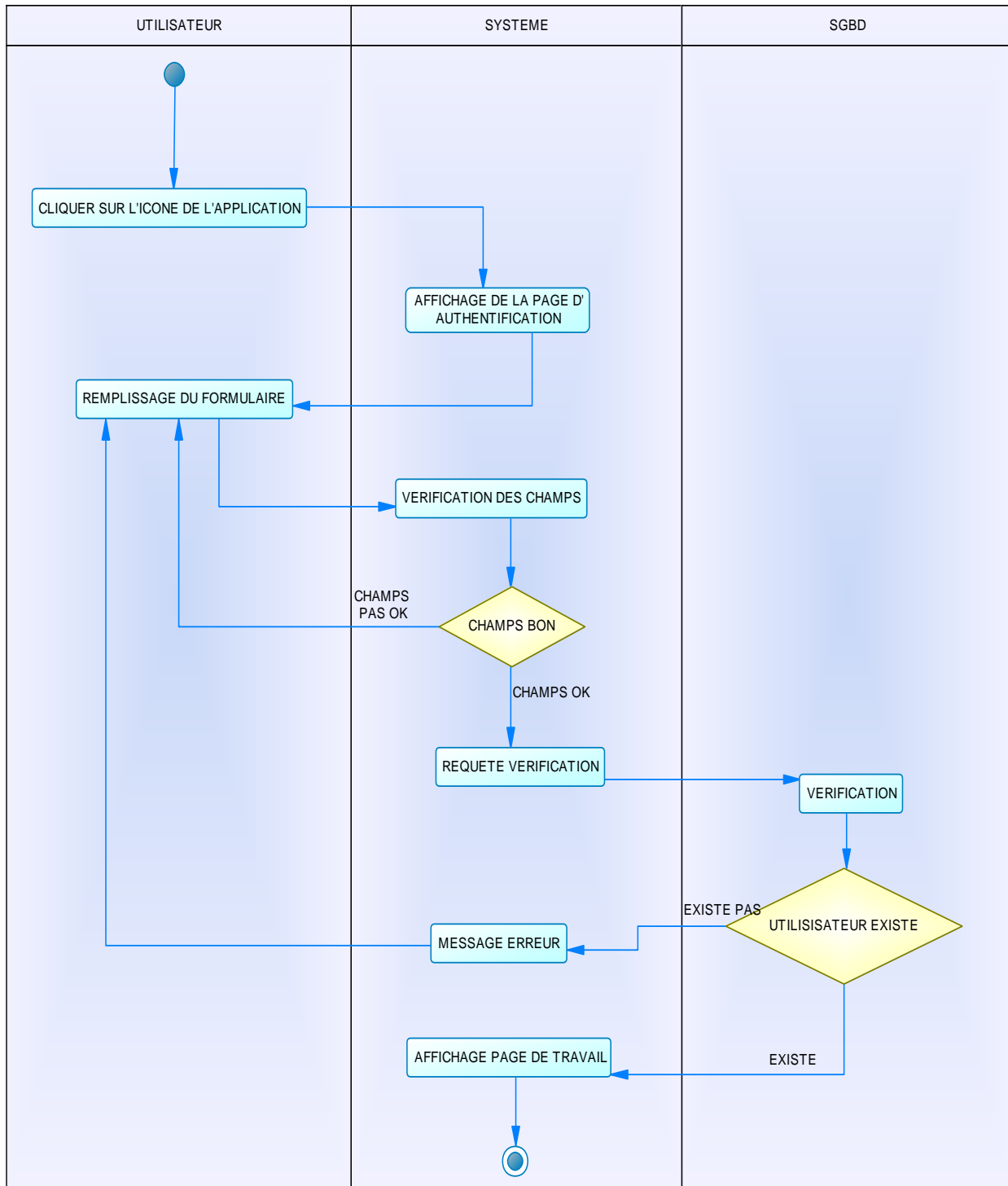


Figure 23: Diagramme d'activité du cas 'S'AUTHTENTIFIER'

❖ Diagramme d'activité du cas 'EFFECTUER DEMANDE CNI'

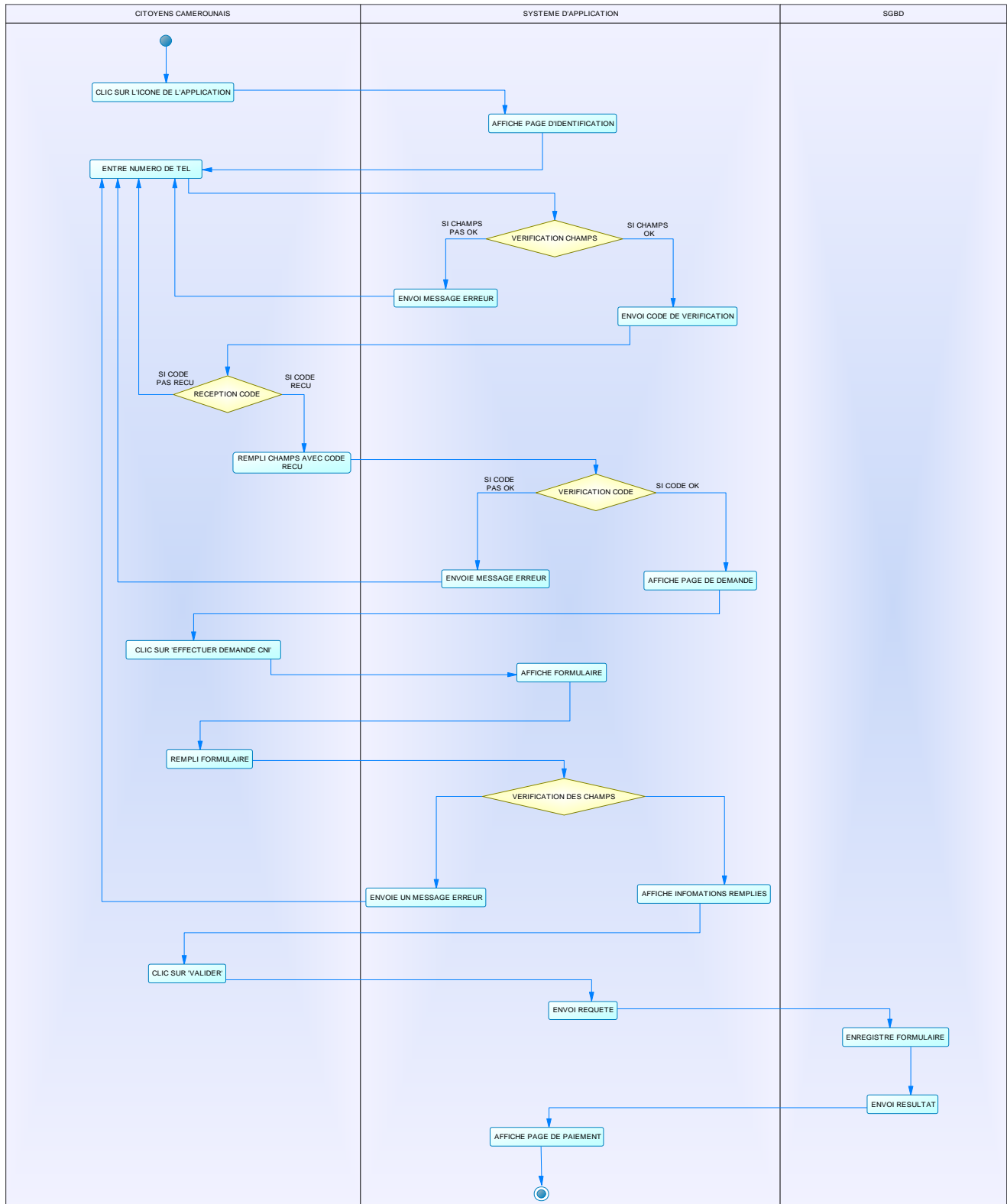


Figure 24: Diagramme d'activité du cas 'EFFECTUER DEMANDE CNI'

Rédigé par **ASSAME NNAMA Orlane Sandy**, élève ingénieur en Deuxième année, option Génie Logiciel,

IAI-CAMEROUN, Centre d'Excellence Technologique PAUL BIYA

Année académique 2023-2024

❖ Diagramme d'activité du cas 'EFFECTUER PAIEMENT'

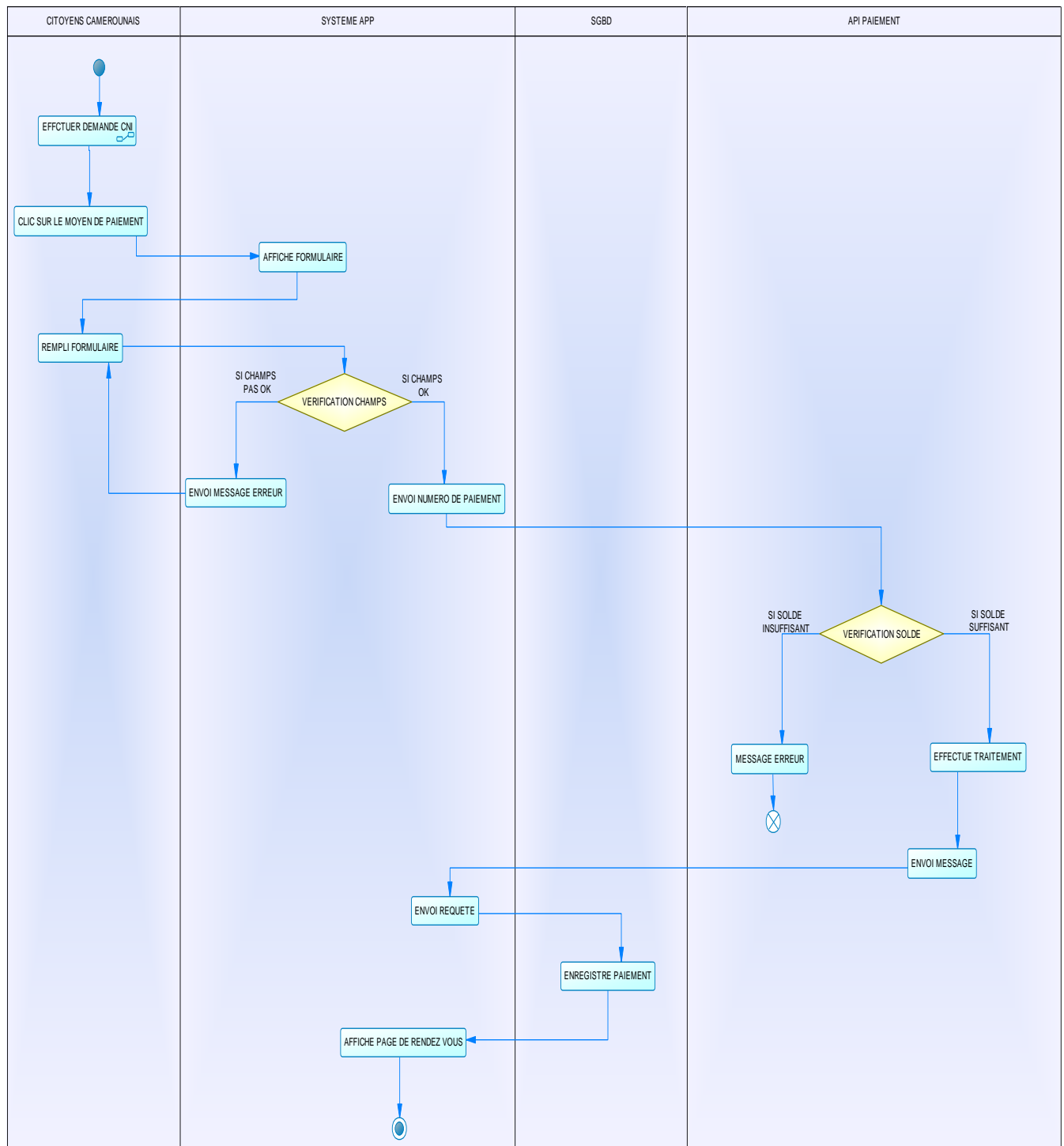


Figure 25: Diagramme d'activité du cas 'EFFECTUER PAIEMENT'

❖ Diagramme d'activité du cas 'ETABLIR PLANNING'

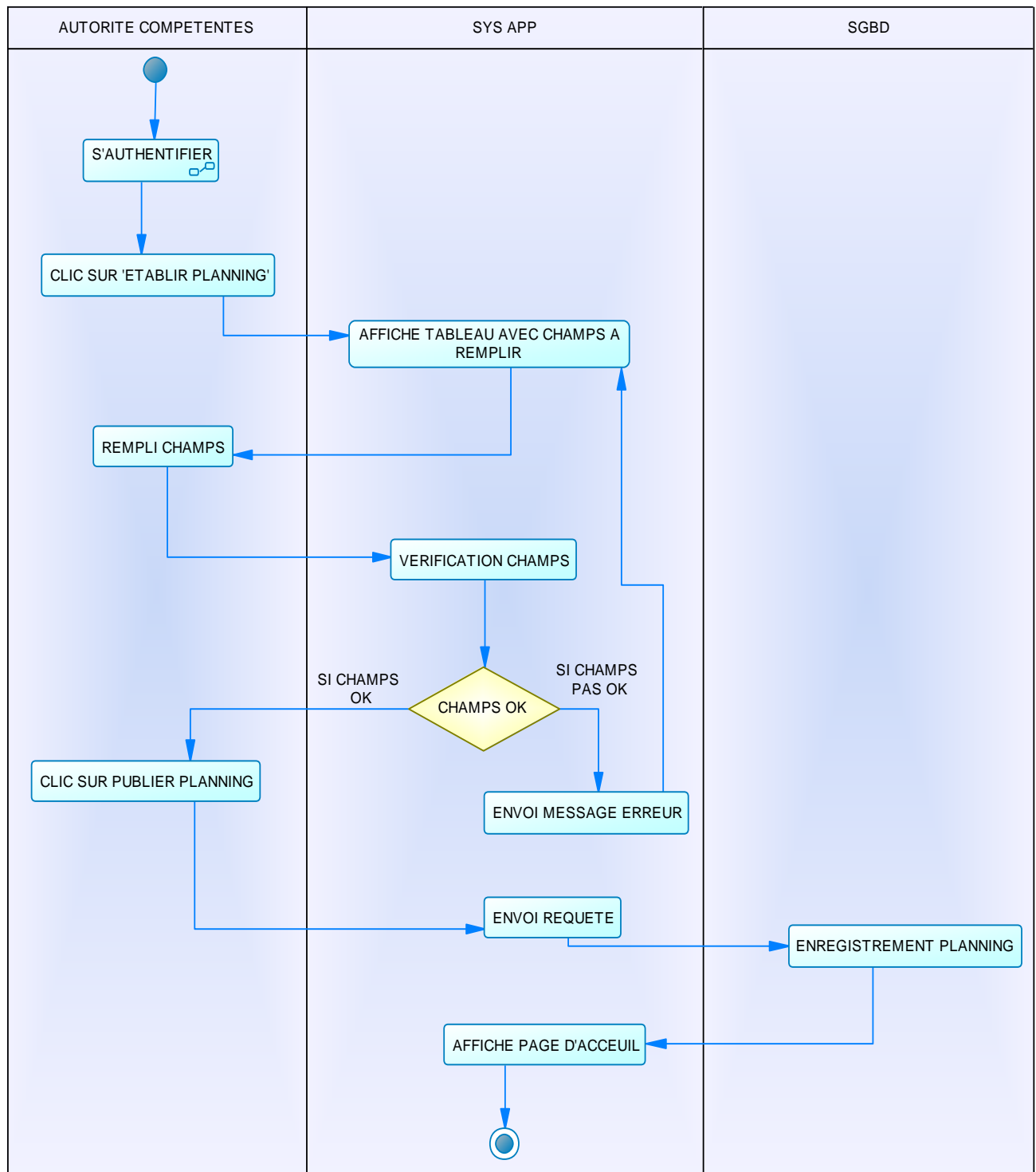


Figure 26: Diagramme d'activité du cas 'ETABLIR PLANNING'

CONCLUSION

En définitive, ce dossier d'analyse nous a permis d'avoir un aperçu détaillé du système à mettre sur pied. Toujours en exploitant le langage de modélisation UML et le processus 2TUP, nous poursuivrons par le dossier de conception dans lequel nous représenterons les diagrammes de la branche technique à savoir le diagramme de classe et le diagramme d'état transition et paquetage.

DOSSIER 4 : LE DOSSIER DE CONCEPTION

RESUME :

Etape non négligeable dans le processus de développement, la conception met en relief le fonctionnement de l'application, c'est d'ailleurs à partir de là que le maître d'œuvre aperçoit clairement ce qui lui sera présenté comme projet.

PLAN

INTRODUCTION

- I. DIAGRAMME DE CLASSE
- II. DIAGRAMME D'ETAT TRANSITION
- III. DIAGRAMME DE PAQUETAGE

CONCLUSION

INTRODUCTION

La phase de conception vise à ressortir les différents contours du modèle statique du système d'information afin d'en dégager un modèle dynamique devant implémenter notre plateforme dans les détails. Dans ce dossier, on retrouve les illustrations simplifiées d'architecture du système, qui font apparaître les différentes composantes de la plateforme, décrit les procédures de ces composantes et fait ressortir les attributs de la plateforme.

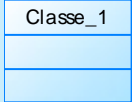


I – DIAGRAMME DE CLASSE


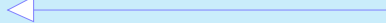
Le diagramme de classe exprime la structure statique du système en termes de classe et de relation entre ces classes. L'intérêt de ce diagramme est de modéliser les entités des systèmes d'informations. Le diagramme de classe permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine en décrivant les attributs et les comportements plutôt que de détailler les méthodes pour atteindre les opérations.

I.1 – Formalisme

Le diagramme de classe met en évidence d'éventuelles relation entre ces classes.il comporte quelques concepts : la classe, les attributs, l'identifiant, les opérations (méthodes) et les relations. Un diagramme de classe est constitué des éléments présentés dans le tableau suivant :

Tableau 20:Eléments de modélisation du diagramme de classe

ELEMENTS DE MODELISATION	DESCRIPTION	REPRESENTATIONS GRAPHIQUES
Classe	C'est une description abstraite d'un ensemble d'objets du domaine	
Association	C'est la relation la plus courante existant entre les classes	
Agrégation	Est une association qui exprime une relation contenant/contenu, agrégat/agrégé, dans laquelle la suppression de l'agrégat n'entraîne	

ELEMENTS DE MODELISATION	DESCRIPTION	REPRESENTATIONS GRAPHIQUES
Composition	pas la suppression de l'agrégé Il s'agit d'une relation d'agrégation forte où la suppression de l'agrégat entraîne automatiquement la suppression de l'agrégé	
Généralisation	C'est une relation qui permet d'identifier parmi les objets d'une classe des sous-ensembles d'objets ayant des définitions spécifiques.	

I.2 – Exemple De Diagramme De Classe

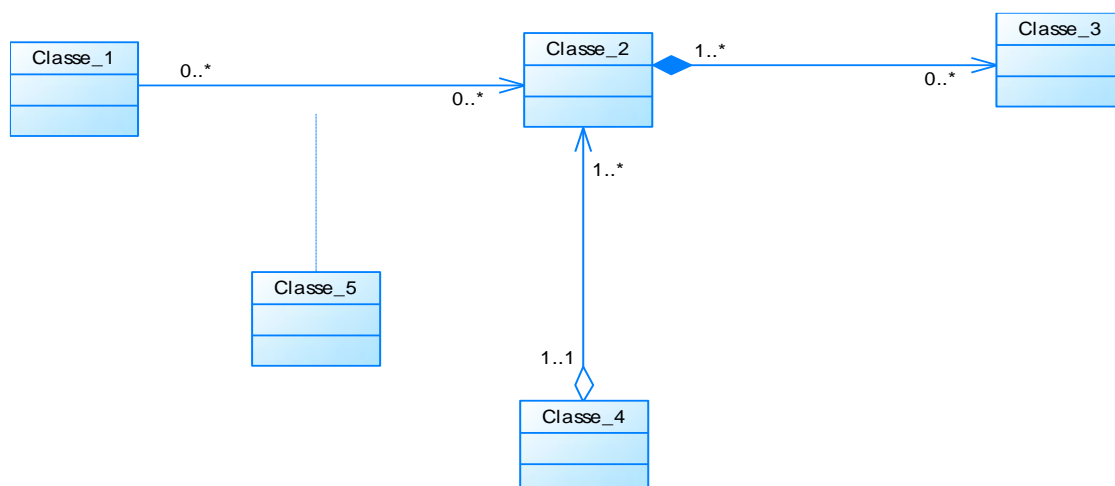


Figure 27: Exemple de diagramme de classe

I.3 – Diagramme De Classe Du Système

Nous pouvons ainsi ressortir le diagramme de classe suivant :

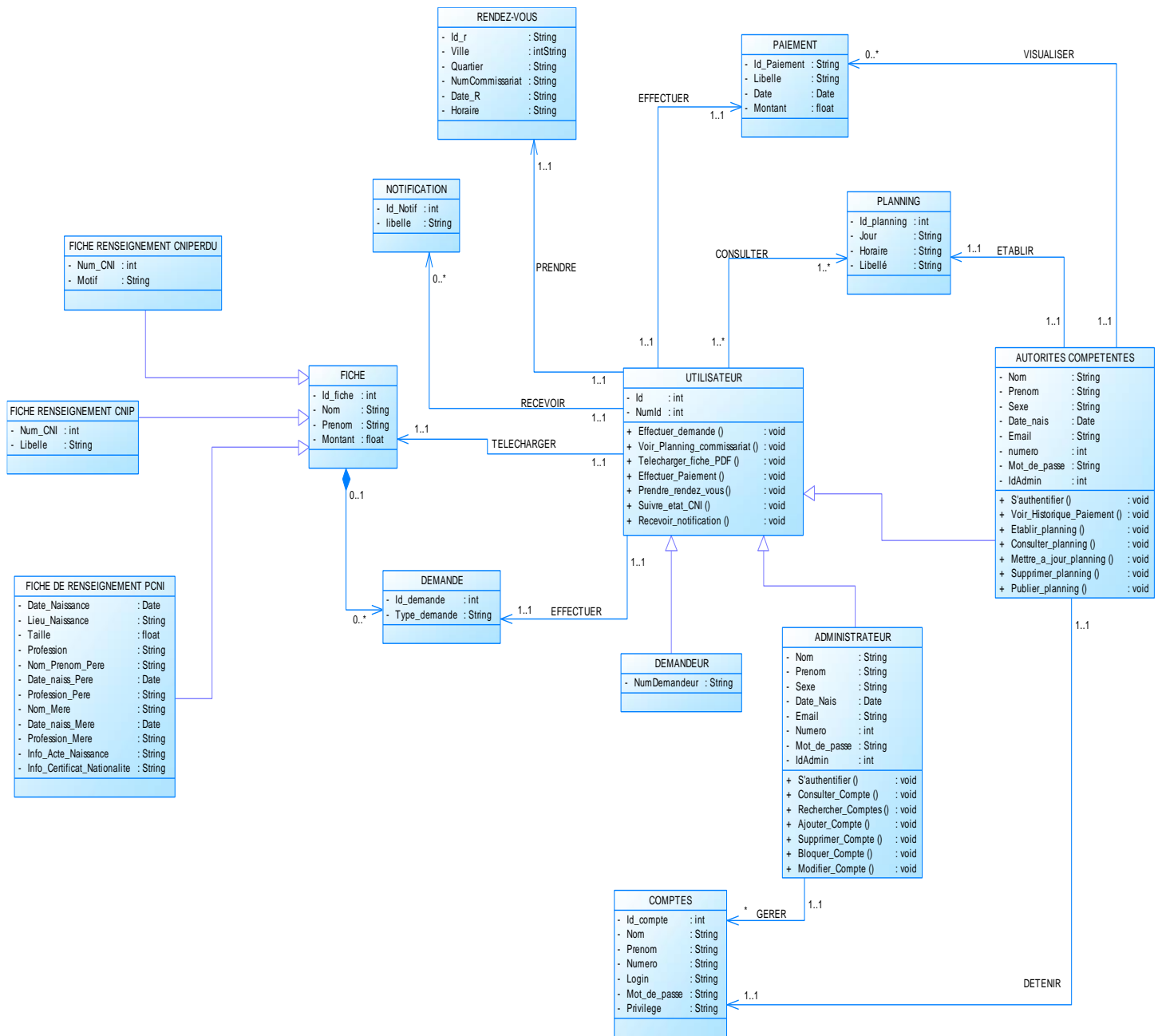


Figure 28: Diagramme de classe du système

REGLES METIERS

Les règles métiers représentent les différentes relations qui existent entre les classes deux à deux (relations binaires) ou en groupes de trois (relations ternaires). A base de notre diagramme de classe, nous pouvons ressortir les règles métiers suivantes :

▪ REGLE N°01

Une instance de classe Utilisateur effectue une seule instance de Demande.

Une instance de classe Demande est effectuée par une seule instance de classe Utilisateur.

▪ REGLE N°02

Une instance de classe Utilisateur peut recevoir aucune ou plusieurs instances de classe Notifications.

Une instance de classe Notification est reçue par une seule instance de classe Utilisateur.

▪ REGLE N°03

Une instance de classe Utilisateur peut consulter une ou plusieurs instances de classe Planning.

Une instance de classe Planning est consultée par une ou plusieurs instances de classe Utilisateur.

▪ REGLE N°04

Une instance de classe Utilisateur peut effectuer aucune ou une seule instance de classe Paiement.

Une instance de classe Paiement est effectuée par une seule instance de classe Utilisateur.

▪ REGLE N°05

Une instance de classe Utilisateur peut télécharger une seule instance de classe Fiche.

Une instance de classe Fiche est téléchargée par une seule instance de classe Utilisateur.

▪ REGLE N°06

Une instance de classe Autorités Compétentes peut visualiser aucune ou plusieurs instances de classe Paiement.

Une instance de classe Paiement est visualisée par une seule instance de classe Autorité Compétente.

▪ **REGLE N°07**

Une instance de classe Autorités Compétentes établie une seule instance de classe Planning.

Une instance de classe Planning est établie par une seule instance de classe Autorités compétentes.

▪ **REGLE N°08**

Une instance de classe Autorités Compétentes détient une seule instance de classe Compte.

Une instance de classe Compte est détenue par une seule instance de classe Autorités Compétentes.

▪ **REGLE N°09**

Une instance de classe Administrateur peut gérer plusieurs instances de classe Compte.

Une instance de classe Compte est géré par une seule instance de classe Administrateur.

▪ **REGLE N°10**

Une instance de classe Utilisateur prend une et une seule instance de classe Rendez-vous.

Une instance de classe Rendez-vous est prise par une seule instance de classe Utilisateur.

II – DIAGRAMME D'OBJET

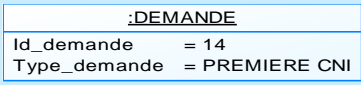

Le diagramme d'objet est une instance spécifique d'un diagramme de classe à un moment précis. Il permet d'exprimer le diagramme de classe au travers d'exemples en mettant en exergue les détails imperceptibles dans un diagramme de classe. En d'autres termes, le diagramme de classe modélise les règles et le diagramme d'objet modélise les faits.

II.1 – FORMALISME

Un diagramme d'objet se matérialise comme un diagramme de classe, cependant le nom de l'objet est précédé de « : » et souligné. Les associations deviennent des liens. Un lien se représente comme une association sans le nom de l'association, et s'il y en a, celui-ci doit être souligné. On ne représente pas de multiplicité dans un diagramme d'objet.

Nous pouvons donc résumer les éléments de modélisation dans le tableau suivant :

Tableau 21: Elément d'un diagramme d'objet

ELEMENTS DE MODELISATION	DESCRIPTION	REPRESENTATIONS GRAPHIQUES
<i>Objet</i>	C'est l'instance d'une classe	
<i>Lien</i>	C'est la relation qui existe entre deux objets	

II.2 – EXEMPLE DE DIAGRAMME D'OBJET

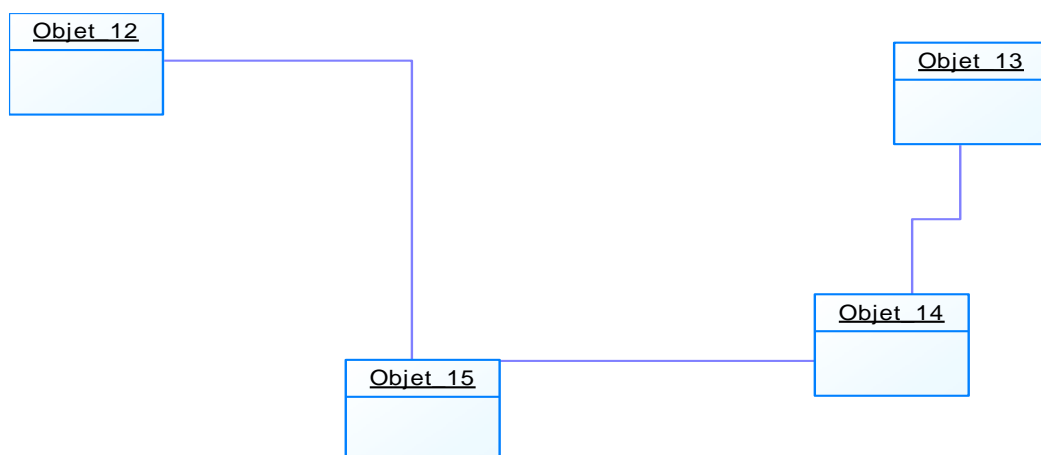


Figure 29: Exemple de diagramme d'objet

II.3 – DIAGRAMME D'OBJET DU SYSTEME

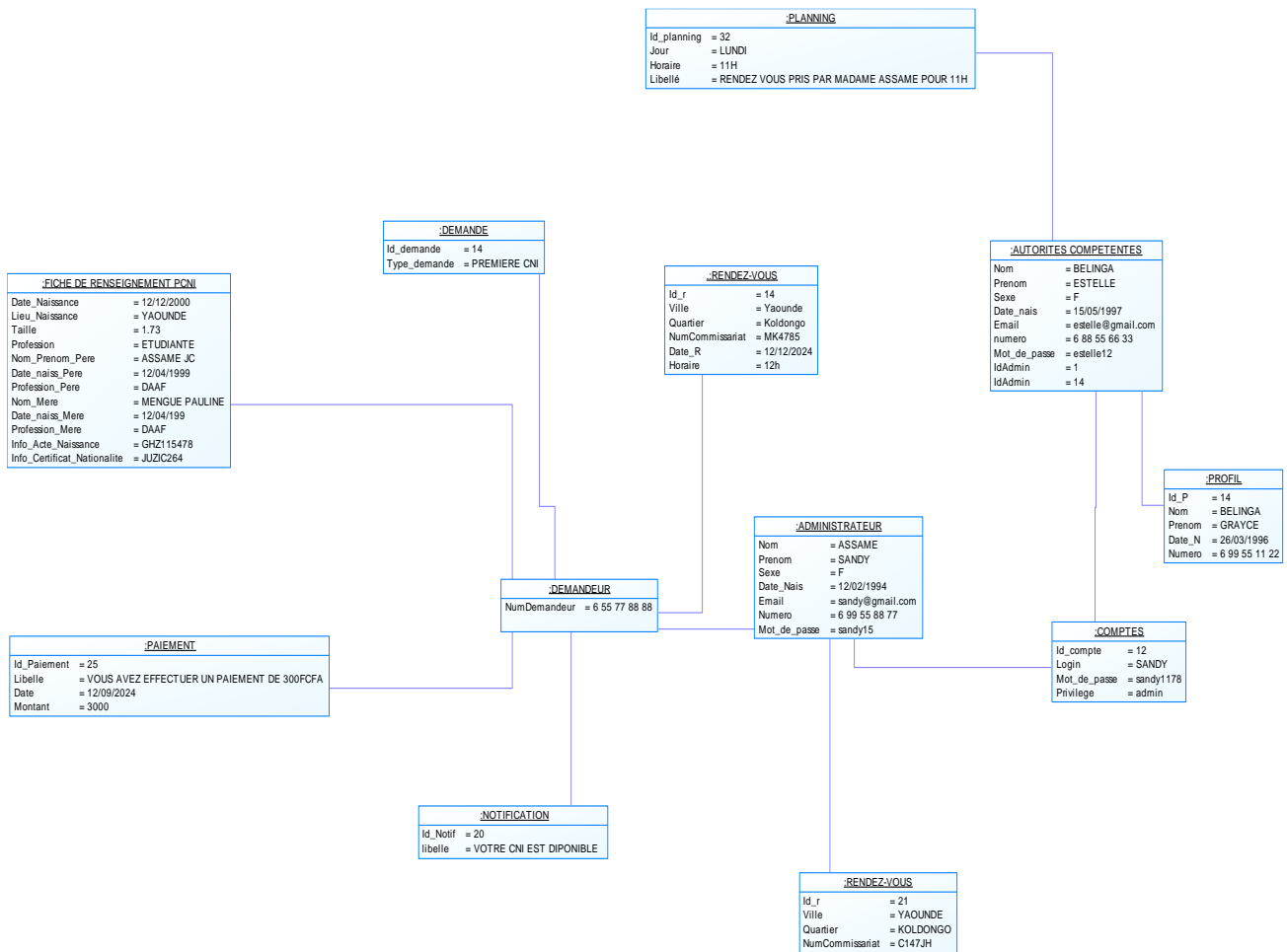


Figure 30: Diagramme d'objet du système

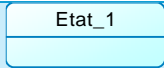
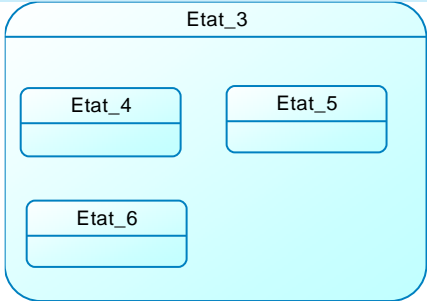


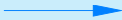
III – DIAGRAMME D'ETAT TRANSITION

Le diagramme d'état transition décrit le comportement interne d'un objet. Il présente les séquences possibles qu'une instance d'une classe peut traiter au cours de son cycle de vie en réaction à des événements. Un objet peut passer par une série d'états durant toute sa vie. Le diagramme d'état transition permet en gros de représenter et de décrire les changements d'état d'un objet liés à des événements.

III.1 – Formalisme

Les éléments qui modélisent un diagramme d'état transition sont :

Tableau 22: Elément de modélisation d'un diagramme d'état transition

<i>ELEMENTS DE MODELISATION</i>	<i>DESCRIPTION</i>	<i>REPRESENTATIONS GRAPHIQUES</i>
<i>Etat simple</i>	C'est un état élémentaire qui ne contient pas de transition interne ni de sous-état	
<i>Etat composite</i>	Est un état qui englobe des sous-états	
<i>Etat initial</i>	Représente l'état de démarrage du système	
<i>Etat final</i>	Représente la fin de fonctionnement du système	
<i>Transition</i>	Représente le passage d'un état à un autre	

III.2 – Exemple De Diagramme D'état Transition

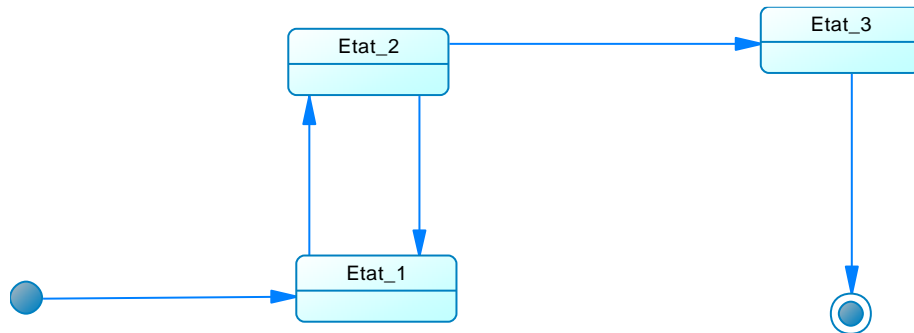


Figure 31: Exemple de diagramme d'état transition

III.3 – Quelques Diagrammes D'état Transition Du Système

- Diagramme d'état-transition de « COMPTE »

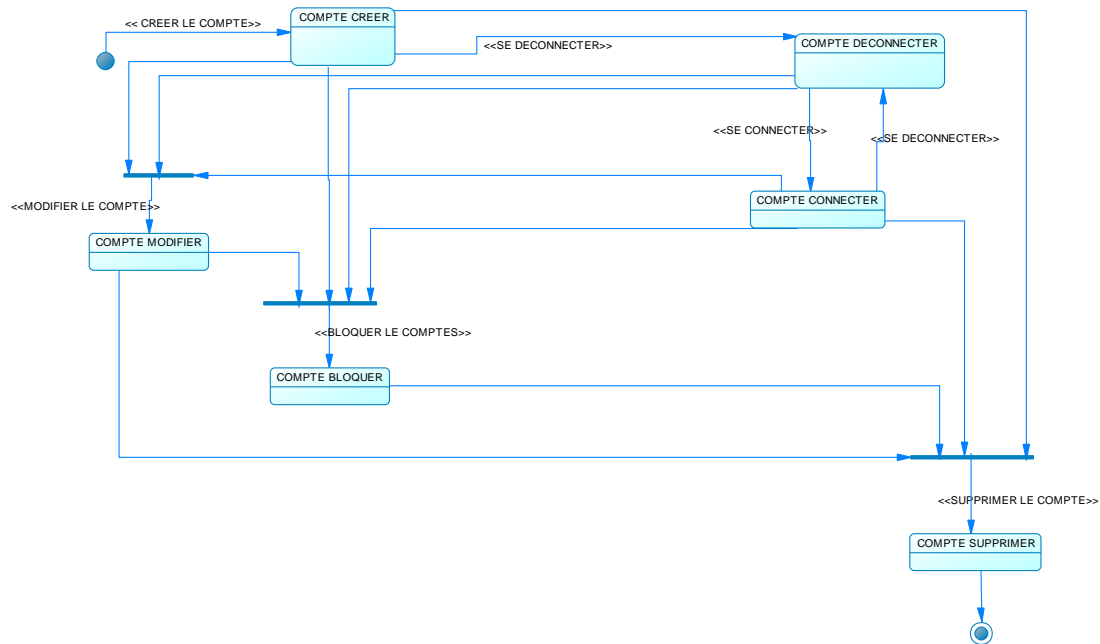


Figure 32: Diagramme d'état transition de "COMPTE"

• Diagramme d'état-transition de « PLANNING »

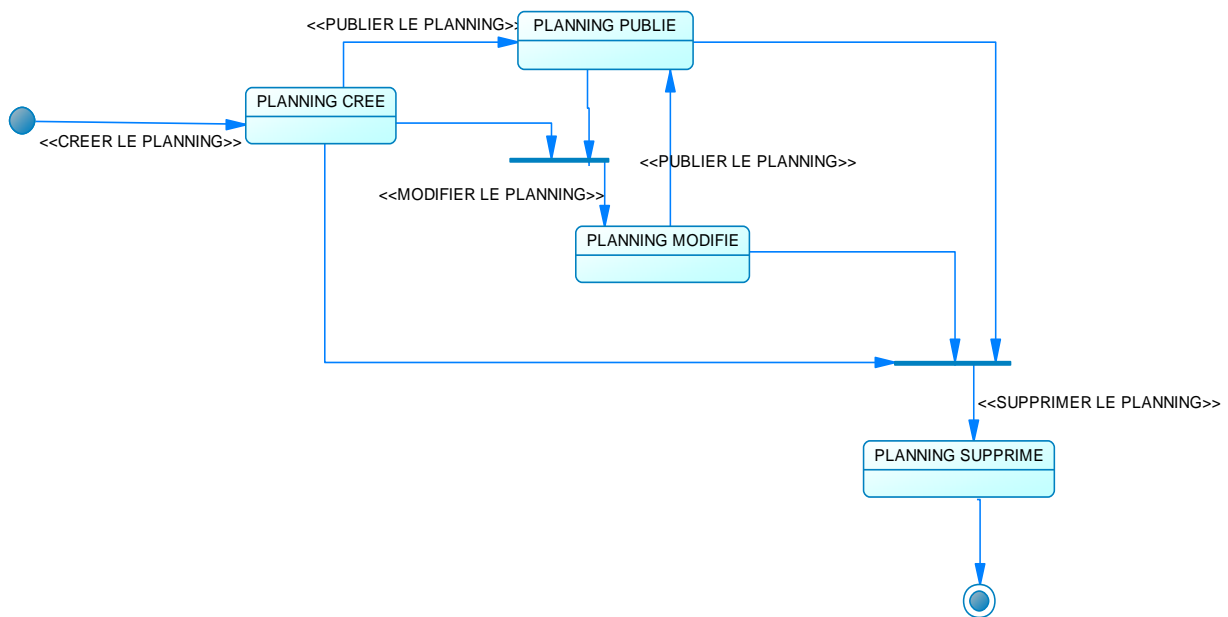


Figure 33: Diagramme d'état transition de "PLANNING"

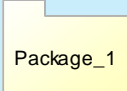
IV – DIAGRAMME DE PAQUETAGE

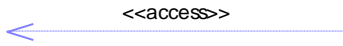
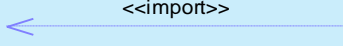
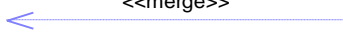
Le diagramme de paquetage permet de décomposer le système en catégories ou parties plus facilement observables appelées packages. C'est une collection d'éléments UML logiquement liés quoi que l'utilisation la plus courante est d'organiser les diagrammes de cas d'utilisation et de classe bien qu'il ne se limite pas à ces éléments. Les packages peuvent être construits pour représenter les opérations physiques ou logiques.

IV.1 – Formalisme

Les éléments qui constituent le diagramme de paquetage sont indiqués dans le tableau qui suit :

Tableau 23: Elément de modélisation d'un diagramme de paquetage

ELEMENTS DE MODELISATION	DESCRIPTION	REPRESENTATIONS GRAPHIQUES
Package	Il regroupe des éléments liés de manière logique au sein du système	

ELEMENTS DE MODELISATION	DESCRIPTION	REPRESENTATIONS GRAPHIQUES
Dépendance « Access »	Relation qui indique qu'un package à besoin du soutien d'un autre package	
Dépendance « import »	Relation qui indique que la fonctionnalité a été importée, d'un package à un autre	
Package « merge »	C'est une relation dirigée dans laquelle le contenu d'un package s'ajoute au contenu d'un autre	

IV.2 – Exemple De Diagramme De Paquetage

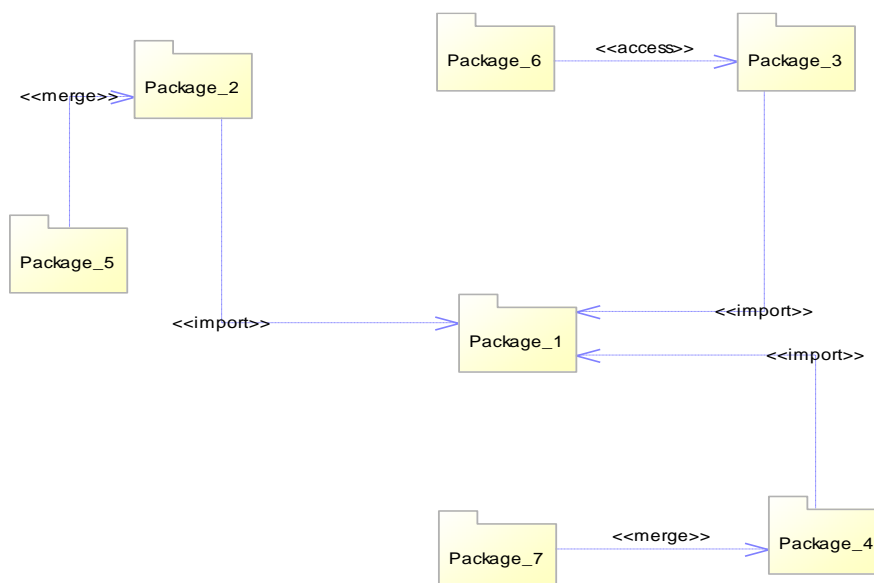


Figure 34 : Exemple de diagramme de paquetage

IV.3 – Diagramme De Paquetage Du Système

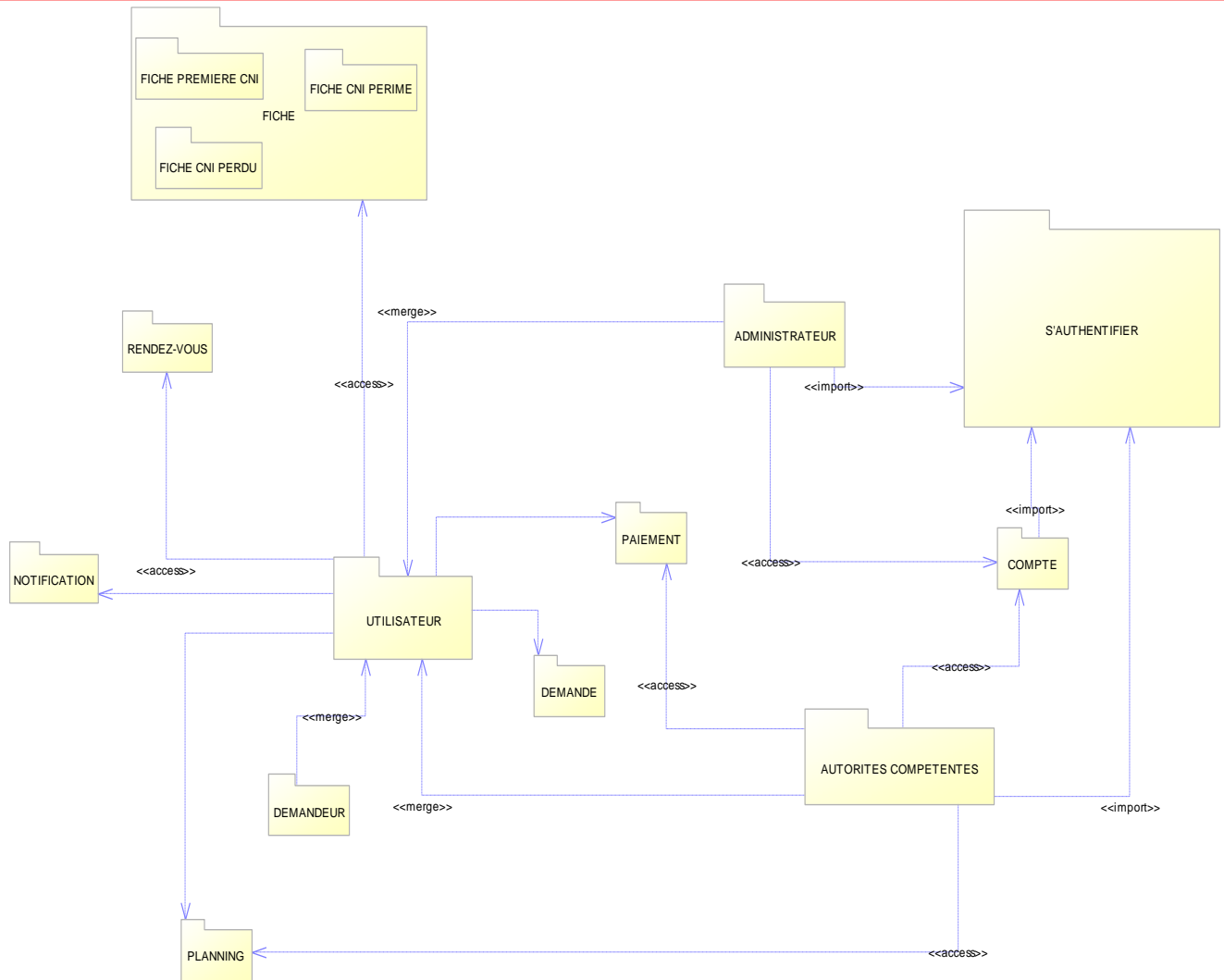


Figure 35: Diagramme de paquetage du système

CONCLUSION

Le dossier de conception nous a permis de ressortir les données nécessaires pour la création de notre base de données et l'implémentation de notre application. Les différents éléments modélisés dans cette partie nous ont permis d'avoir une vue globale sur les différents modules de notre application, dès lors l'étape suivante de notre projet sera la rédaction du dossier de réalisation tenant compte des différents éléments modélisés plus hauts.

DOSSIER 5 : LE DOSSIER DE DEPLOIEMENT

RESUME :

Cette phase consiste à la mise en œuvre du projet dans un langage de programmation conformément aux spécifications définies dans les phases précédentes. Elle renferme en son sein la présentation des outils et langages utilisés, les architectures matérielle et logicielle, le diagramme de composant et celui de déploiement.

PLAN

INTRODUCTION

- I. PRESENTATION DE L'ENVIRONNEMENT LOGICIEL ET DES TECHNOLOGIES
- II. ARCHITECTURE LOGICIELLE ET MATERIELLE
- III. DIAGRAMMES INTERVENANTS

CONCLUSION

INTRODUCTION




La conception et la modélisation étant des étapes importantes dans le processus de développement d'une application car elles permettent de définir et spécifier les différents éléments constituant l'application ou le système étudié, mais la réalisation est l'étape qui permet de reproduire concrètement les différents objets issus de la conception et de donner vie au système.



I – PRESENTATION DE L'ENVIRONNEMENT LOGICIEL ET DES TECHNOLOGIES

I.1 – Environnement Logiciel

Les logiciels qui sont intervenus dans la réalisation de notre application mobile sont présentés dans le tableau suivant :

Tableau 24: Présentation des logiciels utilisés


NOM DU LOGICIEL	VERSION	UTILISATION	LOGO
Système d'exploitation Windows 11 Professionnel	23H2	Système d'exploitation utilisé pour l'exécution des logiciels	
Microsoft Edge	92.0.902.67	Navigateur web utilisé pour les différents tests de la plateforme	
Power AMC	15.1	Atelier de Génie Logiciel, utilisé pour la modélisation de la solution (la réalisation des différents diagrammes)	



NOM DU LOGICIEL	VERSION	UTILISATION	LOGO
Fire Base		Système de Gestion de Base de Données utilisé pour stocker les différentes données	
Emulateur		Téléphone virtuel permettant de compiler l'application	

I.2 – Présentation Des Outils

Pour la réalisation de notre application mobile, nous avons décidé d'utiliser les langages présentés dans le tableau suivant :

Tableau 25: Présentation des langages utilisés

LANGAGE	ROLE	LOGO
Dart	Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, de serveur et web. Dart est un langage orienté objet, basé sur la classe, récupérateur de mémoire avec une syntaxe de	

LANGAGE	ROLE	LOGO
Flutter	<p>type C.</p> <p><u>Flutter</u> est un Framework de développement d'applications multiplateforme, conçu par Google, dont la première version a été publiée sous forme de projet open source à la fin de l'année 2018. Flutter met à disposition une grande variété de bibliothèques d'éléments d'IU standard pour Android et iOS.</p>	
NoSQL	<p>Le NoSQL (Not Only SQL) est un type de base de données qui offre une alternative aux systèmes de gestion de bases de données relationnelles traditionnels (SQL). NoSQL permet de stocker des données non structurées ou semi-structurées, comme des documents, des images, ou des vidéos.</p>	

II – ARCHITECTURE LOGICIELLE ET MATERIELLE

II.1 – Architecture Logicielle

Le modèle MVC (**M**odèle – **V**ue – **C**ontrôleur) est un modèle d'architecture logiciel qui divise une application en trois composants distincts : le modèle, la vue et le contrôleur. Il est largement utilisé dans les applications web et les applications de bureau pour séparer les fonctions de l'interface utilisateur, des données et de la logique de l'application.

Les trois composants du modèle MVC sont :

II.1.1 – Le modèle

C'est la représentation des données et des règles métier de l'application. Le modèle peut être une base de données, un fichier XML, un service web ou toute autre source de données. Le modèle est indépendant de la vue et du contrôleur, ce qui permet de modifier facilement l'un des composants sans affecter les autres.

II.1.2 – La vue

C'est l'interface utilisateur de l'application. La vue est responsable de l'affichage et de la mise à jour des données du modèle. Elle peut être une page web ou une fenêtre de bureau. La vue ne contient pas de code de logique métier, elle n'affiche que les données qui lui sont fournies par le contrôleur.

II.1.3 – Le contrôleur

C'est l'interface entre le modèle et la vue. Le contrôleur est responsable de récupérer les données du modèle et de les envoyer à la vue, ainsi que de recevoir les entrées de l'utilisateur depuis la vue et de les envoyer au modèle pour effectuer des opérations métier. Le contrôleur peut également gérer les événements de l'application (clics de souris, touches de clavier, etc.)

En résumé, le modèle MVC permet de séparer clairement les responsabilités de l'interface utilisateur, la logique de l'application et des données. Cela permet une plus grande modularité, une meilleure réutilisation du code et une facilité de maintenance.

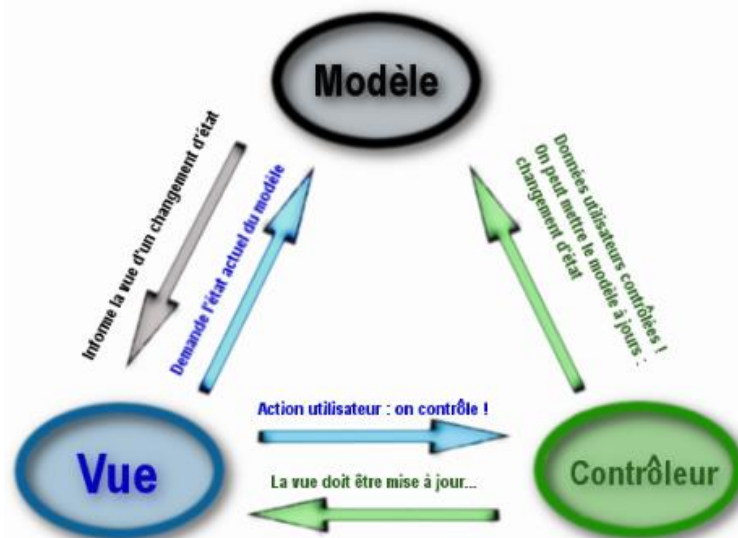


Figure 36: Notre architecture logicielle

II.2 – Architecture Matérielle

L'architecture physique encore appelée architecture technique décrit l'ensemble des composants matériels supportant l'application.

Notre projet consistera à mettre sur pied une plateforme web qui sera structurée en plusieurs couches à savoir : la couche de présentation, la couche de traitement, la couche d'accès aux données et celle réservée aux autres systèmes, notamment pour l'API. Le noyau d'une application est constitué de la logique de présentation et la logique de traitement. C'est la répartition de ce noyau entre les différentes couches qui détermine les architectures caractéristiques des systèmes client/serveurs, nous pouvons ainsi citer l'architecture 1-tiers, 2-tiers, 3-tiers et même n-tiers.

Notre application sera déployée sur une architecture n-tiers. Il s'agit d'une approche de conception logicielle qui divise une application en plusieurs couches indépendantes et interconnectées. Chaque couche a une responsabilité spécifique et communique avec les autres couches via des interfaces bien définies. Cela permet de séparer les préoccupations, d'améliorer la maintenabilité, la scalabilité et la réutilisabilité du logiciel. Cette architecture ajoute davantage de couches à l'architecture 3-tiers, elle permet également de relier plusieurs systèmes ou services ensemble pour former une application plus complexe.

II.2.1 – Couche de présentation

Cette couche est responsable de l'interaction avec l'utilisateur. Elle gère l'affichage des informations et la saisie des données par le biais de menus, de formulaires ou d'autres éléments d'interface utilisateur. Elle peut également gérer la validation des données et interpréter les actions de l'utilisateur pour les transmettre à la couche de logique métier. Les technologies couramment utilisées dans cette couche comprennent les interfaces web, les applications mobiles ou de bureau.

II.2.2 – Couche de logique métier

Cette couche contient la logique métier de l'application. Elle gère les règles métier, les calculs, la logique de traitement des données et la coordination des différentes actions. Elle s'appuie sur les informations fournies par la couche de présentation pour effectuer les opérations nécessaires. Elle interagit également avec la couche de persistance (d'accès aux données) pour récupérer ou enregistrer les données. Cette couche est généralement mise en œuvre à l'aide de langages de programmation tels que Java, C#, Python, PHP, Dart, JavaScript...

II.2.3 – Couche d'accès aux données persistantes

Cette couche est responsable de la gestion des données persistantes, telles que les bases de données, les fichiers, les services web, etc. Elle interagit avec la couche de logique métier pour récupérer ou enregistrer les données. Elle encapsule les détails d'accès aux données, en fournissant des opérations pour créer, lire, mettre à jour ou supprimer les données (CRUD). Les Framework et technologies utilisés dans cette couche incluent les SGBD (Systèmes de Gestion de Bases de Données) tels que MySQL, PostgreSQL, Oracle, MongoDB, etc.

III – DIAGRAMMES INTERVENANTS


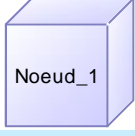

III.1 – Diagramme De Déploiement

Le diagramme de déploiement modélise les composants ou éléments matériels utilisées pour modéliser un système et l'association entre ces composants. Le diagramme de déploiement est généralement mis en œuvre pour documenter l'architecture physique du système.

III.1.1 – Formalisme

Les éléments de modélisation du diagramme de déploiement sont les suivants :

Tableau 26 : Elément de modélisation d'un diagramme de déploiement

<i>ELEMENT DE MODELISATION</i>	<i>DESCRIPTION</i>	<i>REPRESENTATIONS GRAPHIQUES</i>
<i>Composant</i>	Il représente une entité logicielle du système	Composant_1 
<i>Nœud</i>	Il représente un ensemble d'éléments matériels du système	Noeud_1 
<i>Association</i>	Elle est représentée par une ligne pleine entre deux nœuds et indique une communication entre les éléments matériels	
<i>Dépendance</i>	Elle est utilisée pour modéliser la relation entre deux composants	

III.1.2 – Exemple de diagramme de déploiement

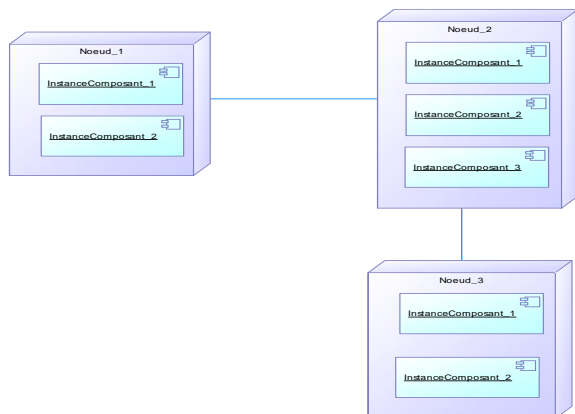


Figure 37: Exemple de diagramme de déploiement

III.1.3 – Diagramme de déploiement du système

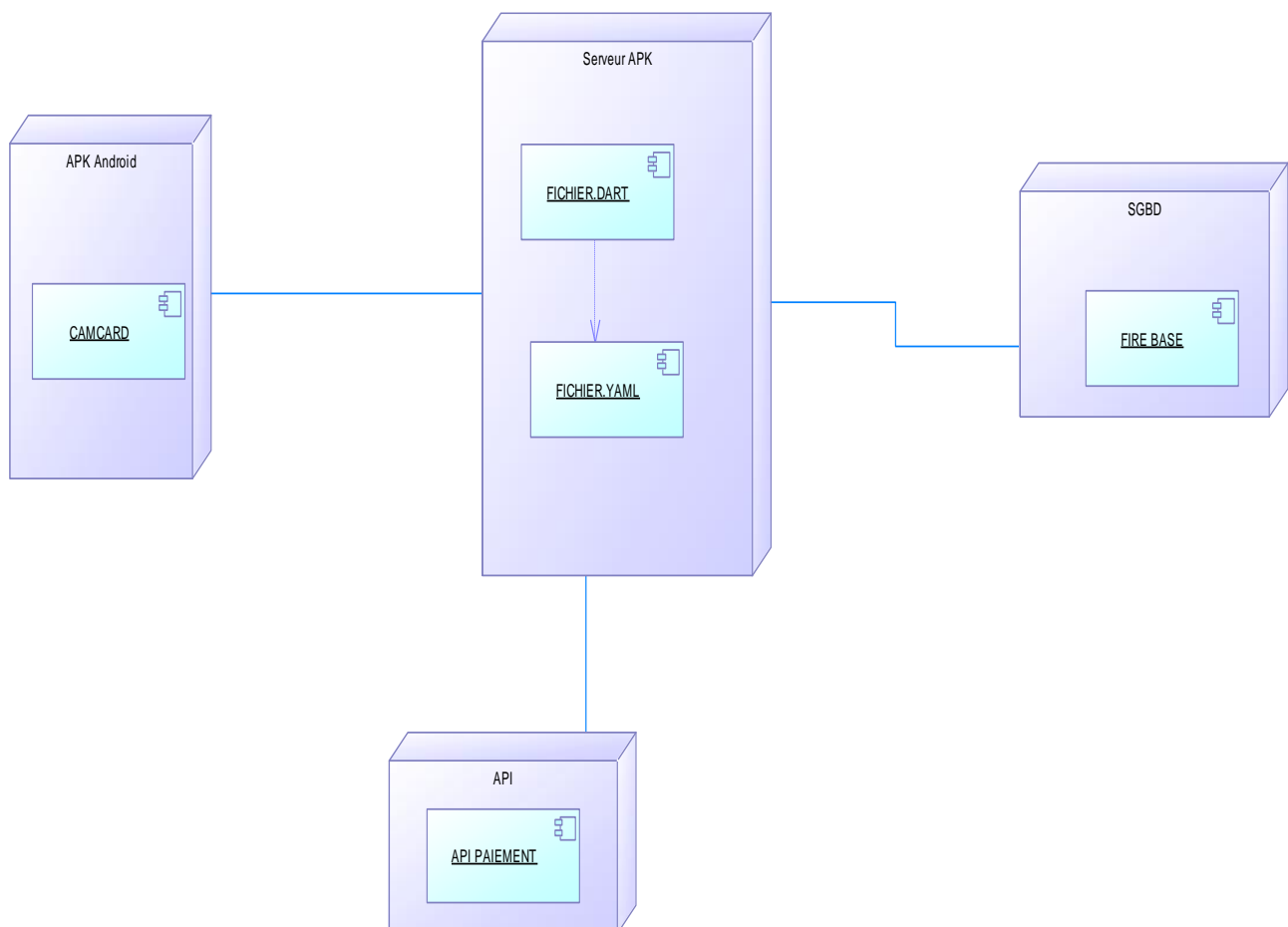


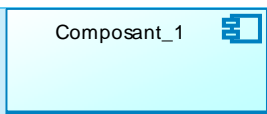

Figure 38 : Diagramme de déploiement du système

III.2 – Diagramme De Composant

III.2.1 – Formalisme

Le diagramme de composant est employé pour décrire les dépendances entre les divers composants logiciels tels que les dépendances entre les fichiers exécutables et les fichiers sources. Il modélise les composants qui contribuent à la réalisation des fonctionnalités.

Tableau 27: Elément de modélisation du diagramme de composant

ELEMENTS DE MODELISATION	DESCRIPTION	REPRESENTATIONS GRAPHIQUES
Composant	Il représente une entité logicielle du système	
Dépendance	Elle est utilisée pour modéliser la relation entre deux composants	

III.2.2 – Exemple de diagramme de composant

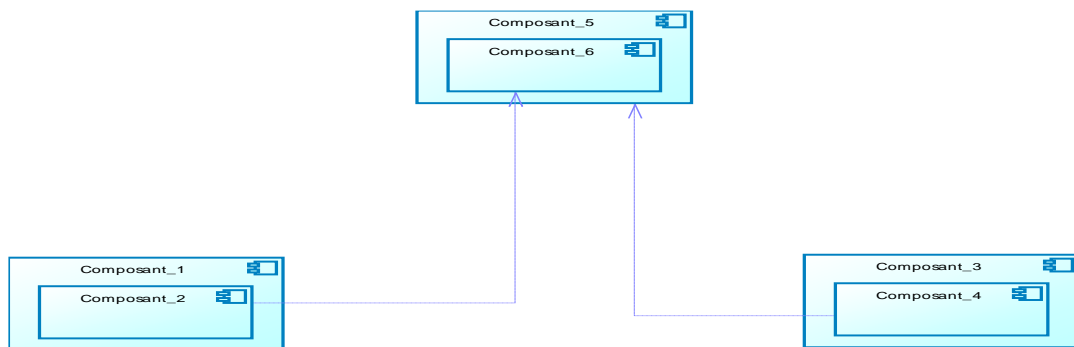


Figure 39: Exemple de diagramme de déploiement

III.2.3 – Diagramme de composant du système

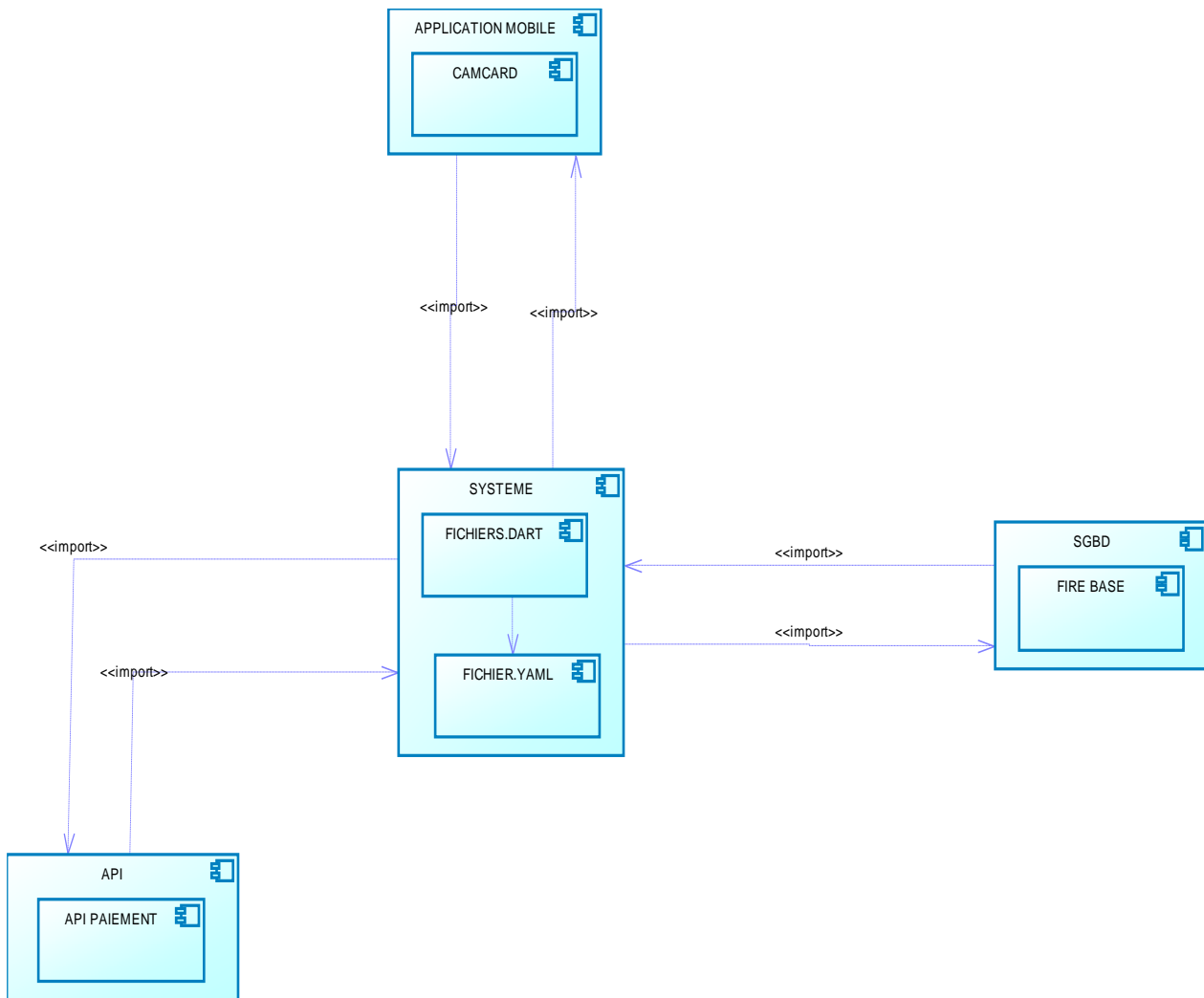


Figure 40: Diagramme de composant du système

CONCLUSION

Le dossier de réalisation nous a permis de choisir les différents outils et technologies à utiliser pour l'implémentation de la solution, ainsi les architectures (physiques et logicielle) de celle-ci. La prochaine étape consistera à présenter un guide d'utilisation de la solution.

DOSSIER 6 : TEST DE FONCTIONNALITES

RESUME

Rédigé par **ASSAME NNAMA Orlane Sandy**, élève ingénieur en Deuxième année, option Génie Logiciel,

IAI-CAMEROUN, Centre d'Excellence Technologique PAUL BIYA

Année académique 2023-2024

Cette phase a pour but de se rassurer de l'efficacité de nos différentes fonctionnalités déployer. Il sera donc question pour nous dans cette partie de faire des tests unitaires des fonctionnalités maitresses dans le but de vérifier que tout fonctionne comme il est attendu.

APERCU

INTRODUCTION

I. FONCTIONNALITES

II. TESTS

CONCLUSION

INTRODUCTION

Les tests représentent une opération importante dans le développement d'applications, consistant à se rassurer qu'une fonction ou qu'un bout de code fonctionne comme il est prévu. Dans cette partie il sera donc question de se rassurer que les fonctionnalités mises sur pied fonctionnent normalement.

I – FONCTIONNALITES

Plusieurs fonctionnalités ont été mises sur pied dans le cadre de la mise sur pied de notre application. Les plus importantes sont :

I.1 – La Fonction D'authentification

La fonction d'authentification est la première fonction qu'on utilisera dans l'application. Elle permettra en effet à tout utilisateur sauf le demandeur d'avoir accès à sa session de travail. Elle validera la connexion si le login et le mot de passe est correct.

```
Future<void> _logins() async{
  try {
    final userCredential = await _auth.signInWithEmailAndPassword(
      email: login.text,
      password: Mdp.text,
    );
    final user = userCredential.user;
    if (user != null){
      Navigator.push(context, MaterialPageRoute(builder: (context) => Paie()));
    }
  } on FirebaseAuthException catch (e){
    setState(() {
      if (e.code == 'user-not-found') {
        _errorMessage = 'Aucun utilisateur trouvé avec cet email.';
      } else {
        _errorMessage = 'Erreur : ${e.message}';
      }
    });
  }
}
```

Figure 41 : Fonction d'authentification.

I.2 – La Fonction Pour Télécharger Une Fiche en PDF

Cette fonction permet de télécharger une fiche précise. En effet, cette fonction téléchargera sous forme de fichier PDF les informations entrées par l'utilisateur courant.

```
import 'dart:io';

class FileService {
  Future<bool> downloadPdf(String url, String filePath) async {
    try {
      // Simuler le téléchargement d'un PDF à partir d'une URL.
      final response = await HttpClient().getUrl(Uri.parse(url));
      final file = File(filePath);
      final sink = file.openWrite();

      await response.pipe(sink);
      await sink.flush();
      await sink.close();

      return true; // Téléchargement réussi
    } catch (e) {
      print('Erreur lors du téléchargement: $e');
      return false; // Erreur de téléchargement
    }
  }
}
```

Figure 42: Fonction télécharger fiche PDF

II – TESTS

Les tests représentent une étape non-négligeable, voire très importante dans le processus de développement d'un logiciel, car elle permet de se rassurer qu'un bout de code fonctionne comme il l'est attendu. Dans la suite, nous ferons les tests unitaires des fonctions présentées plus haut.

II.1 – Test Unitaire De La Fonction D'authentification

```
group('Login', () {  
  final FirebaseAuth _auth = FirebaseAuth.instance;  
  
  test('test login avec email et mot de passe valides', () async {  
    // Arrange  
    final loginController = TextEditingController(text: 'test@example.com');  
    final mdpController = TextEditingController(text: 'password123');  
  
    // Act  
    final loginState = _LoginState();  
    loginState.login = loginController;  
    loginState.Mdp = mdpController;  
    await loginState._logins();  
  
    // Assert  
    expect(loginState._errorMessage, isNull);  
  });  
  
  test('test login avec email invalide', () async {  
    // Arrange  
    final loginController = TextEditingController(text: 'test');  
    final mdpController = TextEditingController(text: 'password123');  
  
    // Act  
    final loginState = _LoginState();  
    loginState.login = loginController;  
    loginState.Mdp = mdpController;  
    await loginState._logins();  
  });  
});
```

Figure 43: Test de la fonction d'authentification

Ce test vérifie que la connexion est réussie ou échoue en fonction des informations saisies par l'utilisateur. Nous avons testé les cas suivants :

- Connexion avec courriel et mot de passe valides
- Connexion avec courriel invalide
- Connexion avec mot de passe invalide
- Connexion avec erreur de serveur

Étape 1 : Définition du groupe de tests : Nous commençons par définir un groupe de tests appelé "Login". Cela nous permet de regrouper les tests liés à la connexion des utilisateurs.

Étape 2 : Définition de la variable d'authentification : Nous définissons une variable qui sera utilisée pour gérer l'authentification des utilisateurs. Cette variable est essentielle pour les tests qui suivent.

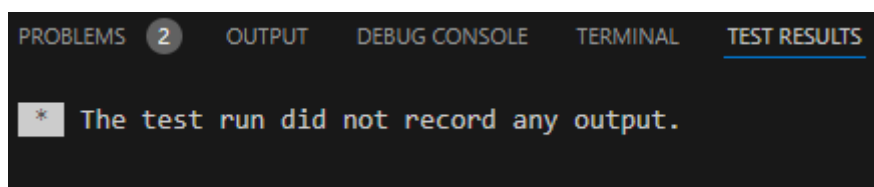
Étape 3 : Premier test - Connexion avec courriel et mot de passe valides : Nous créons un premier test qui vérifie que la connexion est réussie lorsque l'utilisateur saisit un courriel et un mot de passe valides. Nous simulons les champs de saisie du courriel et du mot de passe, et nous appelons la méthode de connexion.

Étape 4 : Deuxième test - Connexion avec courriel invalide : Nous créons un deuxième test qui vérifie que la connexion échoue lorsque l'utilisateur saisit un courriel invalide. Nous simulons les champs de saisie du courriel et du mot de passe, et nous appelons la méthode de connexion.

Étape 5 : Troisième test - Connexion avec mot de passe invalide : Nous créons un troisième test qui vérifie que la connexion échoue lorsque l'utilisateur saisit un mot de passe invalide. Nous simulons les champs de saisie du courriel et du mot de passe, et nous appelons la méthode de connexion.

Étape 6 : Quatrième test - Connexion avec erreur de serveur : Nous créons un quatrième test qui vérifie que la connexion échoue lorsque le serveur renvoie une erreur. Nous simulons les champs de saisie du courriel et du mot de passe, et nous appelons la méthode de connexion.

Le résultat du test (réussi) est présenté dans la figure suivante :



II.2 – Test Unitaire De La Fonction Télécharger Fiche En PDF

```
Run | Debug | Profile
void main() {
  group('PdfGenerator', () {
    test('generateAndPreviewPdf', () async {
      // Créer une instance de Firestore mockée
      final firestore = FirebaseFirestore.instance;
      final mockCollection = MockCollection();
      when(firestore.collection('FichePremiereCni')).thenReturn(mockCollection);

      // Créer une instance de PdfGenerator
      final pdfGenerator = PdfGenerator(firestore: firestore);

      // Créer un context mocké
      final mockContext = MockBuildContext();

      // Appeler la méthode generateAndPreviewPdf
      await pdfGenerator.generateAndPreviewPdf(mockContext);

      // Vérifier que la méthode addPage a été appelée
      verify(mockContext.addPage(any));

      // Vérifier que la méthode save a été appelée
      verify(mockContext.save());
    });
  });
}
```

Figure 44: Test de la fonction télécharger fiche PDF

Ce code définit un test pour la classe PdfGenerator, qui vérifie que la méthode generateAndPreviewPdf fonctionne correctement en générant et prévisualisant un PDF. Le test utilise des classes mockées pour simuler le comportement de Firestore et du contexte de génération de PDF.

Étape 1 : Définition de la méthode main : La méthode main est la méthode d'entrée du programme. Elle est appelée lorsque le programme est lancé.

Étape 2 : Définition d'un groupe de tests : La méthode group est utilisée pour regrouper des tests ensemble. Dans ce cas, le groupe est nommé "PdfGenerator". Les tests à l'intérieur de ce groupe vont vérifier le fonctionnement de la classe PdfGenerator.

Étape 3 : Définition d'un test : La méthode test est utilisée pour définir un test individuel. Dans ce cas, le test est nommé "generateAndPreviewPdf". Ce test va vérifier que la méthode generateAndPreviewPdf de la classe PdfGenerator fonctionne correctement.

Étape 4 : Création d'une instance de Firestore mockée : La ligne final firestore = FirebaseFirestore.instance; crée une instance de la classe FirebaseFirestore, qui est une classe de Firebase pour interagir avec la base de données Firestore.

La ligne final mockCollection = MockCollection(); crée une instance de la classe MockCollection, qui est une classe mockée (c'est-à-dire une classe qui simule le comportement d'une autre classe) pour la collection "FichePremiereCni" dans Firestore.

La ligne when(firestore.collection('FichePremiereCni')).thenReturn(mockCollection); configure la classe mockée MockCollection pour qu'elle soit retournée lorsque la méthode collection de FirebaseFirestore est appelée avec le nom de collection "FichePremiereCni". Cela permet de mocker (simuler) le comportement de Firestore pour les besoins du test.

Étape 5 : Création d'une instance de PdfGenerator : La ligne final pdfGenerator = PdfGenerator(firestore: firestore); crée une instance de la classe PdfGenerator, en passant l'instance de Firestore mockée créée à l'étape 4 comme paramètre.

Étape 6 : Création d'un contexte mocké : La ligne final mockContext = MockBuildContext(); crée une instance de la classe MockBuildContext, qui est une classe mockée pour le contexte de génération de PDF.

Étape 7 : Appel de la méthode generateAndPreviewPdf : La ligne await pdfGenerator.generateAndPreviewPdf(mockContext); appelle la méthode generateAndPreviewPdf de l'instance de PdfGenerator, en passant le contexte mocké créé à l'étape 6 comme paramètre. Cette méthode est censée générer et prévisualiser un PDF.

Étape 8 : Vérification des appels de méthodes : Les lignes verify(mockContext.addPage(any)); et verify(mockContext.save()); vérifient que les méthodes addPage et save du contexte mocké ont été appelées lors de l'exécution de la méthode generateAndPreviewPdf. Les méthodes verify sont utilisées pour vérifier que les méthodes mockées ont été appelées avec les bons paramètres.

CONCLUSION

La phase de tests nous a permis de nous rassurer que les différentes fonctionnalités fonctionnent correctement, comme ils étaient attendus. Dans la prochaine partie, nous présenterons comment installer et utiliser de manière générale notre application.

DOSSIER 7 : GUIDE D'INSTALLATION ET GUIDE D'UTILISATION

RESUME

Dans ce guide d'utilisation nous présenterons le processus d'installation de l'application (application mobile), et nous présenterons par la suite quelques fonctionnalités de la solution finale.

APERCU

INTRODUCTION

- I. GUIDE D'INSTALLATION
- II. GUIDE D'UTILISATION

CONCLUSION

INTRODUCTION

Le manuel d'utilisation est ce document établi après la réalisation d'une application d'un logiciel ou d'une plateforme. Il renseigne sur la question de « comment utiliser le logiciel, l'application ou la plateforme ? », cette question qui hante notamment les personnes analphabètes de l'informatique. De ce fait, il est crucial pour nous de proposer aux différents de notre application un guide leur permettant de pouvoir aisément effectuer diverses opérations sur notre application.

I – GUIDE D'INSTALLATION

I.1 – Matériel Requis

Le projet étant de nature mobile, il fonctionnera donc dans des téléphones.

En termes de qualité du matériel, un téléphone de 1Go de RAM au minimum et 4Go d'espace libre minimum seront amplement suffisants.

I.2 – Système D'exploitation

Le projet étant développé avec Flutter, qui est un Framework multiplateforme, l'application fonctionnera donc sur tous les systèmes d'exploitation mobiles. Il s'agit de :

- Android
- IOS
- Windows Phone

I.3 – Débit Internet

Afin de profiter de la flexibilité et la rapidité de l'application, il est nécessaire de disposer d'une bonne connexion internet, avec les caractéristiques ci-après :

- 1 Mbits/s ou plus (haut débit toujours recommandé) ;
- Connexion 3G minimum (bien) ou 4G (recommandée).

II – GUIDE D'UTILISATION

En ce qui concerne le lancement de l'application, il faudra dans un premier temps l'installer dans son téléphone. L'application installée, il faudra ensuite la lancer en appuyant tout simplement sur son icône, la page suivante s'ouvrira :

II.1 – Page D'identification Par Numéro De Téléphone

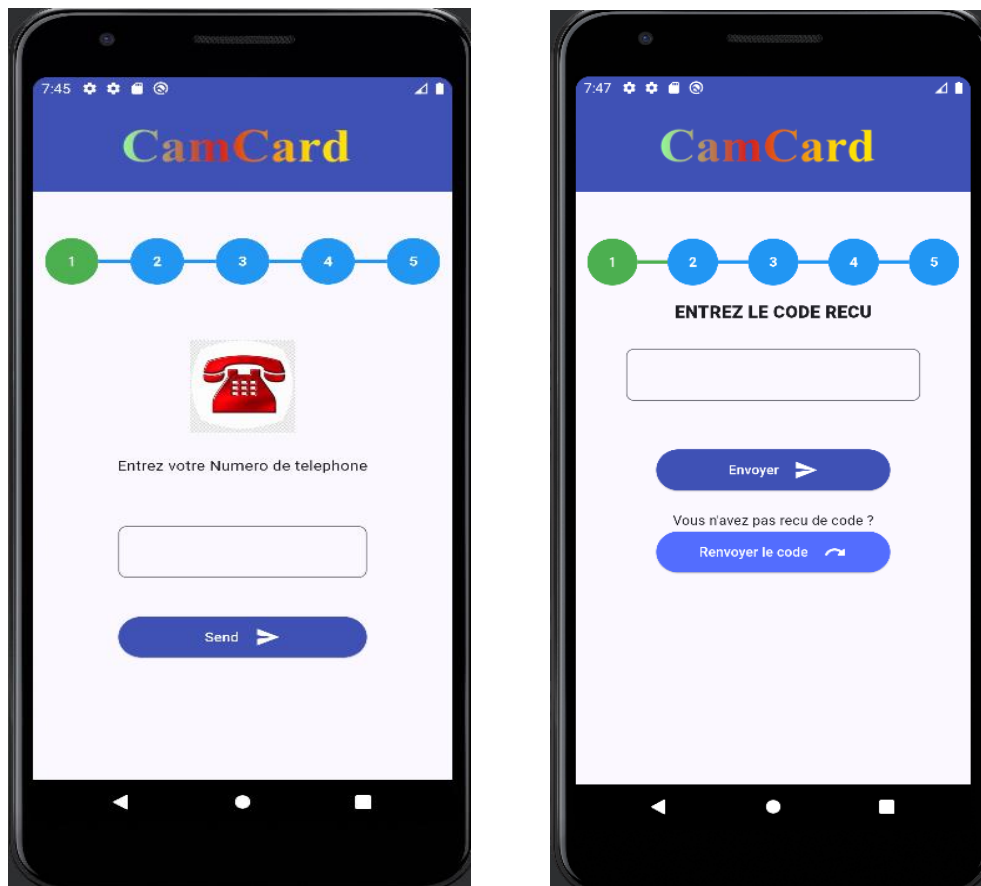


Figure 45: interface d'identification par téléphone

Pour éviter des messages d'erreurs lors du remplissage des champs, nous devons respecter les conditions suivantes :

- **Numéro de téléphone** : ce champ attend un numéro de téléphone constitué de chiffre uniquement. Ce champ est obligatoire.
- **Code de vérification** : ce champ attend un numéro de téléphone constitué de chiffre uniquement. Ce champ est obligatoire.

Après connexion, nous serons redirigés vers notre session de travail.

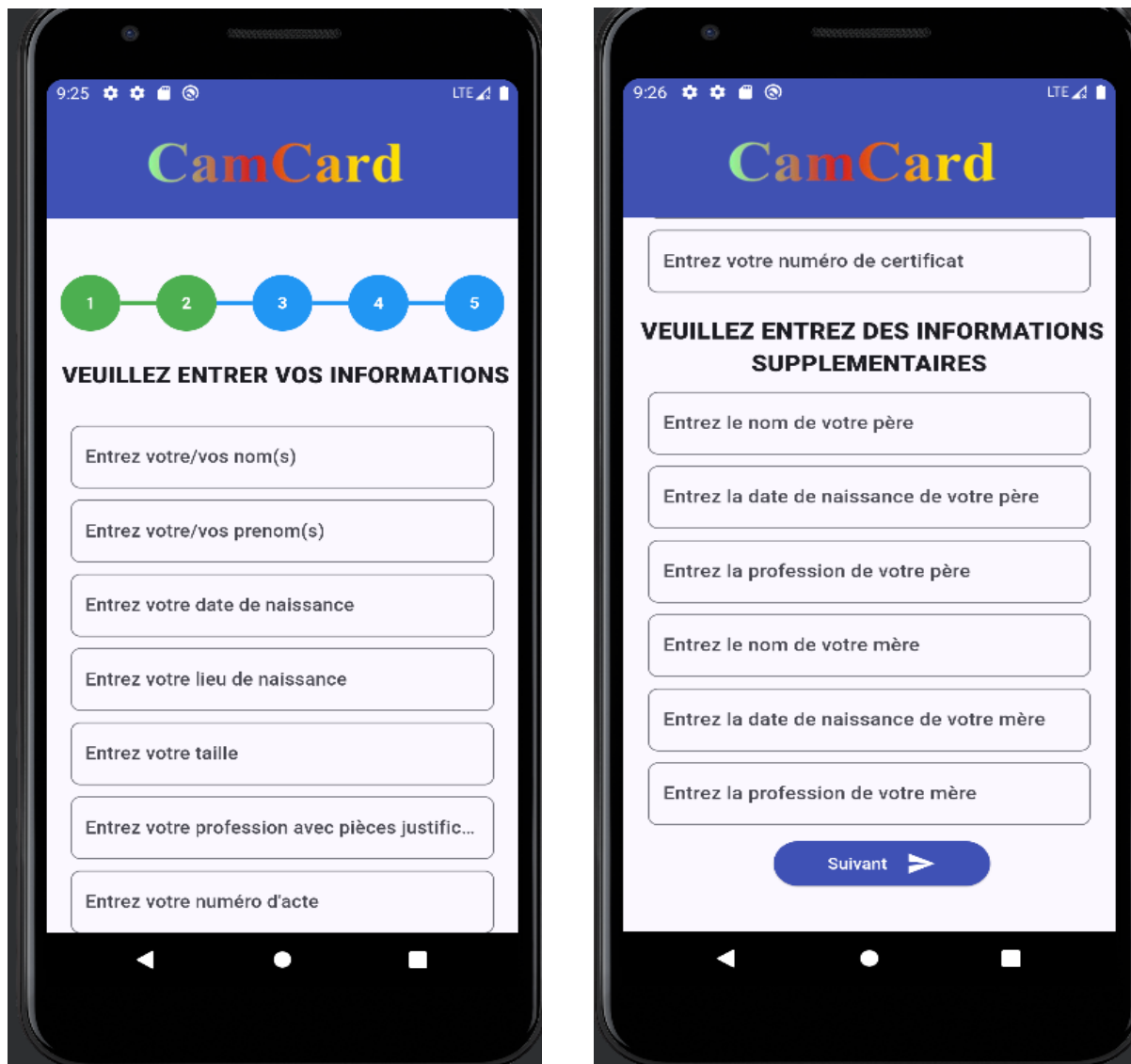
II.2 – Page De Choix De Demande



Figure 46: Interface de choix de demande

Nous pouvons observer sur cette page Trois boutons nous indiquant ici le choix à faire vis-à-vis de la demande.

II.3 – Page De Remplissage Des Formulaires



The image displays two sequential screenshots of the CamCard mobile application interface, showing the form-filling process.

Left Screenshot (Step 1): The screen shows the CamCard logo at the top. Below it is a progress indicator with five numbered circles (1 to 5). Below the progress indicator is the text "VEUILLEZ ENTRER VOS INFORMATIONS". The form contains the following input fields:

- Entrez votre/vos nom(s)
- Entrez votre/vos prenom(s)
- Entrez votre date de naissance
- Entrez votre lieu de naissance
- Entrez votre taille
- Entrez votre profession avec pièces justific...
- Entrez votre numéro d'acte

Right Screenshot (Step 2): The screen shows the CamCard logo at the top. Below it is a text input field for "Entrez votre numéro de certificat". Below this is the text "VEUILLEZ ENTREZ DES INFORMATIONS SUPPLEMENTAIRES". The form contains the following input fields:

- Entrez le nom de votre père
- Entrez la date de naissance de votre père
- Entrez la profession de votre père
- Entrez le nom de votre mère
- Entrez la date de naissance de votre mère
- Entrez la profession de votre mère

At the bottom of the right screenshot is a blue button labeled "Suivant" with a right arrow icon.

Figure 47: Interface de remplissage de formulaire

Il s'agit ici de la page relative au formulaire où les utilisateurs pourront entrer leur information afin de finaliser leur demande.

II.4 – Page De Paiement



Figure 48: Interface de paiement

Ici nous avons affaire à la page de paiement qui permettra à l'utilisateur de payer le service afin de finaliser sa demande.

II.5 – Page De Choix De Demande



Figure 49: interface de téléchargement de fiche PDF

Ici nous avons affaire à la dernière page de notre processus, cette dernière est celle du téléchargement de sa fiche en PDF, il vous suffit juste de cliquer sur le bouton ‘télécharger en PDF’

CONCLUSION

Les guides d'installation et d'utilisation nous ont permis de présenter les conditions nécessaires pour l'utilisation de l'application dans un premier temps et dans un second temps de présenter le fonctionnement général des principales pages et les fonctions mères de ladite application. Nous retiendrons donc qu'avoir un système d'exploitation à jour avec les ressources matérielles sans exagération permettrait d'utiliser de façon optimale l'application.

CONCLUSION GENERALE

L'objectif de ce projet de stage académique était de développer une plateforme mobile de pre-enrolment et de suivi de statut de CNI. Ce projet a nécessité plusieurs étapes, dont la première a été l'insertion dans la structure dans laquelle on devait effectuer nos trois mois de stage, s'est suivie l'élaboration d'un cahier des charges définissant les spécifications et les contraintes du projet. Ensuite, nous avons procédé à une analyse approfondie pour comprendre les enjeux de sécurité et les besoins des utilisateurs en matière de prévention et d'alerte. La phase de conception a permis de définir l'architecture matérielle et logicielle de la plateforme, tandis que la phase de réalisation a consisté à mettre en œuvre les solutions techniques nécessaires. Enfin, des tests ont été effectués pour valider les fonctionnalités principales de l'application. Tout au long de la réalisation de ce projet enrichissant, nous avons pu mettre en pratique les connaissances acquises lors de notre formation en analyse de projets et en développement. Nous avons également acquis une expérience précieuse dans le domaine de l'informatique et du développement de systèmes d'information, notamment en ce qui concerne les méthodologies d'analyse. Cette expérience nous a permis de nous préparer davantage pour le monde professionnel et d'être mieux préparé à faire face aux défis futurs.

BIBLIOGRAPHIE

- **Livre** : Flutter™ Développez vos applications mobiles multiplateformes avec Dart
 - **Auteur** : Julien TRILLARD
 - **Editeur** : Editions ENI
 - **Date de parution** : juin 2020

WEBOGRAPHIE

- Visite du site : <https://www.youtube.com/watch?v=yxiNLruP7CI>
 - **Objectif de la visite** : S'informer en ce qui concerne les calendriers en flutter
 - **Date et heure de la visite** : 28 Aout 2024 à 09h55
 - **Durée de la visite** : 1h

- Visite du site :
https://www.youtube.com/watch?v=YPipIOiJJMc&list=PLjOFHn8uDrvR-nZtbKtV6NX_-4GaBkGNg&index=3
 - **Objectif de la visite** : S'informer en ce qui concerne les opérations telles que create, read, update et delete en flutter
 - **Date et heure de la visite** : 01 septembre 2024 à 19h49
 - **Durée de la visite** : 3h

- Visite du site : [Phone Authentication in Firebase Flutter 2024 | Login with Phone Number Firebase Flutter | Phone OTP - YouTube](#)
 - **Objectif de la visite** : S'informer en ce qui concerne l'authentification par numéro de téléphone en flutter avec Fire base
 - **Date et heure de la visite** : 03 septembre 2024 à 10h40
 - **Durée de la visite** : 2h

- Visite du site : [Comment créer une application Windows avec Flutter, Compiler et préparer son installateur - YouTube](#)
 - **Objectif de la visite** : S'informer en ce qui concerne la compilation d'applications version desktop.
 - **Date et heure de la visite** : 6 septembre 2024 à 09h55

- **Durée de la visite** : 30min

- Visite du site : **Créer une appli mobile avec Flutter #1 - Installation (youtube.com)**
 - **Objectif de la visite** : S'informer en ce qui concerne l'authentification avec flutter
 - **Date et heure de la visite** : 7 septembre 2024 à 12h50
 - **Durée de la visite** : 1h

- Visite du site : **interface admin avec flutter (crud) - YouTube**
 - **Objectif de la visite** : S'informer en ce qui concerne les interfaces administrateurs
 - **Date et heure de la visite** : 28 Aout 2024 à 09h55
 - **Durée de la visite** : 1h

TABLE DE MATIERES

DEDICACE.....	i
REMERCIEMENTS	ii
SOMMAIRE	iii
LISTE DES TABLEAUX	iv
LISTE DES FIGURES	v
GLOSSAIRE	vii
RESUME.....	viii
ABSTRACT	ix
INTRODUCTION GENERALE.....	1
PARTIE I : PHASE D'INSERTION	2
INTRODUCTION.....	4
I – ACCUEIL ET INTÉGRATION EN ENTREPRISE	5
I.1 – Accueil	5
I.2 – Intégration	5
II – PRÉSENTATION DE L'ENTREPRISE	5
II.1 – Historique	5
II.2 – Situation géographique	6
II.3 – Organigramme	7
II.4 – Missions	8
II.5 – Activités	8
II.6 – Ressources matérielles et logicielles	8
II.6.1 – Ressources matérielles de Mercy Innovation Lab	8
II.6.2 – Ressources Logicielles de Mercy Innovation Lab	10
III – PRÉSENTATION DU THÈME	11
CONCLUSION	12
PARTIE II : PHASE TECHNIQUE	13
DOSSIER 1 : L'EXISTANT	14
INTRODUCTION.....	16
I – PRESENTATION DU THEME	17
II –ETUDE DE L'EXISTANT	17
II.1 – Cas 1 étude du site web PassCam.....	17

II.2 – Cas 2 étude sur l'établissement de la cni (Carte Nationale d'Identité) au Cameroun	18
III – CRITIQUE DE L'EXISTANT	19
IV – PROBLEMATIQUE	20
V – PROPOSITION DE LA SOLUTION	20
CONCLUSION	21
DOSSIER 2 : CAHIER DES CHARGES	22
INTRODUCTION	24
I – CONTEXTE ET JUSTIFICATION	25
I.1 – Contexte	25
I.2 – Justification	25
II – BÉNÉFICIAIRES ET CIBLES	26
II.1 – Bénéficiaires	26
II.2 – Cibles	26
III – OBJECTIFS	26
III.1 – Objectif général	26
III.2 – Objectifs spécifiques	26
IV – EXPRESSION DES BESOINS	27
IV.1 – Besoins fonctionnels	27
IV.2 – Besoins non fonctionnels	28
V – CHARTE GRAPHIQUE	30
V.1 – Logo	30
V.2 – Typographie	30
V.3 – Palette de couleurs	31
VI – PLANIFICATION DU PROJET	31
VI.1 – Intervenants	31
VI.2 – Diagramme de Gantt	32
VII – ESTIMATION DU COÛT DU PROJET	33
VII.1 – Ressources Logicielles	33
VII.2 – Ressources matérielles	33
VII.3 – Ressources humaines	34
VII.4 – Tableau récapitulatif	34
VIII – CONTRAINTES DU PROJET	35
IX – LES LIVRABLES	36
CONCLUSION	37

DOSSIER 3 : LE DOSSIER D'ANALYSE	38
INTRODUCTION	40
I – DEMARCHE D'ANALYSE	41
I.1 – Étude comparative MERISE et UML	41
I.2 – Choix Et Présentation De La Démarche D'analyse	42
I.3 – Étude Comparative Des Processus Unifiés	43
I.4 – Choix Et Présentation Du Processus Unifié	46
II – DIAGRAMMES INTERVENANTS	48
II.1 – Diagramme des Cas d'Utilisation	48
II.1.1 – Formalisme	49
II.1.2 – Exemple De Diagramme De Cas D'utilisation	51
II.1.3 – DIAGRAMME DE CAS D'UTILISATION DU SYSTEME	51
II.1.4 – DESCRIPTION TEXTUELLE DE QUELQUES CAS D'UTILISATION	55
II.2 – Diagramme de Communication	62
II.2.1 – Formalisme	62
II.2.2 – Exemple de diagramme de communication	63
II.2.2 – Quelques diagrammes de communication du système	63
II.3 – Diagramme de Séquence	65
II.3.1 – Formalisme	65
II.3.2 – Exemple de diagramme de sequence	68
II.3.3 – Quelques diagrammes de sequence du système	69
II.4 – DIAGRAMME D'ACTIVITE	73
II.4.1 – Formalisme	73
II.4.2 – exemple de diagramme d'activité	75
II.4.3 – Quelques diagrammes d'activité du système	76
CONCLUSION	80
DOSSIER 4 : LE DOSSIER DE CONCEPTION	81
INTRODUCTION	83
I – DIAGRAMME DE CLASSE	84
I.1 – Formalisme	84
I.2 – Exemple De Diagramme De Classe	85
I.3 – Diagramme De Classe Du Système	86
II – DIAGRAMME D'OBJET	88
II.1 – FORMALISME	89

II.2 – EXEMPLE DE DIAGRAMME D’OBJET	89
II.3 – DIAGRAMME D’OBJET DU SYSTEME	90
III – DIAGRAMME D’ETAT TRANSITION	90
III.1 – Formalisme	91
III.2 – Exemple De Diagramme D’état Transition	92
III.3 – Quelques Diagrammes D’état Transition Du Système.....	92
IV – DIAGRAMME DE PAQUETAGE	93
IV.1 – Formalisme	93
IV.2 – Exemple De Diagramme De Paquetage	94
IV.3 – Diagramme De Paquetage Du Système	94
CONCLUSION	96
DOSSIER 5 : LE DOSSIER DE DEPLOIEMENT.....	97
INTRODUCTION.....	99
I – PRESENTATION DE L’ENVIRONNEMENT LOGICIEL ET DES TECHNOLOGIES	100
I.1 – Environnement Logiciel.....	100
I.2 – Présentation Des Outils	101
II – ARCHITECTURE LOGICIELLE ET MATERIELLE	103
II.1 – Architecture Logicielle	103
II.1.1 – Le modèle	103
II.1.2 – La vue	103
II.1.3 – Le contrôleur.....	103
II.2 – Architecture Matérielle	104
II.2.1 – Couche de présentation.....	105
II.2.2 – Couche de logique métier	105
II.2.3 – Couche d’accès aux données persistantes.....	105
III – DIAGRAMMES INTERVENANTS	106
III.1 – Diagramme De Déploiement.....	106
III.1.1 – Formalisme.....	106
III.1.2 – Exemple de diagramme de déploiement	107
III.1.3 – Diagramme de déploiement du système.....	107
III.2 – Diagramme De Composant	108
III.2.1 – Formalisme.....	108
III.2.2 – Exemple de diagramme de composant.....	108
III.2.3 – Diagramme de composant du système	109

CONCLUSION	110
DOSSIER 6 : TEST DE FONCTIONNALITES	111
INTRODUCTION.....	113
I – FONCTIONNALITES.....	114
I.1 – La Fonction D’authentification.....	114
I.2 – La Fonction Pour Télécharger Une Fiche en PDF.....	114
II – TESTS.....	115
II.1 – Test Unitaire De La Fonction D’authentification	116
II.2 – Test Unitaire De La Fonction Télécharger Fiche En PDF.....	118
CONCLUSION	120
DOSSIER 7 : GUIDE D’INSTALLATION ET GUIDE D’UTILISATION	121
INTRODUCTION.....	123
I – GUIDE D’INSTALLATION.....	124
I.1 – Matériel Requis	124
I.2 – Système D’exploitation.....	124
I.3 – Débit Internet	124
II – GUIDE D’UTILISATION	124
II.1 – Page D’identification Par Numéro De Téléphone	125
II.2 – Page De Choix De Demande	126
II.3 – Page De Remplissage Des Formulaires	127
II.4 – Page De Paiement	128
II.5 – Page De Choix De Demande	129
CONCLUSION	130
CONCLUSION GENERALE	131
TABLE DE MATIERES	xv
ANNEXE	xx

ANNEXE