



DÉDICACE

À

la famille GANOTA.



REMERCIEMENTS

C'est avec un grand honneur et un grand plaisir que Nous dédions ces quelques lignes afin de remercier toute personne ayant contribué de près ou de loin à l'accomplissement de ce travail :

- Pr MOHAMMADOU BOUBA Adj, Directeur de l'Institut Universitaire de Technologie de Ngaoundéré pour le travail qu'il effectue pour nous ;
- Dr DANGBE Ezekiel, Chef de la Division de la Formation Initiale à l'Institut Universitaire de Technologie Ngaoundéré pour son travail au niveau de cette division ;
- Pr YENKE Blaise Omer, Chef du Département de Génie Informatique qui a rendu possible ce stage ;
- Dr MBALA Remy, notre encadreur pour son soutien, ses conseils et son aide pour la réussite de ce projet.
- Pr NDAM NJOYA Arouna, Chef de la Division des Stages à l'Institut Universitaire de Technologie de Ngaoundéré, pour avoir rendu le stage possible au niveau de l'entreprise ;
- M WOUPO Cyrille, Directeur d'IMEDIATIS Sarl pour nous avoir accepté dans son entreprise et pour ses conseils ;
- M MAMBOU Jospin, Ingénieur Génie Logiciel pour son souci et ses conseils scrupuleux pour l'avancement de notre travail ;
- M MOUKATE NGUINI ROGER Manfred, Ingénieur en Génie Logiciel pour ses conseils et aide pour l'avancement de notre projet.
- Mes parents M. et Mme GANOTA, pour leurs précieux conseils et leur soutien morale et financier ;
- A tous les enseignants du département de génie logiciel de l'institut Universitaire de Technologie Ngaoundéré pour les cours et les conseils qu'ils nous ont donnés pour parfaire notre formation ;
- A tout le personnel de l'Institut Universitaire de Technologie Ngaoundéré pour leur apport dans le travail que doit effectuer l'IUT pour la formation des étudiants ;
- A tous mes camarades de la promotion et aînés académiques pour leur précieux soutien ;
- A tous mes frères et sœurs pour leur soutien.



TABLE DES MATIÈRES

Contents

DÉDICACE.....	I
REMERCIEMENTS	II
TABLE DES MATIÈRES	III
LISTE DES ABRÉVIATIONS	VI
LISTE DES TABLEAUX.....	VII
LISTE DES FIGURES.....	VIII
Présentation de l'Entreprise	IX
1. Histoire et secteur d'activité.....	IX
2. Structure organisationnelle et situation géographique	IX
RÉSUMÉ.....	1
ABSTRACT	2
INTRODUCTION GÉNÉRALE.....	3
CHAPITRE 1 : PRÉSENTATION DU PROJET ET LANGAGE DE MODÉLISATION.....	4
I. Contexte et justification	4
1. Contexte	4
2. Justification	4
II. Problématique.....	5
III. Objectif de recherche	5
Conclusion.....	6
CHAPITRE II : ANALYSE ET CONCEPTION DE LA SOLUTION	7
I. Spécification des exigences.....	7
I.1. Exigences fonctionnelles	7
I.1.1. La gestion des utilisateurs	7
I.1.2. Gestion des données	7
I.1.3. Évaluation des présences.....	7
I.2. Exigence non-fonctionnelle	8
II. Méthode et langage	8
a. Acteurs.....	11
b. Cas d'utilisation	12
II.1. Diagramme de cas d'utilisation.....	13
II.1.1. Description contextuelle des différents cas d'utilisation	14



II.1.1.1.	14
II.1.1.2. Cas d'utilisation « Authentification »	14
II.1.1.3. Cas d'utilisation « générer rapport de présence »	15
II.1.1.4. Cas d'utilisation Enregistrer un personnel	15
II.1.1.5. Cas d'utilisation modifier les données d'un personnel	15
II.1.1.6. Cas d'utilisation Enregistrer un appareil	16
II.1.1.7. Cas d'utilisation Modifier un appareil	16
II.1.1.8. Cas d'utilisation ajouter un utilisateur	17
II.2. Diagramme de séquence	17
II.2.1. Diagramme de séquence pour créer un compte	18
II.2.2. Diagramme de séquence relative à l'authentification	19
II.2.3. Diagramme de séquence relatifs à la gestion des personnels	19
II.2.4. Diagramme de séquence relatifs à la gestion des appareils	21
II.2.5. Diagramme de séquence relatifs à la gestion des rapports et statistique	22
II.3. Diagramme de classe	23
II.4. Passage au modèle relationnel	25
II.4.1. Fondements	25
II.4.2. Les règles de passage	25
II.5. Modèle relationnel	26
II.6. Architecture MVC	27
II.6.1. Modèle	27
II.6.2. Vue	27
II.6.3. Contrôleur	27
II.7. Avantage de l'architecture MVC	27
conclusion	28
CHAPITRE 3: IMPLÉMENTATION	29
I. Outils de développement	29
I.1 Environnements matériels	29
I.2 Environnements logiciels	30
II. Environnement de programmation	33
II.1. PHP	33
II.2. CSS	33
II.3. HTML	33
II.4. Bootstrap	34
II.5. Javascript	35



II.6.	CodeIgniter	35
III.	Interface de notre application	36
III.1.	Page de l'authentification ou de connexion	36
III.2.	Page gérer les personnels	37
III.3.	Page ajouter un personnel	37
III.4.	Page gérer les appareils.....	38
III.5.	Page ajouter un appareil.....	39
III.6.	Page pour générer le rapport	39
III.7.	Page pour gérer le profil	40
CONCLUSION ET PERSPECTIVE		41
BIBLIOGRAPHIE		42
WEBOGRAPHIE.....		42



LISTE DES ABRÉVIATIONS

API : Application Programming Interface

CMS : Systèmes de gestion de contenu

CSS : Cascading Style Sheets

HTML : Hypertext Markup Language

HTTP : HyperText Transfer Protocol

IDE : Integrated Development Environment ou Environnement de Développement Intégré

IUT : Institut Universitaire de Technologie

JS : JavaScript

MVC : Model View Contrôler

PHP : Hypertext Preprocessor

PME : petites et moyennes entreprises

RH : Ressources Humaines

SARL : Société à Responsabilité Limitée

SGBDR : Système de Gestion de Bse de Données Relationnelles

SQL : Structured Query Language

TIC : Technologie de l'Information et de la Communication

UML : Unied Modeling Language

UP: Processus Unifié

VLAN : Virtual Local Area Networks

VSCode : Visual Studio Code

W3C : Word Wide Web Consortium

XAMPP : Cross- Platform(X) Apache(A) MySQL(M) PHP(P) Perl(P)

XSS : Cross-Site Scripting



LISTE DES TABLEAUX

<i>Tableau 1: Fiche d'identification d'IMEDIATIS Sar</i>	<i>XI</i>
<i>Tableau 2: Vue d'ensemble des diagrammes UML</i>	<i>9</i>
<i>Tableau 3: Cas d'utilisation S'authentifier.....</i>	<i>14</i>
<i>Tableau 4: Cas d'utilisation générer le rapport</i>	<i>15</i>
<i>Tableau 5: Cas d'utilisation enregistrer un personnel.....</i>	<i>15</i>
<i>Tableau 6: Cas d'utilisation modifier les données d'un personnel</i>	<i>15</i>
<i>Tableau 7: Cas d'utilisation ajouter un appareil</i>	<i>16</i>
<i>Tableau 8: Cas d'utilisation modifier un appareil.....</i>	<i>16</i>
<i>Tableau 9: Cas d'utilisation ajouter un utilisateur</i>	<i>17</i>



LISTE DES FIGURES

<i>Figure 1: Organigramme d'IMEDIATIS Sarl</i>	<i>X</i>
<i>Figure 2: Plan de localisation d'IMEDIATIS Sarl</i>	<i>X</i>
<i>Figure 3: Démarche de projet avec le langage UML</i>	<i>11</i>
<i>Figure 4: Représentation d'un acteur</i>	<i>12</i>
<i>Figure 5 Représentation d'un cas d'utilisation</i>	<i>12</i>
<i>Figure 6: Représentation des relations.....</i>	<i>12</i>
<i>Figure 7: Diagramme de cas d'utilisation.....</i>	<i>13</i>
<i>Figure 8: Diagramme de séquence pour créer un compte utilisateur.....</i>	<i>18</i>
<i>Figure 9: Diagramme de séquence pour l'authentification.....</i>	<i>19</i>
<i>Figure 10: Diagramme de séquence pour ajouter un personnel.....</i>	<i>20</i>
<i>Figure 11: Diagramme de séquence pour ajouter un appareil</i>	<i>21</i>
<i>Figure 12: Diagramme de séquence pour gérer le rapport de présences</i>	<i>22</i>
<i>Figure 13: Diagramme de séquence pour consulter le statistique</i>	<i>23</i>
<i>Figure 14: Diagramme de classe</i>	<i>24</i>
<i>Figure 15: Architecture MVC</i>	<i>28</i>
<i>Figure 16: Interface authentification</i>	<i>37</i>
<i>Figure 17: Interface pour gérer les personnels.....</i>	<i>37</i>
<i>Figure 18: Interface ajouter personnel.....</i>	<i>38</i>
<i>Figure 19: Interface gérer un appareil</i>	<i>38</i>
<i>Figure 20: Interface pour ajouter un appareil.....</i>	<i>39</i>
<i>Figure 21: Interface pour générer le rapport</i>	<i>39</i>
<i>Figure 22: Rapport générer</i>	<i>40</i>
<i>Figure 23: Interface pour gérer profil.....</i>	<i>40</i>



Présentation de l'Entreprise

Tout au long de cette partie, nous aurons à présenter l'entreprise IMEDIATIS Sarl en générale (2).

1. Histoire et secteur d'activité

Créée en 2013, IMEDIATIS est une société de conseil en transformation digitale et informatique qui a pour mission le développement des logiciels pour les entreprises locales et l'assistance à de nombreuses petites et moyennes entreprises (PME) dans leur processus de digitalisation. Les éléments entrant dans la motivation de création de cette structure sont les suivants :

- Le besoin d'entreprise faisant dans le développement des solutions informatiques ;
- La demande sans cesse croissante sur le marché de développement des logiciels ;
- La demande d'assistance de nombreuses PME sans service informatique.

La société IMEDIATIS affiche une croissance régulière depuis le début de son existence. Elle a connu une modification de son statut juridique passant ainsi de EURL à sa création à SARL en 2018.

IMEDIATIS SARL, société de conseil en transformation digitale et informatique oriente ses principaux services comme suit :

- Conseil en transformation digitale
- Stratégie digitale des entreprises et audit ;
- Conception et développement de sites Web, applications mobiles et sites d'e-Commerce
- Marketing numérique (Stratégie de marketing numérique, Médias sociaux, référencement, Business Intelligence, Marketing de contenu, SMS Marketing)
- Intégrations d'applications partenaires ;
- Formations professionnelles dans le secteur des TIC ;
- Communication digitale ;
- Création graphique.

2. Structure organisationnelle et situation géographique

IMEDIATIS est composée d'un personnel jeune et dynamique. Bien que n'ayant pas assez de personnel, elle n'échappe pas à la structuration administrative de toute entreprise. Ainsi, elle compte en son sein 2 services (2) :

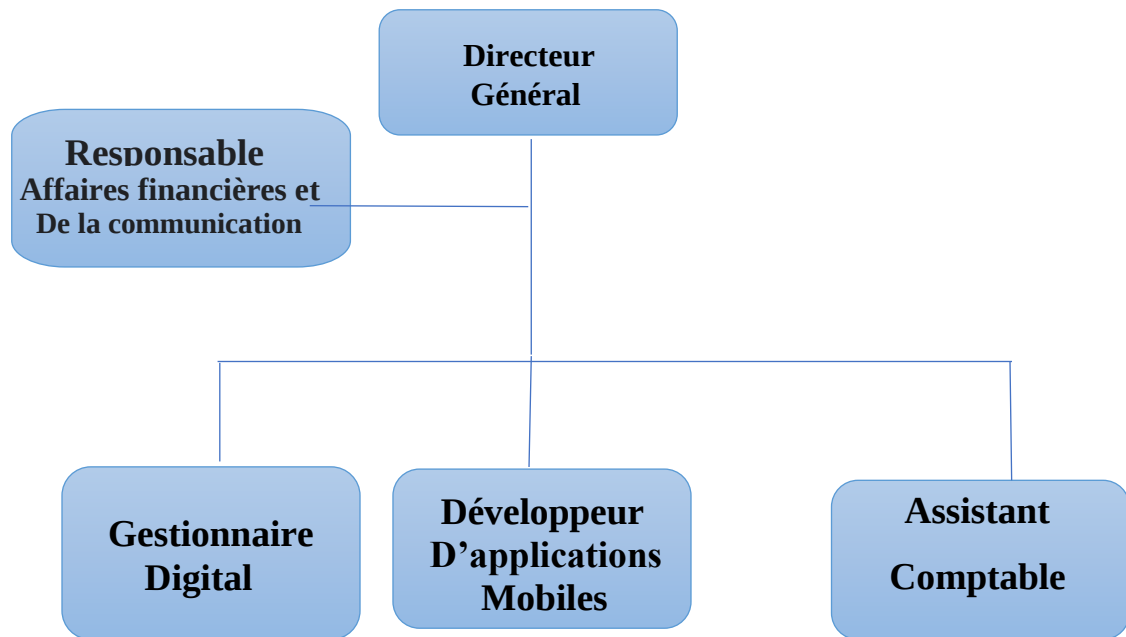


Figure 1: Organigramme d'IMEDIASTIS Sarl

Le reste de l'équipe étant très souvent composé de développeurs freelances ou en contrat à durée déterminée sur des projets ponctuels.

IMEDIATIS est située dans la région du littoral, dans le département du Wouri plus précisément dans l'Arrondissement de Douala 3ème. Elle se trouve à ce jour au 209, BLOC M, MAKEPE. Son plan de localisation est le suivant :

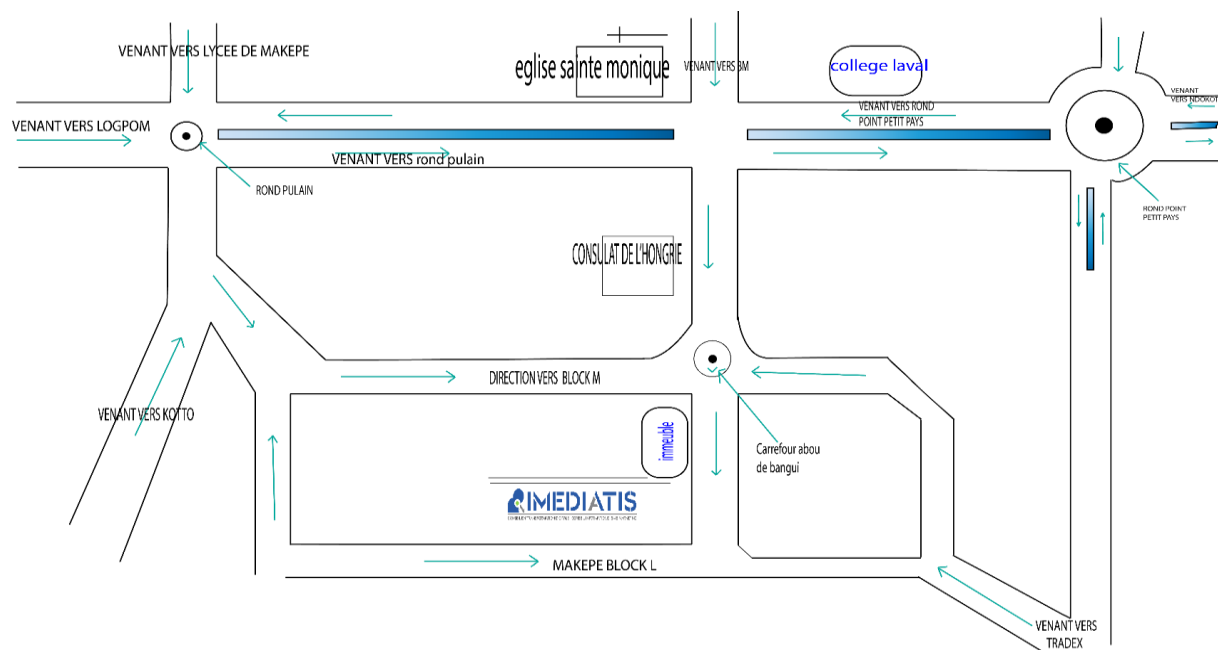


Figure 2: Plan de localisation d'IMEDIATIS Sarl



Tableau 1: Fiche d'identification d'IMEDIATIS Sar

Raison sociale	IMEDIATIS SARL
Logo	 CONSEIL EN TRANSFORMATION DIGITALE - CONSEIL INFORMATIQUE - SMS MARKETING
Siege Sociale	Douala
Boite postal	4 590 Douala – Cameroun
Téléphone	(+237) 233 47 91 84 / 674 17 65 46 / 691 38 84 85
Site internet	http://imediatis.net
Email	hello@imediatis.net
Activités principales	Développement des logiciels, maintenance des applications et à la Formation professionnelle dans le secteur des TIC



RÉSUMÉ

Ce document présente le travail réalisé à IMEDIATIS Sarl pour développer une application de collecte et de traitement de la présence du personnel via des périphériques mobiles.

L'objectif principal était de collecter via un API et de traiter les données de présence des employés connectés au réseau local de l'entreprise, puis de développer une application web en PHP utilisant l'IDE Visual Studio Code et la base de données XAMPP. Le projet s'est concentré sur la création d'une plateforme intuitive permettant à l'entreprises et aux employés de gérer facilement les horaires de chaque employé présent. L'automatisation était la solution retenue pour permettre une gestion efficace et sécurisée des heures de travail.

La méthodologie a inclus une analyse approfondie des besoins des utilisateurs et la modélisation de la base de données avec XAMPP. L'application web sera développée en CodeIgniter. Les principales fonctionnalités développées sont la visualisation des heures de travail, la génération de statistiques générales et génération des rapports.

Cette application de gestion de la présence offre une solution efficace pour aider les gestionnaires à évaluer rapidement la situation de chaque employé. Elle contribue également à améliorer la ponctualité des employés en mettant en valeur leurs compétences clés.

Mots clés : Collecte et de traitement de présence, Compétences, heures de travail, PHP, API.



ABSTRACT

This document presents the work carried out by IMEDIATIS Sarl to develop a staff attendance collection and processing application using mobile devices.

The main objective was to collect and process employee attendance data connected to the company local network, and then develop a web application in PHP using the Visual Studio Code IDE and the XAMPP database. The project focused on creating an intuitive platform allowing companies and employees to easily manage the schedules of each employee present. Automation was the chosen solution to enable efficient and secure management of working hours.

The methodology included a thorough analysis of user needs and database modeling with XAMPP. The web application will be developed using CodeIgniter. The main features developed are the visualization of working hours, the generation of general statistics and report generation.

This attendance management application offers an effective solution to help managers quickly assess the situation of each employee. It also helps improve the employee experience by highlighting their key skills and facilitating their connection with relevant professional opportunities.



INTRODUCTION GÉNÉRALE

IMEDIATIS est une société de conseil en transformation digitale et informatique qui a pour mission le développement des logiciels pour les entreprises locales et l'assistance à de nombreuses petites et moyennes entreprises (PME) dans leur processus de digitalisation. Cette société recoins des nombreuses personnes chaque année comme employés et stagiaires venu de différentes zones. Cependant, le processus actuel de gestion des heures de chaque employé ne permet pas à l'entreprise de bien gérer les heures de chacun et les évaluer en fonction de cela. De plus, les personnes chargées de traiter et gérer les heures ont du mal à identifier rapidement les présences, l'absence et les payer conformément à leur travail.

Pour améliorer l'ancien système de cette société, il est nécessaire de concevoir et de mettre en œuvre un système de collecte et traitement des présences des personnels à partir de périphériques mobiles, qui permettra aux employés de pointer leurs présences en toute sécurité et à l'entreprise de gérer les heures de chacun de ces employés. Le système permettra également à l'entreprise de garder le rapport général de présences de tous les employés avec toutes les détails nécessaires.

En premier lieu, dans le premier chapitre : « présentation du projet et langage de modélisation » nous allons presenter le cadre et le contexte du projet, les raisons qui nous ont poussés à lancer ce projet, la méthodologie optée ainsi le langage de modélisation.

Dans le deuxième chapitre intitulé « Analyse et conception de l'application », nous allons présenter les principaux diagrammes et fonctionnalités de l'application, et ceci après avoir défini les besoins fonctionnels et non fonctionnels afin de bien déterminer et clarifier les principales et importantes fonctionnalités de notre application.

Enfin, dans le chapitre 3 intitulé : « Implémentation » nous allons tout d'abord, énumérer et décrire les outils de développement que nous avons utilisé et illustrer le schéma de navigation de l'application en vous présentant quelques captures d'écrans des interfaces de notre application.



CHAPITRE 1 : PRÉSENTATION DU PROJET ET LANGAGE DE MODÉLISATION

Ce chapitre, nous allons présenter en première temps le contexte du projet, les raisons qui nous ont poussés à lancer ce projet et la solution retenue. Pour terminer ce chapitre, nous définirons notre méthodologie de développement et le langage de modélisation.

I. Contexte et justification

1. Contexte

Imediatiis est une entreprise informatique qui se spécialise dans le développement d'applications mobiles et web, et fournit également des solutions informatiques à des entreprises locales. Dans le cadre de ses activités, Imediatis reçoit régulièrement de nouveaux employés et doit gérer efficacement leurs heures de travail, ce qui permet d'évaluer la performance de chaque employé et de suivre l'évolution de l'entreprise. Le système de gestion des présences qui sera mis en place est un outil très efficace pour améliorer l'ancienne méthode de suivi des heures travaillées par ses membres, permettant ainsi une gestion plus rigoureuse et précise du temps de travail, au bénéfice de l'entreprise et de ses employés.

2. Justification

Une application de gestion des présences des employés, basée sur la collecte automatique des heures de présence, présente de nombreux avantages. Elle permet aux employés de pointer leur présence en toute sécurité à l'aide de leur téléphone portable ou d'objets connectés mobiles au réseau local de l'entreprise, favorisant ainsi le professionnalisme et mettant l'accent sur les heures travaillées plutôt que sur d'autres critères potentiels.

De plus, ce système permet une meilleure adéquation entre les besoins de l'entreprise et le temps de travail effectif de chacun, réduisant ainsi les pertes de temps qui peuvent handicaper l'entreprise. Grâce aux rapports générés par l'application, l'entreprise peut facilement identifier en quelques clics les présences de chaque employé par service, ce qui diminue les risques liés aux retards et absences du personnel.



II. Problématique

Le développement de ce système soulève plusieurs problématiques et défis, mais nous pouvons retenir parmi toutes ceux existants :

Une évaluation précise des heures de travail : comment concevoir des méthodes de collecte et traitement de présences des personnels fiables et objectifs qui permettra de déterminer avec plus et précision les heures de chaque employé ? la résolution de cette problématique sera l'objet pour le développement de cette application et l'adoption innovante de notre plateforme pour la collecte et le traitement de présences de personnel via des périphériques mobiles. Plateforme web innovante pour la gestion des présences base sur la gestion des heures. Elle nécessitera une réflexion approfondie, des recherches complémentaires et la mise en œuvre de meilleures pratiques pour garantir l'efficacité, l'équité et la pertinence de cette approche.

III. Objectif de recherche

L'objectif est de concevoir un système de collecte et traitement de présences des personnels via des périphériques mobiles connecté au réseau local de l'entreprise afin d'évaluer les heures des personnels. Les principaux aspects de l'objectif de recherches sont les suivants :

- Collecter et traiter les données de présence : écrire des scripts qui vont utiliser les APIs du routeur ou du réseau local pour collecter les données de présence des personnels qui sont connecter à ce réseau.
- Concevoir une plateforme web : le développement d'une architecture et une interface conviviales pour une plateformes web dédiée à la collecte et traitement des présences. Cette plateforme.
- Évaluer l'efficacité de la plateforme : évaluer l'impact et l'efficacité de la plateforme en termes de réduction du temps de collecte des présences et heures de travail afin de satisfaire l'entreprise dans ces besoins.



Conclusion

En somme, ce premier chapitre nous a permis de jeter les bases de notre projet sur ce thème qui dit : système de collecte et traitement de la présence du personnel à partir d'un périphérique mobile. Dans le chapitre suivant, nous entrerons en détails dans la conception et développement de cette application, explorer ses fonctionnalités clés et son impact sur la gestion des heures des employés.



CHAPITRE II : ANALYSE ET CONCEPTION DE LA SOLUTION

Ce chapitre présentera ce que nous allons réaliser tout au long de notre projet suivant le processus UP, il exposera donc toute la documentation dégagée. Cette documentation mettra en évidence les objectifs de notre application web, ses acteurs ainsi que leurs besoins, qui sont une nécessité incontestable pour se lancer dans la phase de réalisation. L'assurance d'une telle tâche est menée avant tout par une bonne représentation graphique de ces besoins et de tous les aspects du système, et ceci à travers quelques diagrammes UML. Enfin, nous réaliserons le modèle relationnel qui va permettre de présenter la structure statique du système d'information et l'architecture MVC.

I. Spécification des exigences

I.1. Exigences fonctionnelles

Une exigence fonctionnelle est une spécification détaillée des fonctionnalités et des comportements attendus d'un système, d'une application ou d'un logiciel. Notre système est subdivisé en plusieurs modules à savoir :

I.1.1. La gestion des utilisateurs

On trouve :

Gestion de profils des employés : Enregistrer le personnel du service, consulter la liste des personnels du service, consulter les statistiques de chaque employeur du service concerné ;

Gestion de profils des chefs de service : ajouter un service, ajouter un chef de service, modifier le profil du chef de service ;

I.1.2. Gestion des données

Gestion des données prédéfinies dans la base de données : enregistrer, modifier et supprimer les données.

I.1.3. Évaluation des présences

Participer au pointage : cela permet à l'entreprise de bien évaluer les données de présences de chaque employé

Les exigences fonctionnelles doivent être claires, spécifiques, mesurables et vérifiables. Elles servent de base pour le développement, la mise en œuvre et la validation du



système, en s'assurant que les fonctionnalités attendues sont bien présentes et fonctionnent correctement

NB : Pour que les données soient collectées et traitées, les employés ou personnels doivent être connecter au réseau local de l'entreprise.

I.2. Exigence non-fonctionnelle

Les exigences non fonctionnelles, également appelées critères de qualité, décrivent les aspects et les caractéristiques du système qui ne sont pas liés directement aux fonctionnalités, on peut citer :

- **Performance** : Le système doit être capable de traiter un certain nombre d'utilisateurs simultanément et de fournir des temps de réponse rapides pour les opérations courantes.
- **Sécurité** : le système doit mettre en place des mécanismes de sécurité pour protéger les données des utilisateurs, telles que l'authentification, le chiffrement des données, la gestion des droits d'accès
- **Convivialité** : L'interface utilisateur du système doit être intuitive, facile à utiliser et accessible pour différents types d'utilisateurs, y compris ceux ayant des besoins spéciaux.
- **Maintenabilité** : Le système doit être conçu de manière à faciliter la maintenance, les mises à jour et les corrections d'erreurs, avec un code clair, bien documenté et structuré.

Les exigences non-fonctionnelles sont tout aussi importantes que les exigences fonctionnelles, car elles garantissent la qualité, la performance et l'expérience utilisateur du système. Elles peuvent être utilisées comme critères d'évaluation pour mesurer la réussite de la solution développée et pour guider les décisions de conception et de mise en œuvre.

II. Méthode et langage

Comme la plupart des projets, un projet informatique nécessite une phase d'analyse et une phase de conception. La phase d'analyse a pour but la présentation, l'évaluation et la description de façon précise des besoins des utilisateurs. Cette phase permet de bien comprendre les exigences du projet. La phase de conception apporte plus de détails à la solution et clarifie les aspects techniques, tels que l'installation des différentes parties logicielles. Cette phase permet de préciser la manière dont le projet sera réalisé. Pour mener à bien ces deux phases, les équipes de



	Diagramme de profil	Illustre les relations d'usage au moyen de stéréotypes, de conditions aux limites ...
Diagramme de comportement	Diagramme de cas d'utilisation	Représente divers usages
	Diagramme d'activité	Décrit le comportement de différents processus (parallèle) dans un système.
	Diagramme d'état-transition	Documente la façon dont un objet passe d'un état à un autre par le biais d'un événement .t
Diagramme de comportement – Diagramme d'interaction	Diagramme de séquence	Représente le moment des interactions entre les objets.
	Diagramme de communication	Affiche la répartition des rôles des objets au sein d'une interaction.
	Diagramme de temps	Démontre la limitation temporelle pour les évènements qui mènent à un changement d'état.
	Diagramme d'aperçu d'interaction	Montre comment les séquences et les activités interagissent.

UML est un langage de modélisation qui permet de faire la représentation graphique des besoins des utilisateurs avec des diagrammes. Une procédure de projet à l'aide du langage UML pourrait se résumer en quatre étapes qui sont représentées de la manière suivante (2)

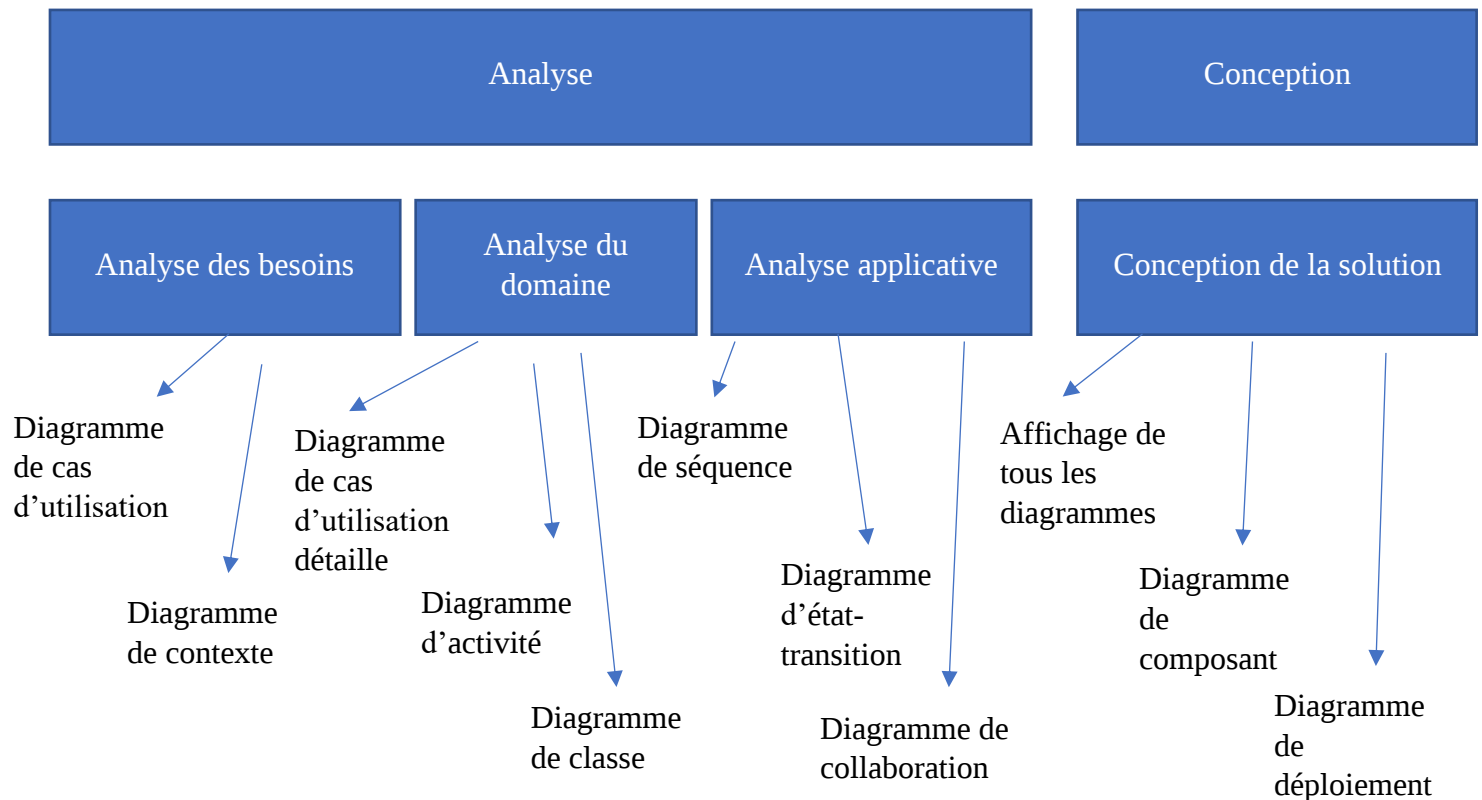


Figure 3: Démarche de projet avec le langage UML

C'est ainsi que pour réaliser notre solution, une démarche qui s'inspire directement du langage UML a été utilisée car d'une part, ce dernier est la base de conception des projets développés par l'IUT et d'autre part car il est un support de communication performant. La conception de notre solution s'inspire donc de sa norme.

a. Acteurs

Un acteur est une personne, un rôle ou une entité (opération, autre système, ...) qui interagit avec le système. Il est représenté par un bonhomme appelé Stikeman ou encore un rectangle. On distingue deux types d'acteurs (1) :

L'acteur principal : c'est celui pour qui le système est construit c'est-à-dire celui pour qui le système rend service,

L'acteur secondaire : est celui qui participe à la réalisation de la tâche dédiée au système ;

On peut représenter un acteur comme suit :

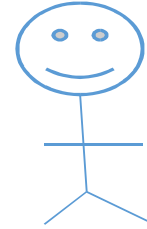


Figure 4: Représentation d'un acteur

b. Cas d'utilisation

Le cas d'utilisation (ou use case) correspond à un objectif du système, motivé par un besoin d'un ou plusieurs acteurs. L'ensemble des use cases décrit les objectifs (le but) du système. Il est formalisé par le diagramme ci-dessous (1):

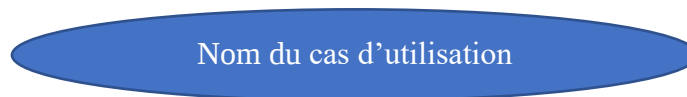


Figure 5 Représentation d'un cas d'utilisation

la relation

Elle exprime l'interaction existant entre un acteur et un cas d'utilisation. Elle est formalisée par le diagramme suivant (3):

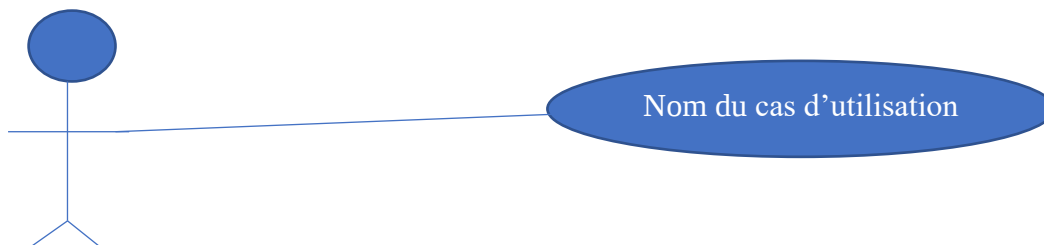


Figure 6: Représentation des relations

Il existe 3 types de relations entre cas d'utilisation :

- La relation de généralisation
- La relation d'extension
- La relation d'exclusion



II.1. Diagramme de cas d'utilisation

Un cas d'utilisation (use case) représente un ensemble de séquences d'actions réalisées par le système, et produisant un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système. L'objectif principal est que l'ensemble des cas d'utilisation doit décrire exhaustivement les exigences fonctionnelles du système. Nous présentons dans cette section le diagramme de cas d'utilisation (2).

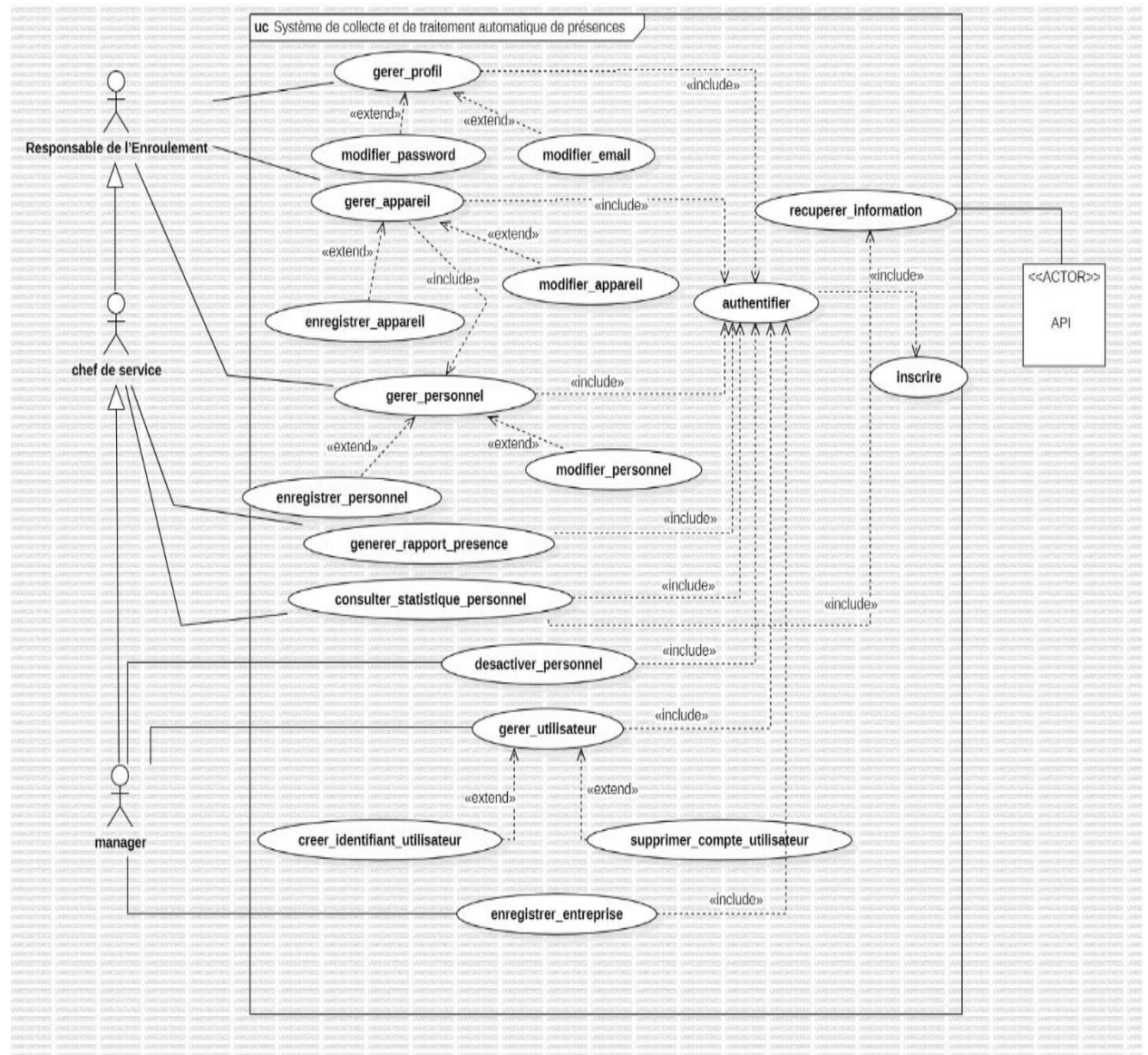


Figure 7: Diagramme de cas d'utilisation



II.1.1. Description contextuelle des différents cas d'utilisation

II.1.1.1.

II.1.1.2. Cas d'utilisation « Authentification »

Tableau 3: Cas d'utilisation S'authentifier

Nom	S'Authentifier
Acteur	Utilisateur
Description	L'utilisateur doit s'authentifier au niveau de la plateforme afin d'accéder aux Fonctionnalités de l'application en fonction de l'utilisateur
Scénario nominal	L'utilisateur renseigne le formulaire d'authentification avec son login et mot de Passe. Le login et le mot de passe sont corrects. Accès à l'application et affichage de l'interface correspondante à chaque acteur.
Scénario alternatif	S'authentifier par un login et un mot de passe. Le login ou/et le mot de passe Sont incorrects. Réaffichage le formulaire d'authentification. Ressaisir du login et le mot de passe.



II.1.1.3. Cas d'utilisation « générer rapport de présence »

Tableau 4: Cas d'utilisation générer le rapport

Nom	Générer rapport de présences
Acteur	Utilisateur
Description	Permet à l'utilisateur de pouvoir : Générer le rapport de présences de chaque employés
Précondition	S'authentifier
Scénario nominal	Après l'authentification, l'utilisateur click sur le bouton rapport et choisi le personnel a la quel il veut générer le rapport, il choit la date du jour et valide l'action.

II.1.1.4. Cas d'utilisation Enregistrer un personnel

Tableau 5: Cas d'utilisation enregistrer un personnel

Nom	Enregistrer un personnel
Acteur	Utilisateur
Description	L'utilisateur doit s'authentifier au niveau de la plateforme afin d'accéder aux Fonctionnalités de l'application qui permettra d'ajouter un personnel
Scénario nominal	L'utilisateur renseigne le formulaire d'ajout de personnel.

II.1.1.5. Cas d'utilisation modifier les donnes d'un personnel

Tableau 6: Cas d'utilisation modifier les données d'un personnel

Nom	Modifier données personnel
Acteur	Utilisateur
Description	Permet aux chefs de services et au manager de modifier les données existantes d'un personnel



Précondition	S'authentifier
Scénario nominal	Après l'authentification, l'utilisateur click sur le bouton Éditer, il modifie les données et clique sur enregistrer et les informations seront mis à jour.

II.1.1.6. Cas d'utilisation Enregistrer un appareil

Tableau 7: Cas d'utilisation ajouter un appareil

Nom	Enregistrer appareil
Acteur	Utilisateur
Description	L'utilisateur doit s'authentifier au niveau de la plateforme afin d'accéder aux fonctionnalités de l'application qui permettra d'ajouter un personnel
Scénario nominal	L'utilisateur renseigne le formulaire d'enregistrement de l'appareil et valide l'action en cliquant sur le bouton enregistrer.

II.1.1.7. Cas d'utilisation Modifier un appareil

Tableau 8: Cas d'utilisation modifier un appareil

Nom	Modifier données appareil
Acteur	Utilisateur
Description	Permet aux chefs de services et au manager de modifier les données existantes d'un appareil
Précondition	S'authentifier
Scénario nominal	Après l'authentification, l'utilisateur click sur le bouton gérer appareil, clique sur l'appareil concerné et modifie ces données. Il clique en fin sur enregistrer et les informations seront mis à jour.



II.1.1.8. Cas d'utilisation ajouter un utilisateur

Tableau 9: Cas d'utilisation ajouter un utilisateur

Ajouter un utilisateur	
Nom	
Acteur	Utilisateur (Super administrateur)
Description	Permet à un super utilisateur de d'ajouter un autre utilisateur qui vont gérer chaque service.
Précondition	S'authentifier
Scénario nominal	Après l'authentification, l'utilisateur clique sur le bouton compte et clique sur ajouter un utilisateur puis renseigne les champs. Il donne le statut a cet utilisateur et peut décider de l'activer ou non et valide.

II.2. Diagramme de séquence

Les descriptions textuelles des cas d'utilisation présentée permettent de communiquer facilement et précisément avec les utilisateurs. En revanche, le texte présente certaines limites puisqu'il est difficile de montrer la succession des enchainements. Il est donc recommandé de compléter la description textuelle par un diagramme de séquence. Dans cette section, nous présentons quelques diagrammes de séquences pour des scénarios susceptibles d'avoir lieu dans notre système, afin de mettre en évidence l'aspect dynamique de l'application (5).



II.2.1. Diagramme de séquence pour créer un compte

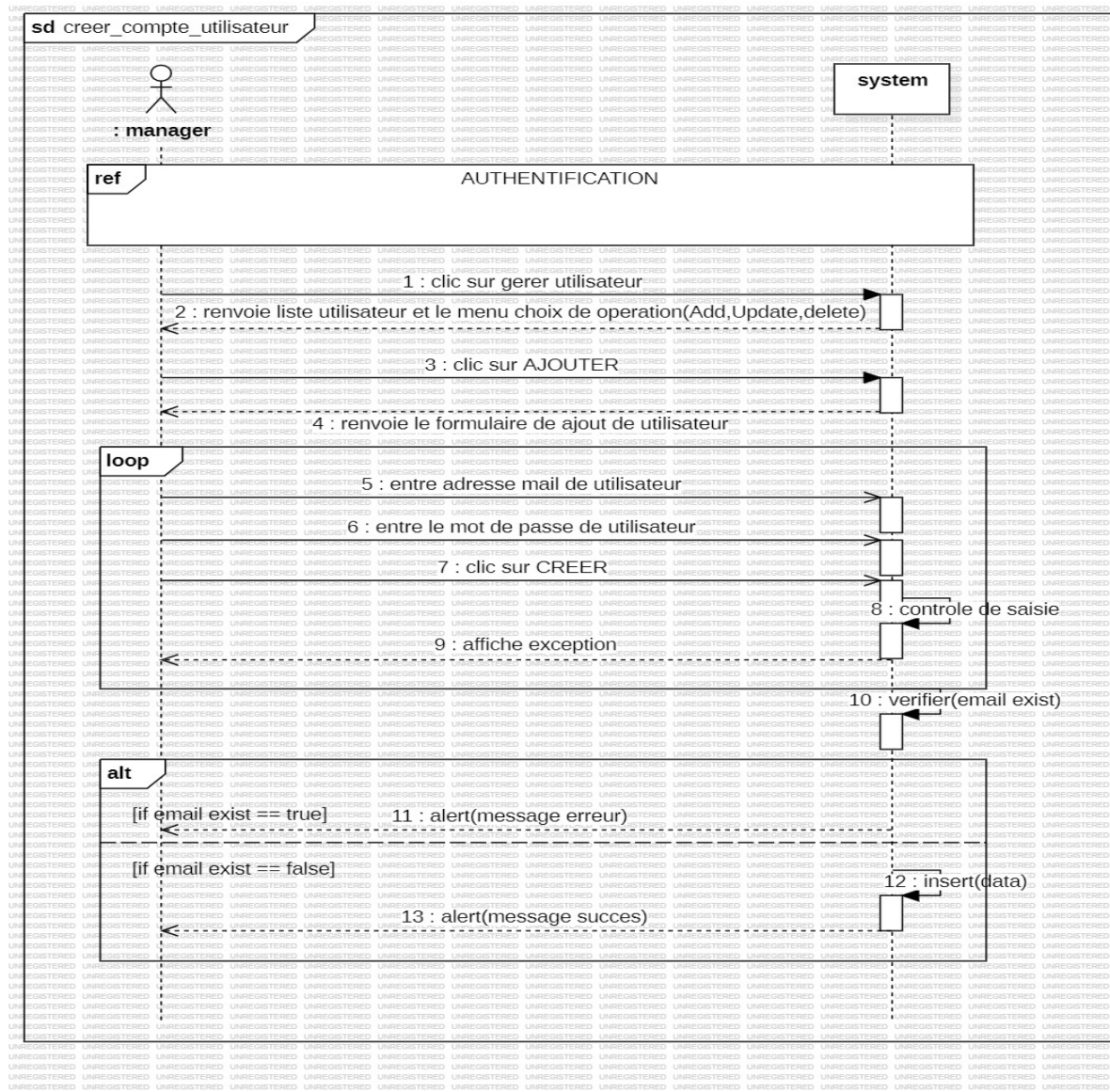


Figure 8: Diagramme de séquence pour créer un compte utilisateur

II.2.2. Diagramme de sequence relative à l'authentification

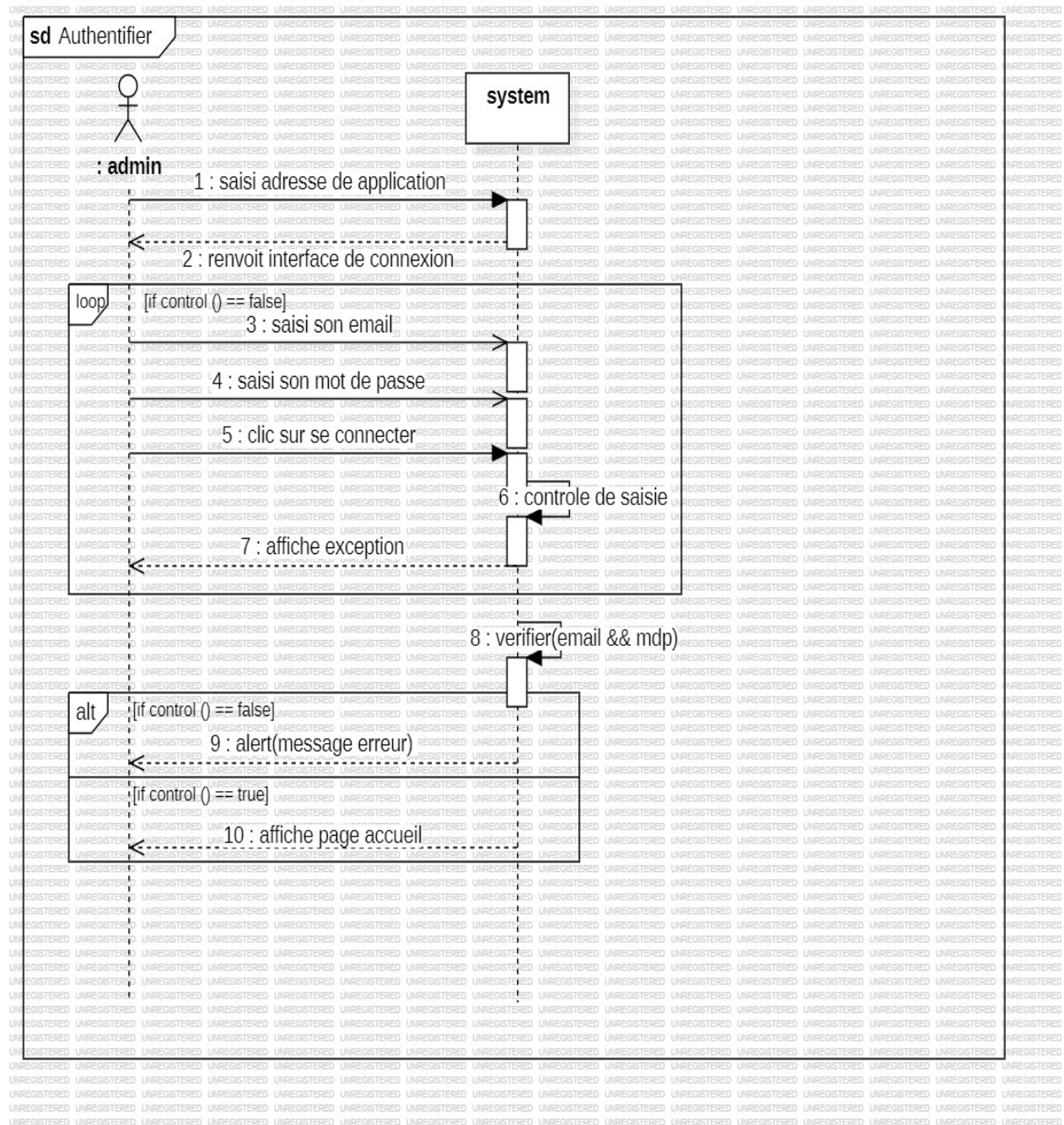


Figure 9: Diagramme de séquence pour l'authentification

II.2.3. Diagramme de séquence relatifs à la gestion des personnels

Dans cette partie nous présentons quelques diagrammes de séquences pour des scénarios relatifs à la gestion des personnels.

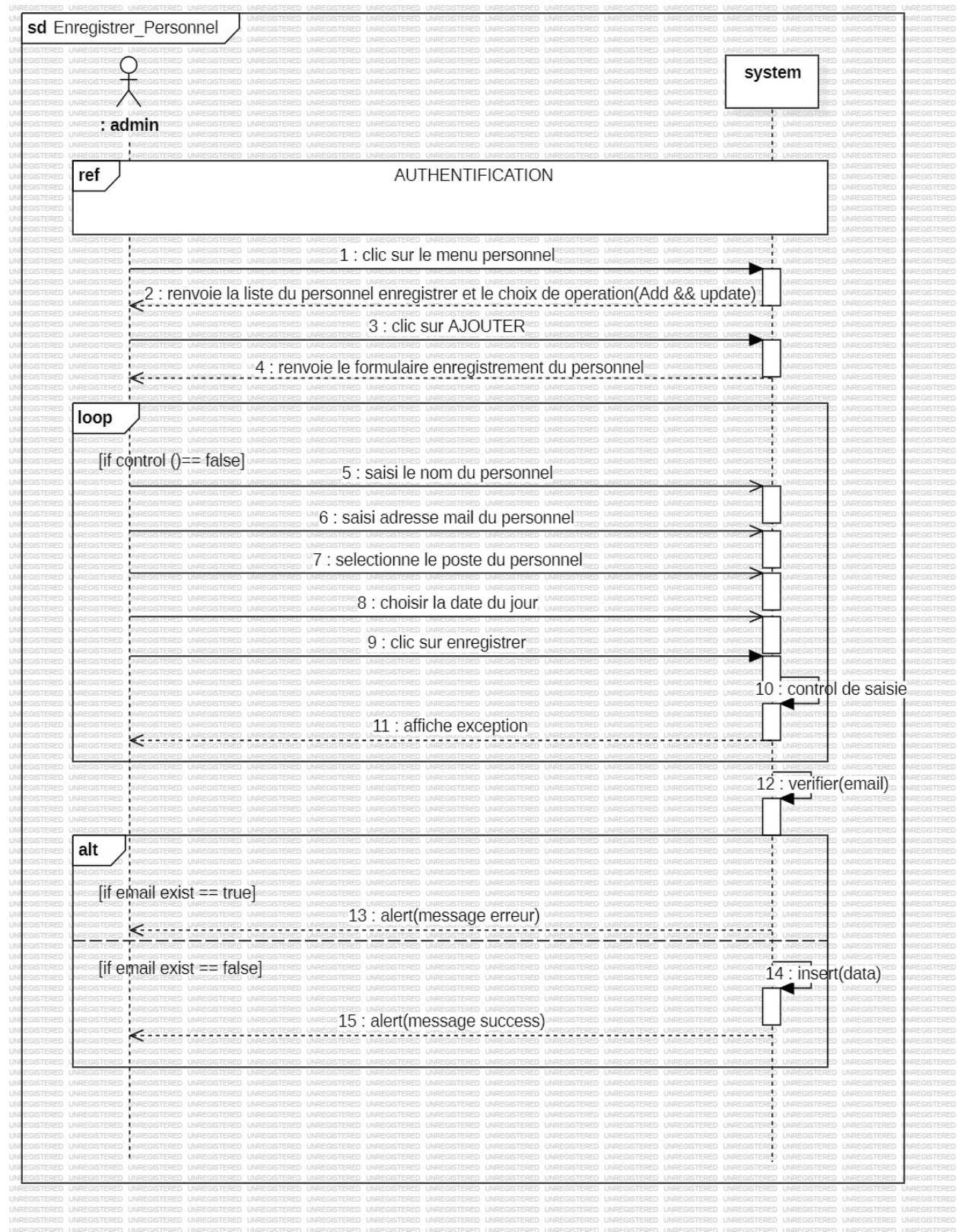


Figure 10: Diagramme de séquence pour ajouter un personnel



II.2.4. Diagramme de séquence relatifs à la gestion des appareils

➤ Ajouter un appareil

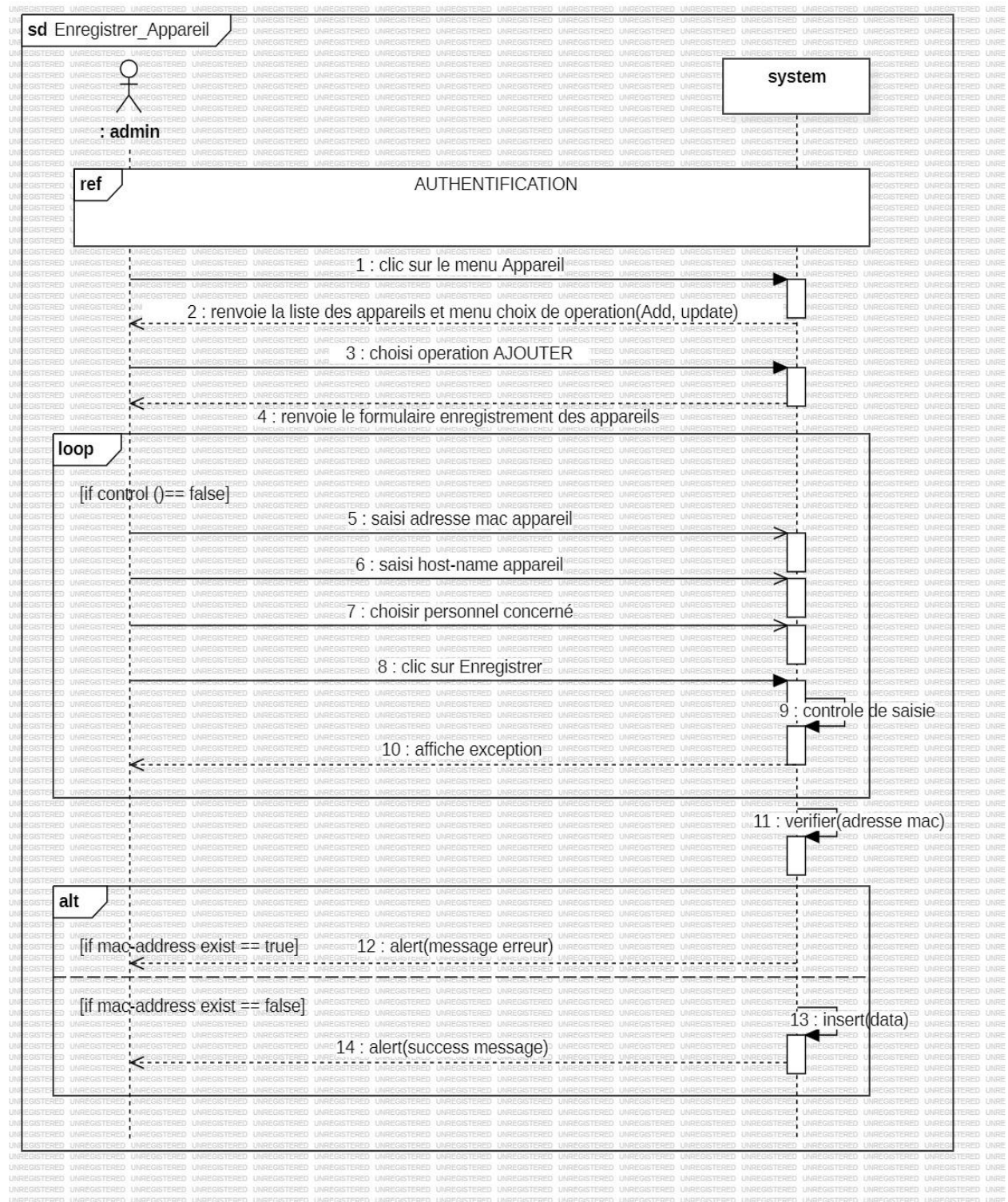


Figure 11: Diagramme de séquence pour ajouter un appareil



II.2.5. Diagramme de séquence relatifs à la gestion des rapports et statistique

➤ Générer le rapport de présences

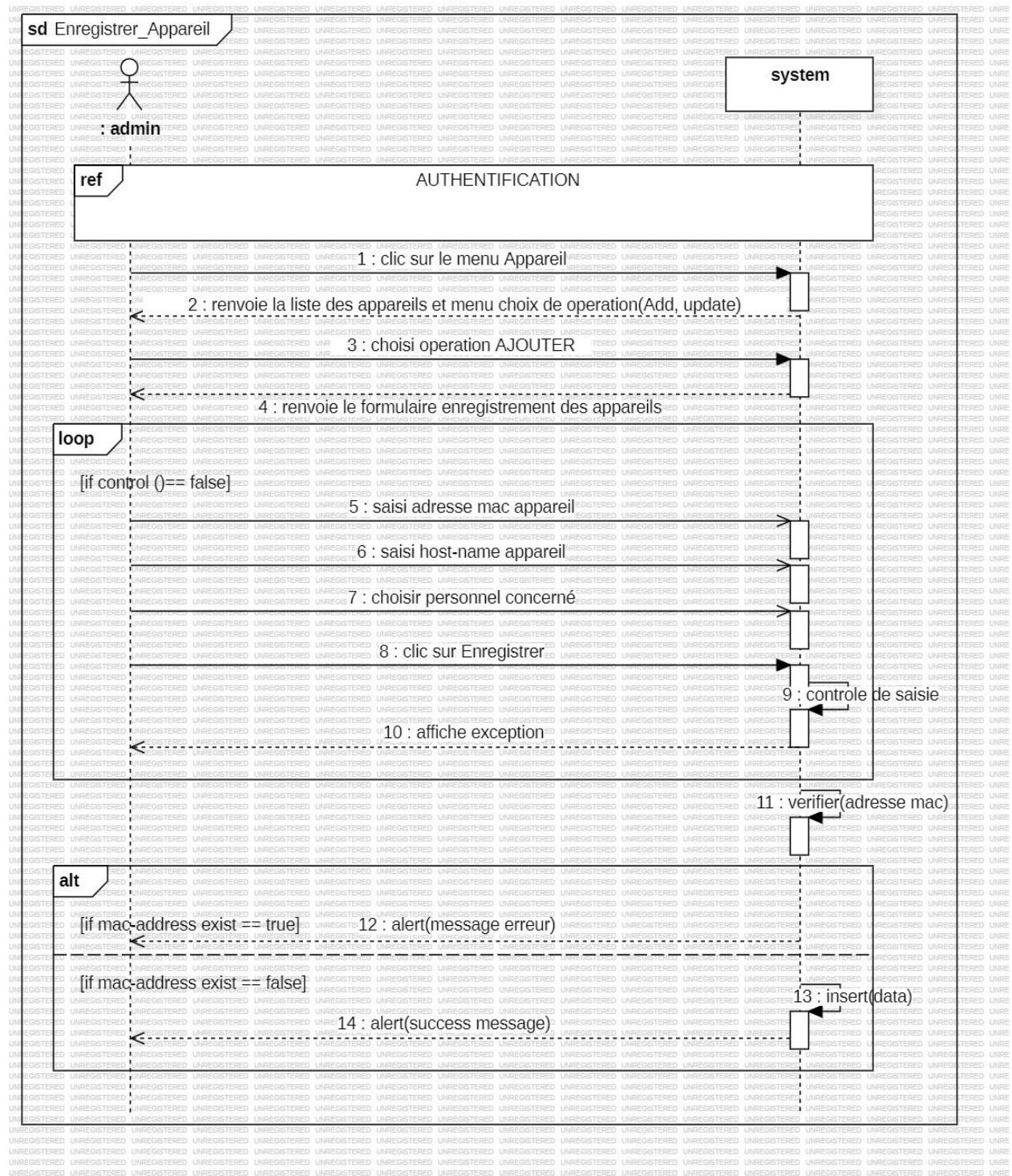


Figure 12: Diagramme de séquence pour gérer le rapport de présences



➤ Consulter le statistique

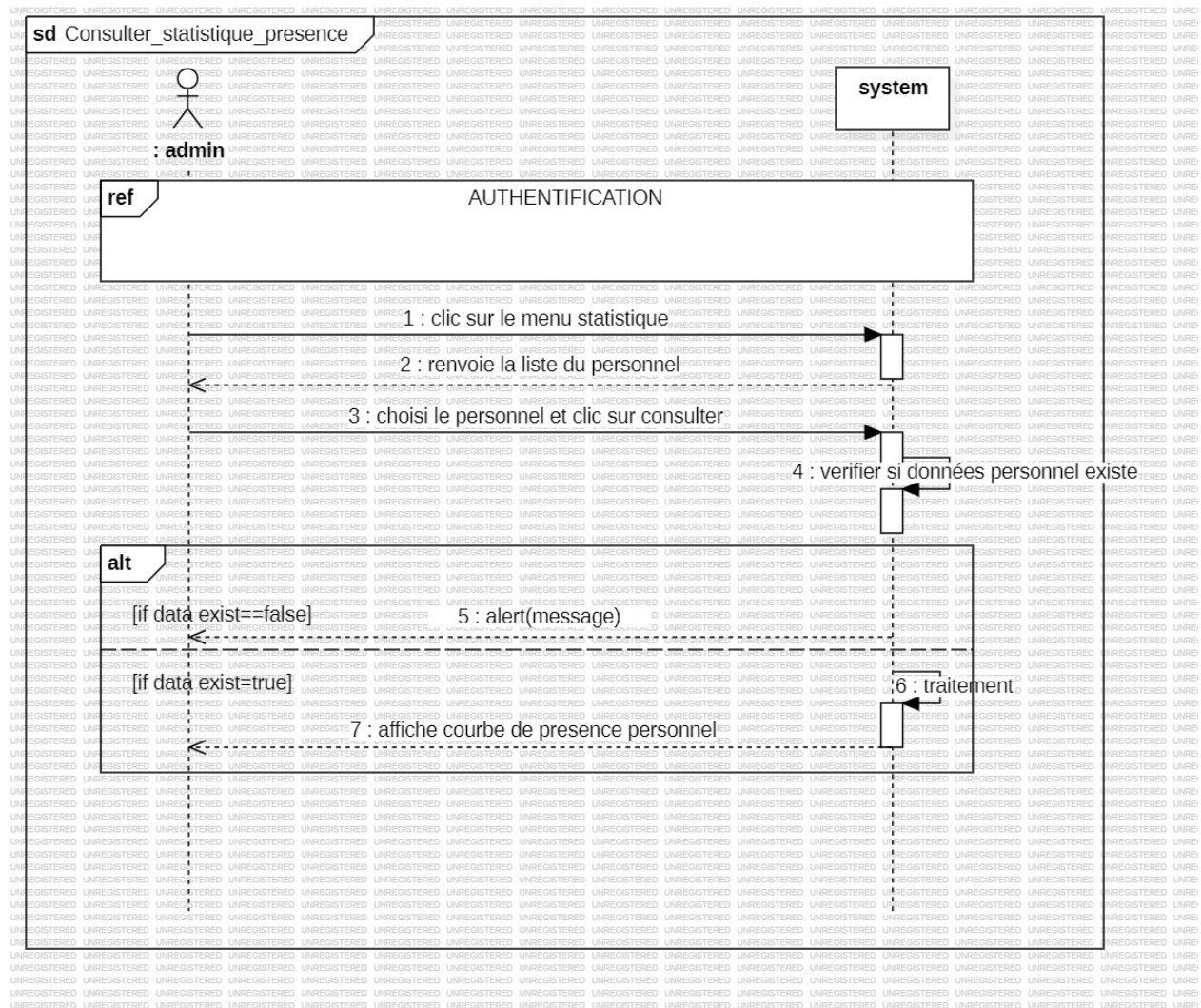


Figure 13: Diagramme de séquence pour consulter le statistique

II.3. Diagramme de classe

Le diagramme de classes fournit une vue globale d'un système en présentant ses classes, ses interfaces, ses collaborations, et les relations entre elles. La figure 14 ci-dessous représente le diagramme des classes de notre application (1).

Dans cette partie nous décrivant les cardinalités entre les différentes entités.

- Un utilisateur est un personnel
- Un utilisateur est identifié par son email et mot de passe
- Un personnel peut n'est pas être un utilisateur



- Un personnel appartient à un service
- Un personnel peut avoir un ou plusieurs téléphones
- Un personnel a un horaire de travail
- Un service contient zéro ou plusieurs personnels
- Chaque service a un horaire lier à l'horaire du service
- Un appareil peut emmètre plusieurs un ou plusieurs fois sa présences
- Chaque personnel peut générer plusieurs rapports de présences pour chaque personnel de son service

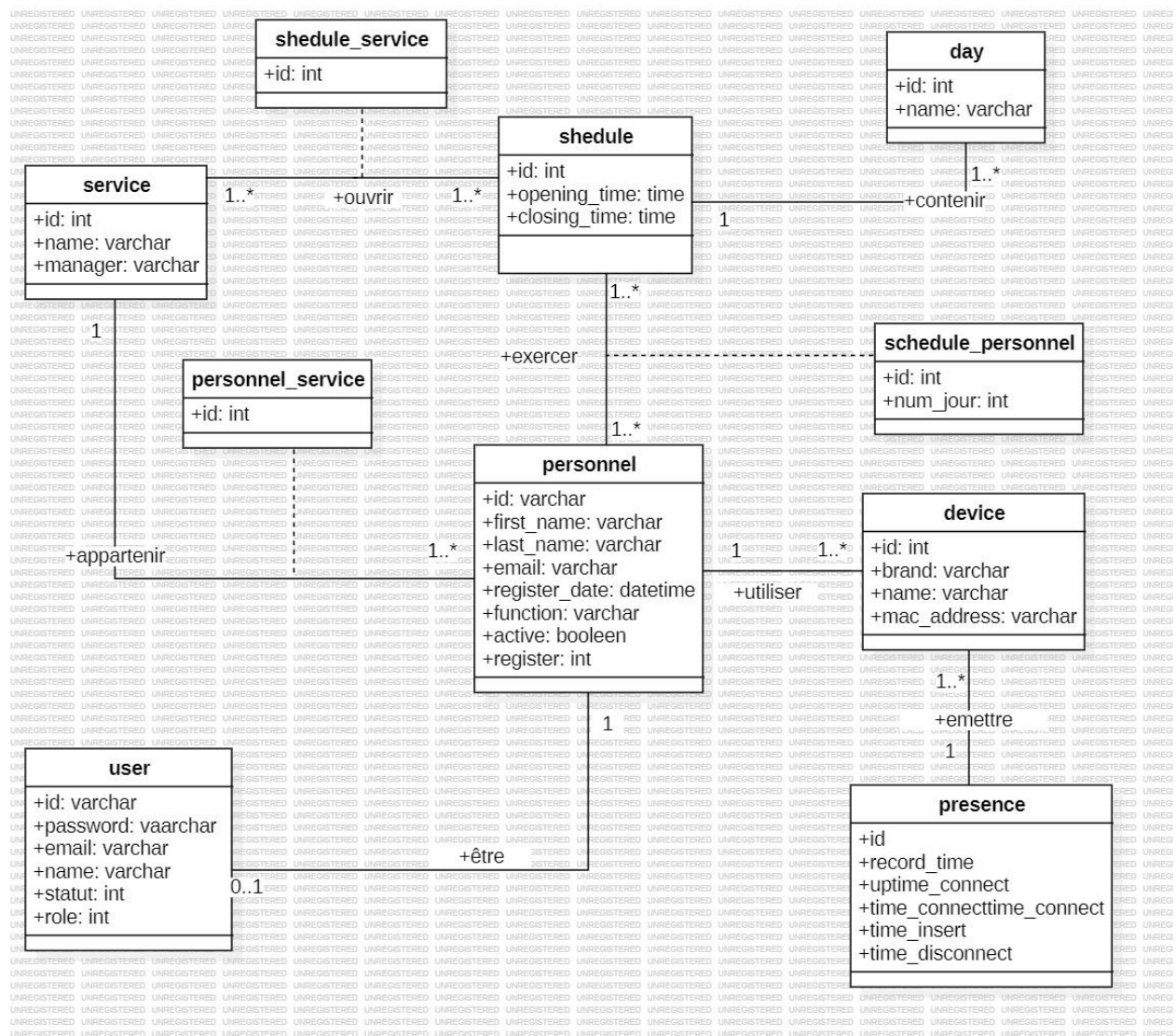


Figure 14: Diagramme de classe



II.4. Passage au modèle relationnel

Le modèle relationnel est un modèle de données qui permet de représenter et de manipuler des données sous forme de tables. Il fournit un cadre conceptuel pour représenter et manipuler des données de manière structurée, indépendamment de leur implémentation physique. Cela permet d'assurer la cohérence et l'intégrité des données, tout en offrant des moyens puissants pour les interroger et les traiter (1).

II.4.1. Fondements

Le modèle relationnel permet de représenter les données que l'on va gérer à l'aide d'un très petit nombre de concepts très simples :

- Les domaines de valeur : représente l'ensemble des valeurs d'un attribut.
- Les relations ou tables : tableau de deux dimensions où les colonnes représentent les domaines et où les lignes contiennent les tuples.
- Les clés : certaines cases doivent avoir des valeurs uniques et non nulles, tel une notion d'identifiant.
- Les clés étrangères : attributs qui sont représentés en une clé primaire dans une autre entité et ceci afin de gérer les relations entre plusieurs tables.

II.4.2. Les règles de passage

La transition du diagramme de classe au modèle relationnel suit des règles bien définies, utilisées par les outils de modélisation pour effectuer cette opération (5).

- Objets :
 - Chaque objet devient une table.
 - Les propriétés de l'objet deviennent des attributs de la table.
 - L'identifiant de l'objet devient la clé primaire de la table.
- Associations sans propriétés propres :
 - Cardinalités (0,1) ou (1,1) vers (0, n) ou (1, n) : la clé de la relation avec la cardinalité (0,1) ou (1,1) est transférée vers la relation à la cardinalité (0, n) ou (1, n), et l'association disparaît. Cette clé est appelée "clé étrangère".
 - Cardinalités (0, n) ou (1, n) vers (0, n) ou (1, n) : l'association se transforme en une relation avec comme clé la concaténation des clés des 2 relations.



- Associations avec propriétés propres : L'association se transforme en une relation ayant comme clé la concaténation des clés des relations associées à chaque individu, les attributs de cette nouvelle relation sont les propriétés propres de l'association précédente.
- Transformation de l'héritage :
 - Décomposition par distinction : Chaque sous-classe est transformée en une relation. La clé primaire de la surclasse migre dans la (les) relation(s) issue(s) de la (des) sous-classe(s) et devient à la fois clé primaire et clé étrangère.
 - Décomposition descendante (push-down) : Tous les attributs de la surclasse sont transférés dans la ou les relation(s) issue(s) de la (des) sous-classe(s).
 - Décomposition ascendante (push-up) : La (les) relation(s) issue(s) de la (des) sous-classe(s) est (sont) supprimée(s) et ses (leurs) attributs sont transférés dans la relation issue de la surclasse.

II.5. Modèle relationnel

En appliquant les règles bien établies pour transformer un diagramme de classe en un modèle relationnel, telles que la correspondance entre les objets et les tables, la migration des propriétés des objets vers les attributs des tables, et les différents modes de traitement des associations et de l'héritage, nous avons pu aboutir à la conception du schéma relationnel final.

User (id, name, password);

- Personnel (idPersonnel, first_name, last_name, email, register_date, function, active, register);
- Service (idService, name, manager);
- Presence (idPresence, #idDevice, register_time, uptime_connect, time_connect, time_insert, time_disconnect)
- day (idDay, name);
- device(idDevice, #idPersonnel, brand, name, mac_address);
- schedule (idSch, #idDay, opening_time, closing_time);
- schedule_personnel (idScheP, #idPersonnel, #idSch, num_day);
- personnel_service(id, #idPersonnel, #idService)
- schedule_service(id, #idService, #idSch)



II.6. Architecture MVC

Une architecture est un modèle de conception de logiciel. Pour une bonne organisation, un développement rapide et flexible et une maintenabilité, il faut choisir de manière minutieuse l'architecture de l'application (3).

MVC, abréviation de Model View Controller ou (Modèle, Vue, Contrôleur en Français) est un modèle de conception très répandu de nos jours créé pour le développement d'applications spécifiquement des applications Web.

II.6.1. Modèle

Le modèle, ou model, représente les données qui vont être utilisées dans l'application web. C'est ici que va être stockée la data, et tout ce qui permet de la modifier (getters, setters, etc.), que ça soit en local en en distant (base de données). Ce sont littéralement des modèles de données.

II.6.2. Vue

La vue, ou view, c'est l'interface graphique de l'application. C'est via cet élément que vont se faire les interactions entre l'utilisateur et le code métier. Elle ne contient presque aucune logique (contrairement à l'architecture concurrente de MVC dont on parlera plus loin), son but est de construire, à partir de ce que renvoie le serveur, une interface et de l'afficher à l'utilisateur.

II.6.3. Contrôleur

Le contrôleur, ou controller, est l'élément qui contient la logique métier. C'est ici que sont la plupart des algorithmes, calculs, etc. C'est aussi l'intermédiaire principal entre la vue et le modèle. Par exemple, la vue soumet un formulaire au contrôleur, qui gère sa validation via du code métier, et demande au modèle de faire des modifications dans la base de données.

II.7. Avantage de l'architecture MVC

Faisons une liste des avantages qu'à l'architecture MVC (5):

- Facile à maintenir, de par sa séparation obligatoire en fichiers et logique ;
- Le développement peut se faire à plusieurs niveaux en parallèle (création des vues par un développeur front pendant qu'un développeur back travailler sur les contrôleurs);
- La décorrélation ou séparation des vues et de la logique permet de tester le code de manière plus simple et efficace, du moins en théorie.

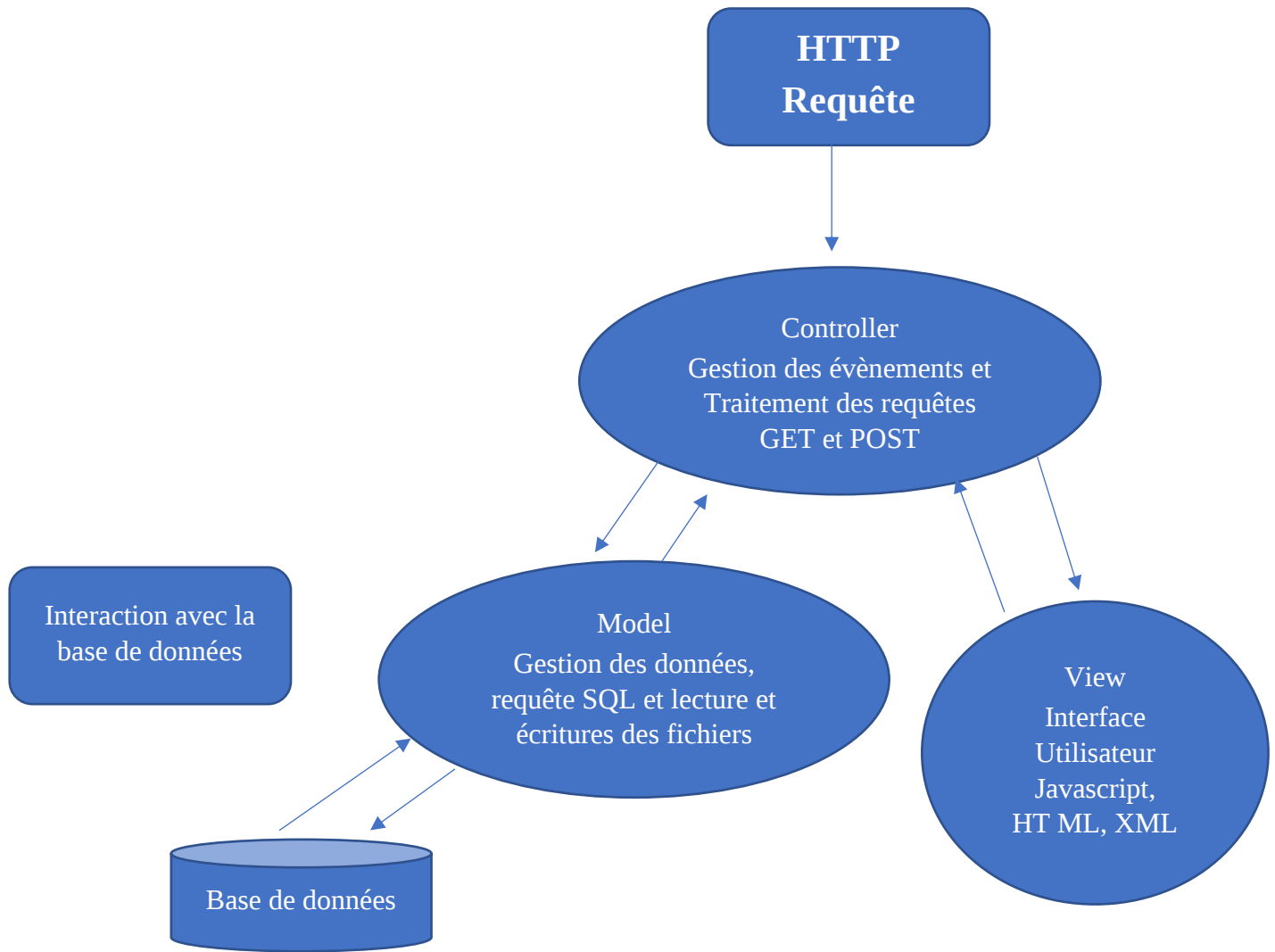


Figure 15: Architecture MVC

conclusion

Ce chapitre a débuté par l'étude des exigences fonctionnelles, suivie de la présentation détaillée des cas d'utilisation, des diagrammes de séquence correspondants ainsi que du diagramme de classes et terminé par l'architecture MVC. Finalement, nous avons conclu ce travail conceptuel par la définition du modèle de données relationnel, fournissant ainsi le schéma de la base de données de l'application. Le prochain chapitre sera consacré à la présentation des outils et des environnements de développement qui seront utilisés pour l'implémentation de notre application, accompagnés de captures d'écran illustrant les différentes fonctionnalités mises en œuvre.



CHAPITRE 3: IMPLÉMENTATION

Tout projet informatique nécessite de choisir les technologies adaptées à sa mise en œuvre. C'est en définissant les outils de développement que nous entamons ce dernier volet. Par la suite, nous allons présenter les interfaces de notre application Web afin de mettre en évidence leur aspect pratique et intuitif, qui a été l'un de nos principaux objectifs.

I. Outils de développement

I.1 Environnements matériels

❖ Ordinateur

- Nom de l'appareil : DESKTOP-KTVP4UP;
- Edition du système installé : Windows 10 Professionnel;
- Type de système : Système d'exploitation 64 bits, processeur x64 ;
- Processeur : Intel(R) Core (TM) i7-6600U CPU @ 2.60GHz 2.80 GHz ;
- Mémoire : 8,00 GB.
- Carte mère : 0T6HHJ Version A00.

❖ Routeur

- Nom du routeur : RouteurBoard
- Marque : Mikrotik
- Model : 951 Ui-2HnD
- Version : 7.1

Description du routeur



Le RouteurBoard 951 Ui-2HnD est un routeur sans fil haut de gamme de la marque Mikrotik, doté d'une API riche et flexible (10).

❖ **Caractéristiques principales :**

- Processeur puissant avec architecture ARM 600 MHz



- 64 Mo de RAM
- 4 ports Ethernet Gigabit
- 1 ports USB
- 1 port PoE in
- Interface sans fil Wi-Fi 802.11b/g/n dual-band 2,4 GHz et 5 GHz
- Prise en charge du système d'exploitation RouterOS version 7.1

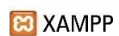
❖ API et fonctionnalités logicielles

- API complète et bien documentée permettant le contrôle et la configuration du routeur via des scripts ou des applications tierces
- Prise en charge des principaux protocoles d'API tels que l'API Winbox, l'API Web, l'API JSON-RPC et l'API Terminal
- Possibilité de développer des scripts et des extensions personnalisés en utilisant les langages supportés (Perl, Python, etc.)
- Nombreuses fonctionnalités avancées de routage, de pare-feu, de gestion VLAN, de qualité de service, etc.
- Surveillance et gestion du réseau à distance via l'interface web ou l'application mobile
- Mises à jour régulières du système d'exploitation RouterOS apportant de nouvelles fonctionnalités et améliorations

I.2 Environnements logiciels

Tout projet informatique doit s'appuyer sur des technologies performantes pour garantir une mise en œuvre efficace des besoins identifiés dans le chapitre précédent. Pour le développement de cette application, nous avons utilisé les outils logiciels suivants (7) :

➤ XAMPP



XAMPP

Les principaux outils utilisés sont :

❖ Apache



Apache est un serveur Web libre et open source gérée par la Fondation Apache. Malgré sa gratuité, il s'impose comme une référence en termes de fiabilité. Son principal rôle est de permettre aux propriétaires de sites Web de diffuser leur contenu en ligne. Lorsqu'un utilisateur demande des fichiers, des images ou des documents via son navigateur, le serveur Apache les transmet au client en utilisant le protocole HyperText Transfer Protocol (HTTP).

❖ MYSQL



Développé en 1995 par la société suédoise MySQL AB, le système de gestion de bases de données MySQL s'est imposé comme le plus répandu sur le Web grâce à sa robustesse et sa fiabilité. En tant que système de gestion de bases de données relationnelles (SGBDR), MySQL offre une organisation efficace et rapide pour accéder aux informations. De plus, il est open source, multiplateforme et s'intègre facilement à de nombreux serveurs Web (7).

➤ Visual Studio Code



Visual Studio Code est un éditeur de code open source développé par Microsoft. Lancé en 2015, il est devenu l'un des éditeurs les plus populaires auprès des développeurs grâce à ses nombreuses fonctionnalités et son interface intuitive. Vs code est développé avec le Framework Électron et conçu principalement pour développer des projets avec JavaScript, Node.js ou encore TypeScript. L'utilisation de Visual Studio Code (VSCode) dans notre projet présente plusieurs avantages (4) tel que : Muti-plateforme, éditeur de code avancé, intégration avec les outils Git etc.

➤ Postman



Postman est un outil de test et de développement d'APIs très populaire. Ses principales caractéristiques et avantages sont (5) :



1. Test d'APIs : Postman permet de tester facilement des APIs REST, GraphQL ou WebSocket en envoyant des requêtes HTTP personnalisées (GET, POST, PUT, DELETE, etc.) et en visualisant les réponses.
2. Environnements et collections : Postman permet d'organiser ses requêtes dans des collections, et de définir des environnements pour gérer différents paramètres (variables d'environnement, authentification, etc.).
3. Documentation et collaboration : Postman génère automatiquement une documentation lisible pour les APIs testées, facilitant la collaboration entre équipes de développement et de test.
4. Scripts et automatisation : Postman permet d'écrire des scripts JavaScript pour ajouter de la logique et de l'automatisation à ses tests, comme la validation de réponses ou la génération de données.
5. Intégration continue : Postman s'intègre avec des outils d'intégration continue comme Jenkins ou Travis CI, permettant d'automatiser les tests d'APIs dans les pipelines.
6. Interface intuitive : L'interface de Postman est très intuitive et conviviale, facilitant la prise en main pour les développeurs et les testeurs.
7. Multiplateforme : Postman est disponible sur Windows, macOS et Linux, ainsi qu'en version web, assurant une accessibilité maximale.

➤ Winbox



Winbox est une interface graphique conviviale développée par Mikrotik pour configurer et administrer ses routeurs, points d'accès et autres équipements réseau (10).

Winbox expose une API permettant d'interagir avec les équipements Mikrotik de manière programmatique. Cette API offre un accès complet à la configuration et aux fonctionnalités des routeurs, y compris la possibilité de configurer les interfaces, les règles de pare-feu, les services réseau, etc.



L'API Winbox est basée sur un protocole propriétaire Mikrotik et peut être utilisée à partir de plusieurs langages de programmation tels que C#, Python, Perl, etc. Elle permet d'automatiser de nombreuses tâches d'administration réseau, de développer des applications tierces intégrant les équipements Mikrotik, et d'écrire des scripts complexes de gestion du parc réseau. La documentation et les exemples d'utilisation de l'API Winbox sont fournis par Mikrotik pour faciliter son adoption par les administrateurs et les développeurs.

II. Environnement de programmation

II.1. PHP



PHP (Hypertext Preprocessor) est un langage de programmation côté serveur, populaire et largement utilisé pour le développement web. Créé en 1994 par Rasmus Lerdorf, PHP est un langage open source qui s'exécute sur le serveur web et génère du contenu dynamique pour les pages web (9).

II.2. CSS



CSS est l'acronyme de « Cascading Style Sheets » ce qui signifie « feuille de style en cascade ». Le CSS correspond à un langage informatique permettant de mettre en forme des pages web (HTML ou XML). Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000 (6).

II.3. HTML



HTML (HyperText Markup Language) est un langage de balisage utilisé pour créer et structurer des pages web. Il permet de définir la structure sémantique du contenu, d'ajouter des



éléments multimédias comme des images ou des formulaires, et de rendre les sites web accessibles à tous les utilisateurs (5).

L'utilisation de HTML dans notre projet présente plusieurs avantages, notamment une structure claire et organisée pour votre contenu web, une compatibilité multiplateforme, des fonctionnalités d'accessibilité, une amélioration du référencement naturel et une facilité d'utilisation.

II.4. Bootstrap



Bootstrap est un framework CSS open-source très populaire qui facilite la création de sites web responsives et mobiles-first. Principales caractéristiques et avantages de Bootstrap (8) :

1. Structure et grille : Bootstrap fournit une structure de base avec un système de grille puissant, permettant de créer facilement des mises en page complexes et réactives.
2. Composants prédéfinis : Bootstrap propose une vaste bibliothèque de composants HTML/CSS/JS préconçus, comme des menus, des boutons, des formulaires, des alertes, etc. Cela accélère le développement.
3. Responsive design : Bootstrap est conçu pour être mobile-first et s'adapter automatiquement à différentes tailles d'écran, grâce à son système de grille fluide et ses classes de mise en page responsives.
4. Personnalisation : Bien que Bootstrap fournisse un style par défaut, il est très facile de le personnaliser avec ses propres styles CSS pour correspondre à l'identité de votre projet.
5. Compatibilité multiplateforme : Bootstrap est compatible avec tous les navigateurs web modernes, garantissant un affichage cohérent sur ordinateurs, tablettes et smartphones.
6. Communauté et écosystème : Bootstrap bénéficie d'une très large communauté d'utilisateurs et de développeurs, ce qui se traduit par de nombreuses ressources, outils et extensions disponibles.



II.5. Javascript



JavaScript est un langage de programmation web très populaire utilisé pour ajouter de l'interactivité et des fonctionnalités dynamiques aux pages web. Ses principales caractéristiques et avantages sont :

1. Interactivité : JavaScript permet de rendre les pages web interactives en réagissant aux actions de l'utilisateur, comme des clics, des saisies de formulaire ou des mouvements de souris.
2. Dynamisme : Grâce à JavaScript, les pages web peuvent être modifiées de manière dynamique, sans avoir à recharger la page. Cela permet de créer des expériences utilisateur fluides et réactives.
3. Fonctionnalités côté client : JavaScript s'exécute directement dans le navigateur de l'utilisateur, permettant d'ajouter des fonctionnalités avancées sans avoir à solliciter le serveur à chaque fois.
4. Intégration avec HTML et CSS : JavaScript s'intègre parfaitement avec le HTML et le CSS, permettant de combiner la structure, la présentation et l'interactivité des pages web.
5. Multiplateforme : JavaScript est pris en charge par tous les navigateurs web modernes, assurant un fonctionnement cohérent sur ordinateurs, tablettes et smartphones.
6. Écosystème riche : JavaScript bénéficie d'une très large communauté de développeurs qui a créé de nombreuses bibliothèques et Frameworks facilitant le développement web.

II.6. CodeIgniter

CodeIgniter est un framework PHP open source, léger et rapide, conçu pour le développement d'applications web. Il a été initialement développé en 2006 par Rick Ellis et est maintenu par la communauté CodeIgniter.

Principales caractéristiques de CodeIgniter :



1. Architecture MVC : CodeIgniter suit le modèle architectural Modèle-Vue-Contrôleur (MVC), ce qui facilite la séparation des préoccupations et l'organisation du code.
2. Simplicité et légèreté : CodeIgniter a été conçu pour être un framework léger et simple à utiliser, avec une empreinte minimale sur le serveur.
3. Performance élevée : Grâce à son architecture optimisée, CodeIgniter offre des performances élevées et une exécution rapide du code.
4. Bibliothèques intégrées : CodeIgniter fournit une série de bibliothèques et de classes utilitaires intégrées, couvrant de nombreuses fonctionnalités web courantes (authentification, envoi d'e-mails, gestion des sessions, etc.).
5. Configuration minimale : CodeIgniter nécessite une configuration minimale pour être opérationnel, ce qui le rend facile à déployer sur différents environnements.
6. Sécurité renforcée : CodeIgniter accorde une grande importance à la sécurité, avec des fonctionnalités intégrées pour prévenir les attaques courantes comme les injections SQL et les failles XSS.
7. Communauté active : CodeIgniter bénéficie d'une communauté active de développeurs qui contribuent à son développement, partagent des ressources et fournissent un support.

III. Interface de notre application

III.1. Page de l'authentification ou de connexion

L'utilisateur introduit ses données de connexion et clique sur « Connecter ». Si les informations saisies sont conformes à son profil, il sera dirigé sur son profil, sinon un message d'erreur s'affiche :



Se connecter

Email

Mot de passe

Se connecter

Figure 16: Interface authentication

III.2. Page gérer les personnels

Après l'authentification, nous avons la page de gestion des personnels dont nous avons la liste des personnel et deux action actions qui sont ajouter un personnel et editer un personnel.

ImediatiS Sarl

Rechercher

Kenananaiko869@gmail.com

Personnel enregistré

Rechercher

AJOUTER

NOM ET PRENOM	EMAIL	POSTE	ENREGISTRÉ LE	SERVICE	OPÉRATION
Ganota Anaiko Kenan	ganotakenan@gmail.com	poste	21/06/24	informatique	
Mambou Jospin	jospinmambou@gmail.com	poste1	21/06/24	informatique	
Moukante Manfred	manfred120mouk@gmail.com	poste2	21/06/24	genie logiciel	
fomo samuel kerian	princefomo49@gmail.com	poste2	21/06/24	genie logiciel	

Dashboard

SERVICE

GERER LE PERSON...

GERER LES APPAR...

HORAIRES

RAPPORTS

STATISTIQUES

PROFIL DU COMPTE

Figure 17: Interface pour gérer les personnels

III.3. Page ajouter un personnel

Dans cette interface nous avons un formulaire à remplir pour ajouter un personnel. Après avoir remplir les champs du formulaire, on clique sur ajouter et le personnel est enregistrer dans le système



ImediatiS Sarl

Rechercher

Kenananaiko869@gmail.com

Gerer mon personnel

Accueil / Ajout de personnel

AJOUTER UN PERSONNEL

FIST-NAME:

LAST-NAME*:

Email*:

Poste*:

Figure 18: Interface ajouter personnel

III.4. Page gérer les appareils

Cette interface nous presente la gestion des appareils, tous les appareils enregister sont represente ici et on peut y ajouter des appareils. Apres avoir cliquer le le bouton ajouter cela nous dirige sur la page III.5

ImediatiS Sarl

Rechercher

Kenananaiko869@gmail.com

Consulter les appareils

Accueil / Appareil

Appareil enregistré

Rechercher

AJOUTER

NOM	MARQUE	ADRESSE MAC	PERSONNEL	OPERATION
android 572 xp 30 pro max	DESKTOP	A8:6B:AD:68:A4:E7	Mambou Jospin	
TECHNO TOP 4	Android	7E:F9:16:60:6E:99	fomo samuel kerian	
Galaxy	Android	B8:D7:AF:6B:A3:C3	Woupo Cyrille	
Samsung S8	Samsung	B9:D7:EF:6B:A3:C5	Ganota Anaiko Kenan	

Figure 19: Interface gérer un appareil



III.5. Page ajouter un appareil

Cette interface nous montre un formulaire pour l'ajout d'un appareil du personnel.

Imediati Sarl

Rechercher

Kenananaiko869@gmail.com

Gerer les appareils

Accueil / Ajout d'appareil

Ajouter un Appareil

Nom*
le host-name de votre appareil

Mac-Address*
adresse mail de l'appareil
Veuillez remplir ce champ.

Marque*
marque de l'appareil

Personnel*

Annuler Enregistrer

Copyright © 2023-2024 IMEDIATIS SARL. Version 1.0

Figure 20: Interface pour ajouter un appareil

III.6. Page pour générer le rapport

Les présentes interfaces nous montrent comment procéder pour générer un rapport pour un personnel. Après avoir choisis la date devant un personnel concerné, on clique sur générer pour voir tout le rapport du jour du personnel a la date correspondante.

Imediati Sarl

Rechercher

Kenananaiko869@gmail.com

Rapport

Date de début :
aaaa-mm-DD

Date de fin :
aaaa-mm-DD

PERSONNEL	EMAIL	POSTE	RAPPORT
Ganota Anaiko Kenan	ganotakenan@gmail.com	poste	
Mambou Jospin	jospinmambou@gmail.com	poste1	
Moukante Manfred	manfred120mouk@gmail.com	poste2	
fomo samuel kerian	princefomo49@gmail.com	poste2	
Woupo Cyrille	imediatisar@gmail.com	pdg	

Copyright © 2023-2024 IMEDIATIS SARL. Version 1.0

Figure 21: Interface pour générer le rapport



Rapport du Personnel

Tableau de Présence du personnel *Woupo Cyrille*

pour la période allant du: 2024-05-27 au 2024-07-17

Date	Appareil	Marque	Arrivée	Durée	Absence	Durée	Retour	Durée	Départ
2024-06-28	Galaxy	Android	19:53:02	1s	19:53:03				
					20:10:14	14m9s	20:06:45	27s	20:10:14

Imprimer

PROFIL DU COMPTE Woupo Cyrille imediatissarl@gmail.com pdg

Figure 22: Rapport générer

III.7. Page pour gérer le profil

Cette interface permet de gérer le profil d'utilisateur. L'utilisateur peut modifier ces données déjà existantes tels que son nom, prénom, email et son mot de passe et valider les modifications.

Imediatissarl

Profil du Compte

Rechercher

Accueil / Profil Kenananaiko869@gmail.com

Dashboard

SERVICE

GERER LE PERSON...

GERER LES APPAR...

HORAIRES

RAPPORTS

STATISTIQUES

PROFIL DU COMPTE

PROFILER

CHEF DE SERVICE TEL

Email: kenananaiko869@gmail.com

Nom: Your name

Fermer

Changer le mot de passe

Email: kenananaiko869@gmail.com

Mot de passe actuel:

Nouveau mot de passe:

Mettre à jour Annuler

Figure 23: Interface pour gérer profil



CONCLUSION ET PERSPECTIVE

Ce projet de fin d'étude avait pour but la conception et le développement d'un système de collecte et de traitement de présences du personnel à partir de périphérique mobile .

Pour réaliser ce travail, nous avons présenté dans un premier temps le cadre général du projet. Ensuite, nous avons procédé à l'analyse et la spécification des besoins de notre application. Cette analyse a enfin servi dans la conception et l'implémentation de la nouvelle application.

Grace à cette application, les entreprises peuvent gérer efficacement les heures de présence et générer régulièrement des rapports pour un personnel de l'entreprise afin de l'évaluer en fonction de ces heures de présence. Les personnels pourront faire leur pointage en toute sécurité et bien gérer leurs heures de présence.

Pour garantir la sécurité et la confidentialité du système, les protocoles de cryptage et des mesures de sécurités avancées pourraient être conçu. Les données de pointage enregistrées sont gardées précieusement pour permettre une analyse ultérieure si nécessaire.

L'intégration de l'intelligence artificielle (IA) pourrait permettre de rendre très efficace l'application faisant l'analyse des tendances pour détecter les anomalies ou encore l'intégration avec d'autres systèmes de RH pour une gestion plus fluide des absences, des heures supplémentaires



BIBLIOGRAPHIE

- [1] AMADOU JEAN ESAIE, conception et réalisation d'un système de gestion des offres au fne de maroua, Rapport Ingenieur, IUT de Ngaoundéré 2023;
- [2] MOUKANTE NGUINI ROGER MANFRED, recrutement en ligne basé sur les compétences, Rapport Ingenieur, IUT de Ngaoundéré 2023;
- [3] NDAM NJOYA AROUNA, support de cours sur UML, cours de 2e année, IUT de Ngaoundere, 2020-2021 ;

WEBOGRAPHIE

- [4] <https://visualstudio.microsoft.com/fr/downloads> , (consulté du 16 mai au 28 juillet)
- [5] <https://www.google.com/search> (consulté tout au long du stage)
- [6] <https://www.w3.org> (consulté tout au long du stage)
- [7] <https://dev.mysql.com/doc/> (consulté tout au long du stage)
- [8] <https://getbootstrap.com/docs> (consulté tout au long du stage)
- [9] <https://www.php.net/docs.php> (Consulté tout au long du stage)
- [10] <https://mikrotik.com> (consulté du 25 avril au 16 mai)