

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A. Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de Fin de Cycle

En vue de l'obtention du diplôme de Master Professionnel en Génie Logiciel

Thème

Conception et réalisation d'un système de pointage biométrique

Réalisé par

M. MAOUCHI Mohamed Djamil M. ZADIR Azeddine

Devant le jury composé de

Examinateur : Dr. Farid KACIMI Université de Béjaïa
Examinatrice : Dr. Sofia ZEBBOUDJ Université de Béjaïa
Encadrant : Dr. Karim AKILAL Université de Béjaïa

Remerciements

Nos remerciements s'adressent à notre encadrant Monsieur AKILAL, pour avoir accepté de diriger ce travail. Son soutien, sa clairvoyance, ses compétences, ainsi que son infinie disponibilité nous ont été d'une aide inestimable.

Nous remercions également M^{me} Siham BOUCHELAGHEM et M^{me} Sofia ZEBBOUDJ, pour leurs disponibilité, leurs gentillesse, et leurs précieuses directives tout au long de la réalisation de ce travail.

Qu'ils puissent trouver dans ce travail le témoignage de notre sincère gratitude et de notre profond respect.

Nous tenons également à remercier sincèrement les membres du jury ; Monsieur Farid KACIMI, et M^{me} Sofia ZEBBOUDJ ; qui nous font l'honneur d'évaluer ce travail.

Nous remercions également nos familles et amis pour leur soutien permanent qui nous a été bien utile.

Nous remercions tout particulièrement Monsieur BELAID Yacine pour nous avoir consacré de son temps, afin de nous aider dans la réalisation de la pointeuse biométrique.

Dans l'impossibilité de citer tous les noms, que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail trouvent ici l'expression de notre sincère gratitude.

Table des matières

Liste des figures	vii
Liste des tableaux	ix
Liste des abréviations	ix
Introduction générale	1
1 Contexte du projet et méthodologie de conception	3
1.1 Introduction	3
1.2 Contexte du projet	3
1.3 Problématique	4
1.4 Les applications Web	4
1.4.1 Définition	4
1.4.2 Quelle est la différence entre les applications Web et les applications natives ?	4
1.4.3 Pourquoi une application Web ?	5
1.5 Empreinte digitale	6
1.5.1 Caractéristique d'une empreinte digitale	6
1.5.2 Pourquoi l'empreinte digitale ?	6
1.6 Processus de développement	6
1.6.1 Processus unifié (UP)	6
1.6.2 Les méthodes agiles	7
1.6.3 Le processus choisi	8
1.7 Conclusion	9
2 Spécification et analyse des besoins	10
2.1 Introduction	10
2.2 Identification des acteurs	10
2.3 Identification des cas d'utilisation	10
2.4 Spécification des exigences	12
2.4.1 Exigences fonctionnelles	12
2.4.2 Exigences non fonctionnelles	13
2.5 Diagrammes des cas d'utilisation	13
2.5.1 Diagramme de cas d'utilisation de l'employé	14
2.5.2 Diagramme de cas d'utilisation du manager	15

2.5.3	Diagramme de cas d'utilisation du responsable	16
2.5.4	Diagramme de cas d'utilisation de l'administrateur	17
2.6	Descriptions des cas d'utilisation	18
2.7	Diagrammes de séquence système DSS	24
2.7.1	Cas d'utilisation « Se pointer »	24
2.7.2	Cas d'utilisation « Consulter mon profil »	25
2.7.3	Cas d'utilisation « Consulter ma fiche de pointage »	26
2.7.4	Cas d'utilisation « Consulter tableau de bord manager »	26
2.7.5	Cas d'utilisation « Ajouter membre »	27
2.7.6	Cas d'utilisation « Ajouter planning »	28
2.7.7	Cas d'utilisation « Ajouter équipe »	28
2.7.8	Cas d'utilisation « Ajouter employé »	29
2.7.9	Cas d'utilisation « Consulter le profil d'un employé »	30
2.8	Les maquettes IHM	31
2.8.1	Employé	31
2.8.2	Manager	33
2.8.3	Responsable	34
2.8.4	La navigation	36
2.9	Conclusion	36
3	Conception	37
3.1	Introduction	37
3.2	Modèle de domaine	37
3.3	Diagrammes de classes participantes	44
3.3.1	Définitions et formalisme	44
3.4	Diagrammes de séquence	54
3.5	Diagramme de classes conception préliminaire	63
3.6	Diagramme de classes de conception	67
3.7	Modèle relationnel	68
3.8	Conclusion	69
4	Réalisations de la pointeuse	70
4.1	Introduction	70
4.1.1	Logiciel	70
4.2	ESP32	70
4.2.1	Caractéristiques techniques	71
4.3	Capteur d'empreinte DY 50	72
4.3.1	Principe de fonctionnement	74
4.3.2	Caractéristiques techniques du capteur optique	74
4.4	Schémas et branchement	75
4.5	Prototype	77
4.6	Programmation	79
4.6.1	Environnement de travail	79
4.6.2	Code et fonctionnement	81
4.7	Conclusion	85

5	Implémentations, mise en service et test	86
5.1	Introduction	86
5.2	Environnement de développement	86
5.2.1	Plateformes utilisées	86
5.2.2	Logiciels utilisés	87
5.2.3	Technologie utilisées	88
5.2.4	HTML	88
5.2.5	CSS	88
5.2.6	Sass	88
5.2.7	JavaScript	89
5.2.8	Python	89
5.2.9	Django	89
5.3	Persistance des données	90
5.3.1	MySQL	90
5.3.2	ORM	90
5.4	Les librairies utilisées	91
5.4.1	Bootstrap	91
5.4.2	JQuery	91
5.4.3	Crispy forms	91
5.4.4	Pillow	91
5.4.5	Django import/export	91
5.4.6	Popper	91
5.4.7	Font Awesome	92
5.4.8	Chart.js	92
5.5	Architecture du système	92
5.6	Présentation de l'application	93
5.6.1	Nom et logo de l'application	93
5.6.2	Palette de couleurs	94
5.6.3	Typographie	94
5.6.4	Squelette de l'application web	94
5.7	Présentation des interfaces	96
5.7.1	Espace employé	96
5.7.2	Espace manager	99
5.7.3	Espace responsable	101
5.7.4	Espace administrateur	104
5.8	Sécurité de l'application	104
5.8.1	Protection contre Cross Site Scripting	105
5.8.2	Protection contre Cross site request forgery	105
5.8.3	Protection contre SQL injection	105
5.8.4	Protection contre le Clickjacking	105
5.9	Tests	105
5.10	Conclusion	106

Conclusion générale et perspectives	107
Bibliographie	109
Annexe A	112
Annexe B	131

Table des figures

1.1	Récapitulatif du processus de développement [5]	9
2.1	Diagramme de cas d'utilisation associé à l'acteur «Employé»	14
2.2	Diagramme de cas d'utilisation associé à l'acteur «Manager»	15
2.3	Diagramme de cas d'utilisation associé à l'acteur «Responsable»	16
2.4	Diagramme de cas d'utilisation associé à l'acteur «Administrateur»	17
2.5	Diagramme de séquence système « Se pointer »	24
2.6	Diagramme de séquence système « Consulter mon profil »	25
2.7	Diagramme de séquence système « Consulter ma fiche de pointage »	26
2.8	Diagramme de séquence système « Consulter tableau de bord manager »	27
2.9	Diagramme de séquence système « Ajouter membre »	27
2.10	Diagramme de séquence système « Ajouter planning »	28
2.11	Diagramme de séquence système « Ajouter équipe »	29
2.12	Diagramme de séquence système « Ajouter employé »	30
2.13	Diagramme de séquence système « Consulter profil d'un employé »	31
2.14	Maquette tableau de bord Employé	32
2.15	Maquette profil Employé	32
2.16	Maquette tableau de bord Manager	33
2.17	Maquette des équipes d'un Manager	34
2.18	Maquette création d'un planning	35
2.19	Maquette affectations d'un employé a une équipe	35
3.1	Modèle du domaine « Se pointer »	38
3.2	Modèle du domaine « Consulter ma fiche de pointage »	39
3.3	Modèle du domaine « Ajouter planning »	39
3.4	Modèle du domaine « Ajouter équipe »	40
3.5	Modèle du domaine « Ajouter employé »	41
3.6	Diagramme de séquence système « Ajouter membre »	42
3.7	Modèle du domaine « Consulter profil d'un employé »	43
3.8	Modèle du domaine « Consulter tableau de bord manager »	44
3.9	Représentation d'une classe dialogue	45
3.10	Représentation d'une classe contrôle	45
3.11	Représentation d'une classe entité	46
3.12	Modèle d'un diagramme de classe de conception	46
3.13	Diagrammes de classes participantes « Consulter ma fiche de pointage »	47
3.14	Diagrammes de classes participantes « Consulter mon profil »	48
3.15	Diagrammes de classes participantes « Consulter ma fiche de pointage »	48

3.16	Diagrammes de classes participantes « Consulter tableau de bord manager »	49
3.17	Diagrammes de classes participantes « Ajouter planning »	50
3.18	Diagrammes de classes participantes « Ajouter équipe »	51
3.19	Diagrammes de classes participantes « Ajouter membre »	52
3.20	Diagrammes de classes participantes « Ajouter employé »	53
3.21	Diagrammes de classes participantes « Consulter profil employé »	53
3.22	Formalisme du diagramme de séquence	54
3.23	Diagramme d'interaction « Se pointer »	55
3.24	Diagramme d'interaction « Consulter mon profil »	56
3.25	Diagramme d'interaction « Consulter ma fiche de pointage »	57
3.26	Diagramme d'interaction « Consulter tableau de bord manager »	58
3.27	Diagramme d'interaction « Ajouter équipe »	59
3.28	Diagramme d'interaction « Ajouter planning »	60
3.29	Diagramme d'interaction « Ajouter membre »	61
3.30	Diagramme d'interaction « Ajouter employé »	62
3.31	Diagramme d'interaction « Consulter profil d'un employé »	63
3.32	Diagramme de classes conception préliminaire « Consulter ma fiche de pointage » .	64
3.33	Diagramme de classes conception préliminaire « Ajouter une équipe »	65
3.34	Diagramme de classes conception préliminaire « Ajouter un planning»	65
3.35	Diagramme de classes conception préliminaire « Ajouter un employé»	66
3.36	Diagramme de classes conception préliminaire « Consulter profil d'un employé» .	66
3.37	Diagramme de classes de conception	67
3.38	Le père du modèle relationnel	68
4.1	ESP32 DevKit	71
4.2	Schéma fonctionnel	73
4.3	Capteur optique DY50	73
4.4	Schéma expliquant le montage préliminaire des différents modules	75
4.5	Schéma expliquant le montage préliminaire des différents modules	76
4.6	Pointeuse vue de face	77
4.7	Pointeuse vue de haut	77
4.8	Côté gauche extérieure	78
4.9	Côté gauche intérieure	78
4.10	Pointeuse ouverte	78
4.11	Pointeuse fermée	78
4.12	interface Arduino IDE et un programme qui clignote une LED	80
5.1	Fonctionnement de l'ORM de Django [49]	90
5.2	Architecture du système	92
5.3	Nom de l'application	93
5.4	Logo avec couleur uniforme	93
5.5	Logo sur fond bleu	93
5.6	Logo sur fond blanc	93
5.7	Palette de couleur utilisée	94
5.8	Squelette de l'application web	95
5.9	User flow de l'employé	96
5.10	Interface dashboard employé	97

5.11 Interface ma fiche de pointage	98
5.12 Interface consulter mon profil	98
5.13 User flow du manager	99
5.14 Interface dashboard manager	100
5.15 Interface résumé d'une équipe	101
5.16 User flow du responsable	102
5.17 interface créer un planning	103
5.18 Interface affecter un employé	103
5.19 User flow de l'administrateur	104
5.20 Diagramme de séquence système « Consulter résumé de pointage des collaborateur »	131
5.21 Diagramme de séquence système « Consulter la liste des collaborateurs »	132
5.22 Diagramme de séquence système « Consulter mes équipes »	132
5.23 Diagramme de séquence système « Consulter mon profil »	133
5.24 Diagramme de séquence système « Modifier mon profil »	134
5.25 Diagramme de séquence système « Affectation d'un planning »	135
5.26 Diagramme de séquence système « Consulter liste des équipes »	136
5.27 Diagramme de séquence système « Ajouter empreinte »	137
5.28 Diagramme de séquence système « Supprimer empreinte »	138

Liste des tableaux

1.1	Tableau comparatif entre les applications Web et natives.	5
2.1	Identification des cas d'utilisation	12
2.2	Description du cas d'utilisation « S'authentifier »	18
2.3	Description du cas d'utilisation « Se pointer »	19
2.4	Description du cas d'utilisation « Consulter mon profil »	19
2.5	Description du cas d'utilisation « Consulter ma fiche de pointage »	20
2.6	Description du cas d'utilisation « Consulter tableau de bord manager »	20
2.7	Description du cas d'utilisation « Ajouter membre »	21
2.8	Description du cas d'utilisation « Ajouter un planning »	22
2.9	Description du cas d'utilisation « Ajouter équipe »	22
2.10	Description du cas d'utilisation « Ajouter un employé »	23
2.11	Description du cas d'utilisation « Consulter profil d'un employé »	23
4.1	Interface de connexion [34]	74
5.1	Endpoint de l'api RESTful	92
5.2	Description du cas d'utilisation « Consulter mon tableau de bord »	112
5.3	Description du cas d'utilisation « Consulter le tableau de bord responsable »	113
5.4	Description du cas d'utilisation « Rechercher période »	113
5.5	Description du cas d'utilisation « Rechercher employé »	114
5.6	Description du cas d'utilisation « Modifier mon profil »	114
5.7	Description du cas d'utilisation « Consulter profil d'un collaborateur »	115
5.8	Description du cas d'utilisation « Consulter le résumé de pointage des collaborateurs »	115
5.9	Description du cas d'utilisation « Consulter feuille de pointage d'un collaborateur »	116
5.10	Description du cas d'utilisation « Consulter résumer de pointage des employés »	116
5.11	Description du cas d'utilisation « Consulter feuille de pointage d'un employé »	117
5.12	Description du cas d'utilisation « Consulter la liste des collaborateurs »	118
5.13	Description du cas d'utilisation « Consulter la liste des employés »	118
5.14	Description du cas d'utilisation « Modifier employé »	119
5.15	Description du cas d'utilisation « Supprimer un employé »	120
5.16	Description du cas d'utilisation « Ajouter une empreinte »	120
5.17	Description du cas d'utilisation « Supprimer une empreinte »	121
5.18	Description du cas d'utilisation « Consulter la liste de mes équipes »	122
5.19	Description du cas d'utilisation « Consulter le résumé de mon équipe »	122
5.20	Description du cas d'utilisation « Consulter la liste des équipes »	123
5.21	Description du cas d'utilisation « Consulter le résumé d'une équipe »	123
5.22	Description du cas d'utilisation « Modifier équipe »	124

5.23 Description du cas d'utilisation « Supprimer équipe »	125
5.24 Description du cas d'utilisation « Supprimer membre »	125
5.25 Description du cas d'utilisation « Consulter liste des plannings »	126
5.26 Description du cas d'utilisation « Consulter mon planning »	126
5.27 Description du cas d'utilisation « Consulter planning d'un collaborateur »	127
5.28 Description du cas d'utilisation « Consulter planning d'un employé »	127
5.29 Description du cas d'utilisation « Modifier un planning »	128
5.30 Description du cas d'utilisation « Supprimer un planning»	129
5.31 Description du cas d'utilisation « Affectation d'un planning »	129
5.32 Description du cas d'utilisation « Consulter le journal des affectations »	130
5.33 Description du cas d'utilisation « Importer/Exporter »	130

Liste des abréviations

3D	3 Dimensions
API	Application Programming Interface
AES	Advanced Encryption Standard
CAN	Controller Area Network
CRUD	Create, Read, Update, Delete
CSRF	Cross Site Request Forgery
CSS	Cascading Style Sheets
CPU	Central Processing Unit
DCC	Diagramme de Classe Conception
DCP	Diagramme de Classe Participante
DSS	Diagramme de Séquence Système
HTML	HyperText Markup Language
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IHM	Interface Homme-Machine
LED	Light-Emitting Diode
MAC	Media Access Control
MDD	Modèle Du Domaine
MVT	Model View Template
NTIC	Nouvelles Technologies de l'Information et de la Communication
ORM	Object Relational Mapping
PWM	Pulse Width Modulation
PDG	Président-Directeur Général
PME	Petites et Moyennes Entreprise
SASS	Syntactically Awesome StyleSheets
SHA	Secure Hash Algorithm

SQL	Structured Query Language
TTL	Transistor-Transistor Logic
UI	User Interface
UML	Unified Modeling Language
UP	Unified Process
USB	Universal Serial Bus
UX	User eXperience
W3C	World Wide Web Consortium
WPA	Wi-Fi Protected Access
XML	eXtensible Markup Language
XP	eXtrem Programing

Introduction générale

Une entreprise est une organisation qui rassemble des moyens matériels ainsi que des personnes qui mobilisent leurs talents et leurs énergies afin de fournir un service ou un produit à ses clients. Avec l'avènement des Nouvelles Technologies de l'information et de la Communication (NTIC), de plus en plus d'entreprises font appel aux nouvelles technologies pour rester dans l'air du temps et faire face à la concurrence.

En effet avec leurs anciennes méthodes de gestion, les entreprises sont dans l'obligation de déployer des moyens humains et financiers pour faire face à des tâches répétitives, mais nécessaires. Dans cette optique, ce projet a pour but d'offrir un système de gestion de pointage qui sera composé d'une application Web disposant de fonctionnalités très étendues destinées à différents types d'employés, et d'une pointeuse biométrique afin d'automatiser l'une des tâches centrales de toute entreprise, et ce en gardant l'heure de pointage.

Le premier chapitre introduit le contexte du projet et sa problématique, ainsi que les concepts clés tels que les applications Web et les empreintes digitales. Par la suite, nous définirons la méthodologie avec laquelle nous avons organisé notre projet de façon rationalisée et structurée pour nous aider à accomplir chaque étape du projet, de la planification à la mise en œuvre de façon efficace.

Le deuxième chapitre qui s'intitule spécification et analyse des besoins, permettra d'identifier les différents acteurs et leurs cas d'utilisation respectifs afin de modéliser leurs diagrammes et de les décrire de façon détaillée ainsi qu'exprimer les exigences fonctionnelles et non fonctionnelles. Nous modéliserons, ensuite, les diagrammes de séquence système pour enfin réaliser le prototype de l'application Web.

Le troisième chapitre a pour but de détailler la phase de conception de notre système. Nous élaborerons d'abord les modèles du domaine, puis les diagrammes de classes participantes afin de détailler les diagrammes de séquence système réalisés dans le chapitre précédent. Ensuite, nous passerons aux diagrammes de classes de conception préliminaire qui nous permettront de réaliser le diagramme de classe de conception. Pour enfin implémenter notre base de données à l'aide du modèle relationnel.

Dans le quatrième chapitre, nous aborderons l'aspect matériel et logiciel qui nous permettra de réaliser la pointeuse biométrique, puis nous présenterons le prototype et le fonctionnement de cette dernière.

Quant au dernier chapitre, il expose l'environnement de travail qui est composé des plateformes, logiciels, et technologies utilisés pour la réalisation de l'application Web. Nous clôturerons ce chapitre par la présentation de cette dernière.

Enfin, nous conclurons ce travail en résumant les connaissances acquises durant la réalisation de ce projet et nous dégagerons quelques perspectives.

Chapitre 1

Contexte du projet et méthodologie de conception

1.1 Introduction

Dans ce premier chapitre, nous exposerons le contexte du projet et la problématique à résoudre. Ensuite, nous aborderons quelques définitions sur les applications Web et les empreintes digitales. Enfin nous définirons le processus de développement entrepris afin de faciliter l’élaboration du projet.

1.2 Contexte du projet

L’entreprise est une organisation qui mobilise des ressources dans le but de produire ou fournir un service, dans un souci vital de rentabilité. Or, le climat économique actuel se distingue par des marqueurs qui rendent la survie des entreprises difficile. Parmi ces derniers, on peut citer la forte concurrence, l’extrême évolution des marchés et leurs imprévisibilité ainsi que la mondialisation du secteur économique.

Toute entreprise voulant être prospère se doit de garder les coûts au minimum et les profits au maximum, tout en ayant une administration qui veille à son bon fonctionnement. Cependant, certaines tâches sont répétitives et chronophages, mais ne peuvent pas être négligées. Ceci pousse l’entreprise à déployer des ressources humaines et matérielles considérables dans ces tâches, ce qui ne représente pas la valeur ajoutée réelle que génère l’entreprise pour son environnement dans son domaine d’expertise.

Dans l’optique de minimiser les dépenses et de mieux utiliser leurs moyens, les entreprises ont eu recours aux technologies de l’information et de la communication ainsi qu’aux systèmes d’information. Du fait de simplement vouloir garder les informations des employés dans une base de données pour y accéder plus facilement, jusqu’à l’utilisation des algorithmes d’intelligence artificielle, ou des big data pour l’aide à la décision. Du simple employé au PDG, tous ont recours aux nouvelles technologies pour mieux accomplir leurs tâches et être plus efficaces et efficientes. Parmi

ces tâches, nous avons choisi de traiter la gestion de pointage des employés ainsi que leurs temps de travail.

1.3 Problématique

Le contexte du projet étant établi, dans cette section nous allons décrire la problématique de notre projet afin de poser les conditions-cadres ainsi que les attentes de ce dernier.

Afin de répondre aux besoins des différentes PME et offrir une solution adapter à ces dernières, nous avons pour but de concevoir et de réaliser une application Web qui permet une gestion précise du pointage des employés et de leurs temps de travail, grâce à une pointeuse biométrique que nous allons réaliser et qui sera capable d'identifier de manière unique un individu déjà enregistré et de communiquer avec l'application Web.

Nous espérons une fois ce projet à terme, inciter les entreprises à abandonner leurs anciennes méthodes de pointage et de gestion des plannings pour gagner en efficacité et réduire les ressources allouées à ces tâches. Cela en offrant, un outil de supervision simple et ergonomique et en collectant les informations qui sont primordiales pour faciliter l'utilisation aux responsables, ainsi qu'un espace individuel dédié à chaque employé dans le but d'avoir son planning et ses informations de pointage de manière transparente.

1.4 Les applications Web

1.4.1 Définition

Une application Web (ou Web App) est un logiciel applicatif hébergé sur un serveur et accessible depuis un navigateur Web (Google Chrome, Mozilla Firefox, Safari...). Contrairement à une application native, aucune installation n'est nécessaire ouvrant la porte à de nombreux avantages.

1.4.2 Quelle est la différence entre les applications Web et les applications natives ?

Une application Web fonctionne généralement comme une application native installée sur votre machine à la différence que celle-ci s'exécute directement sur le navigateur Web, ce qui lui permet d'être disponible partout avec des données synchronisées, le tableau 1.1 présente une comparaison entre les deux types d'application [1] :

	Application native	Application Web
Plateforme	Dépendant de la plate-forme utilisée	Indépendant de la plate-forme utilisée
Stockage de données	Sur l'appareil de l'utilisateur / ou sur un serveur	En général sur le serveur
Utilisation des fonctions de l'appareil de l'utilisateur	Totale	Limitée
Source	Téléchargement via le fournisseur	Directement sur le navigateur
Installation	Nécessaire.	Pas nécessaire.
Mise à jour	Doit être téléchargée puis installée	Est intégrée par les fournisseurs et disponible immédiatement après le déploiement
Connexion Internet	Pas nécessaire la plupart du temps	pas tout le temps

TABLE 1.1 – Tableau comparatif entre les applications Web et natives.

1.4.3 Pourquoi une application Web ?

Les applications Web ont considérablement évolué au cours des dernières années avec des améliorations en matière de sécurité avec des technologies de plus en plus flexibles, ce qui permet de développer presque toutes les applications natives en tant qu'applications Web et de bénéficier des nombreux avantages offerts par le Web [2] :

- **Accessibilité optimisée** : Les applications Web n'ont pas besoin d'être installées, cela permet un accès universel depuis n'importe quel type de poste.
- **Développement rentable** : Il n'est pas nécessaire de programmer et de tester sur toutes les versions et configurations des systèmes d'exploitation possibles, cela rend le développement moins coûteux et réduit les délais.
- **Installation et maintenance simplifiées** : Avec l'approche basée sur le Web, l'installation et la maintenance deviennent également moins compliquées. Une fois qu'une nouvelle version ou mise à niveau est installée sur le serveur, elle sera accessible sur n'importe quel type de poste.
- **Technologies de base flexibles** : Chacune des technologies de base peut être utilisée pour créer des applications Web, en fonction des exigences de l'application.

1.5 Empreinte digitale

1.5.1 Caractéristique d'une empreinte digitale

Une empreinte digitale se compose d'un ensemble de stries (ici définies comme étant les reliefs positifs qui rentrent en contact avec la surface du capteur) et de sillons définissant le relief de la surface du doigt.

Les caractéristiques topologiques de l'empreinte restent constantes tout au long de la vie d'un individu et ne peuvent être que partiellement altérées par de profondes coupures laissant apparaître des cicatrices. Le caractère permanent de l'empreinte digitale permet ainsi d'extraire une signature mathématique donnant la possibilité de l'identifier de manière extrêmement fiable[3].

1.5.2 Pourquoi l'empreinte digitale ?

Dans un monde en constante évolution en matière d'innovation technologique, la biométrie s'est rapidement distinguée comme la plus pertinente pour identifier et authentifier les personnes de manière fiable et rapide en fonction de leurs caractéristiques biologiques uniques.

Dans la perspective de réaliser un système de pointage fiable pour mesurer le temps de travail et gérer la présence des employés, tout en minimisant les pertes de l'entreprise, il est nécessaire d'avoir un bon système de reconnaissance fiable et simple d'utilisation. Ainsi, nous considérons que l'empreinte biométrique est la solution la plus intéressante, tant du point de vue technique et économique.

1.6 Processus de développement

Un processus définit une séquence d'étapes, partiellement ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. L'objectif d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles.

Après avoir analysé de manière globale notre projet, nous avons décidé de travailler selon le processus de développement proposé dans le livre qui s'intitule « UML 2 modéliser une application Web » de Pascal Roques [5]. Un processus que décrit l'auteur à mi-chemin entre UP (Unified Process) et les méthodes agiles telles que XP et Scrum, et qui s'inspire également des bonnes pratiques prônées par les tenants de la modélisation agile.

1.6.1 Processus unifié (UP)

Le processus unifié est un processus de développement logiciel « itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques » [5] :

- **Itératif et incrémental** : le projet est découpé en itérations de courte durée (environ 1 mois) qui aident à mieux suivre l'avancement global. À la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale.
- **Centré sur l'architecture** : tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et pas seulement documentée en texte.
- **Piloté par les risques** : les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.
- **Conduit par les cas d'utilisation** : le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés.

1.6.2 Les méthodes agiles

La notion de méthode agile est née à travers un manifeste signé en 2001 par 17 personnalités du développement logiciel dont Ward Cunningham, Alistair Cockburn, Kent Beck, Martin Fowler, Ron Jeffries, Steve Mellor, Robert C. Martin, Ken Schwaber, Jeff Sutherland, etc [5]. Ce manifeste prône quatre valeurs fondamentales [5] :

- « **Personnes et interactions plutôt que processus et outils** » : dans l'optique agile, l'équipe est bien plus importante que les moyens matériels ou les procédures. Il est préférable d'avoir une équipe soudée et qui communique, composée de développeurs moyens, plutôt qu'une équipe composée d'individualistes, même brillants. La communication est une notion fondamentale.
- « **Logiciel fonctionnel plutôt que documentation complète** » : il est vital que l'application fonctionne. Le reste, et notamment la documentation technique, est secondaire. Même si une documentation succincte et précise est utile comme moyen de communication. La documentation représente une charge de travail importante et peut être néfaste si elle n'est pas à jour. Il est préférable de commenter abondamment le code lui-même, et surtout de transférer les compétences au sein de l'équipe (on en revient à l'importance de la communication).
- « **Collaboration avec le client plutôt que négociation de contrat** » : le client doit être impliqué dans le développement. On ne peut se contenter de négocier un contrat au début du projet, puis de négliger les demandes du client. Le client doit collaborer avec l'équipe et fournir un feedback continu sur l'adaptation du logiciel à ses attentes.
- « **Réagir au changement plutôt que suivre un plan** » : la planification initiale et la structure du logiciel doivent être flexibles afin de permettre l'évolution de la demande du

client tout au long du projet. Les premières releases du logiciel vont souvent provoquer des demandes d'évolution.

1.6.3 Le processus choisi

Après une brève présentation des concepts majeurs dont s'inspire le processus choisi, nous allons le décrire d'une manière plus détaillée et citer d'où vient chaque caractéristique.

- Un processus conduit par cas d'utilisation, comme UP.
- Relativement léger et restreint, comme les méthodes agiles néanmoins sans négliger les activités de modélisations en analyse et conception.
- Utilise un sous-ensemble nécessaire et suffisant du langage UML.
- Veille à modéliser tous les aspects critiques du système.

Les besoins sont modélisés en cas d'utilisation UML pour être représentés de façon plus concrète par des maquettes IHM, dans le but de les présenter aux futurs utilisateurs. Puis nous allons produire des diagrammes de séquence système pour décrire le système comme une boîte noire tout en représentant graphiquement la chronologie des interactions entre les acteurs et le système dans le cadre d'un scénario nominal. Grâce aux diagrammes de cas d'utilisation ainsi qu'aux maquettes, on pourra modéliser les diagrammes de classes participantes qui décriront les cas d'utilisation, en ayant recours aux trois principales classes d'analyse : les classes dialogues, contrôles, entités ainsi que leurs relations.

À la suite de cela, nous modéliserons les différents diagrammes d'interactions où chaque cas d'utilisation est décrit en détail dans le but de mettre en évidence l'allocation de responsabilités de chaque objet intervenant dans le cas d'utilisations traité dans les différents scénarios possibles (nominale/erreur). Enfin, nous pourrons définir les diagrammes de classes de conception qui représentent la structure statique du code par le biais des attributs et des relations entre classes, ainsi que des opérations décrivant la responsabilité dynamique des classes logicielle. La figure 1.1 suivante résume la totalité des diagrammes à modéliser dans le processus choisi.

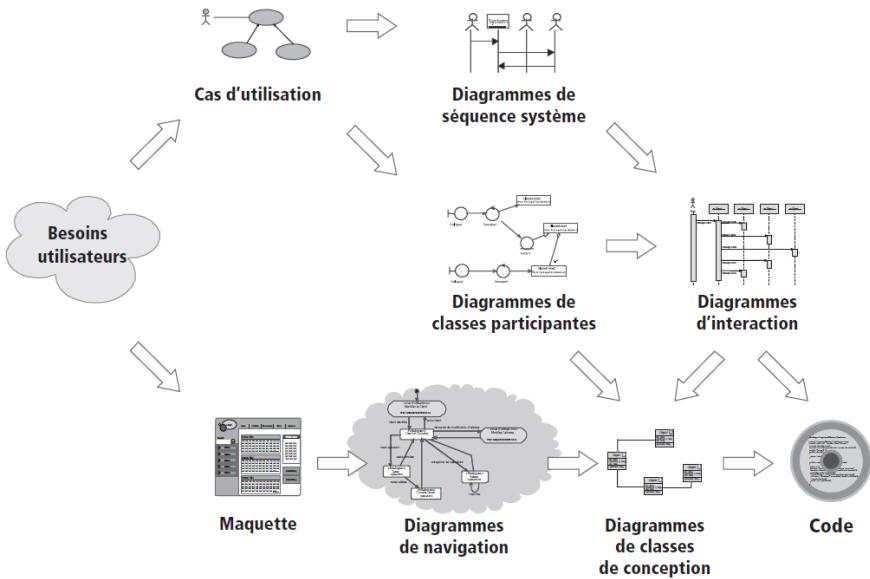


FIGURE 1.1 – Récapitulatif du processus de développement [5]

1.7 Conclusion

Ce premier chapitre nous a permis de présenter le cadre général du projet, à savoir le contexte et la problématique en élaborant une solution à cette dernière. Nous avons aussi brièvement défini les applications Web et les empreintes digitales. Nous avons conclu par la description du processus de développement à suivre tout au long du projet.

Chapitre 2

Spécification et analyse des besoins

2.1 Introduction

Ce chapitre sera consacré à la phase de spécification et d'analyse des besoins. Dans cette étape, nous allons tout d'abord identifier les différents acteurs du système, ainsi que leurs actions qui seront représentées en cas d'utilisation, puis nous allons décrire les spécifications des exigences fonctionnelles et non fonctionnelles, ainsi que leurs contraintes. Ceci nous permettra de modéliser les diagrammes de cas d'utilisation.

2.2 Identification des acteurs

Un acteur représente une entité extérieure au système modélisé, qui interagit directement avec lui afin d'atteindre des objectifs [4]. Dans notre système, on peut identifier 4 acteurs humains :

- **L'Employé** : Joue le rôle d'une personne qui occupe un emploi dans l'entreprise et qui interagit avec le système, pour un pointage des horaires de travail réalisé, ainsi que la possibilité de s'authentifier sur la plateforme pour consulter les informations qui le concernent.
- **Le Manager** : Joue le rôle d'un employé très important qui est le lien entre la direction et les autres employés. Les managers s'occupent de l'organisation et du contrôle de leurs équipes et employés à travers l'application web.
- **Le Responsable** : Joue le rôle d'un responsable qui gère les employés ainsi que les différentes équipes de l'entreprise.
- **L'administrateur** : Joue le rôle d'une personne vitale pour le bon fonctionnement du système via l'espace d'administration.

2.3 Identification des cas d'utilisation

Un cas d'utilisation représente une série d'interactions d'un acteur avec un système. Cette interaction est destinée à fournir des résultats à l'acteur. Le tableau ci-dessous présente les différents cas d'utilisations avec leurs acteurs respectifs.

N	Cas d'utilisation	Acteur
1	S'authentifier	Employé/Manager Responsable
2	Consulter mon tableau de bord	
3	Consulter ma fiche de pointage	
4	Rechercher période	
5	Consulter mon profile	
6	Modifier mon profile	
7	Consulter mon planning	
8	Se pointer	
9	Consulter tableau de bord manager	Manager
10	Consulter liste des collaborateurs	
11	Consulter profil d'un collaborateur	
12	Consulter planning d'un collaborateur	
13	Consulter liste de mes équipes	
14	Consulter résumé de mon équipe	
15	Consulter résumé de pointage des collaborateurs	
16	Consulter feuille de pointage d'un collaborateurs	
17	Rechercher employé	Manager/Responsable
18	Consulter le tableau de bord responsable	Responsable
19	Consulter liste des équipes	
20	Consulter résumé d'une équipe	
21	Consulter liste des employés	
22	Consulter profil d'un employé	
23	Consulter liste des plannings	
24	Consulter planning d'un employé	
25	Consulter résumé de pointage des employés	
26	Consulter feuille de pointage d'un employé	Responsable/Administrateur
27	Consulter journal d'affectation	
28	Gérer les plannings	
	Ajouter un planning Modifier un planning Supprimer un planning Affectation d'un planning	
29	Gérer les équipes	
	Ajouter une équipe Modifier une équipe Supprimer une équipe Ajouter membre Supprimer membre	

30	Importer/exporter		Responsable/Administrateur	
31	Ajouter une empreinte			
32	Supprimer une empreinte			
33	Gérer les employés	Ajouter un employé Modifier un employé Supprimer un employé		
34	S'authentifier		Administrateur	

TABLE 2.1 – Identification des cas d'utilisation

2.4 Spécification des exigences

2.4.1 Exigences fonctionnelles

Les besoins fonctionnels expriment une action que doit effectuer le système en réponse à une demande (sorties qui sont produites pour un ensemble donné d'entrées). Par exemple, le système doit stocker les informations de pointage des employés. Le système à concevoir devra répondre aux besoins que nous allons citer. Afin de mettre en évidence les différentes informations accessibles selon le rôle, nous avons décidé de les regrouper par acteur.

- **Employé** : Accès à son profil ainsi qu'aux informations de pointage et au planning qui le concerne, ainsi qu'aux informations publiques et aux statuts de présence des collaborateurs faisant partie de la même équipe que lui.
- **Manager** : En plus des informations accessibles à l'employé, le manager doit avoir accès aux informations des équipes dont il est responsable ainsi qu'aux informations des membres qui les composent, aux plannings et aux informations de pointage y compris.
- **Responsable** : Celui-ci doit avoir accès à la totalité des informations du système concernant tous les employés, quels que soient leurs statuts. Il doit être capable d'ajouter, modifier, supprimer des équipes ou des plannings, affecter un manager à une équipe, un planning à un employé, ajouter ou supprimer un employé à une équipe. Importer/exporter des informations de pointage est une fonctionnalité primordiale que doit avoir ce rôle.
- **Administrateur** : La personne responsable de maintenir le système et de veiller à son bon fonctionnement a naturellement accès à l'espace d'administration du système à partir duquel elle pourra gérer les comptes des différents utilisateurs et leur assigner le bon rôle. En cas d'incohérences ou d'erreurs, ce dernier devrait avoir la possibilité d'intervenir de manière directe sur la base de données afin de résoudre ces cas. Ce dernier doit pouvoir aussi gérer les empreintes (ajout et suppression).

En plus des besoins précédemment cités, il existe des besoins partagés entre tous les acteurs tels que :

- Authentification : Chaque utilisateur du futur système devra être authentifié au préalable avec son nom d'utilisateur et son mot de passe avant tout accès à la plateforme. Le système devra ensuite rediriger l'utilisateur vers son espace utilisateur qui varie selon son rôle.
- Récupérations de mot de passe : Tous les utilisateurs pourront réinitialiser leurs mots de passe grâce à leurs adresses mail.

2.4.2 Exigences non fonctionnelles

Ce sont des besoins qui caractérisent le système, elles sont liées aux contraintes pesant sur les fonctionnalités. La liste suivante représente les besoins non fonctionnels de notre système :

- Performance : l'application doit assurer un temps de réponse minime, tout en répondant aux exigences de l'utilisateur.
- Sécurité : aucune opération ne doit être possible sans authentification préalable. Une protection contre CSRF est impérative. Les informations doivent être accessibles aux utilisateurs possédant les bons droits d'accès seulement. Les mots de passe doivent être hashés puis stockés et jamais stockés en clair. Le système doit identifier la pointeuse afin d'accepter les informations reçues.
- Ergonomie : les informations doivent être lisibles et facilement interprétables. Des graphiques résumant les informations pertinentes doivent être présentés à l'utilisateur. Une navigation simple et efficace. L'utilisation d'icônes explicites permettant de comprendre le sens des actions que l'utilisateur s'apprête à effectuer. Les actions critiques doivent toujours être confirmées avant leurs exécutions. Le système ne doit pas nécessiter une formation longue durée pour être utilisé, il doit offrir une prise en main rapide et facile sans beaucoup d'efforts.
- Modularité du code : compte tenu du contexte et de la problématique traitée. On se doit de penser à l'évolution potentielle du système. Dans cette optique, il est primordial d'écrire un code lisible, modulaire et bien documenté pour faciliter la maintenance, la personnalisation, et l'évolutivité de la solution proposée.

2.5 Diagrammes des cas d'utilisation

Les diagrammes de cas d'utilisation représentent la structure des fonctionnalités nécessaires aux utilisateurs du système [6], nous allons ici modéliser les diagrammes de cas d'utilisation des 4 acteurs du système pour avoir une vue globale de ce dernier.

2.5.1 Diagramme de cas d'utilisation de l'employé

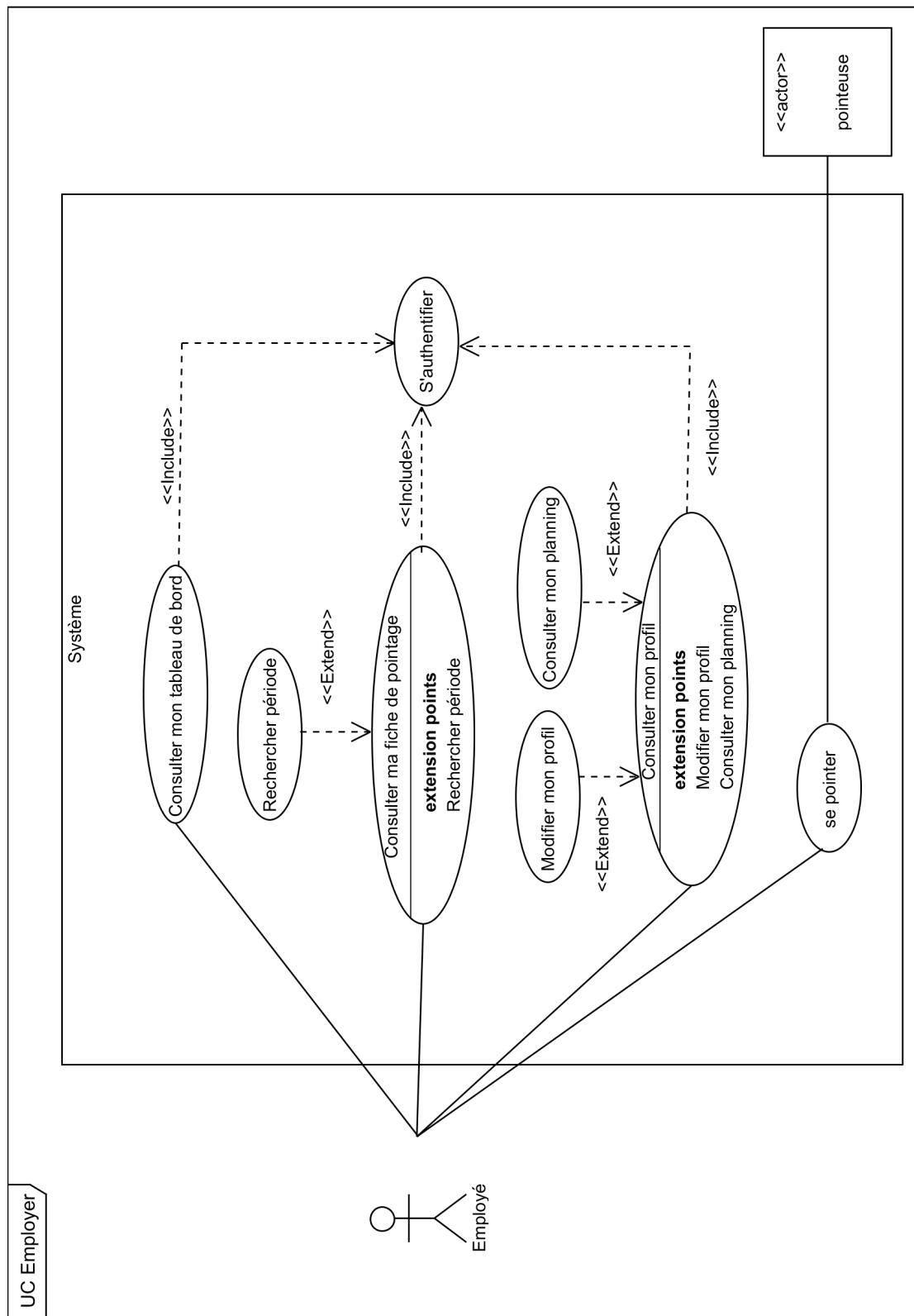


FIGURE 2.1 – Diagramme de cas d'utilisation associé à l'acteur «Employé»

2.5.2 Diagramme de cas d'utilisation du manager

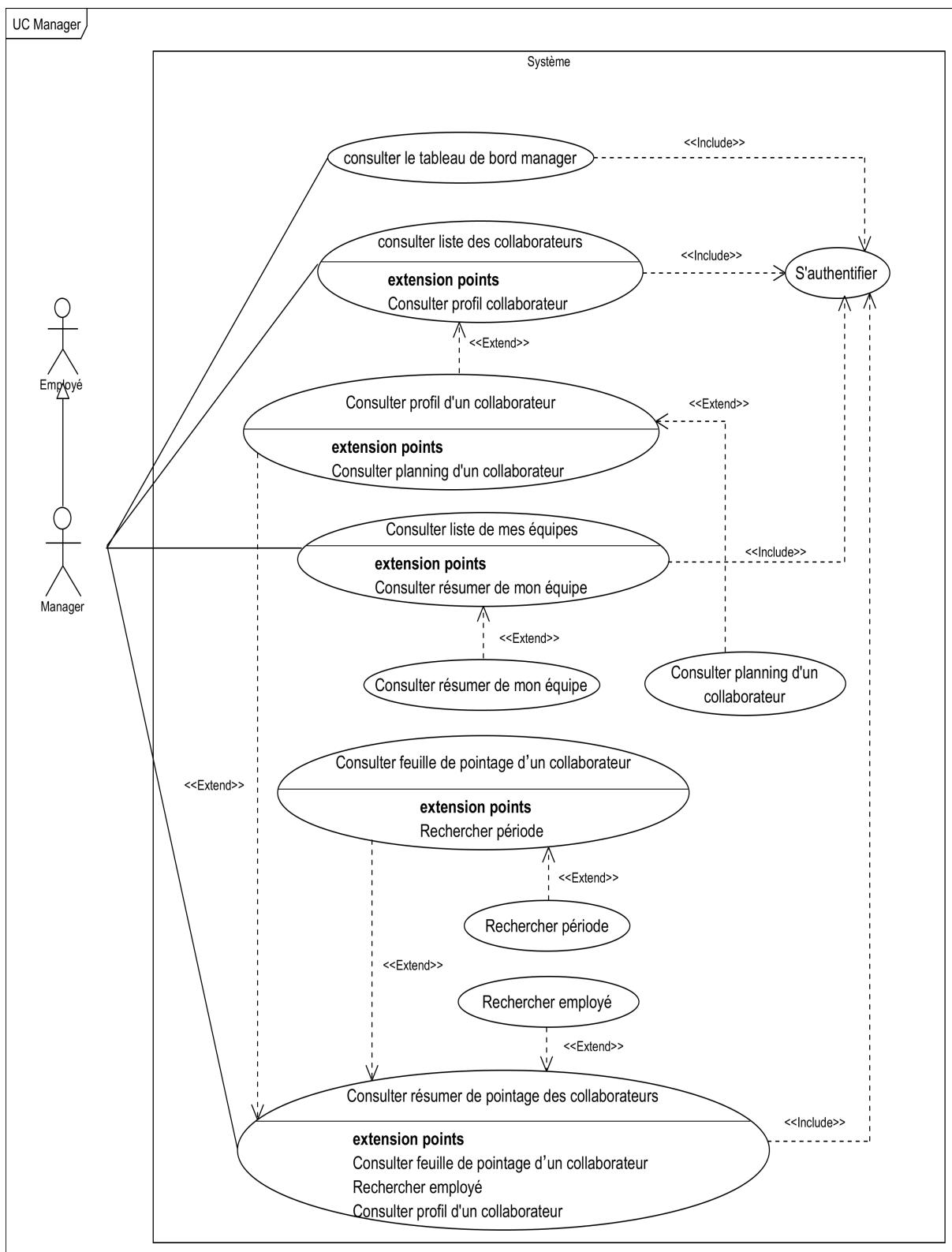


FIGURE 2.2 – Diagramme de cas d'utilisation associé à l'acteur «Manager»

2.5.3 Diagramme de cas d'utilisation du responsable

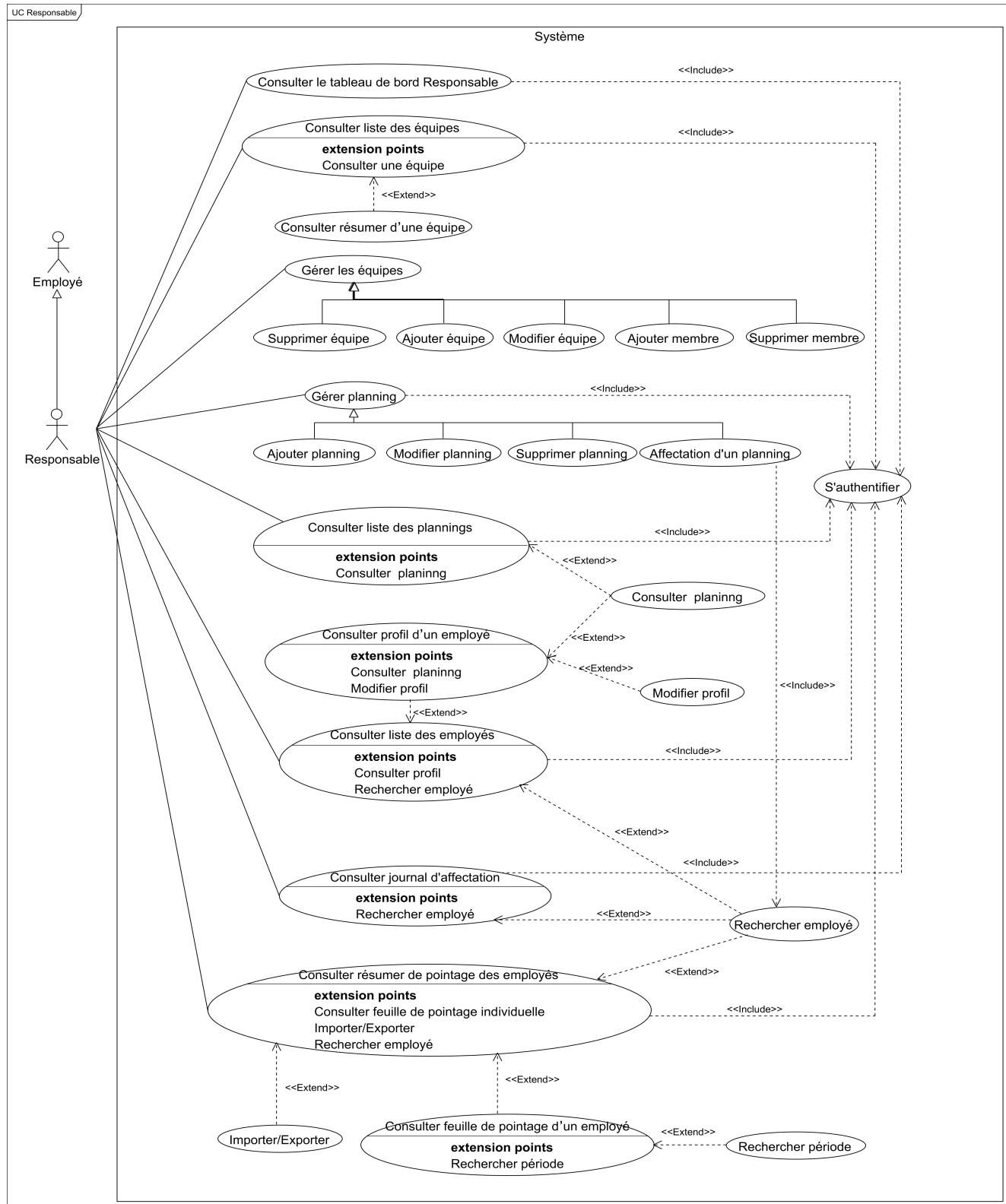


FIGURE 2.3 – Diagramme de cas d'utilisation associé à l'acteur «Responsable»

2.5.4 Diagramme de cas d'utilisation de l'administrateur

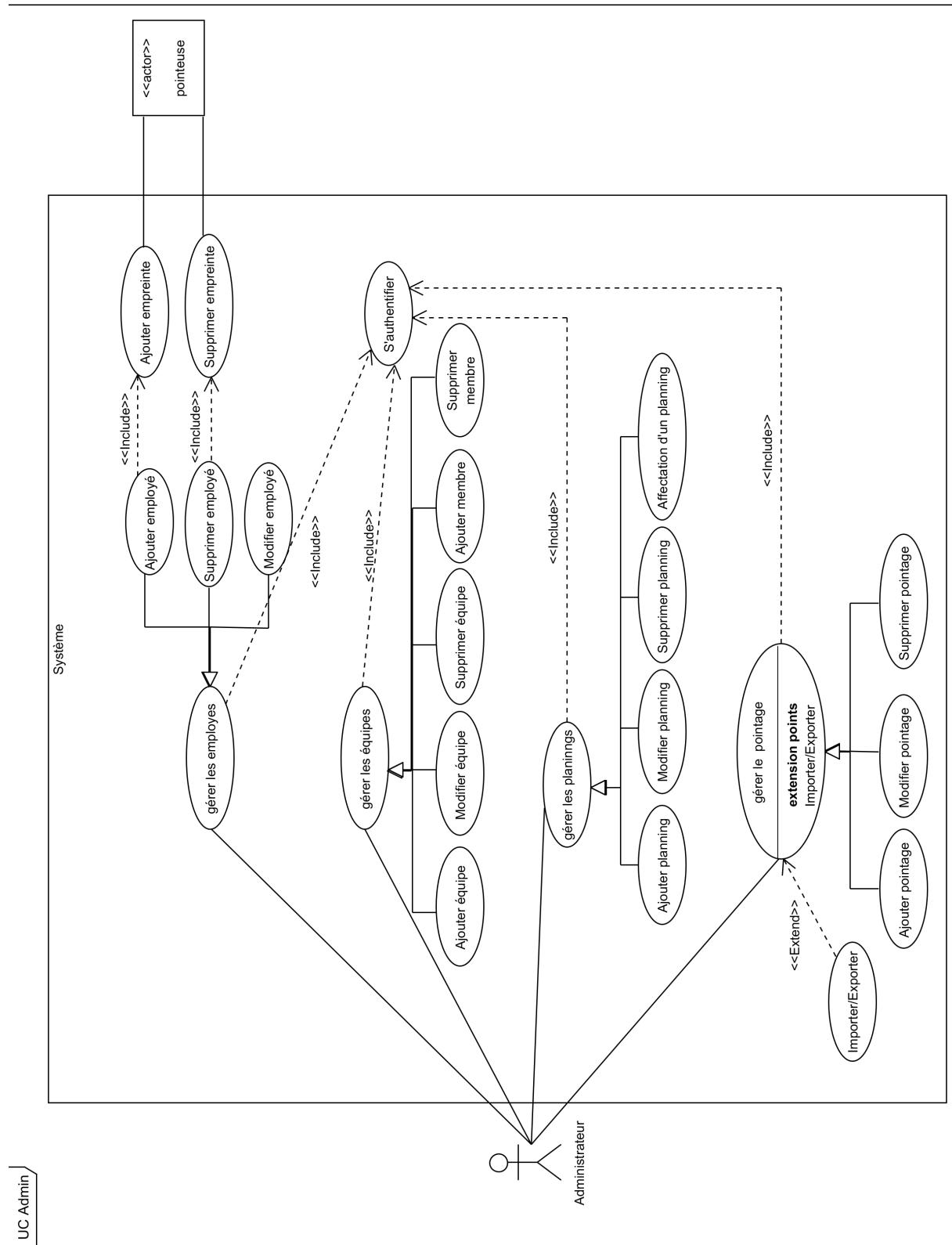


FIGURE 2.4 – Diagramme de cas d'utilisation associé à l'acteur «Administrateur»

Bien qu'exécutés par des acteurs différents, les objectifs de certains cas d'utilisation sont identiques. Ceci peut être illustré par les cas d'utilisation « Consulter mon profil », « Consulter profil d'un collaborateur » et « Consulter Profil d'un employé »

Le premier cas concerne un employé qui consulte son propre profil en ayant la possibilité de modifier certaines informations. Le deuxième cas est déclenché par le manager pour consulter le profil d'un collaborateur sans pouvoir le modifier. Le troisième est utilisé par un administrateur pour consulter le profil d'un employé (quel que soit son rôle) avec la possibilité de modifier toutes les informations.

2.6 Descriptions des cas d'utilisation

Les cas d'utilisation ne représentent pas uniquement les interactions avec les acteurs, mais ils ajoutent également les prés et post conditions ainsi que les enchaînements alternatifs. Dans cette partie, nous allons décrire les diagrammes de cas d'utilisation des différents acteurs.

Cas d'utilisation « S'authentifier »

Sommaire d'identification	
Titre	S'authentifier
Acteur	Employé, Manager, Responsable, Administrateur
Résumé	L'acteur doit s'identifier en saisissant son nom d'utilisateur et mot de passe pour accéder à son espace personnel.
Description des scénarios	
Pré-conditions	L'utilisateur doit être créé par l'administrateur.
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche le formulaire d'authentification. 2. L'acteur saisit le nom d'utilisateur ainsi que son mot de passe. 3. Le système vérifie si les identifiants saisis sont corrects. 4. Le système affiche le tableau de bord (voir le cas d'utilisations « Consulter mon tableau de bord »)
Enchaînement alternatif	<ol style="list-style-type: none"> 2a. Les identifiants saisis par l'acteur sont incorrects. <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur pour signaler que les identifiants sont incorrects. 2. Le cas d'utilisation reprend de l'étape 1 du scénario nominale.
Postconditions	L'utilisateur est authentifié et accède aux fonctionnalités qui lui sont dédiées.

TABLE 2.2 – Description du cas d'utilisation « S'authentifier »

Cas d'utilisation « Se pointer »

Sommaire d'identification	
Titre	Se pointer
Acteur	Employé, Manager, Responsable, Pointeuse
Résumé	Un acteur signale son entrée ou sa sortie de l'entreprise en posant son index sur la pointeuse.
Description des scénarios	
Pré-conditions	Être enregistré dans le système (empreinte et profil)
Scénario nominal	<ol style="list-style-type: none"> L'acteur pose son index sur le lecteur d'empreinte. La pointeuse reconnaît l'acteur et envoie un signal au système. Le système accueille la réception du signal. La pointeuse reçoit un accusé de réception et fait clignoter une LED, pour signaler le bon déroulement de l'opération de pointage.
Enchaînement alternatif	<p>2a La pointeuse ne reconnaît pas l'acteur.</p> <ol style="list-style-type: none"> La LED de la pointeuse ne clignotera pas. Le cas d'utilisation reprend de l'étape 1 du scénario nominale.
Postconditions	Le système enregistre l'heure et l'ID de l'employé responsable de l'événement.

TABLE 2.3 – Description du cas d'utilisation « Se pointer »

Cas d'utilisation « Consulter mon profil »

Sommaire d'identification	
Titre	Consulter mon profil
Acteur	Employé, Manager, Responsable
Résumé	L'acteur accède aux informations qui constituent son profil.
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	<ol style="list-style-type: none"> Les différentes informations du profil sont affichées.
Enchaînement alternatif	<p>1a. L'employé peut modifier ses informations (voir le cas d'utilisations « modifier mon profil »)</p>
Postconditions	

TABLE 2.4 – Description du cas d'utilisation « Consulter mon profil »

Cas d'utilisation « Consulter ma fiche de pointage »

Sommaire d'identification	
Titre	Consulter ma fiche de pointage
Acteur	Employé, Manager, Responsable
Résumé	L'acteur accède aux informations relatives à son pointage (Heures d'arrivée et de sortie)
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	<ol style="list-style-type: none"> Le système affiche les informations de pointage de l'acteur concerné pour une semaine.
Enchaînement alternatif	<ol style="list-style-type: none"> l'acteur choisi d'avoir un affichage par mois. <ol style="list-style-type: none"> Le système affiche les informations de pointage de l'acteur concerné pour un mois.
Postconditions	

TABLE 2.5 – Description du cas d'utilisation « Consulter ma fiche de pointage »

Cas d'utilisation « Consulter tableau de bord manager »

Sommaire d'identification	
Titre	Consulter tableau de bord manager
Acteur	Manager
Résumé	Le manager consulte son tableau de bord, qui est constitué de deux parties. La première représente le tableau de bord du manager en tant qu'employé tandis que la deuxième est un récapitulatif des informations de pointage de son/ses équipes et de tous ses collaborateurs.
Description des scénarios	
Pré-conditions	Le manager doit être authentifié.
Scénario nominal	<ol style="list-style-type: none"> Le système affiche un résumé des informations de pointage concernant le manager et son planning du jour.
Enchaînement alternatif	<ol style="list-style-type: none"> Le manager décide de consulter la deuxième partie de son tableau de bord. <ol style="list-style-type: none"> Le système affiche la partie qui concerne ses équipes ainsi que les collaborateurs encadrés par le manager en question.
Postconditions	

TABLE 2.6 – Description du cas d'utilisation « Consulter tableau de bord manager »

Cas d'utilisation « Ajouter membre »

Sommaire d'identification	
Titre	Ajouter membre
Acteur	Responsable, Administrateur
Résumé	L'acteur accède à une interface qui lui permet d'affecter des employés à l'équipe.
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur recherche un employé (voir le cas d'utilisations <u>Recherche employé</u>). 2. Le système retourne une liste d'employés. 3. L'acteur sélectionne un employé et l'ajoute. 4. Le système affiche l'interface d'affectation avec les données mise à jour.
Enchaînement d'erreur	<p>2a. Aucun employé trouvé.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur.
Postconditions	

TABLE 2.7 – Description du cas d'utilisation « Ajouter membre »

Cas d'utilisation « Ajouter un planning »

Sommaire d'identification	
Titre	Ajouter planning
Acteur	Responsable, Administrateur
Résumé	L'acteur accède à une interface qui lui permet la création d'un nouveau planning.
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche le formulaire. 2. L'acteur saisit le nom du planning ainsi que la description et les horaires de travail. 3. Le système vérifie la conformité des informations saisies. 4. Le système affiche la liste des plannings (voir cas d'utilisations <u>Consulter liste des plannings</u>.)

Enchaînement alternatif	2a. Le nom de l'équipe est déjà existant. 1. Le système affiche un message d'erreur pour signaler que le nom du planning existe dans la base de données. 2. Le cas d'utilisation reprend de l'étape 1 du scénario nominale.
Postconditions	

TABLE 2.8 – Description du cas d'utilisation « Ajouter un planning »**Cas d'utilisation « Ajouter équipe »**

Sommaire d'identification	
Titre	Ajouter équipe
Acteur	Responsable, Administrateur
Résumé	L'acteur accède à une interface qui lui permet la création d'une nouvelle équipe.
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	1. Le système affiche le formulaire. 2. L'acteur saisit le nom de l'équipe ainsi que la description et sélectionne le manager de cette dernière. 3. Le système vérifie la conformité des informations saisies. 4. Le système affiche l'interface d'affectation des membres de l'équipe (voir le cas d'utilisations « affectation des membres »).
Enchaînement alternatif	2a. Le nom de l'équipe est déjà existant. 1. Le système affiche un message d'erreur pour signaler que le nom de l'équipe existe dans la base de données. 2. Le cas d'utilisation reprend de l'étape 1 du scénario nominale.
Postconditions	

TABLE 2.9 – Description du cas d'utilisation « Ajouter équipe »**Cas d'utilisation « Ajouter un employé »**

Sommaire d'identification	
Titre	Ajouter un employé
Acteur	Administrateur

Résumé	L'administrateur ajoute un employé.
Description des scénarios	
Pré-conditions	Être authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur saisit l'ensemble des informations de l'employé. 2. L'administrateur enregistre l'employé. 3. L'administrateur enregistre l'empreinte de l'employé (Cas d'utilisation <u>Ajouter une empreinte</u>).
Enchaînement alternatif	<p>2a Les informations saisies ne sont pas valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message qui spécifie les informations incorrectes et demande à l'administrateur de les corriger. 2. Le cas d'utilisation reprend de l'étape 1 du scénario nominal.
Postconditions	Mise à jour des données présentes dans la base de données.

TABLE 2.10 – Description du cas d'utilisation « Ajouter un employé »

Cas d'utilisation « Consulter profil d'un employé »

Sommaire d'identification	
Titre	Consulter profil d'un employé
Acteur	Responsable
Résumé	Le responsable consulte le profil d'un employé.
Description des scénarios	
Pré-conditions	Le responsable doit être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur sélectionne un employé (Cas d'utilisation <u>Consulter liste des employés</u>). 2. Le système affiche le profil de l'employé sélectionné. 3. L'acteur peut modifier le profil (Cas d'utilisation <u>Modifier profil</u>).
Enchaînement alternatif	
Postconditions	

TABLE 2.11 – Description du cas d'utilisation « Consulter profil d'un employé »

Nous avons établi la totalité des descriptions des cas d'utilisation, mais nous avons décidé de ne pas inclure tous les tableaux afin de ne pas encombrer le lecteur. Néanmoins, le reste des descriptions est cité dans l'annexe A.

2.7 Diagrammes de séquence système DSS

Nous utilisons le terme de diagramme de séquence « système » pour souligner le fait que nous considérons le système informatique comme une boîte noire, nous ouvrirons la boîte noire seulement en conception.[5]

2.7.1 Cas d'utilisation « Se pointer »

L'employé doit marquer ses heures de travail via la pointeuse biométrique, qui vérifie son empreinte puis enregistre son pointage et lui signal le bon déroulement de l'opération en allumant une LED.

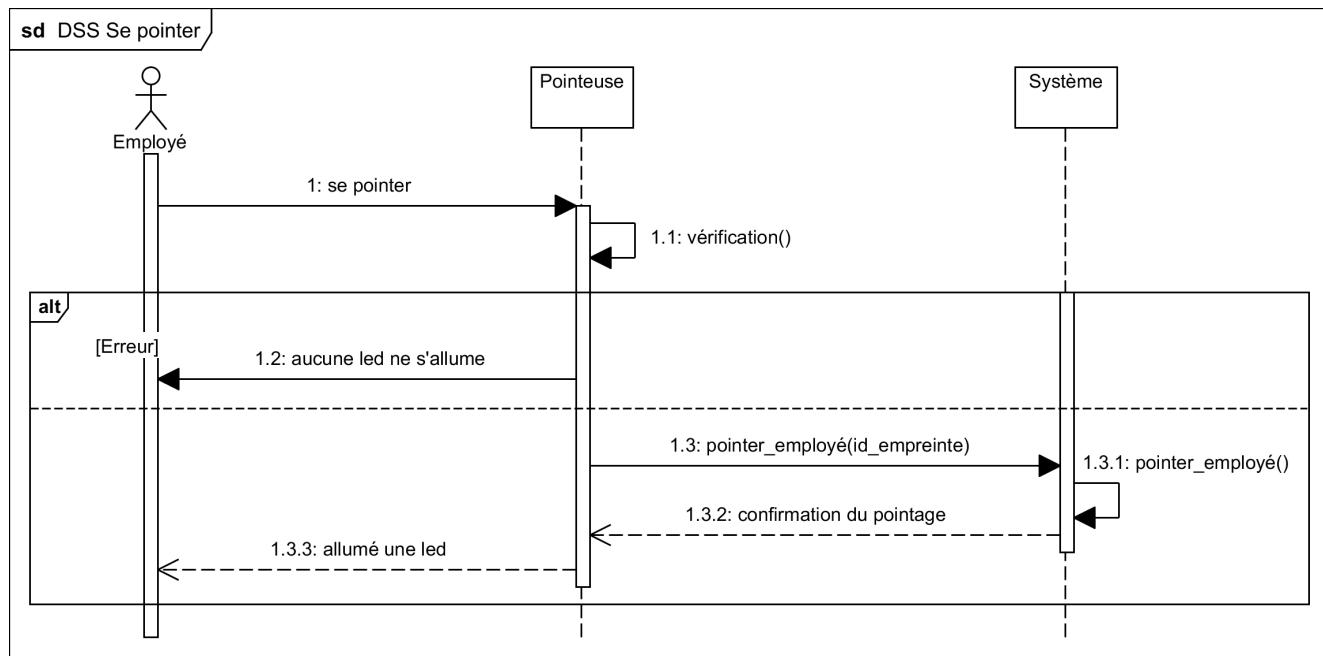


FIGURE 2.5 – Diagramme de séquence système « Se pointer »

2.7.2 Cas d'utilisation « Consulter mon profil »

L'employé peut décider de consulter son profil et avoir accès aux différentes informations qui le concernent et modifier certaines valeurs s'il le souhaite, ou accéder à son planning.

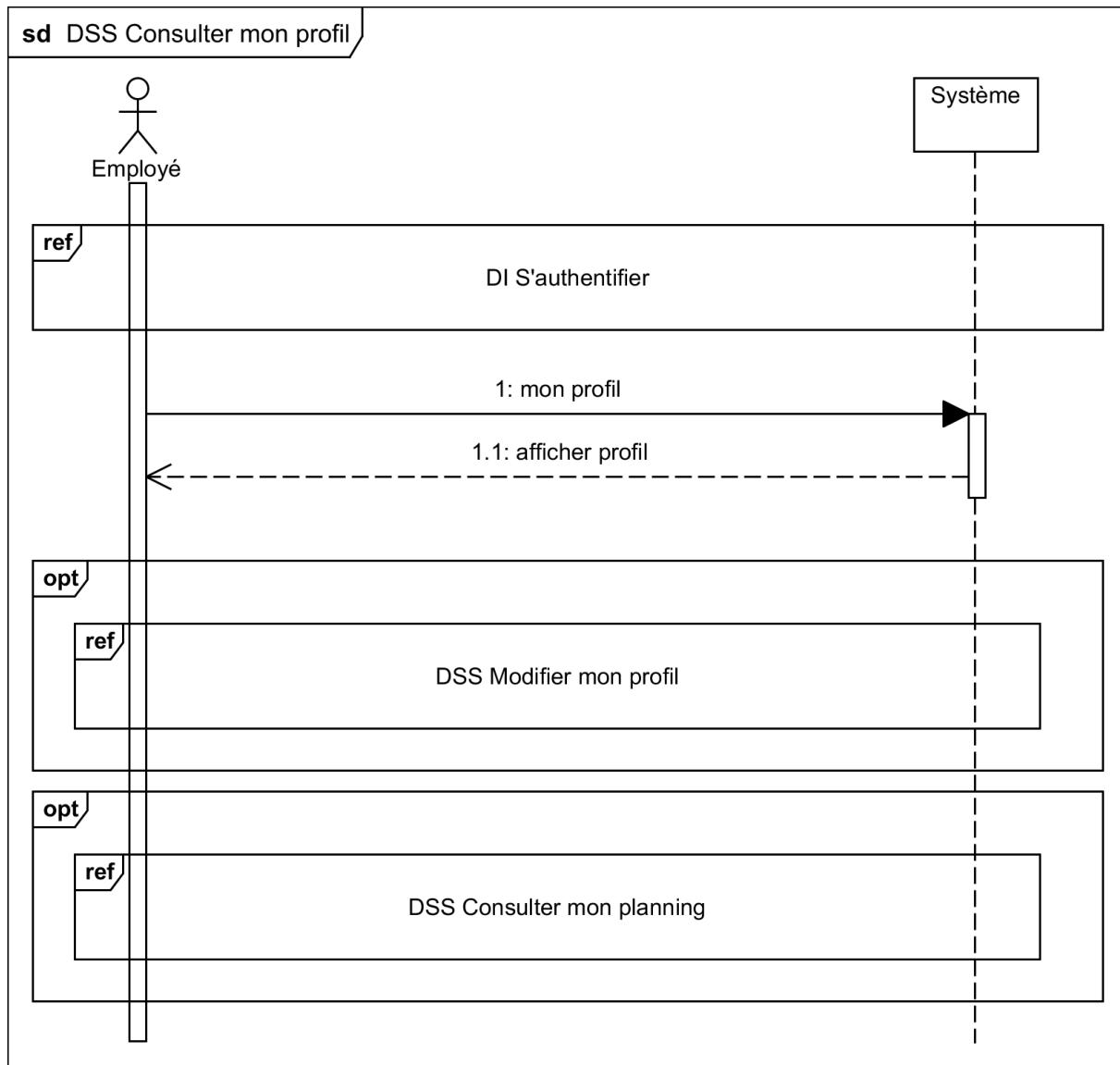


FIGURE 2.6 – Diagramme de séquence système « Consulter mon profil »

2.7.3 Cas d'utilisation « Consulter ma fiche de pointage »

Un employé peut consulter sa propre fiche de pointage qui comporte toutes les informations concernant ses heures d'arrivée et de sortie des jours précédents. Il peut aussi avoir un affichage par semaine ou par mois.

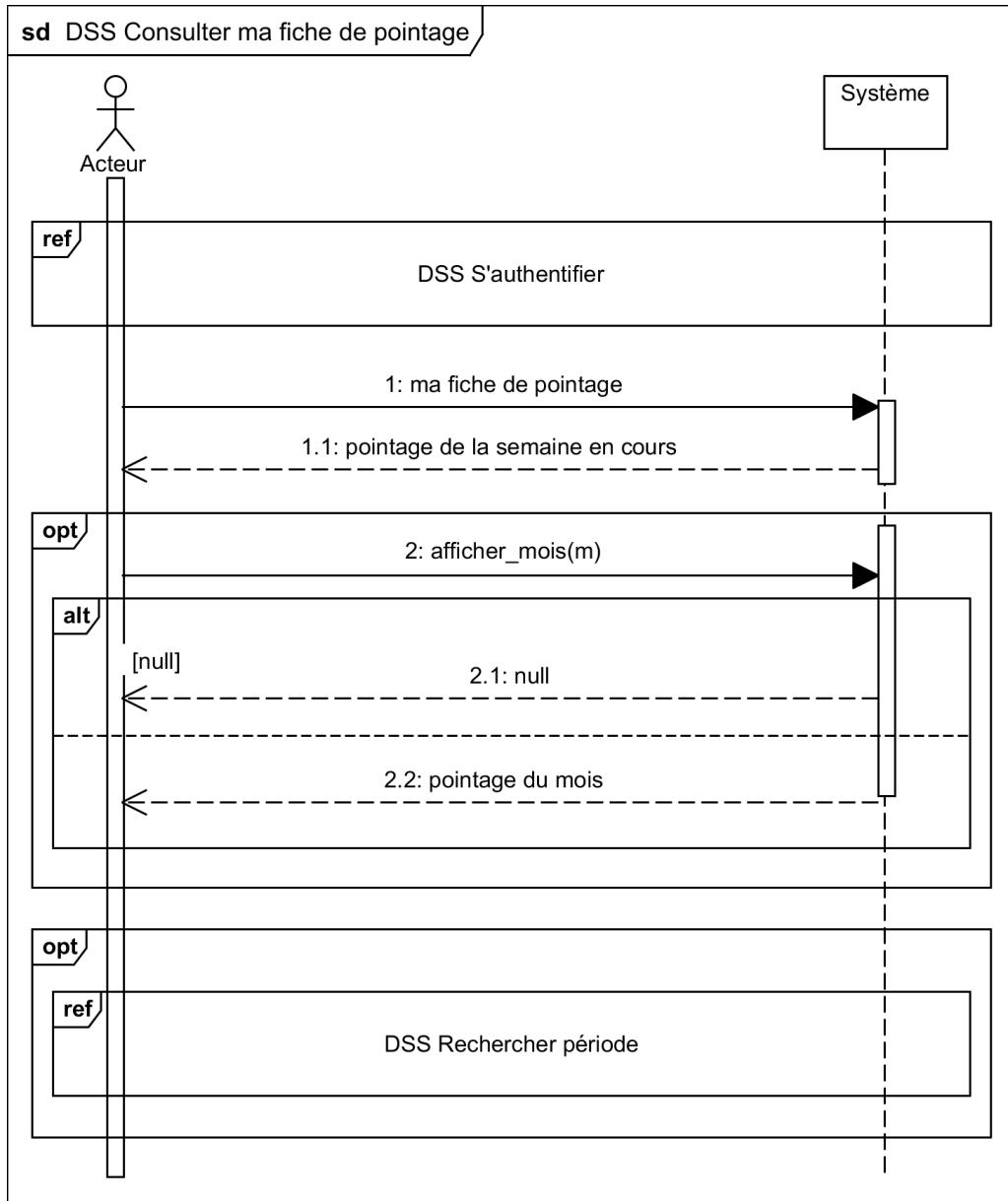


FIGURE 2.7 – Diagramme de séquence système « Consulter ma fiche de pointage »

2.7.4 Cas d'utilisation « Consulter tableau de bord manager »

Une fois authentifié, le manager sera redirigé directement vers son tableau de bord qui est composé d'un espace en tant qu'employé et d'un autre consacré à ses équipes. Grâce auquel il aura un récapitulatif de présence de ses collaborateurs, leurs nombres, ainsi que leur dernier pointage et le temps passé en poste.

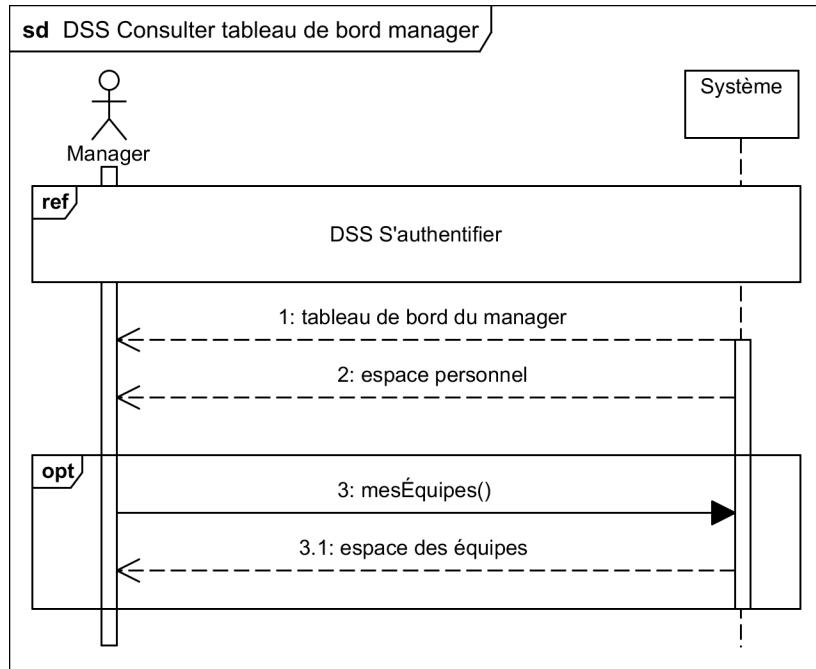


FIGURE 2.8 – Diagramme de séquence système « Consulter tableau de bord manager »

2.7.5 Cas d'utilisation « Ajouter membre »

Un responsable décide d'ajouter un membre à une équipe, il commence par sélectionner l'équipe puis il effectue une recherche pour trouver l'employé pour enfin l'ajouter en tant que membre de cette dernière.

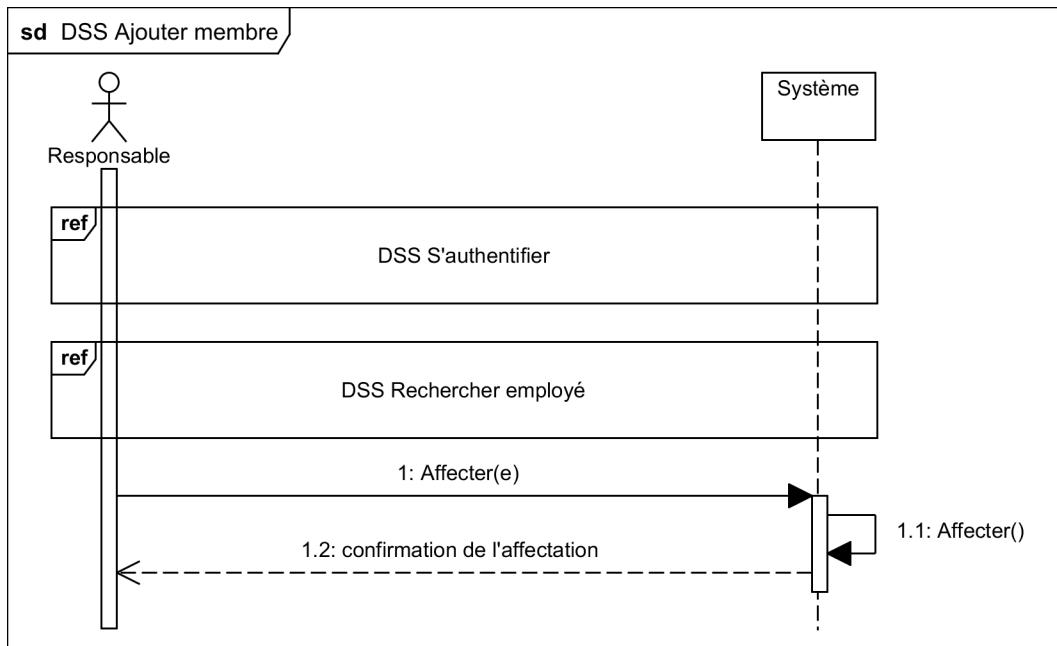


FIGURE 2.9 – Diagramme de séquence système « Ajouter membre »

2.7.6 Cas d'utilisation « Ajouter planning »

Un responsable peut créer un planning d'une semaine où il définit les jours de travail ainsi que les horaires pour chaque jour. Pour ce, il doit saisir un intitulé pour le planning et une description ainsi que les horaires d'entrée et de sortie.

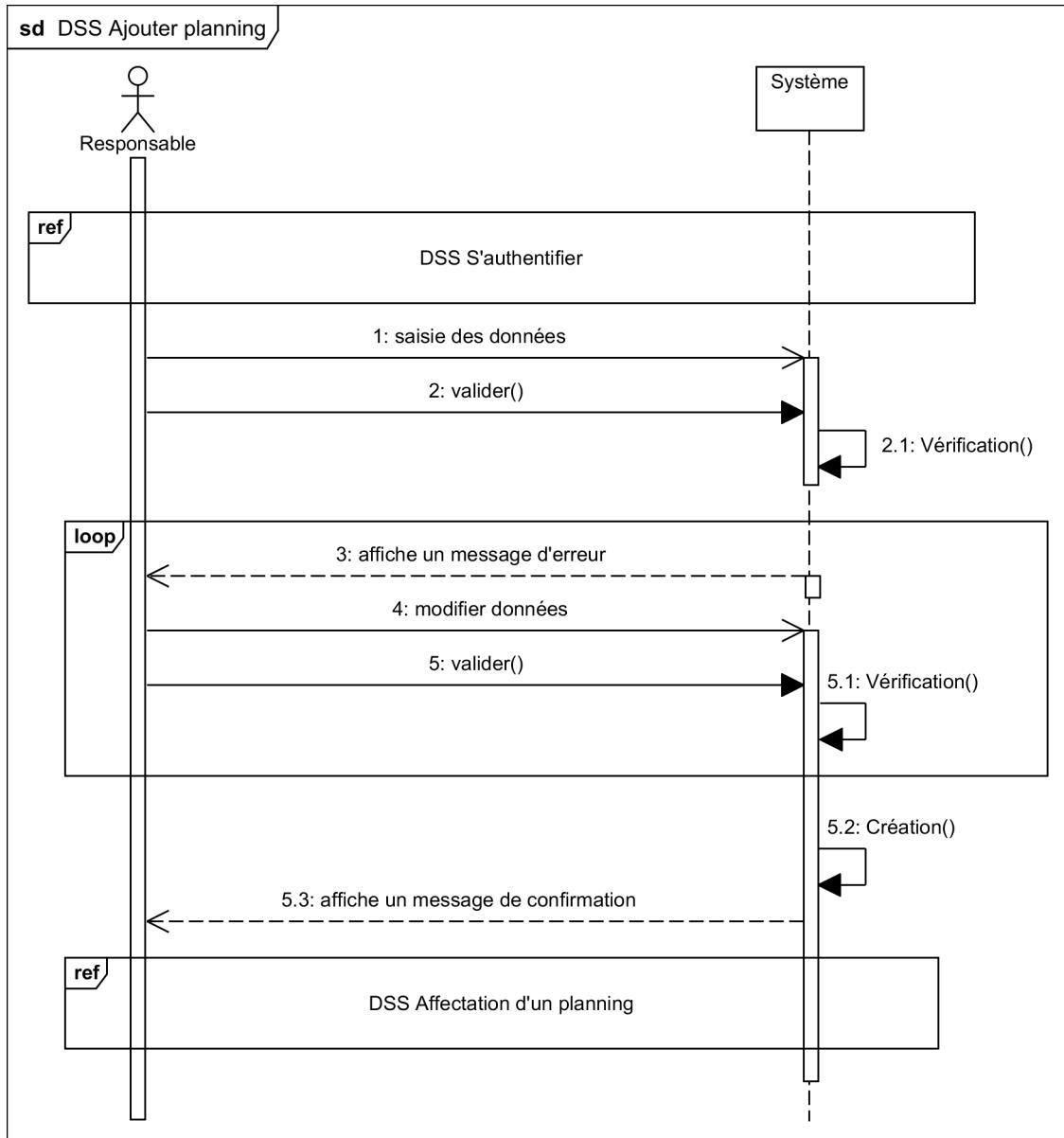


FIGURE 2.10 – Diagramme de séquence système « Ajouter planning »

2.7.7 Cas d'utilisation « Ajouter équipe »

Pour créer une équipe, le responsable doit saisir l'intitulé et la description de l'équipe puis lui attribuer un manager. Une fois l'équipe créée, il pourra ajouter des membres à cette équipe.

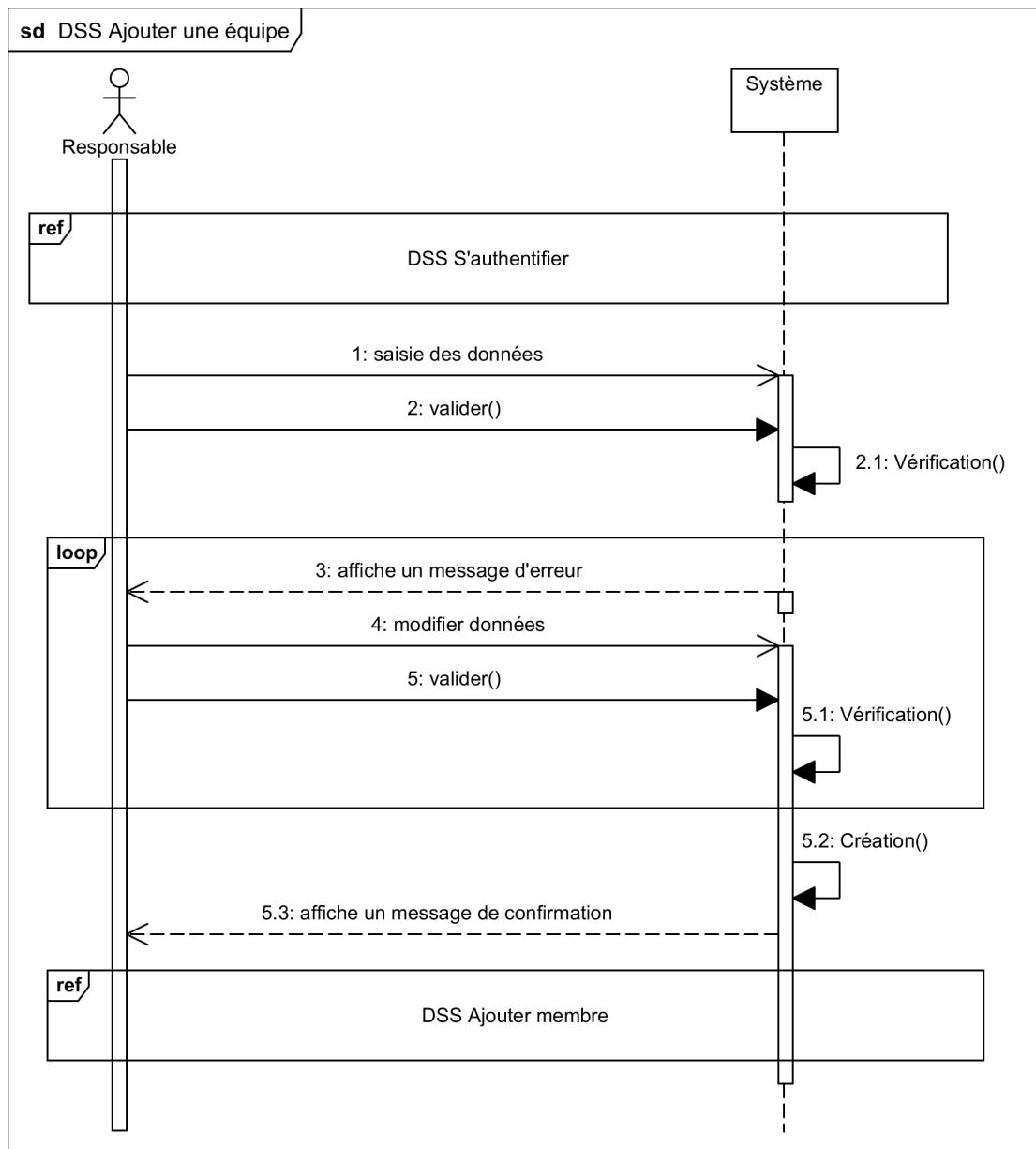


FIGURE 2.11 – Diagramme de séquence système « Ajouter équipe »

2.7.8 Cas d'utilisation « Ajouter employé »

L’administrateur peut créer un employé, il doit saisir les informations le concernant. Une fois l’employé crée, il doit ajouter son empreinte.

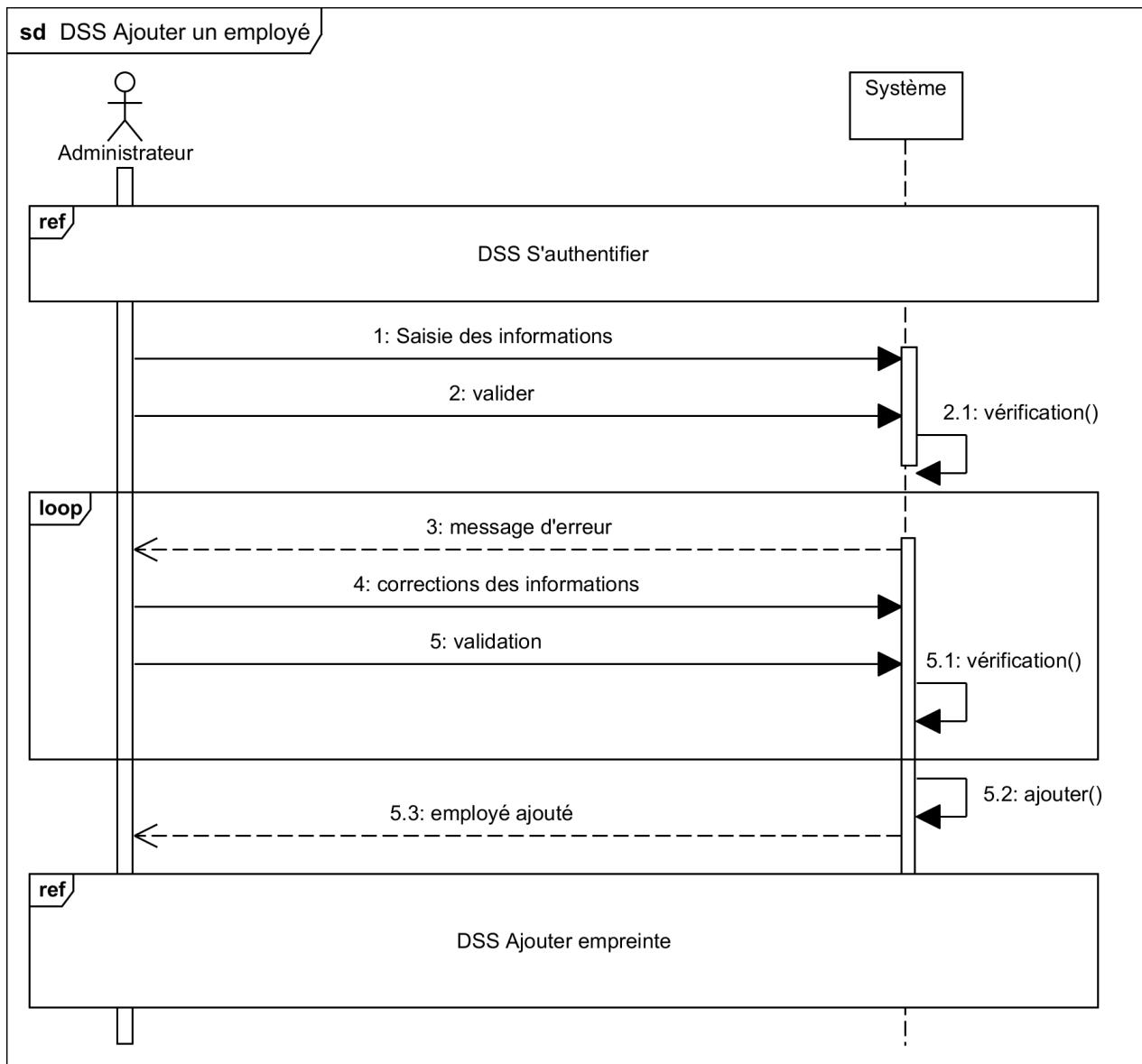


FIGURE 2.12 – Diagramme de séquence système « Ajouter employé »

2.7.9 Cas d'utilisation « Consulter le profil d'un employé »

Le responsable peut consulter le profil d'un employé qui contient les informations de ce dernier. Il aura aussi la possibilité de le modifier ainsi que de consulter son planning.

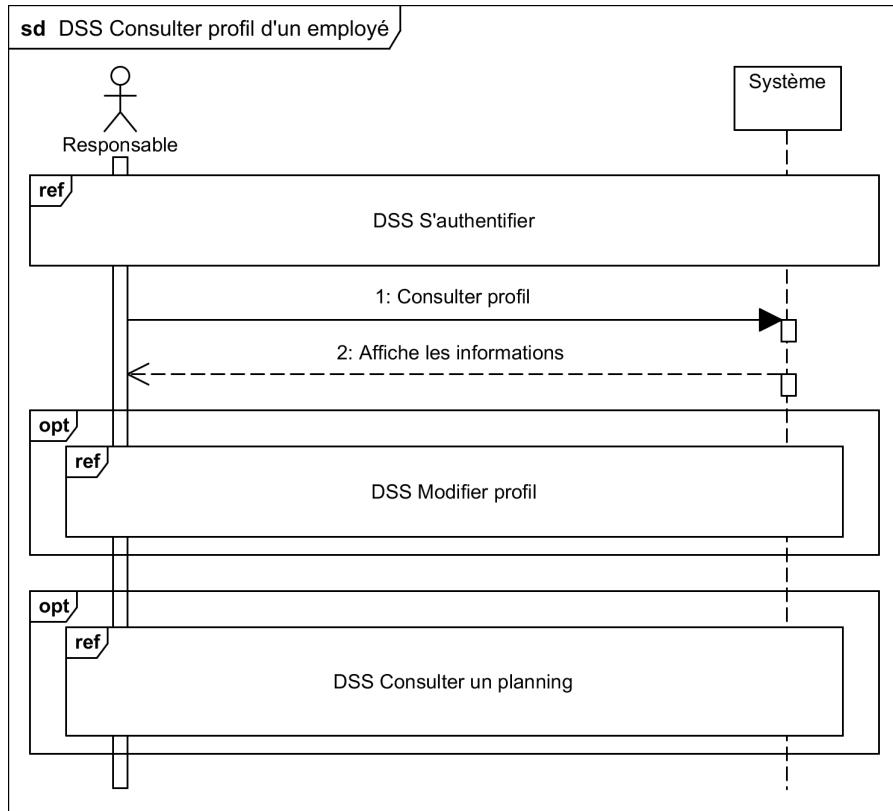


FIGURE 2.13 – Diagramme de séquence système « Consulter profil d'un employé »

Dans un souci de lisibilité, nous avons déplacé le reste des diagrammes de séquence système vers l'annexe B.

2.8 Les maquettes IHM

2.8.1 Employé

La figure 2.14 : Une fois, authentifié, cette interface sera la première affichée à l'utilisateur ayant le rôle d'un employé. On remarque un menu sur le côté gauche, un composant qui sera présent sur la totalité des interfaces du projet, et qui permet une navigation entre les différentes fonctionnalités qu'offre le système tout en s'adaptant au rôle de l'utilisateur connecté. L'utilisateur peut à tout moment se déconnecter grâce au bouton présent au bas du menu.

Cette interface représente le tableau de bord de l'employé, elle regroupe les informations les plus récentes et les plus pertinentes, comme ses horaires du jour ou son dernier pointage ainsi que le temps passé en poste. Une seconde partie est dédiée à des informations concernant ses collaborateurs faisant partie de son équipe.

La figure 2.15 : représente l'affichage du profil de l'employé lui-même, où ses informations sont regroupées en catégories selon leurs utilité et leurs importance. Il existe un bouton qui permet à l'employé de visualiser son planning à partir de cette interface directement.

LAVANCE-

MON TABLEAU DE BORD | MON PROFIL | MA FICHE DE POINTAGE | MON PLANNING | DÉCONNEXION | Username

En poste depuis :
2h:05m

Vous sortez dans
1h:05m

Le planning du jour :

Entrée 1: 08:00	Sortie 1: 12:00	Entrée 2: 13:00	Sortie 2: 17:00
--------------------	--------------------	--------------------	--------------------

Liste des collaborateurs présents :

#	Nom de l'employé	Présent	08:02	03h05m	Homme	⋮
#1	Nom de l'employé	Présent	08:02	03h05m	Homme	⋮
#1	Nom de l'employé	Présent	08:02	03h05m	Homme	⋮
#1	Nom de l'employé	Présent	08:02	03h05m	Homme	⋮
#1	Nom de l'employé	Présent	08:02	03h05m	Homme	⋮
#1	Nom de l'employé	Présent	08:02	03h05m	Homme	⋮

FIGURE 2.14 – Maquette tableau de bord Employé

LAVANCE-

MON TABLEAU DE BORD | MON PROFIL | MA FICHE DE POINTAGE | MON PLANNING | DÉCONNEXION | Username

INFORMATION PERSONNEL

NOM D'UTILISATEUR	
Username	
NOM	PRÉNOM
Nom de l'employé	Prénom de l'employé
TYPE DE CONTRAT	NUMERO DE L'EMPLOYÉ
Type	XXXXXXXXXXXXXXXXXXXXXX
DATE DE NAISSANCE	DATE DE RECRUTEMENT
10/01/1970	12/05/2020
GENRE	ETAT CIVIL
Genre	Etat civil

INFORMATION PUBLIQUE

FONCTION	TELÉPHONE PROFESSIONNEL
Fonction de l'employé	+213XXXXXXXX
DÉSCRIPTION	
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.	

COORDONNÉES

ADRESSE	Adresse
CODE POSTAL	VILLE
Code postal	Ville
TÉLÉPHONE	
+213XXXXXXXXXXXX	
EMAIL	
Nom de l'employé	
CONTACT D'URGENCE	
Nom du contact	Lien du contact
+213XXXXXXXXXXXX	

Afficher le Planning

FIGURE 2.15 – Maquette profil Employé

2.8.2 Manager

La figure 2.16 : représente le tableau de bord du manager qui est composé de deux onglets. Le premier onglet est identique au tableau de bord de l'employé simple, tandis que l'onglet « Mes équipes » récapitule l'ensemble des informations concernant les équipes sous sa responsabilité. Il met en évidence le nombre total des équipes et des collaborateurs, le récapitulatif de présence de ces derniers sous forme de graphe, ainsi qu'une liste des employés avec leurs derniers pointages

La figure 2.17 : cette interface affiche les informations qui concernent une équipe sélectionnée au préalable par le manager. On peut y voir le titre, la description de l'équipe ainsi que des statistiques. Dans la partie inférieure, on liste l'ensemble des employés appartenant à cette équipe avec la possibilité d'un accès direct vers le profil de chacun d'eux.

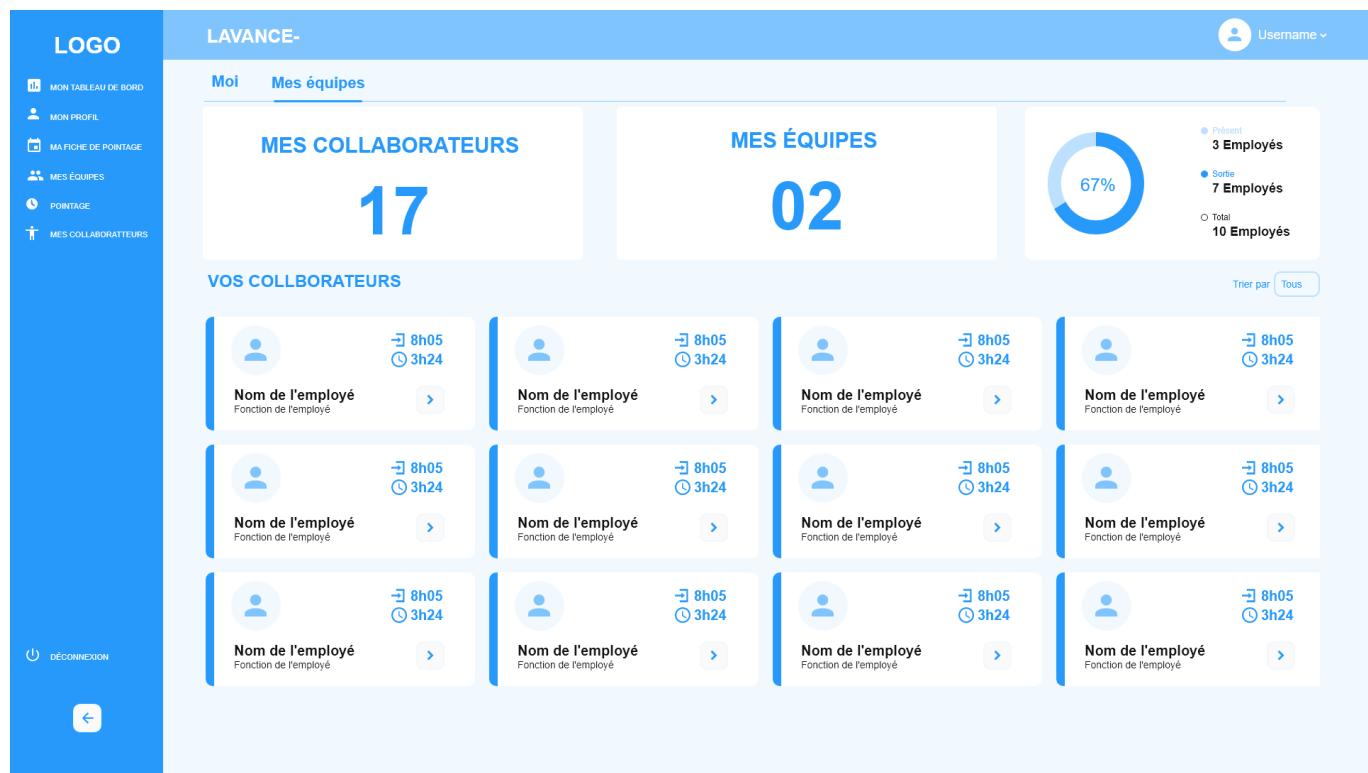


FIGURE 2.16 – Maquette tableau de bord Manager

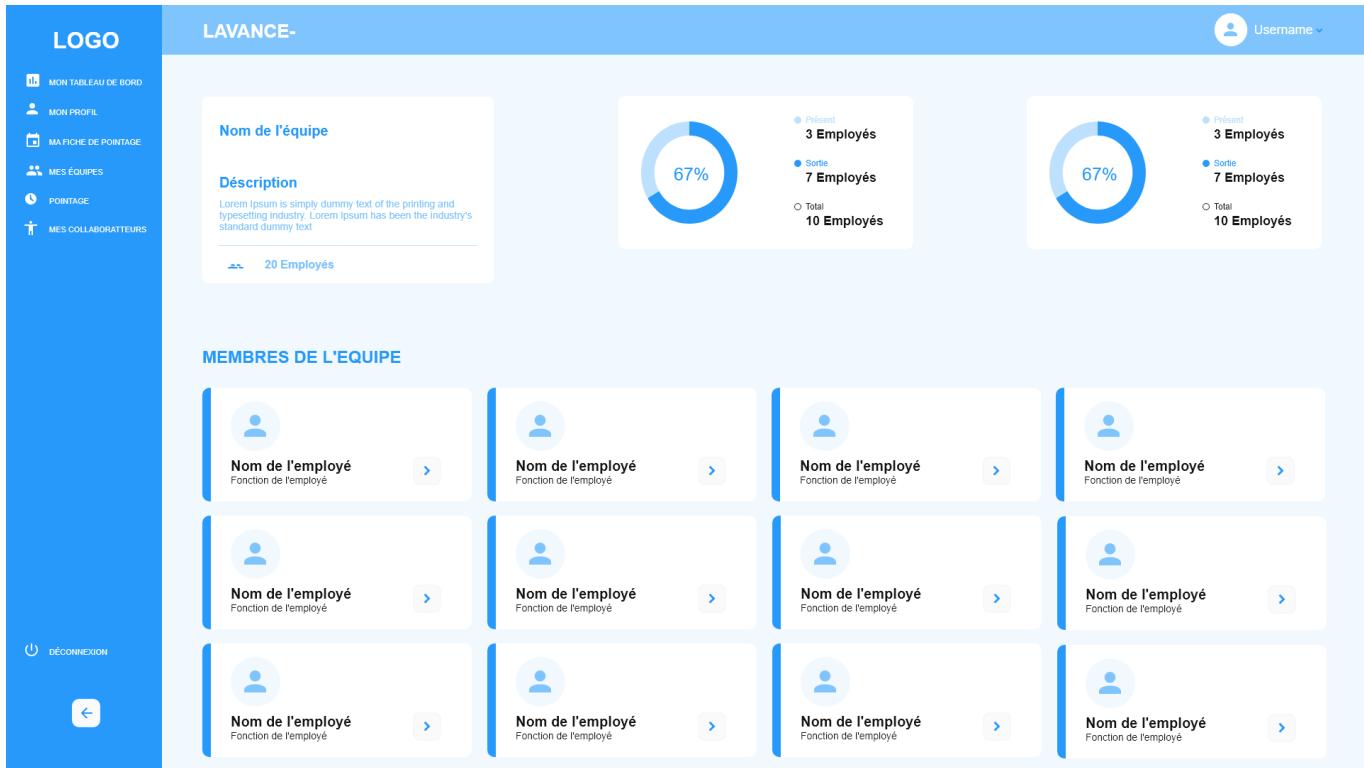


FIGURE 2.17 – Maquette des équipes d'un Manager

2.8.3 Responsable

La figure 2.18 : représente l'interface dans laquelle le responsable crée un planning en lui attribuant un nom et une description et les horaires de début et de fin des périodes de travail tout au long de la semaine, avec chaque jour étant composé de deux parties.

La figure 2.19 : représente l'interface avec laquelle un responsable affecte un planning à un ou des employés en effectuant une recherche par lettre qui affiche une liste des employés ayant un nom ou un prénom correspondant aux caractères saisis, et un bouton qui permet d'affecter l'employé sélectionné. L'interface comporte aussi une liste des employés ayant déjà été affectés à ce planning avec la possibilité de détacher un employé du planning.

Logo

LAVANCE-

Crée un nouveau planning

Nom du planning

Décription du planning

Dimanche

Lundi

Mardi

Mercredi

Jeudi

Vendredi

Samedi

Partie 1

Partie 2

DÉCONNEXION

Créer le planning

Figure 2.18 : Maquette de création d'un planning. La page affiche un formulaire pour créer un nouveau planning. Il y a un champ pour le nom du planning, une zone de description avec un placeholder en français, et une grille pour les journées de la semaine (dimanche à samedi) divisée en deux parties. Chaque jour a deux cases vides dans chaque partie. En bas à droite, il y a un bouton bleu "Créer le planning".

FIGURE 2.18 – Maquette création d'un planning

Logo

LAVANCE-

Crée une nouvelle équipe

Nom de l'équipe

Description de l'équipe

Manager de l'équipe

Manager

Membres

Membres

AJOUTER

Liste des membres

Nom de l'employé	username	administrateur	Homme	+213XXXXXXXXXX	Présent	⋮
Nom de l'employé	username	administrateur	Homme	+213XXXXXXXXXX	Présent	⋮
Nom de l'employé	username	administrateur	Homme	+213XXXXXXXXXX	Présent	⋮
Nom de l'employé	username	administrateur	Homme	+213XXXXXXXXXX	Présent	⋮
Nom de l'employé	username	administrateur	Homme	+213XXXXXXXXXX	Présent	⋮

DÉCONNEXION

Figure 2.19 : Maquette d'affectations d'un employé à une équipe. La page affiche un formulaire pour créer une nouvelle équipe. Il y a des champs pour le nom de l'équipe, une description, et le manager. Ensuite, il y a un champ pour les membres et un bouton "AJOUTER". Enfin, il y a une liste des membres actuels, qui sont tous marqués comme "Présent".

FIGURE 2.19 – Maquette affectations d'un employé à une équipe

À travers ces quelques maquettes, nous avons essayé de vous transmettre notre vision générale du projet sur son aspect graphique, et la composition globale de certaines interfaces. Le reste des maquettes sera présenté de manière détaillée dans l'annexe.

2.8.4 La navigation

L'outil utilisé pour réaliser les maquettes offre la possibilité de créer un prototype complet et très représentatif du résultat final. Ci-dessous, des liens vers les prototypes réalisés comportant toutes les interfaces et les liens de navigations.

Prototype employé : lien du prototype 1 URL :<<https://xd.adobe.com/view/9af474c-7053-4831-4a92-bb9d2f4d8c65-eccc/?fullscreen&hints=off>>

Prototype manager : lien du prototype 2 URL :<<https://xd.adobe.com/view/855f8afc-8c35-451d-7b50-838460e72c7e-f36e/?fullscreen&hints=off>>

Prototype responsable : lien du prototype 3 URL :<<https://xd.adobe.com/view/8c62407e-81ae-4950-60f2-88006b652695-fa53/?fullscreen&hints=off>>

2.9 Conclusion

Ce chapitre nous a permis d'exprimer et d'analyser les besoins permettant de décrire les fonctionnalités du système de manière globale. Aussi, grâce à l'identification des acteurs et des cas d'utilisations, nous avons formalisé les différents besoins fonctionnels et non fonctionnels, défini les diagrammes de cas d'utilisations, réalisé les maquettes IHM et la navigation. Tout ceci nous permettra, dans le chapitre suivant, de réaliser les différents modèles de la phase de conception.

Chapitre 3

Conception

3.1 Introduction

Dans ce chapitre, nous allons nous intéresser à la conception de notre système. Le but est de définir et de mettre en place les choix d'architecture technique, et de compléter la description du système sous l'angle technique. Nous étendrons donc la représentation des diagrammes effectuée au niveau de l'analyse en y intégrant les aspects techniques les plus proches des préoccupations physiques.

3.2 Modèle de domaine

C'est le résultat d'une analyse du domaine. Il est considéré comme étant la première version du diagramme de classe. Ce modèle doit définir les classes qui modélisent les entités ou concepts présents dans le domaine (on utilise aussi le terme de métier) de l'application.

Il s'agit donc de produire un modèle des objets du monde réel dans un domaine donné. Ces entités ou concepts peuvent être identifiés directement à partir de la connaissance du domaine ou par des entretiens avec des experts du domaine. Il faut absolument utiliser le vocabulaire du métier pour nommer les classes et leurs attributs. Les classes du modèle de domaine ne doivent pas contenir des opérations, mais uniquement des attributs[7]. Les étapes à suivre pour établir ce diagramme sont :

- Identifier les entités ou concepts du domaine.
- Identifier et ajouter les associations et les attributs.
- Organiser et simplifier le modèle en éliminant les classes redondantes et en utilisant l'héritage.
- Le cas échéant, structurer les classes en paquetage selon les principes de cohérence et d'indépendance.

Comment identifier les concepts du domaine ? Plutôt que de partir à l'aveugle et nous heurter à la taille du problème à résoudre, nous allons prendre les cas d'utilisation un par un et nous poser pour chacun la question suivante : quels sont les concepts métier qui participent à ce cas

d'utilisation ? En suivant cette méthode, nous avons sélectionné des cas d'utilisation différents afin d'avoir une vue globale sur les différentes entités qui composent le système sans répétition.

Modèle du domaine du cas d'utilisation « Se pointer »

Pour marquer les horaires de travaux d'un employé, nous avons besoin d'une entité employé, d'une empreinte pour l'identifier ainsi qu'une entité shift qui permet de sauvegarder ses pointages.

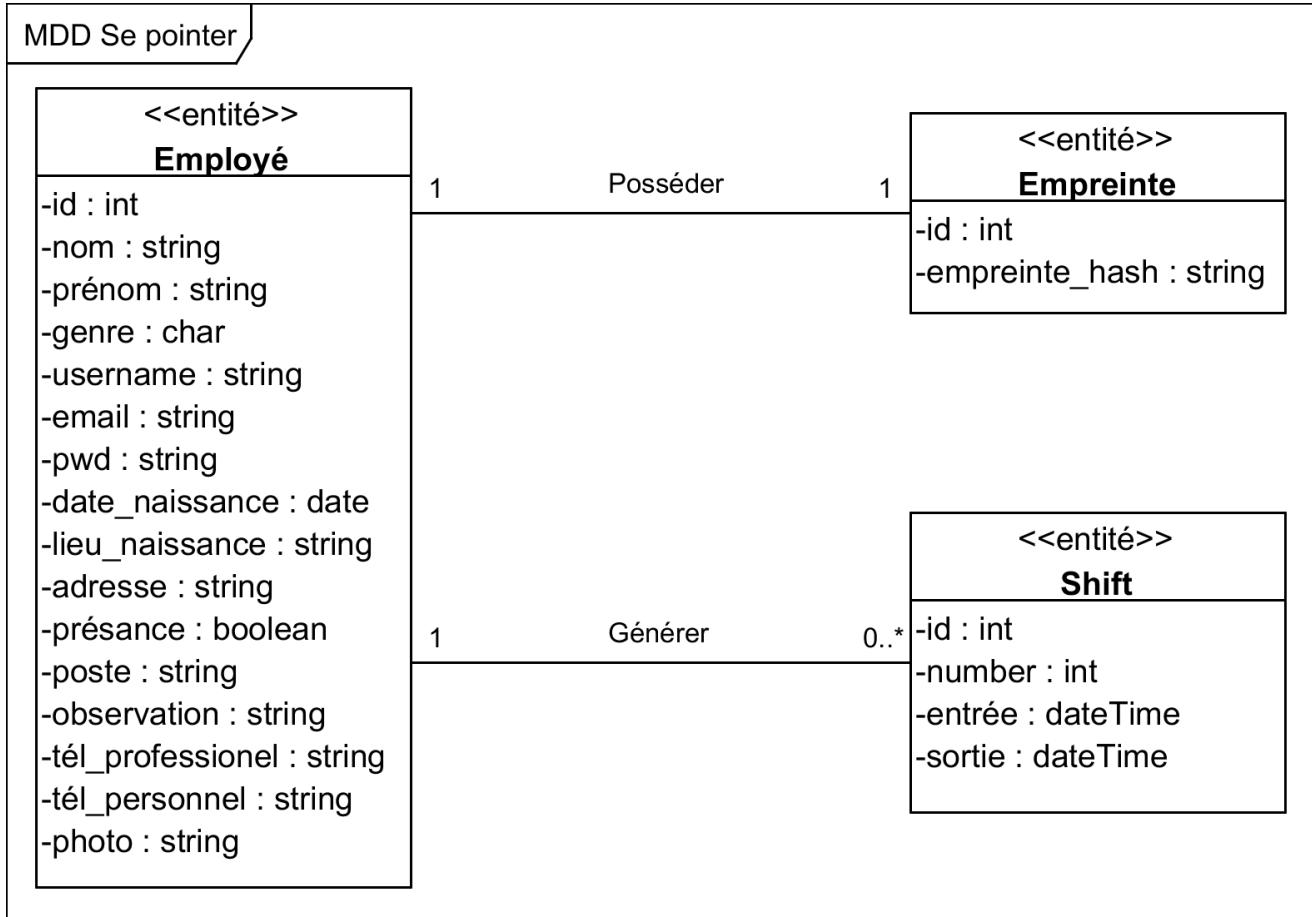


FIGURE 3.1 – Modèle du domaine « Se pointer »

Modèle du domaine du cas d'utilisation « Consulter ma fiche de pointage »

Ce diagramme du modèle du domaine représente les entités concernées par le cas « Consulter ma fiche de pointage ». On peut distinguer deux entités : employé et Shift (qui représente une période de temps délimitée par un temps d'entrée et un temps de sortie et la date de ces deux événements).

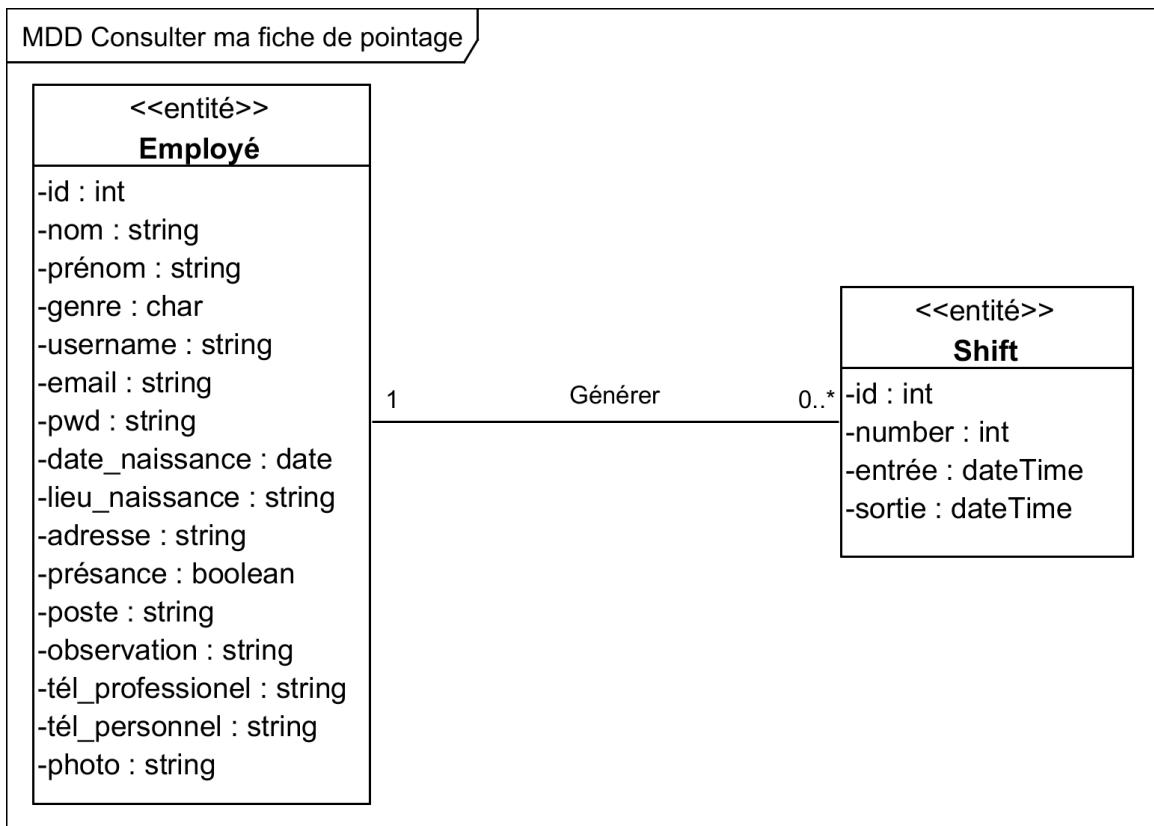


FIGURE 3.2 – Modèle du domaine « Consulter ma fiche de pointage »

Modèle du domaine du cas d'utilisation « Ajouter planning »

Ce modèle du domaine exprime le fait qu'un planning est composé de sept jours. Dans lesquels nous retrouvons deux parties qui représente une journée de travail d'un employé.

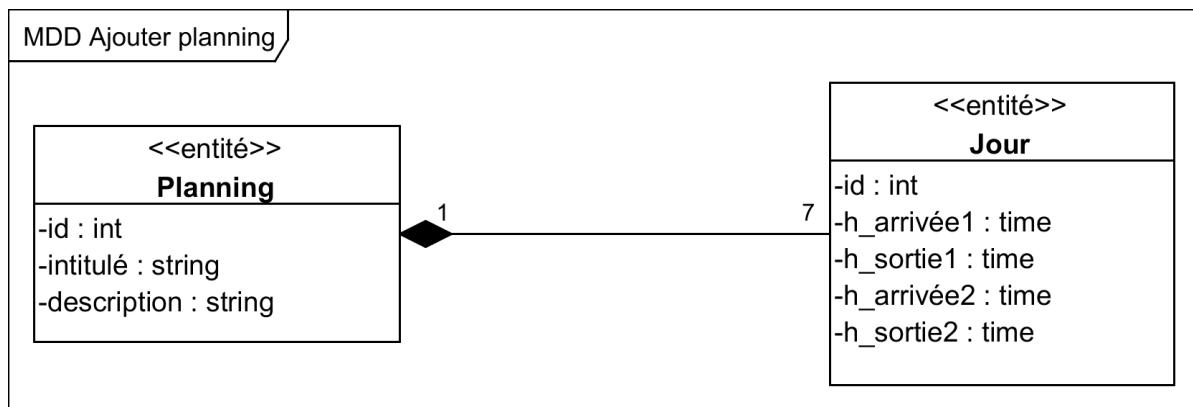


FIGURE 3.3 – Modèle du domaine « Ajouter planning »

Modèle du domaine du cas d'utilisation « Ajouter équipe »

Sur ce modèle du domaine on remarque que les entités sont reliées par deux associations. Une représente le fait d'être membre d'une équipe, et l'autre le fait de gérer (ou manager) une équipe. Nous avons décidé de relier cette association à l'employé et non au manager (qui hérite de l'employé), car nous estimons qu'à ce stade il n'y a pas d'attributs liés exclusivement au manager (il se pourrait que cela se fera dans les prochains diagrammes en raison des méthodes).

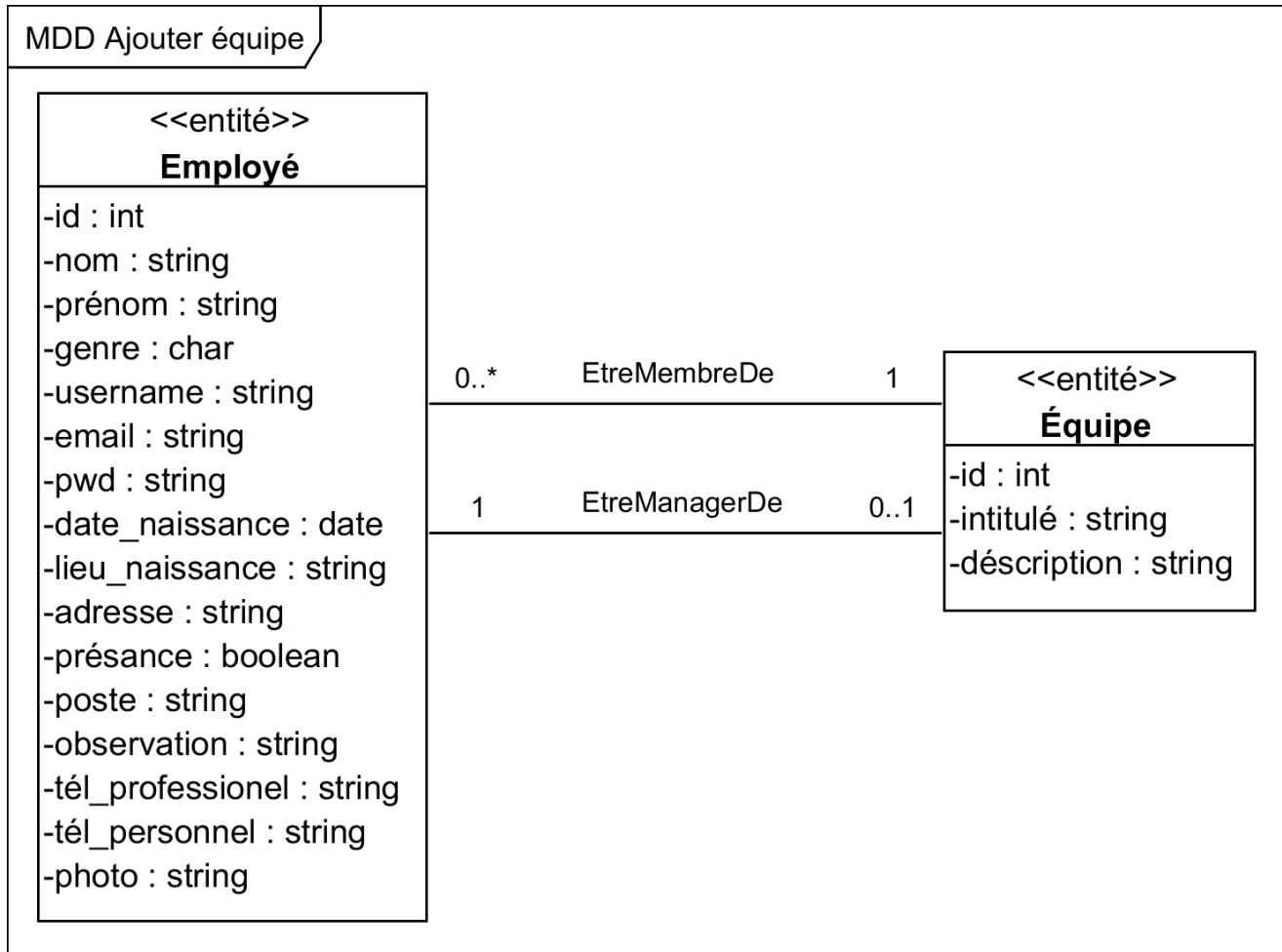


FIGURE 3.4 – Modèle du domaine « Ajouter équipe »

Modèle du domaine du cas d'utilisation « Ajouter employé »

Sur ce modèle du domaine figurent deux entités capitales de notre système qui sont l'employé et son empreinte. Il est important de noter qu'un employé ne peut avoir qu'une seule empreinte dans le système (index droit).

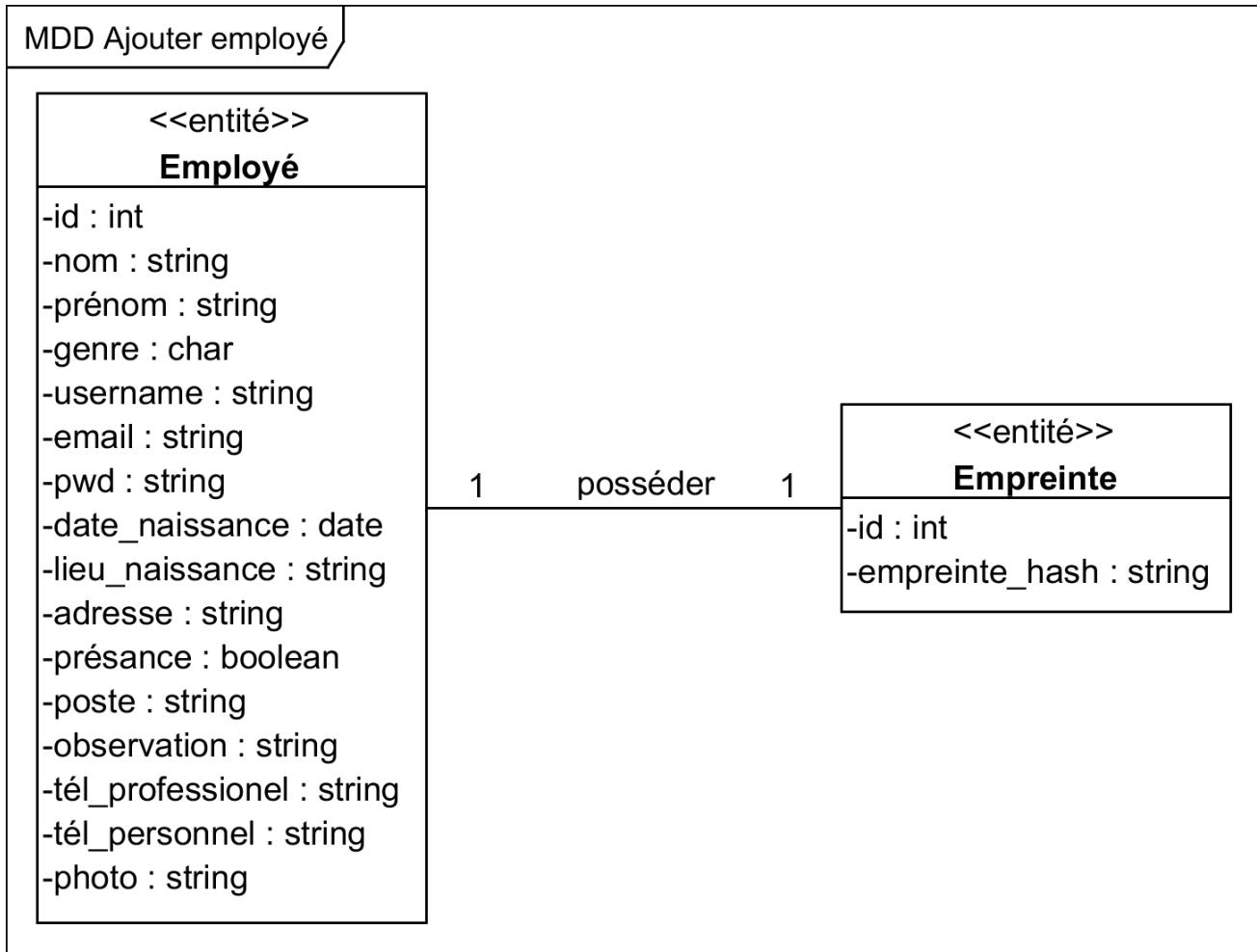


FIGURE 3.5 – Modèle du domaine « Ajouter employé »

Modèle du domaine du cas d'utilisation « Ajouter membre »

En plus des deux entités, précédemment citées dans le modèle du domaine du cas d'utilisation « Ajouter équipe », on retrouve une classe d'association qui nous permet de garder une trace du changement d'équipe pour chaque employé.

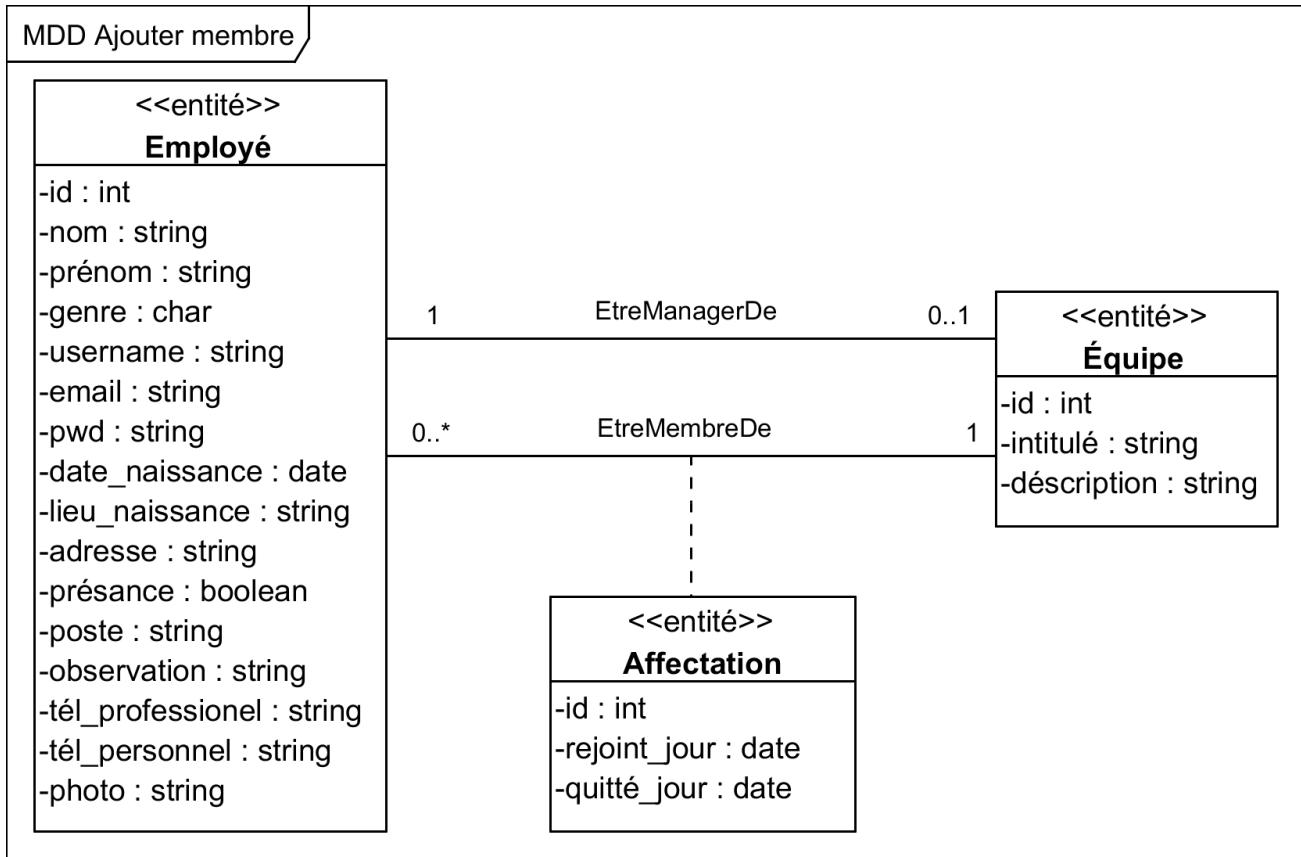


FIGURE 3.6 – Diagramme de séquence système « Ajouter membre »

Modèle du domaine du cas d'utilisation « Consulter profil d'un employé »

Sur ce modèle du domaine, le but est d'offrir aux employés un planning flexible. Pour ce faire, nous avons une composition entre 2 entités : un *planning* qui est la classe mère avec comme attributs un intitulé du planning ainsi qu'une description, et une classe *jour* qui modélise les jours de la semaine avec comme attributs les horaires de travaux.

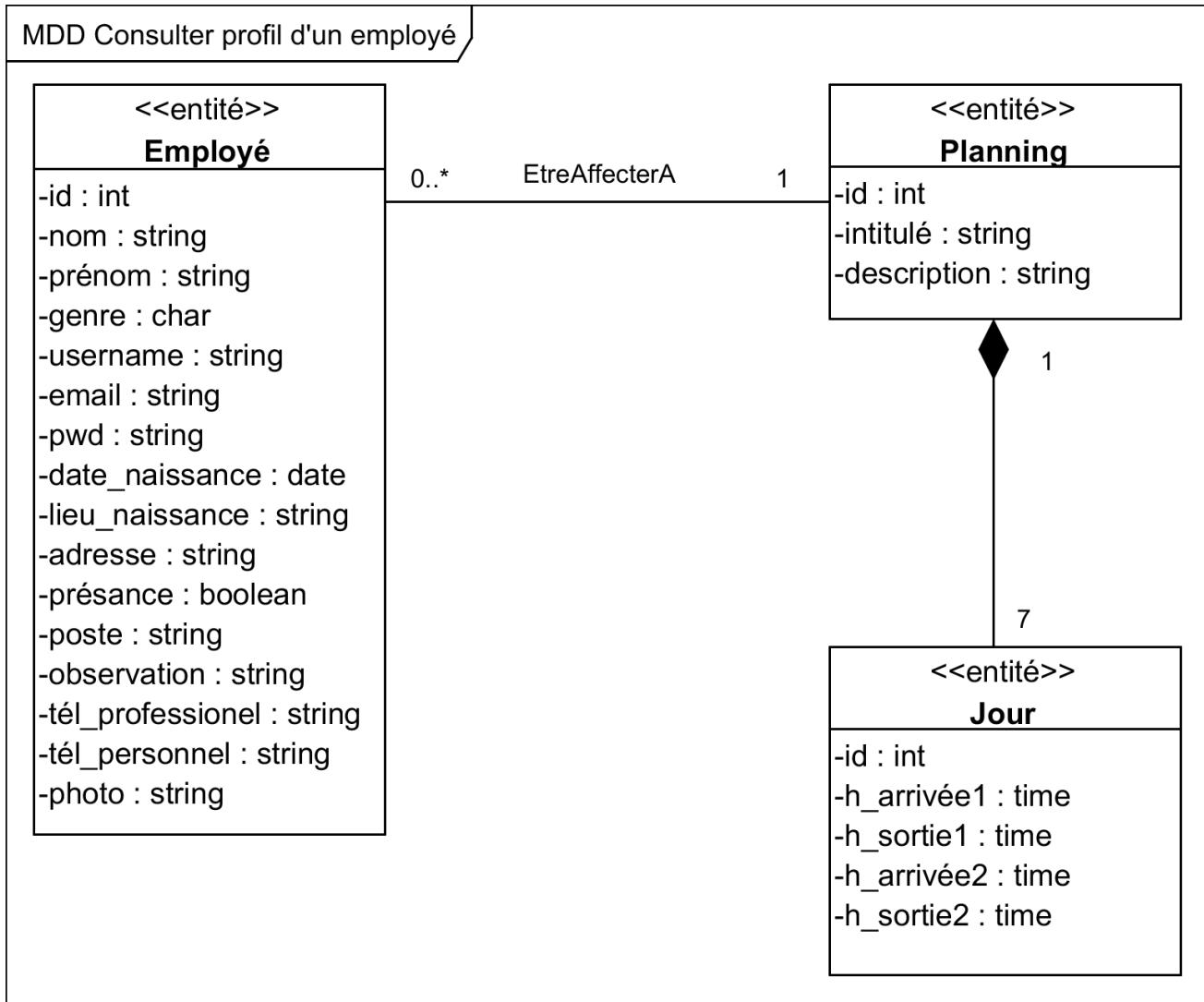


FIGURE 3.7 – Modèle du domaine « Consulter profil d'un employé »

Modèle du domaine du cas d'utilisation « Consulter tableau de bord manager »

Sur ce modèle du domaine, on peut observer que plusieurs entités participent à la réalisation du cas d'utilisation. Ceci est dû à sa nature sommaire. Un tableau de bord se doit d'être récapitulatif et offrir un maximum d'informations pertinentes en un minimum d'espace. Pour avoir la liste des collaborateurs, nous avons besoins des deux entités : *employé* et *équipe*. Afin de savoir si un collaborateur est en retard ou absent, on doit accéder à son planning.

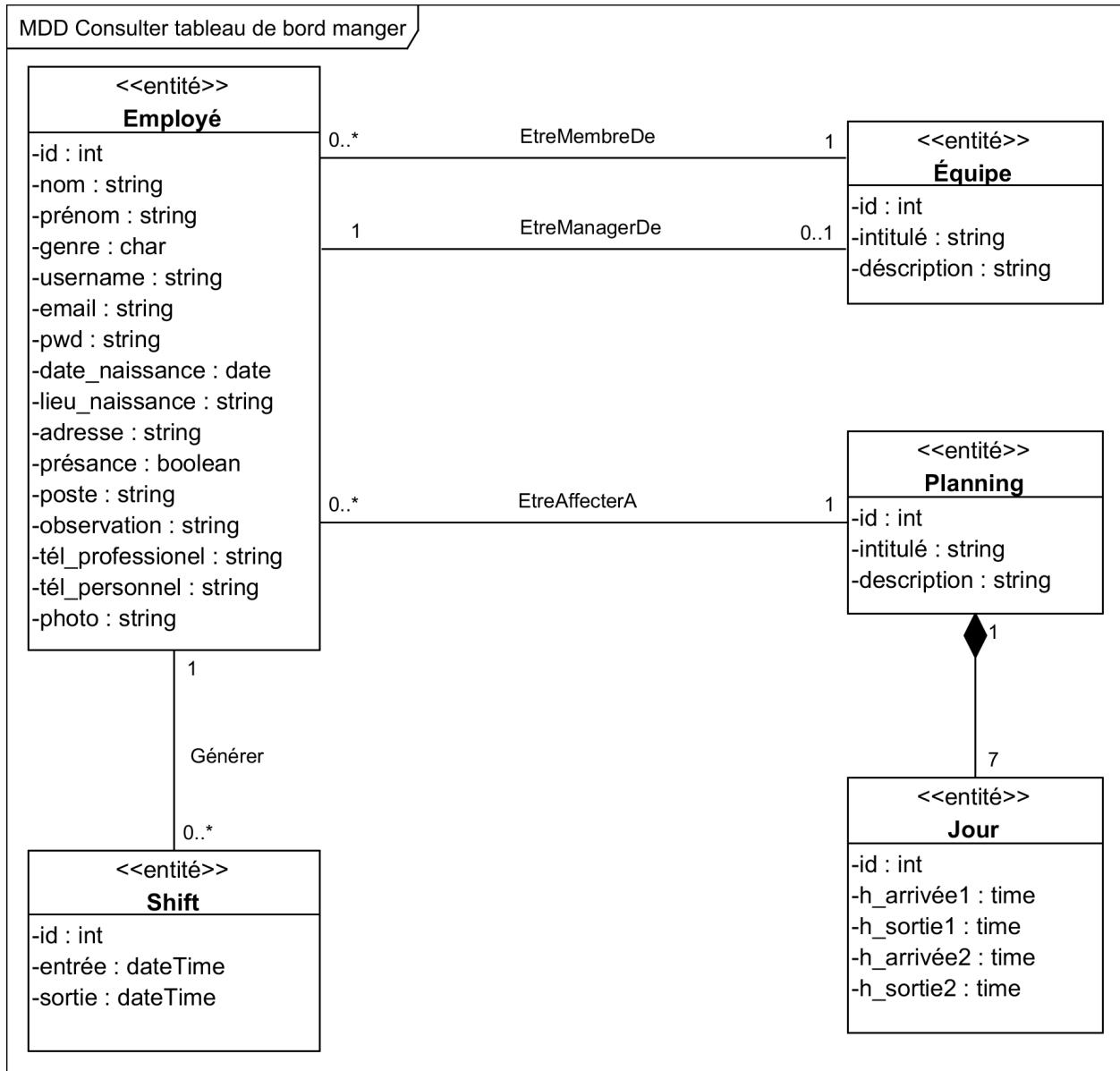


FIGURE 3.8 – Modèle du domaine « Consulter tableau de bord manager »

3.3 Diagrammes de classes participantes

L’importance du diagramme de classes participantes réside dans le fait qu’il effectue la jonction entre les cas d’utilisation, les classes d’analyse et l’interface avec l’utilisateur, de plus, il permet l’enrichissement des diagrammes de séquence système.[8]

3.3.1 Définitions et formalisme

Les classes composant un diagramme de classes participantes se répartissent en trois catégories :[6]

Les classes de dialogue

Ces classes permettent les interactions entre l'IHM et les utilisateurs. Il s'agit typiquement des écrans proposés à l'utilisateur : les formulaires de saisie, les résultats de recherche, etc. En général, les dialogues vivent seulement le temps du déroulement du cas d'utilisation concerné, afin de modéliser une classe de dialogue nous allons utiliser le modèle de la figure3.9.

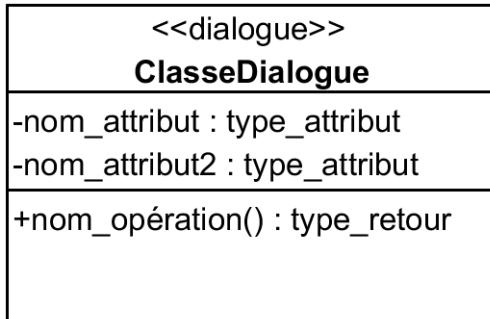


FIGURE 3.9 – Représentation d'une classe dialogue

Les classes de contrôle

Les classes qui modélisent la cinématique de l'application sont appelées contrôles. Elles font la jonction entre les dialogues et les classes métier en permettant aux différentes vues de l'application de manipuler des informations détenues par des objets métier. Elles contiennent les règles applicatives et les isolent à la fois des dialogues et des entités, elles seront représentées comme sur la figure3.10.

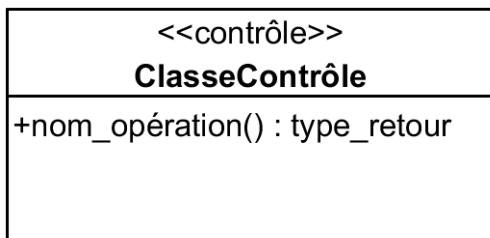


FIGURE 3.10 – Représentation d'une classe contrôle

Les classes entités

Les classes métier, qui proviennent directement du modèle du domaine sont qualifiées d'entités. Ces classes sont généralement persistantes, c'est à dire qu'elles survivent à l'exécution d'un cas d'utilisation particulier et qu'elles permettent à des données et à des relations d'être stockées dans des fichiers ou des bases de données. Elles seront représentées comme sur la figure3.11.

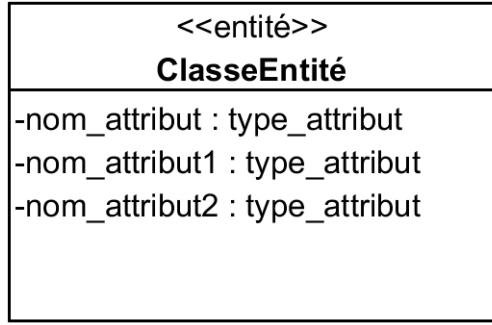


FIGURE 3.11 – Représentation d'une classe entité

Afin d'établir un DCP, nous ajouterons des associations entre les classes tout en respectant les règles ci-après : [5]

- Les dialogues ne peuvent être reliés qu'aux contrôles ou à d'autres dialogues, mais pas directement aux entités.
- Les entités ne peuvent être reliées qu'aux contrôles ou à d'autres entités.
- Les contrôles ont accès à tous les types de classes, y compris d'autres contrôles.
- Enfin, un acteur ne peut être lié qu'à un dialogue.

Un exemple de diagramme de classe participante est représenté dans la figure ci-dessous :

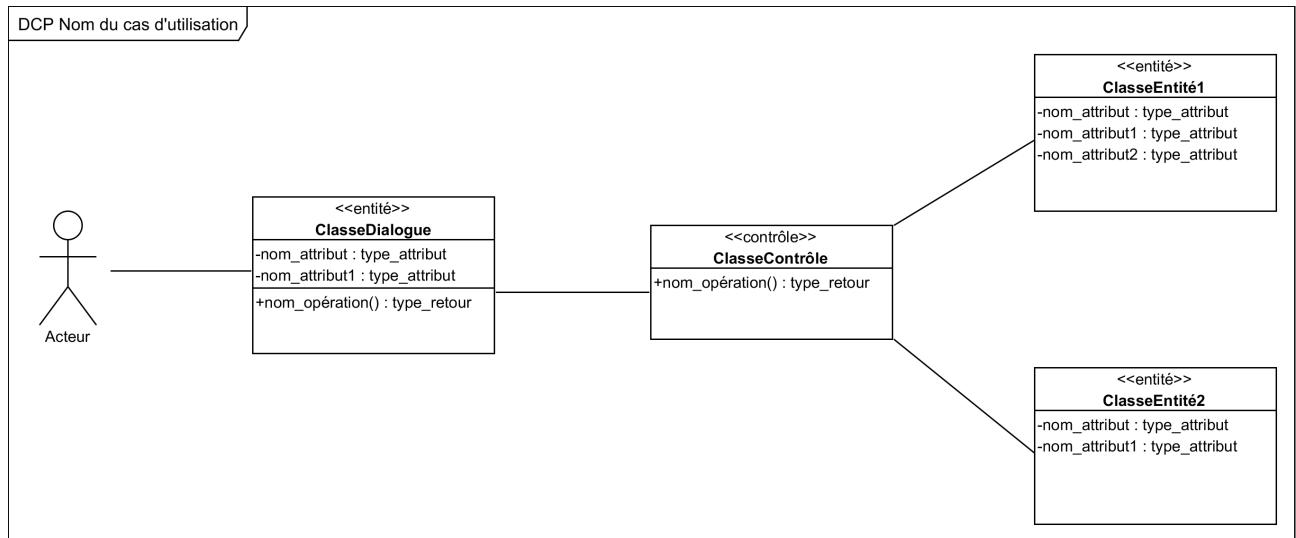


FIGURE 3.12 – Modèle d'un diagramme de classe de conception

Diagrammes de classes participantes du cas d'utilisation « Se pointer »

La pointeuse biométrique permet à l'employé d'enregistrer ses horaires de travaux le plus facilement possible. Pour ce qui est de la logique concernant les actions effectuées par l'employé, elle est déléguée à trois contrôleurs : CtrShift et CtrEmployé et CtrEmpreinte.

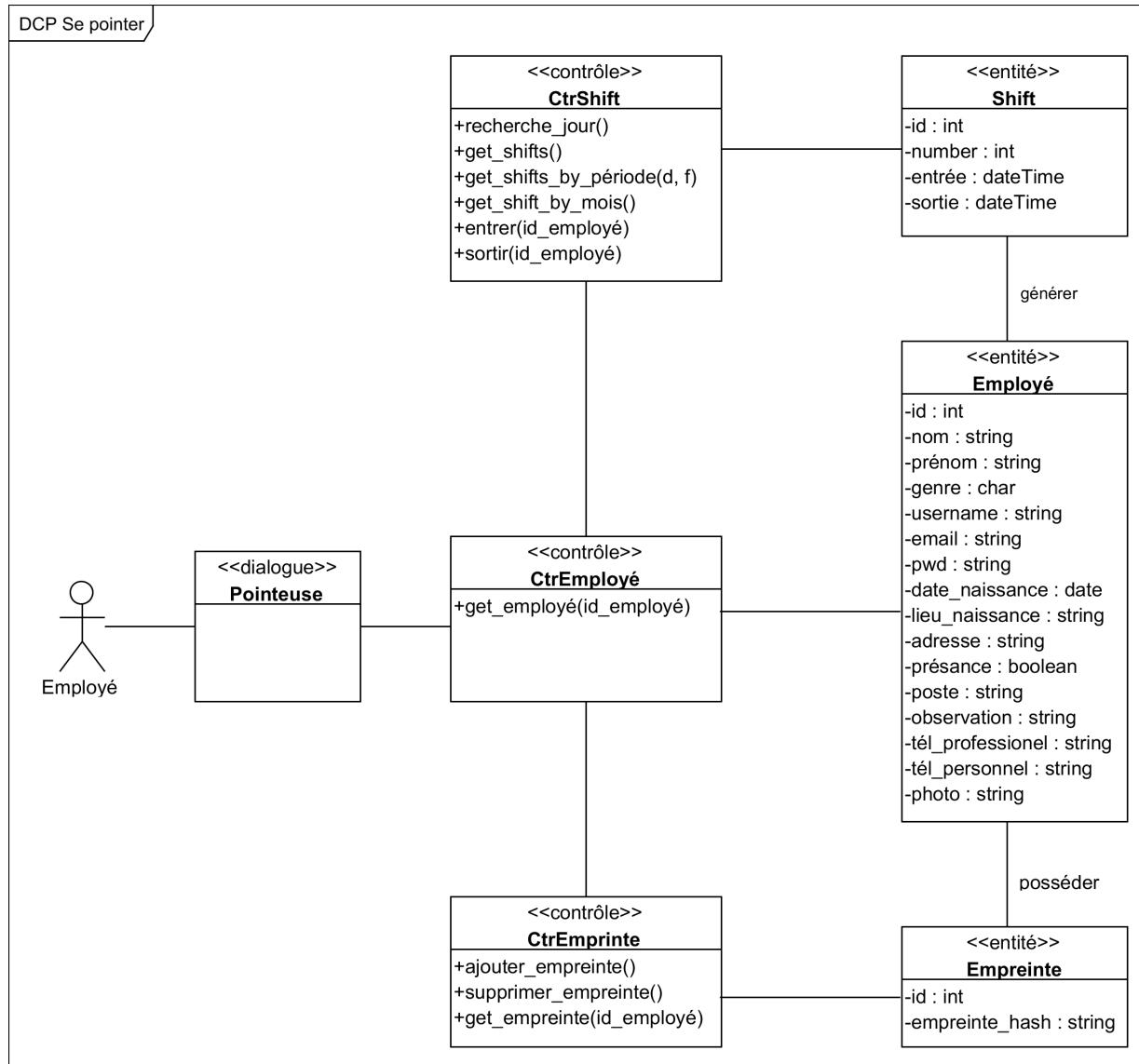


FIGURE 3.13 – Diagrammes de classes participantes « Consulter ma fiche de pointage »

Diagrammes de classes participantes du cas d'utilisation « Consulter mon profil »

Un employé peut consulter son profil sur l'interface « InterfaceMonProfil » qui regroupe ses informations, il peut aussi accéder à l'interface de modification de son profil, ainsi qu'afficher son planning. Les actions effectuées sont gérées par le contrôleur CtrEmployé.

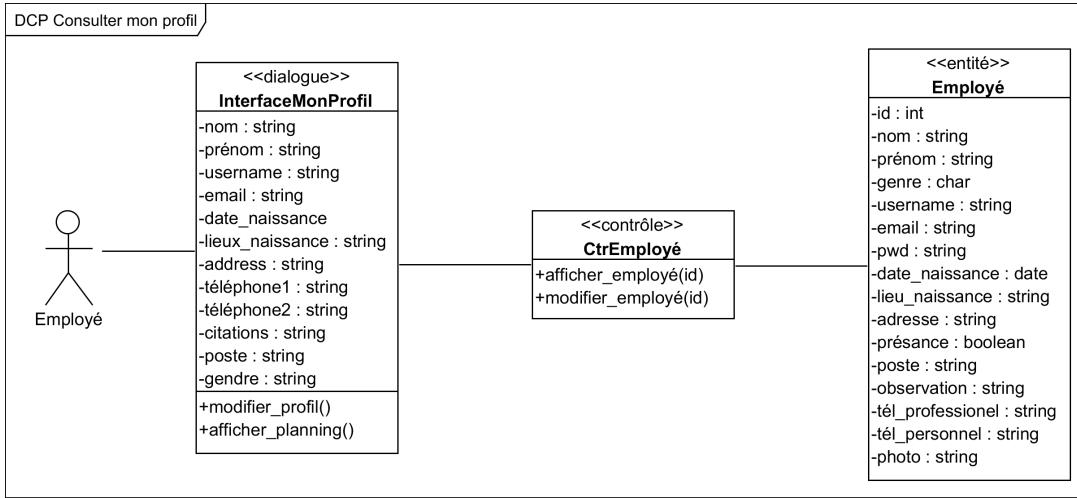


FIGURE 3.14 – Diagrammes de classes participantes « Consulter mon profil »

Diagrammes de classes participantes du cas d'utilisation « Consulter ma fiche de pointage »

Un employé peut consulter sa fiche de pointage sur l'interface « InterfaceConsulterMaFichePointage ». Par défaut, l'affichage de la liste des pointages est par semaine, s'il le souhaite il peut l'afficher par mois ou faire une recherche précise par période. La logique concernant les actions effectuées par l'employé est déléguée à deux contrôleurs : CtrShift et CtrEmployé.

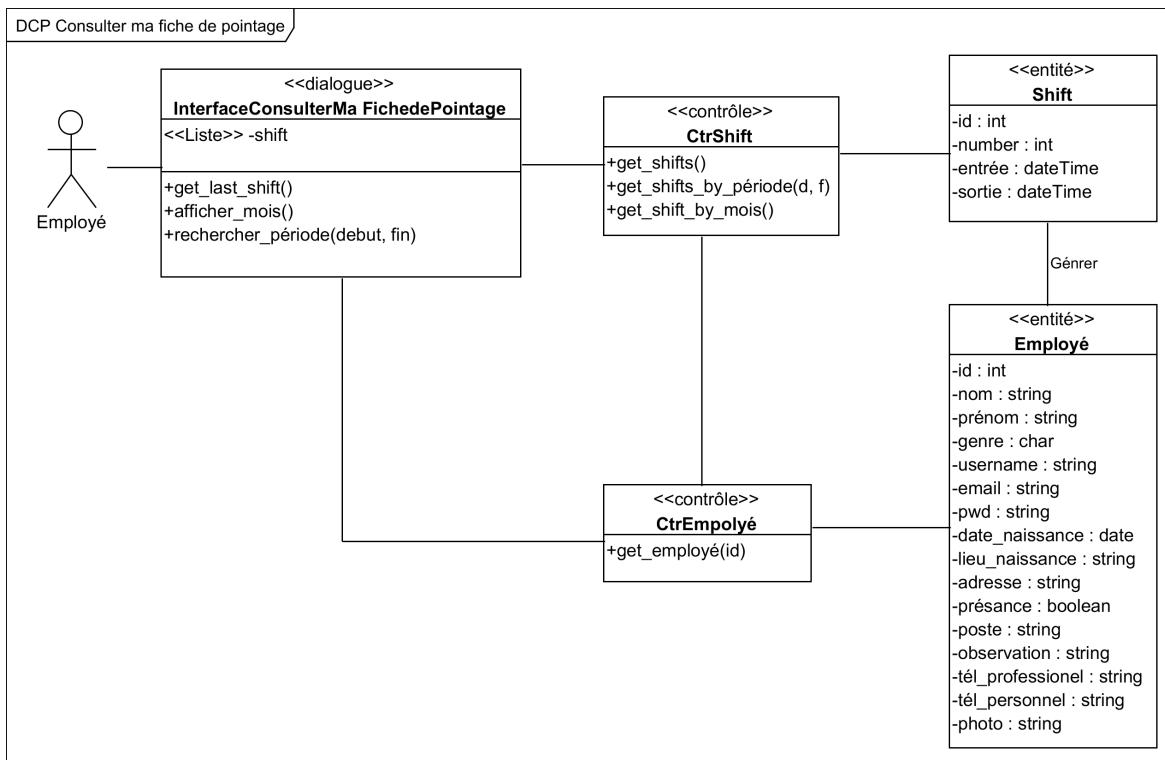


FIGURE 3.15 – Diagrammes de classes participantes « Consulter ma fiche de pointage »

Diagrammes de classes participantes du cas d'utilisation « Consulter tableau de bord manager »

Une fois authentifié, le manager est redirigé vers « InterfaceTableauBordManager » où il aura accès à son planning du jour, une liste de ces collaborateurs, ainsi qu'un résumé sur l'ensemble de ces équipes. Les 2 contrôles CtrEmployés et CtrEquipe traitent les actions du manager.

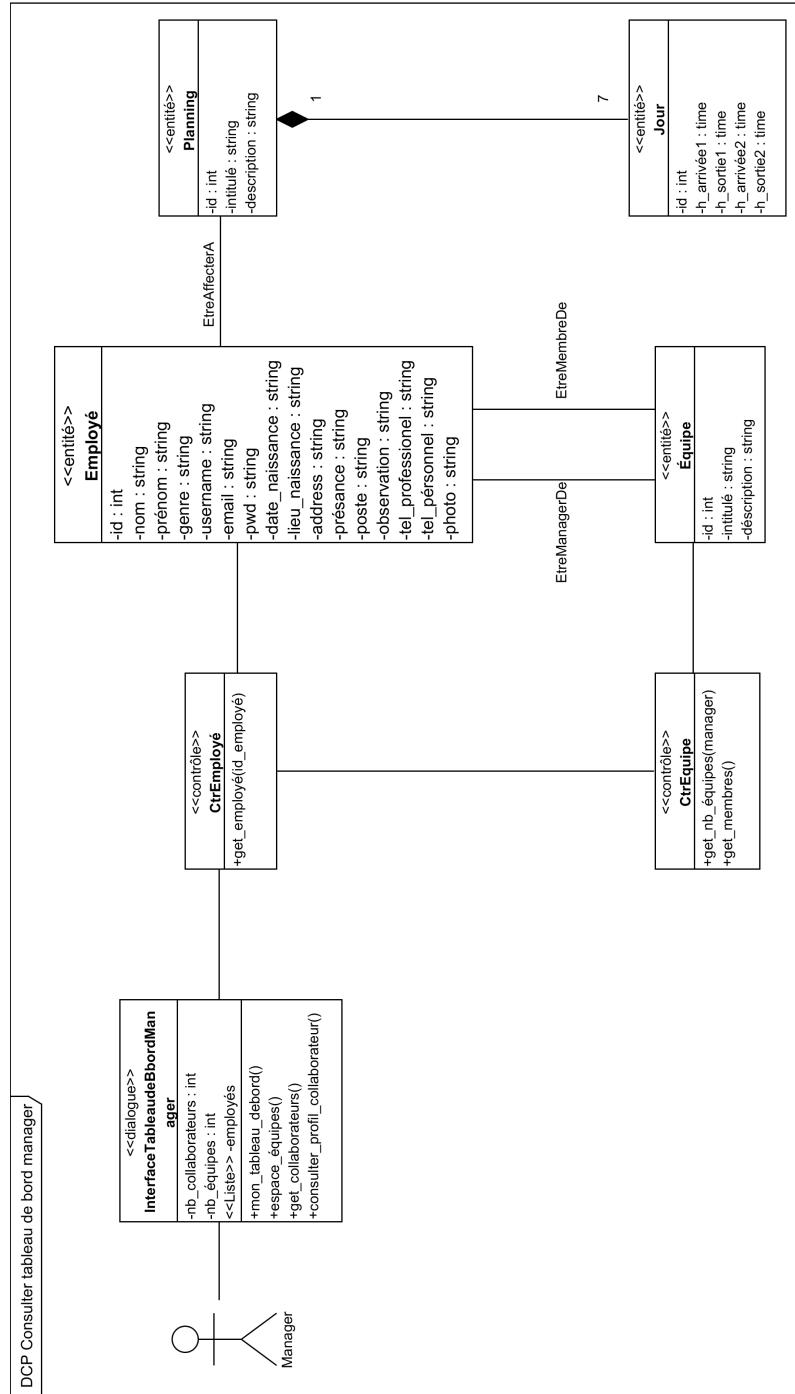


FIGURE 3.16 – Diagrammes de classes participantes « Consulter tableau de bord manager »

Diagrammes de classes participantes du cas d'utilisation « Ajouter planning »

L'interface « InterfaceAjouterPlanning » affiche un formulaire qui permet à l'acteur de créer un nouveau planning. Il doit saisir le nom, la description, ainsi que les différents horaires de travail. Une fois validé par l'acteur, CtrPlanning vérifie les informations saisies pour enfin pouvoir ajouter le planning à son entité correspondante.

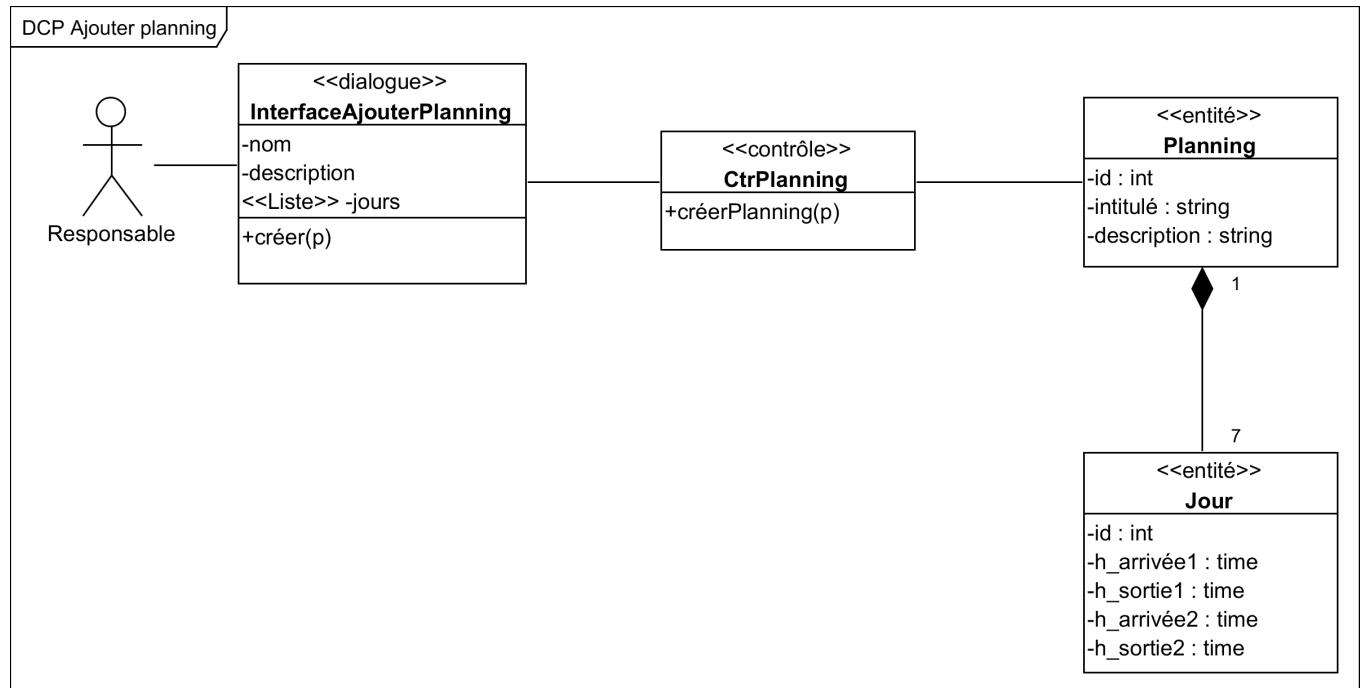


FIGURE 3.17 – Diagrammes de classes participantes « Ajouter planning »

Diagrammes de classes participantes du cas d'utilisation « Ajouter équipe »

L'interface « InterfaceAjouterEquipe » affiche un formulaire qui permet à l'acteur de créer une nouvelle équipe. Il doit saisir le nom, la description ainsi que chercher et sélectionner un manager qui sera le responsable de cette dernière.

La logique concernant les actions effectuées par l'acteur est déléguée à deux contrôleurs, CtrEmployé qui va gérer la recherche du manager et CtrEquipe qui vérifiera les informations saisies pour pouvoir ajouter l'équipe à son entité correspondante.

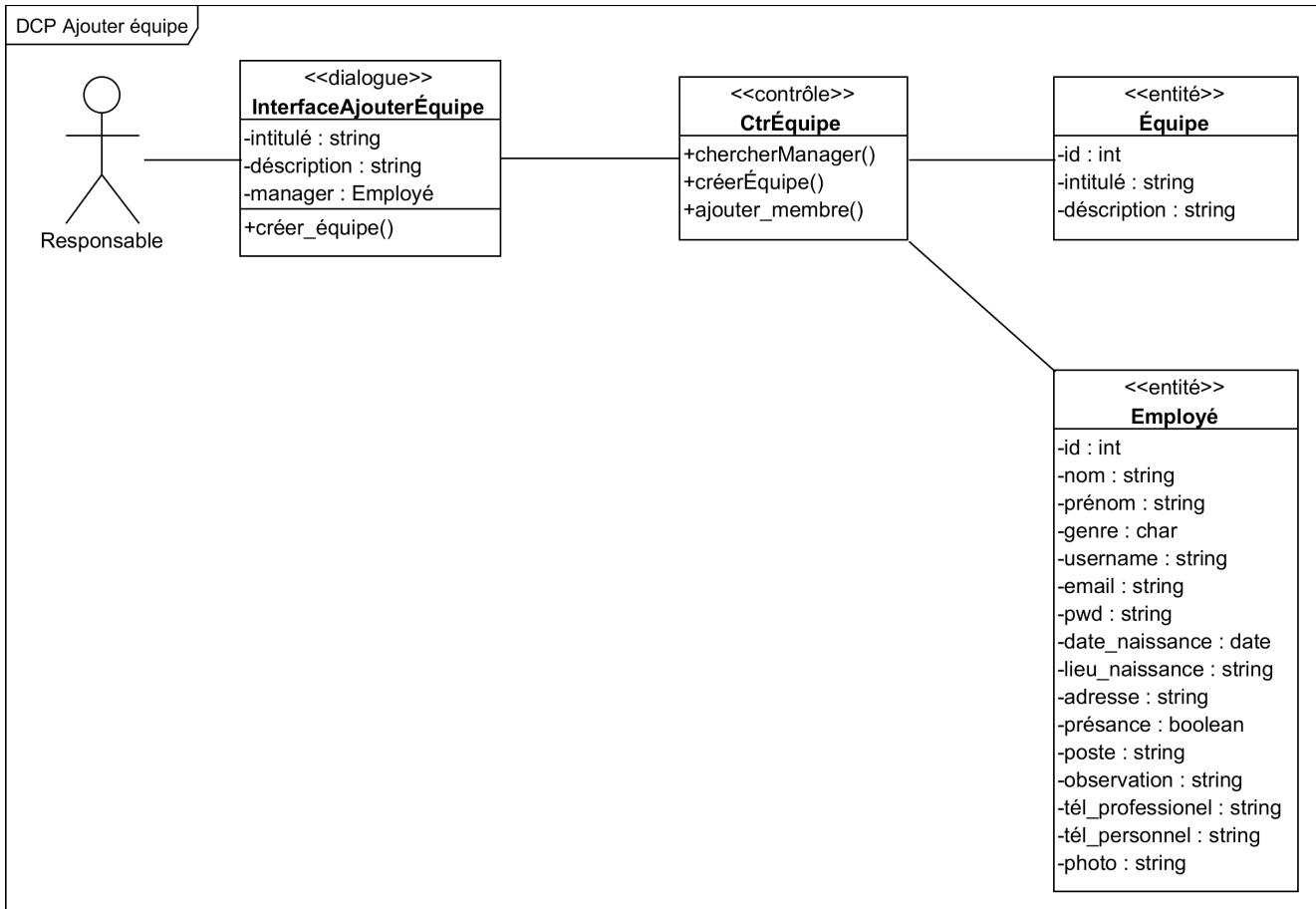


FIGURE 3.18 – Diagrammes de classes participantes « Ajouter équipe »

Diagrammes de classes participantes du cas d'utilisation « Ajouter membre »

L’interface « InterfaceAjouterMembre » affiche à l’acteur une liste des membres de l’équipe, ainsi qu’un champ qui lui permettra de faire une recherche d’un employé afin de l’ajouter à l’équipe.

Pour ce qui est des classes de contrôles, la première classe CtrEquipe aura la responsabilité de gérer la recherche, et l’ajout des membres de l’équipe. Quant à la deuxième classe CtrAffectation, elle permettra de créer une affectation pour chaque employé ajouté.

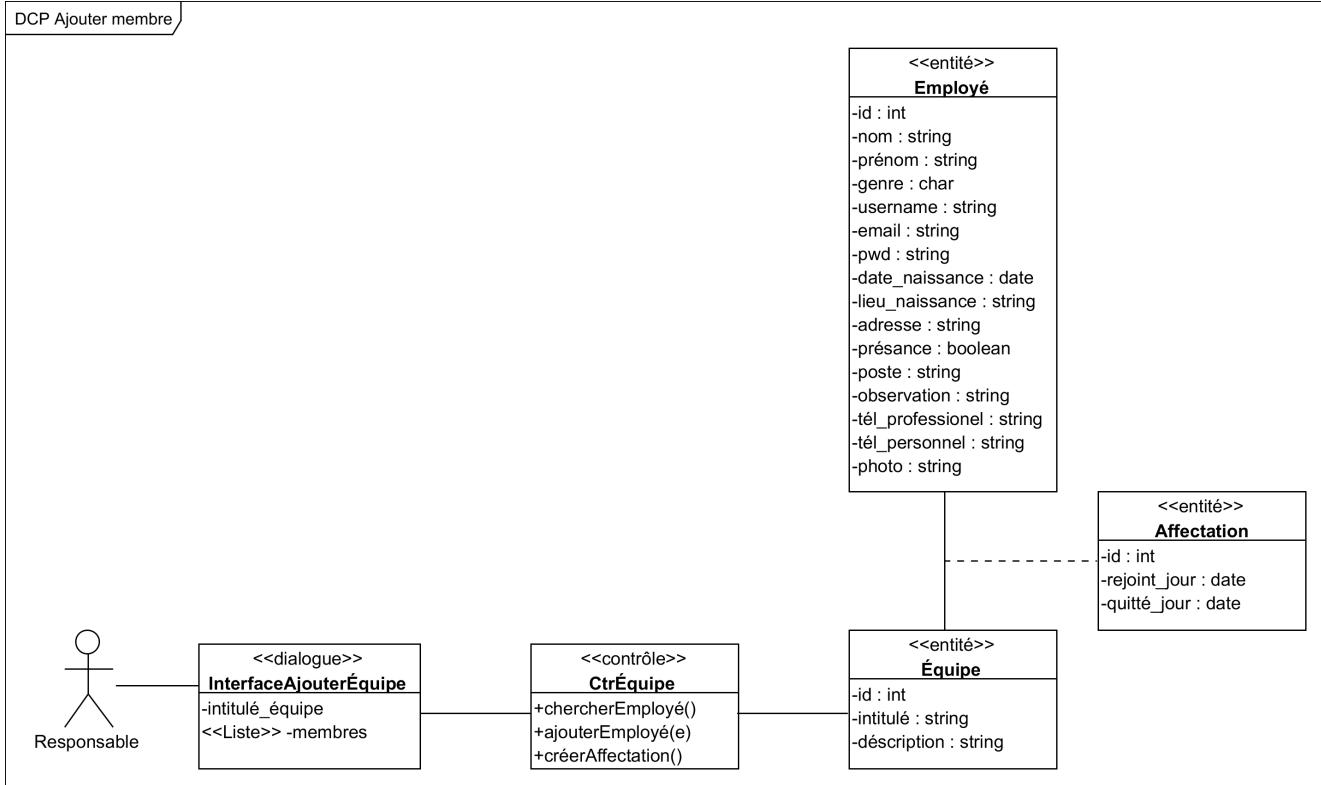


FIGURE 3.19 – Diagrammes de classes participantes « Ajouter membre »

Diagrammes de classes participantes du cas d'utilisation « Ajouter employé »

L’administrateur à travers l’interface « InterfaceAjouterEmployé » peut ajouter un employé en remplissant le formulaire. Après avoir validé, l’administrateur devra ajouter l’empreinte de l’employé qui sera gérée par la classe CtrEmpreinte. Une fois enregistrée, la classe CtrEmployé récupèrera l’empreinte pour finaliser l’enregistrement de l’employé.

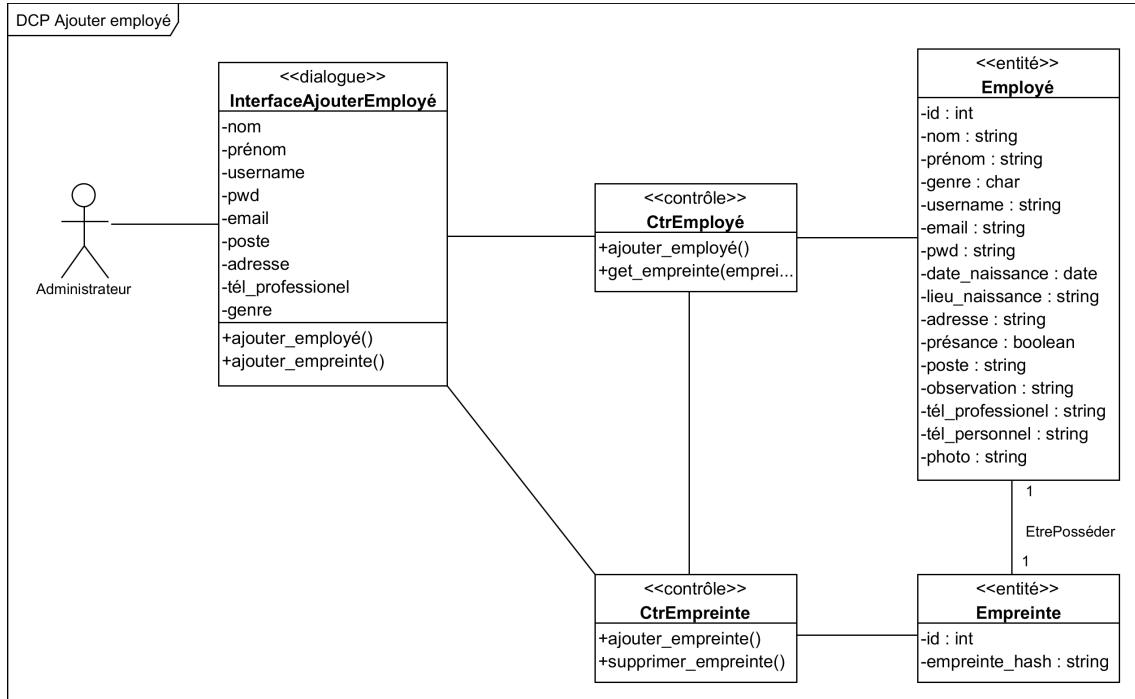


FIGURE 3.20 – Diagrammes de classes participantes « Ajouter employé »

Diagrammes de classes participantes du cas d'utilisation « Consulter profil d'un employé »

L’interface « InterfaceConsulterProfilEmployé » permet au responsable de consulter le profil d’un employé où il aura accès à toutes ses informations. Il pourra aussi à partir de cette interface accéder à l’interface de modification du profil et du planning de l’employé. Quant à la classe CtrEmployé, elle permet de récupérer l’employé sélectionné.

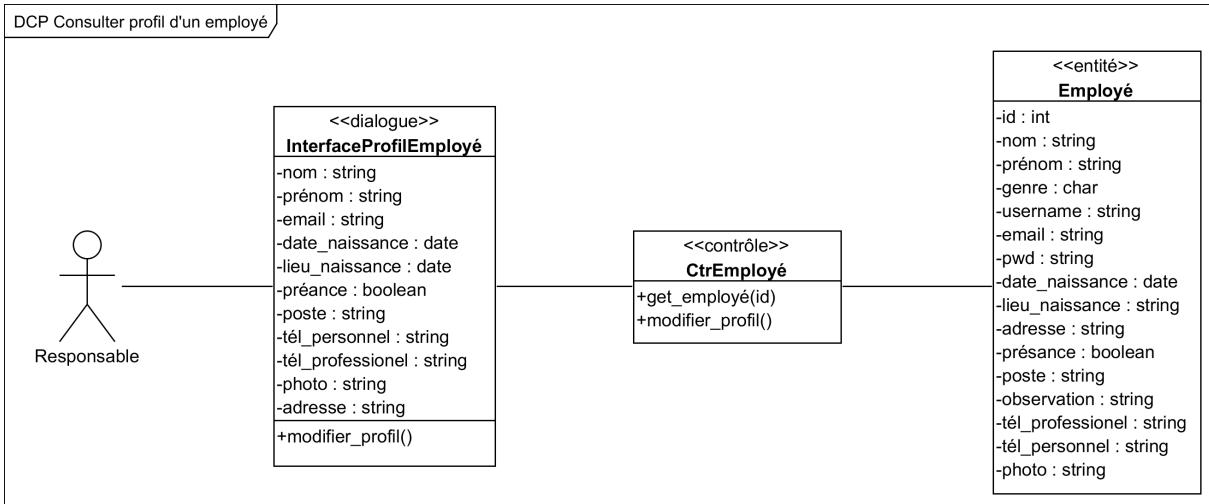


FIGURE 3.21 – Diagrammes de classes participantes « Consulter profil employé »

3.4 Diagrammes de séquence

L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés.[9]

Dans cette partie, nous allons détailler les diagrammes de séquence système élaborés dans le chapitre 2, en remplaçant le système vu comme une boîte noire par un ensemble d'objets de classes différentes tout en respectant les règles suivantes :

- Les acteurs ne peuvent interagir (envoyer des messages) qu'avec les dialogues.
- Les dialogues peuvent interagir avec les contrôles.
- Les contrôles peuvent interagir avec les dialogues, les entités, ou d'autres contrôles.
- Les entités ne peuvent interagir qu'entre elles.

Le formalisme est donné dans l'exemple type présenté à la figure 3.22

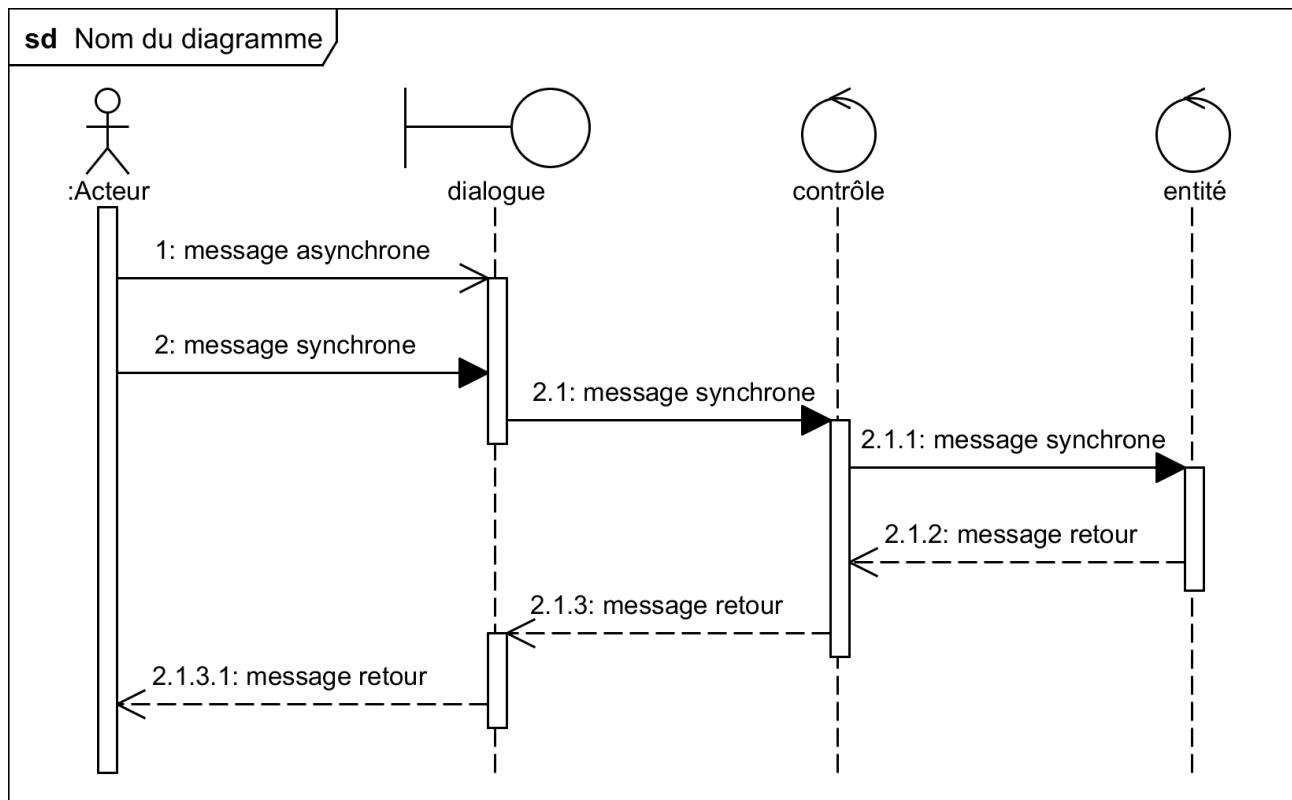


FIGURE 3.22 – Formalisme du diagramme de séquence

Diagramme d'interaction du cas d'utilisation « Se pointer »

Avant chaque entrée ou sortie de l'entreprise, l'employé doit se pointer avec son empreinte digitale afin que la pointeuse l'identifie. Après une courte vérification, si l'empreinte n'est pas reconnue aucune LED ne s'allume, par contre si la pointeuse la reconnaît, elle signale au contrôle CtrEmployé qui récupère son identifiant, puis délègue au contrôle CtrShift la recherche de son dernier pointage et la vérification du type de pointage. Une fois effectué, le contrôle CtrEmployé déclenche l'enregistrement du pointage, si le type de pointage est une sortie alors le contrôle CtrShift insérera l'heure de sortie dans le dernier pointage, si c'est une entrée alors un nouveau pointage est créé avec l'heure d'entrée et l'identifiant de l'employé.

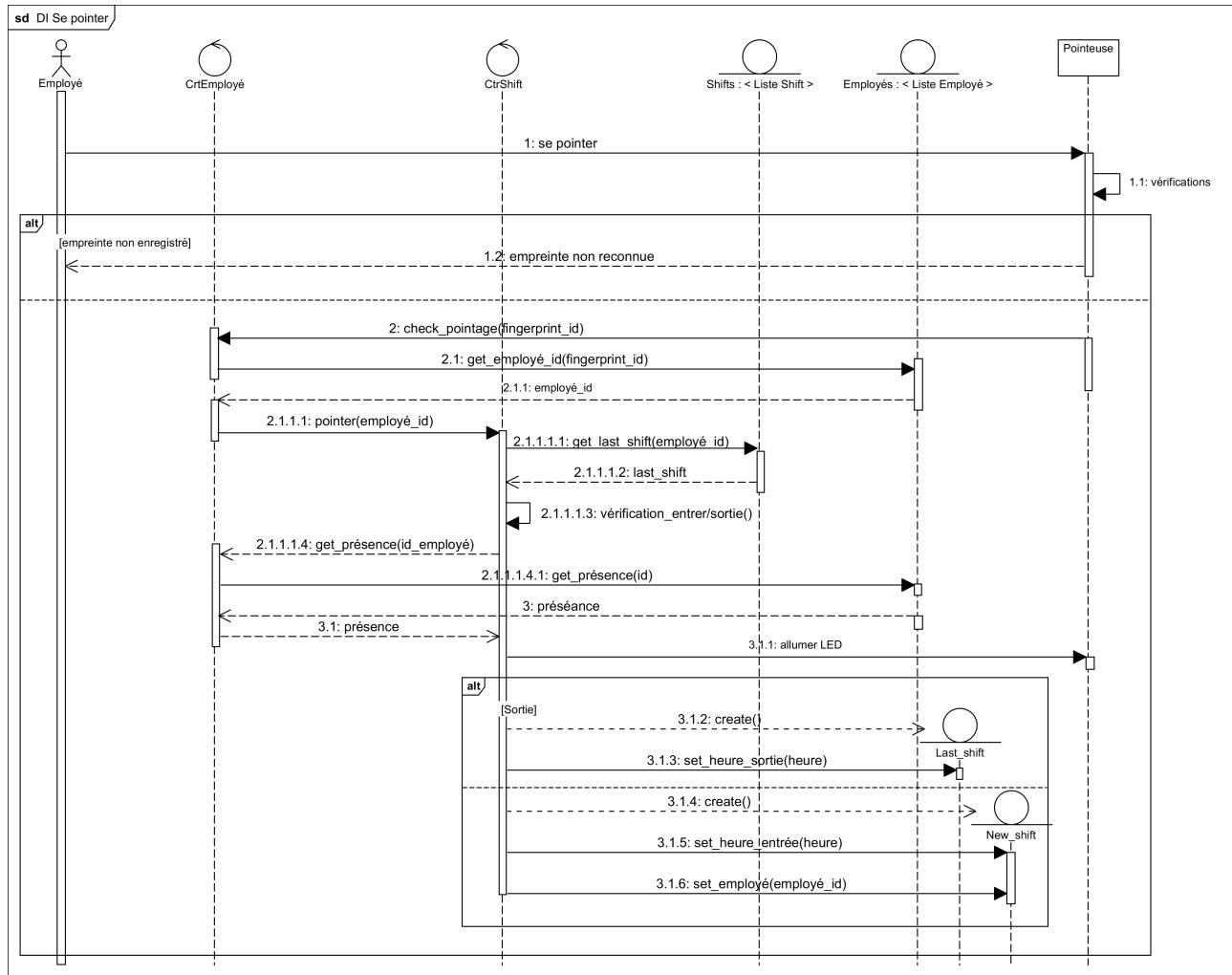


FIGURE 3.23 – Diagramme d'interaction « Se pointer »

Diagramme d'interaction du cas d'utilisation « Consulter mon profil »

Une fois authentifié, l'employé peut consulter son profil à travers n'importe quelle interface, il est redirigé vers l'interface MonProfil qui délègue la recherche avec l'identifiant au contrôle CtrEmployé. Une fois la recherche effectuée, les informations sont affichées. Il pourra aussi à travers cette interface modifier son profil. Une fois les informations saisies, il doit valider ce qui déclenchera une vérification, si les informations saisies sont correctes les modifications seront enregistrées. Dans le cas contraire, un message d'erreur sera affiché.

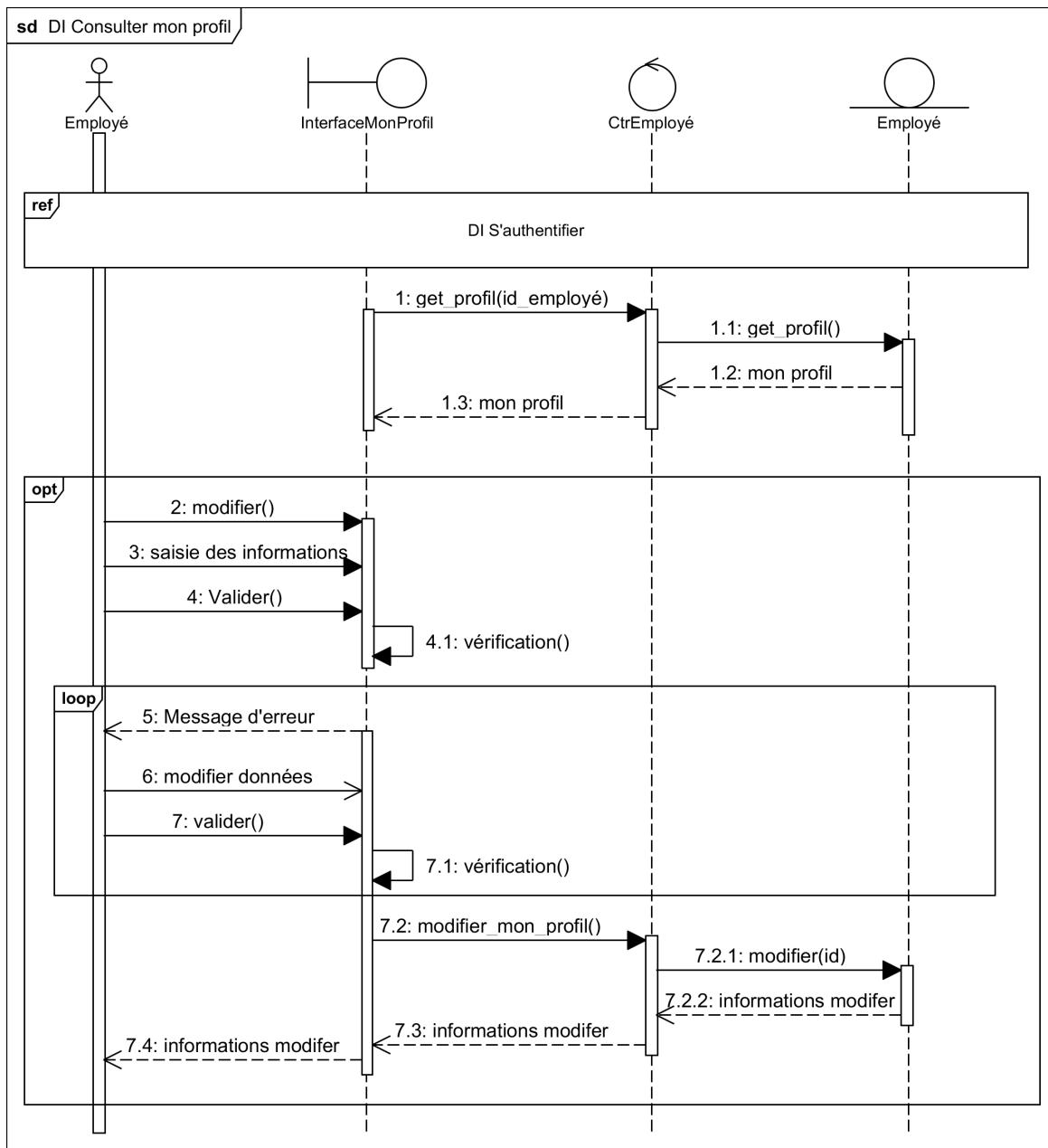


FIGURE 3.24 – Diagramme d'interaction « Consulter mon profil »

Diagramme d'interaction du cas d'utilisation « Consulter ma fiche de pointage »

L'employé devra s'authentifier pour consulter sa fiche de pointage. L'interface InterfaceMaFicheDePointage déléguera la recherche de la liste des derniers pointages au contrôle CtrShift qui seront affichés par semaine. L'employé aura aussi la possibilité de les afficher par mois, ainsi que de faire une recherche par période.

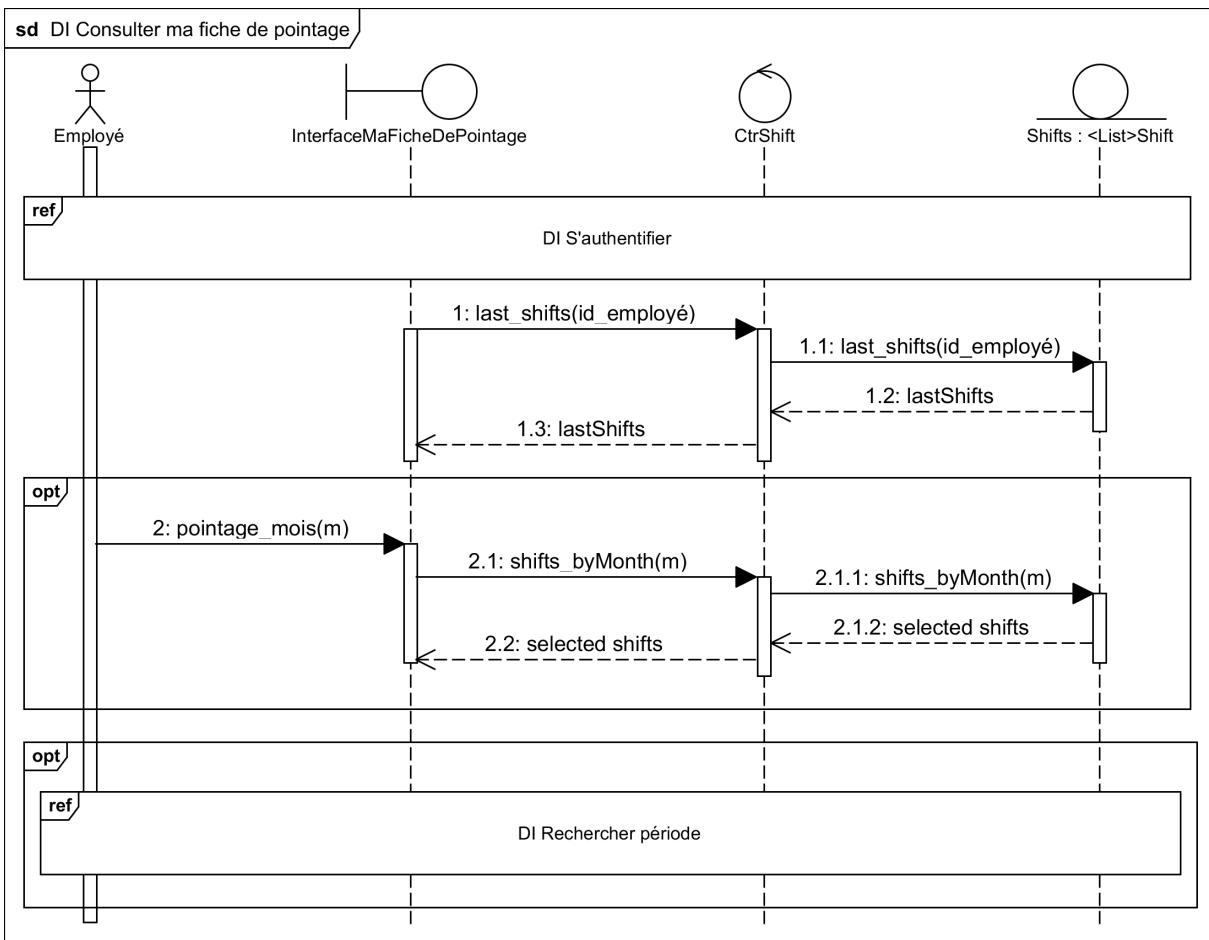


FIGURE 3.25 – Diagramme d’interaction « Consulter ma fiche de pointage »

Diagramme d'interaction du cas d'utilisation « Consulter tableau de bord manager »

Une fois authentifié, le manager est redirigé vers son tableau de bord. En premier lieu son espace personnel sera affiché, il pourra sélectionner l'espace manager, 3 contrôleurs seront nécessaires, le contrôle CtrEmployé pour récupérer la liste de ses collaborateurs, le contrôle CtrEquipe pour récupérer la liste des équipes du manager, et le dernier contrôle CtrShift pour récupérer la liste des pointages de ses collaborateurs.

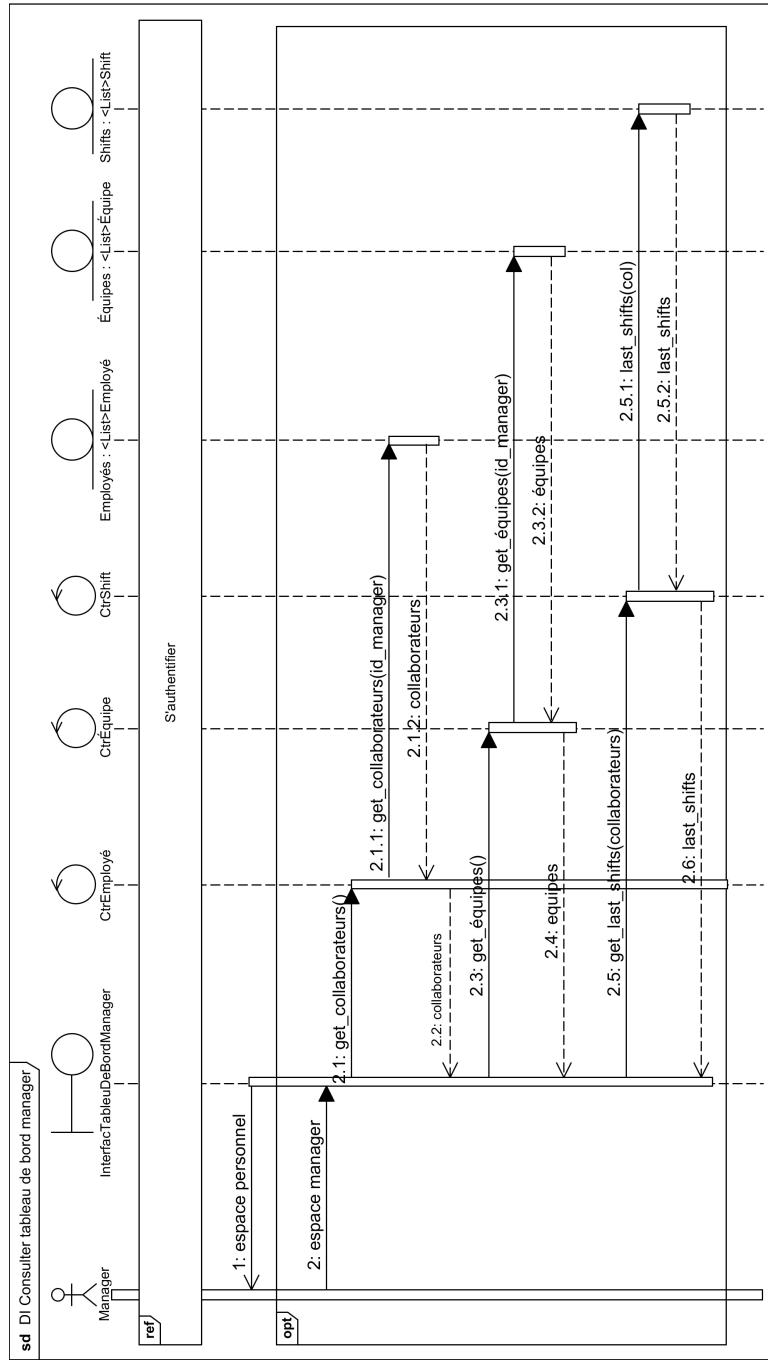


FIGURE 3.26 – Diagramme d’interaction « Consulter tableau de bord manager »

Diagramme d’interaction du cas d’utilisation « Ajouter équipe »

Le responsable pourra ajouter une équipe. Après avoir saisi les informations et sélectionné un manager, une vérification est effectuée. Si aucune erreur n'est détectée il pourra l'enregistrer, le contrôle CtrEquipe s'occupera de la création et il déléguera la recherche du manager au contrôle CtrManager, une fois terminée, il sera redirigé vers l'interface InterfaceGérerEquipe, où il pourra affecter des membres.

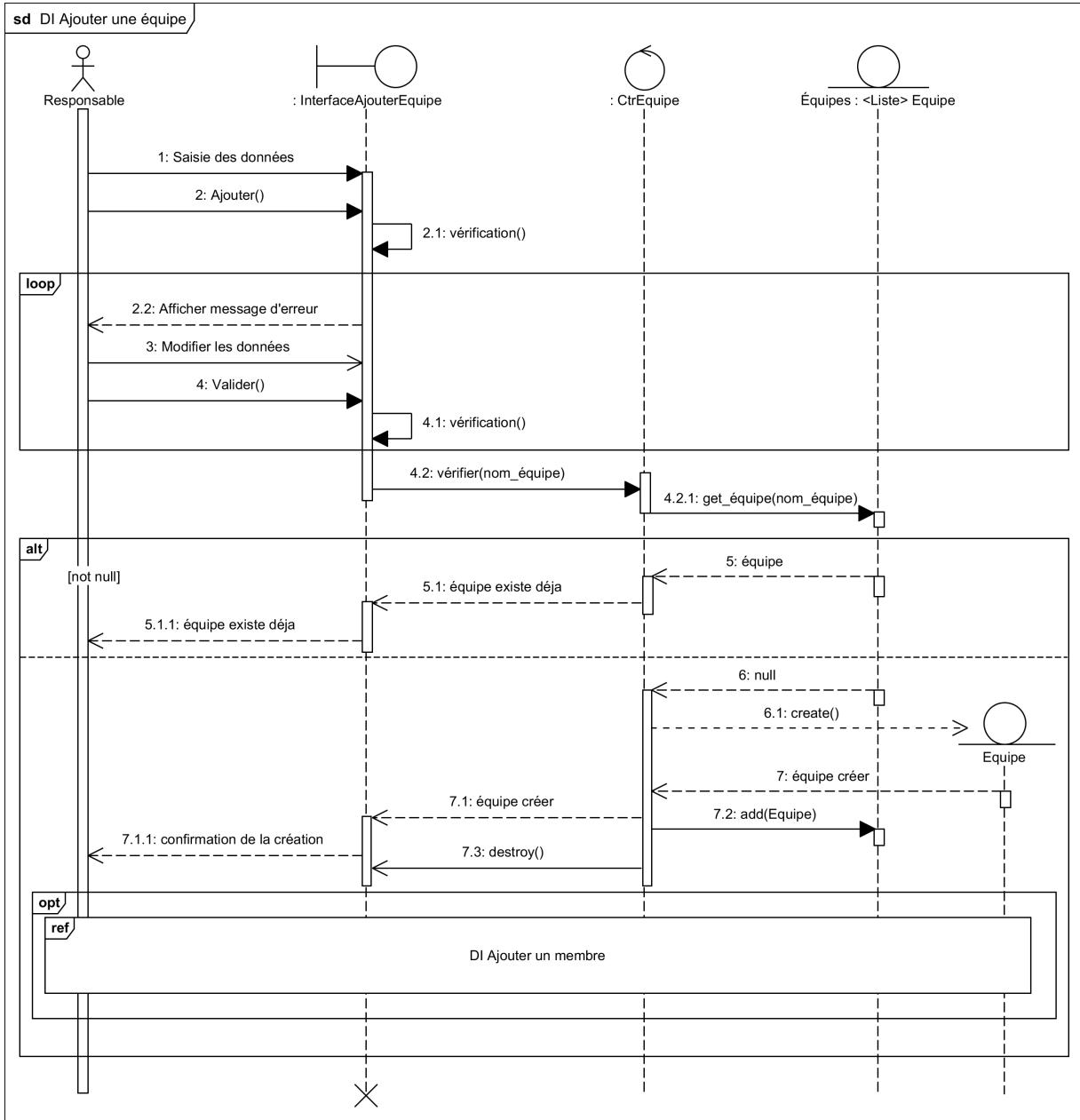


FIGURE 3.27 – Diagramme d’interaction « Ajouter équipe »

Diagramme d’interaction du cas d’utilisation « Ajouter planning »

Le responsable aura la possibilité d’ajouter un planning. Pour ce, il doit saisir les horaires de travail de la semaine ainsi que l’intitulé et la description du planning. Après la validation, une vérification des informations sera effectuée. Si les données sont invalides un message d’erreur sera affiché et il devra les corriger. Si les informations sont valides, le contrôle CtrPlanning vérifiera si aucun planning n’existe avec l’intitulé saisi afin de valider l’ajout du planning. Une fois terminé, il pourra affecter un planning aux employés.

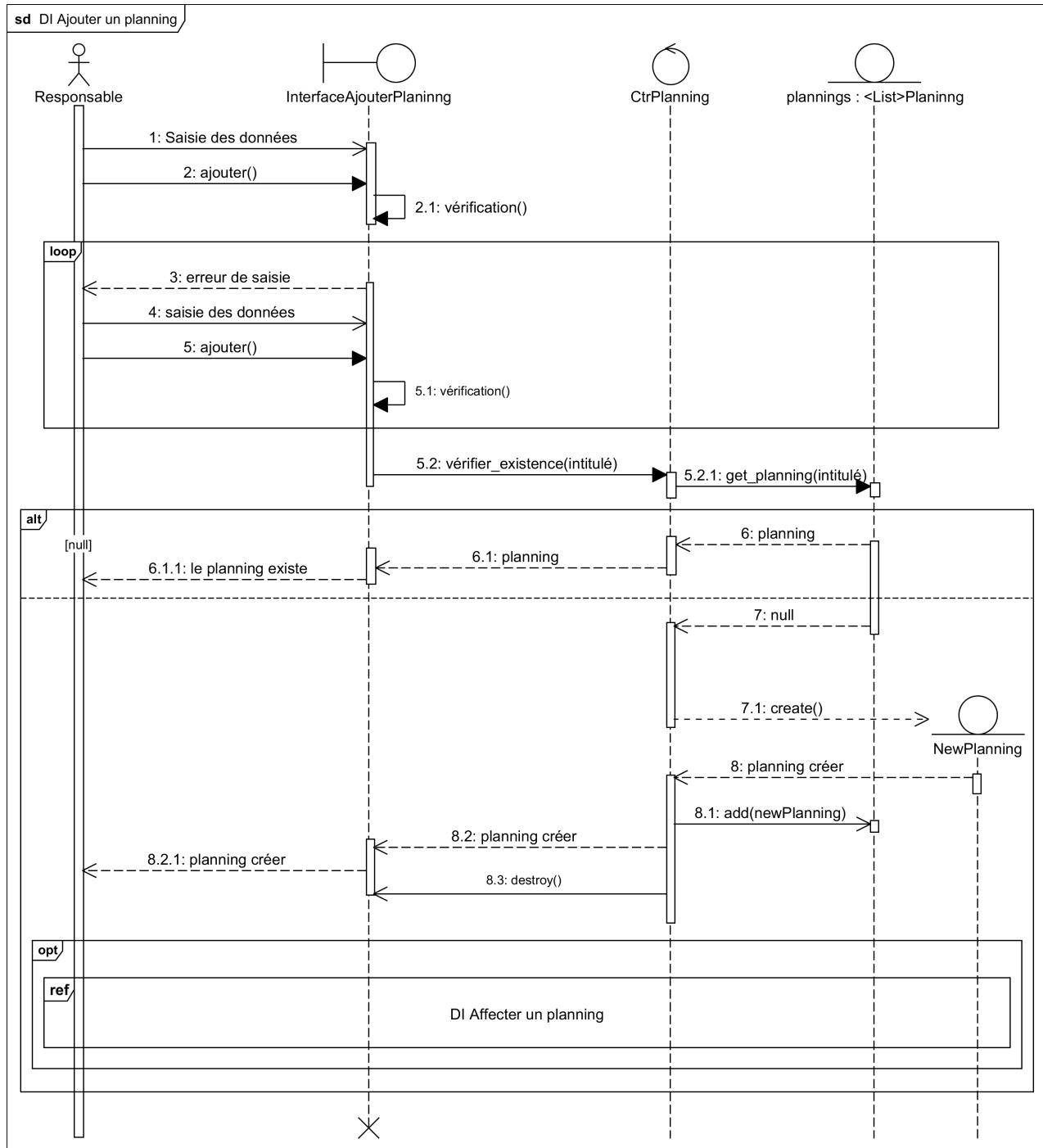


FIGURE 3.28 – Diagramme d’interaction « Ajouter planning »

Diagramme d’interaction du cas d’utilisation « Ajouter membre »

Le responsable aura la possibilité d’ajouter un membre à une équipe, il devra effectuer une recherche qui lui permettra à travers l’interface InterfaceAjouterMembre d’affecter un employé s’il existe à une équipe. Une fois terminé, le contrôle CtrEmployé retournera un message de confirmation.

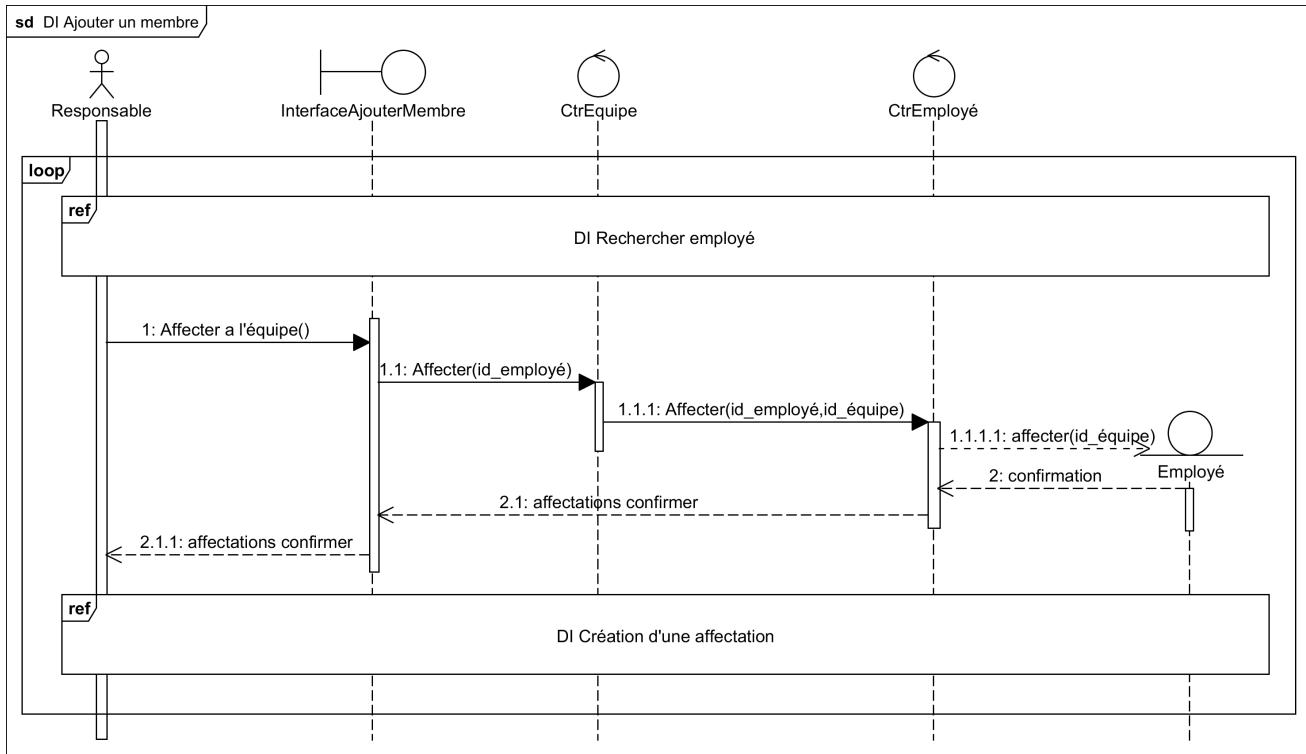


FIGURE 3.29 – Diagramme d’interaction « Ajouter membre »

Diagramme d’interaction du cas d’utilisation « Ajouter employé »

À travers l’interface InterfaceGestionEmployé, l’administrateur aura la possibilité d’ajouter un employé en saisissant les informations. Après la validation, une vérification des données est effectuée avant l’ajout de l’employé. Une fois créé, le contrôle ctrEmpreinte se chargera d’ajouter son empreinte qui déclenchera la redirection vers l’interface InterfaceGestionEmployé avec la nouvelle liste des employés.

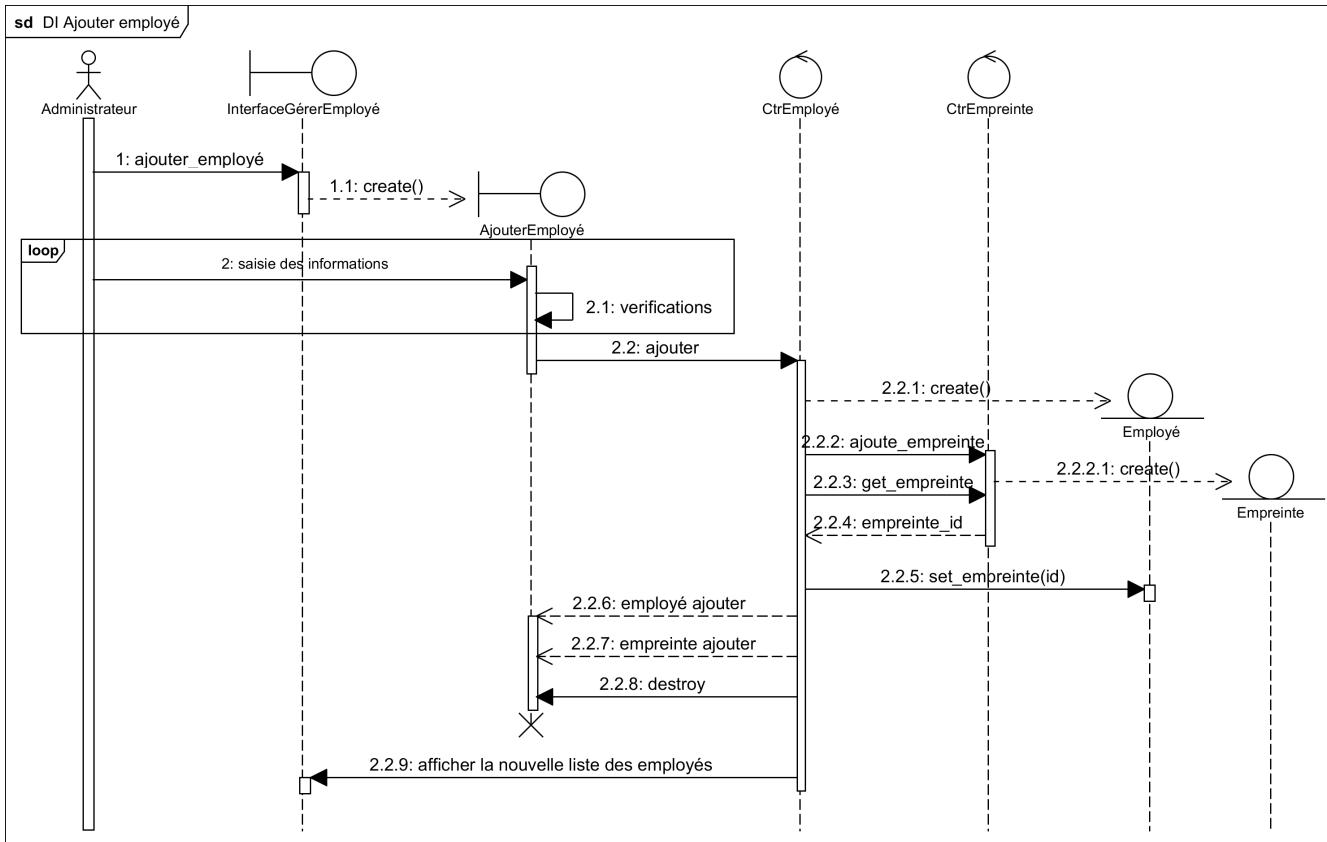


FIGURE 3.30 – Diagramme d’interaction « Ajouter employé »

Diagramme d’interaction du cas d’utilisation « Consulter profil d’un employé »

Une fois authentifié, le responsable aura la possibilité de consulter le profil d’un employé. L’interface `InterfaceProfileEmployé` déléguera au contrôle `CtrEmployé` la recherche de l’employé sélectionné, puis lui retournera les informations concernant ce dernier. Il pourra aussi à travers cette interface être redirigé vers l’interface de modification du profil ainsi que vers l’interface qui lui permettra de consulter le planning de cet employé.

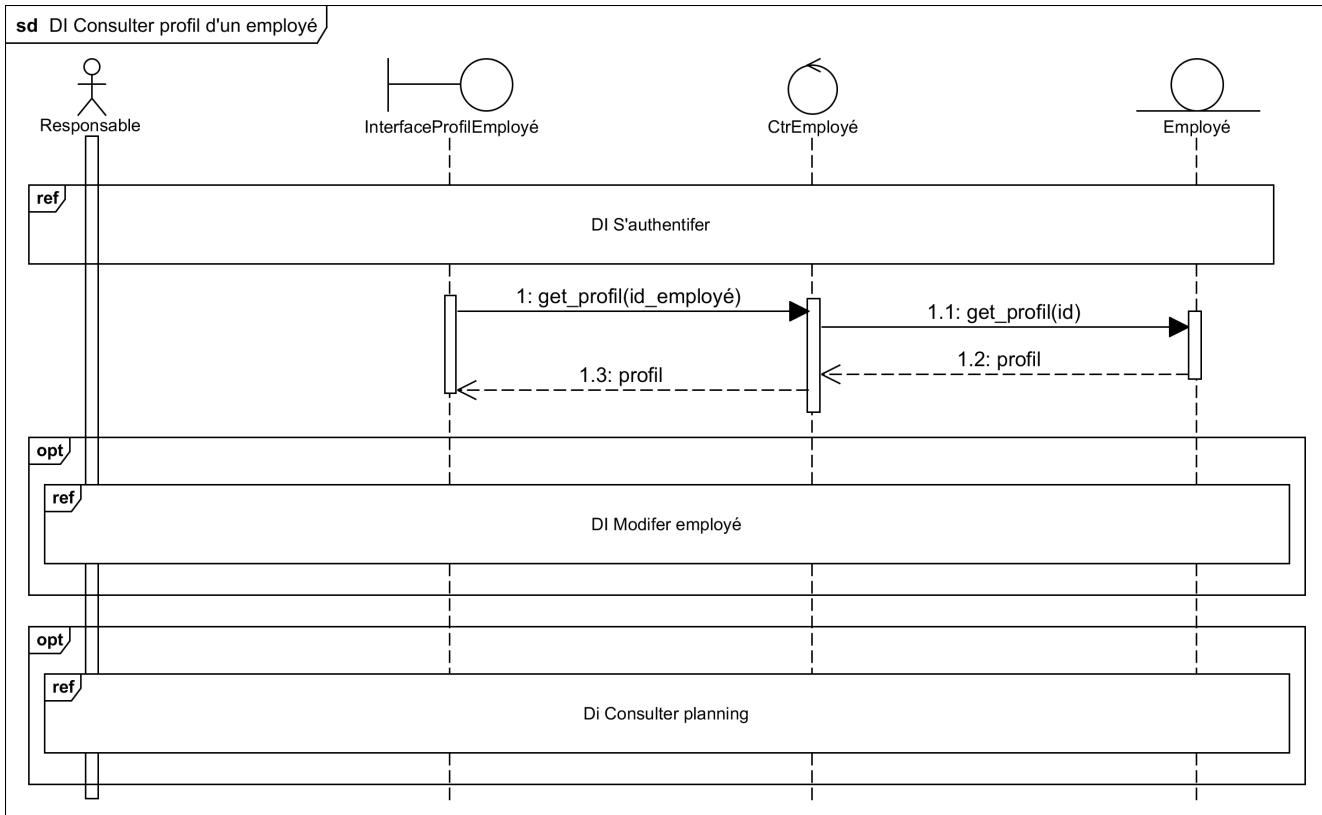


FIGURE 3.31 – Diagramme d’interaction « Consulter profil d’un employé »

3.5 Diagramme de classes conception préliminaire

En partant du modèle d’analyse, nous allons affiner et compléter les diagrammes de classes participantes obtenus précédemment. Pour cela, nous utiliserons les diagrammes de séquence que nous venons de réaliser pour :

- Ajouter ou préciser les opérations dans les classes (un message ne peut être reçu par un objet que si sa classe a déclaré l’opération publique correspondante).
- Ajouter des types aux attributs et aux paramètres et retours des opérations.
- Affiner les relations entre classes : associations (avec indication de navigabilité), généralisations ou dépendances.[5]

Diagramme de classes conception préliminaire du cas d'utilisation « Consulter ma fiche de pointage »

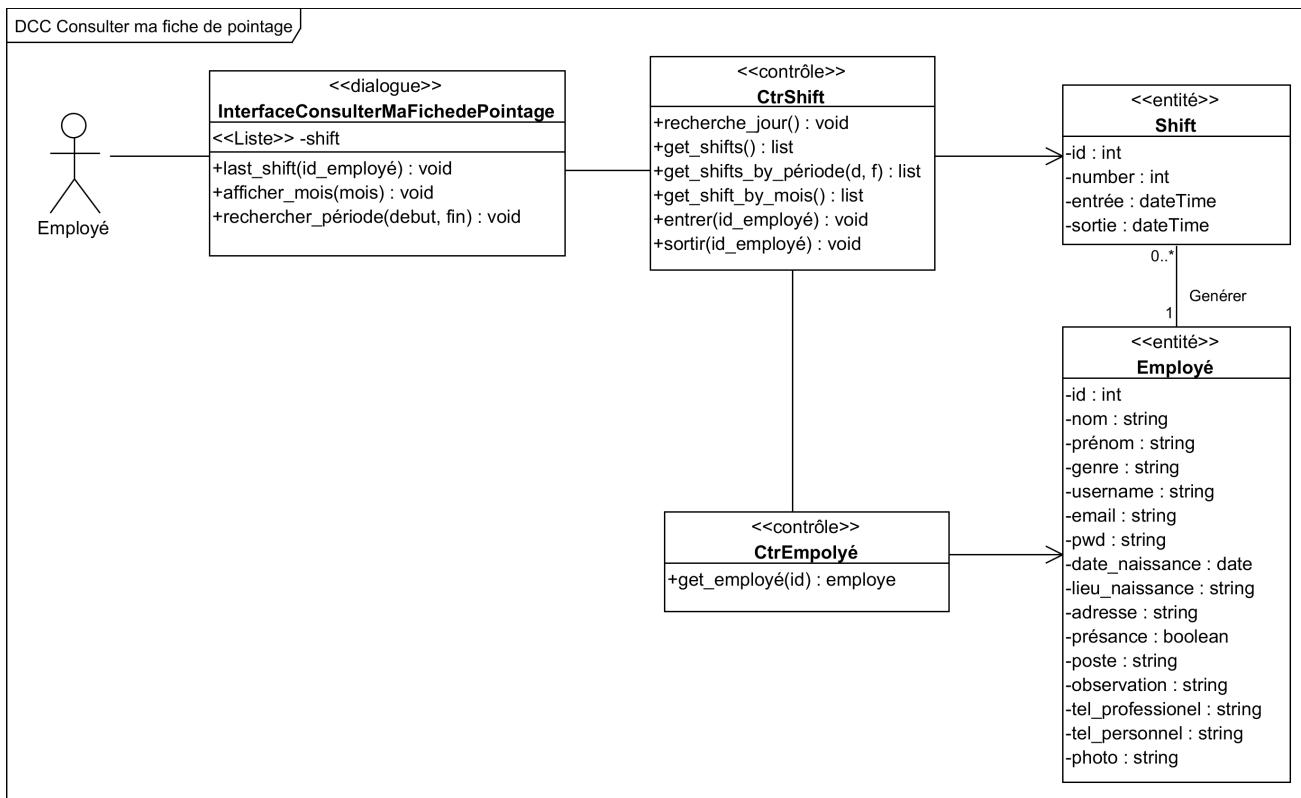


FIGURE 3.32 – Diagramme de classes conception préliminaire « Consulter ma fiche de pointage »

Diagramme de classes conception préliminaire du cas d'utilisation « Ajouter une équipe »

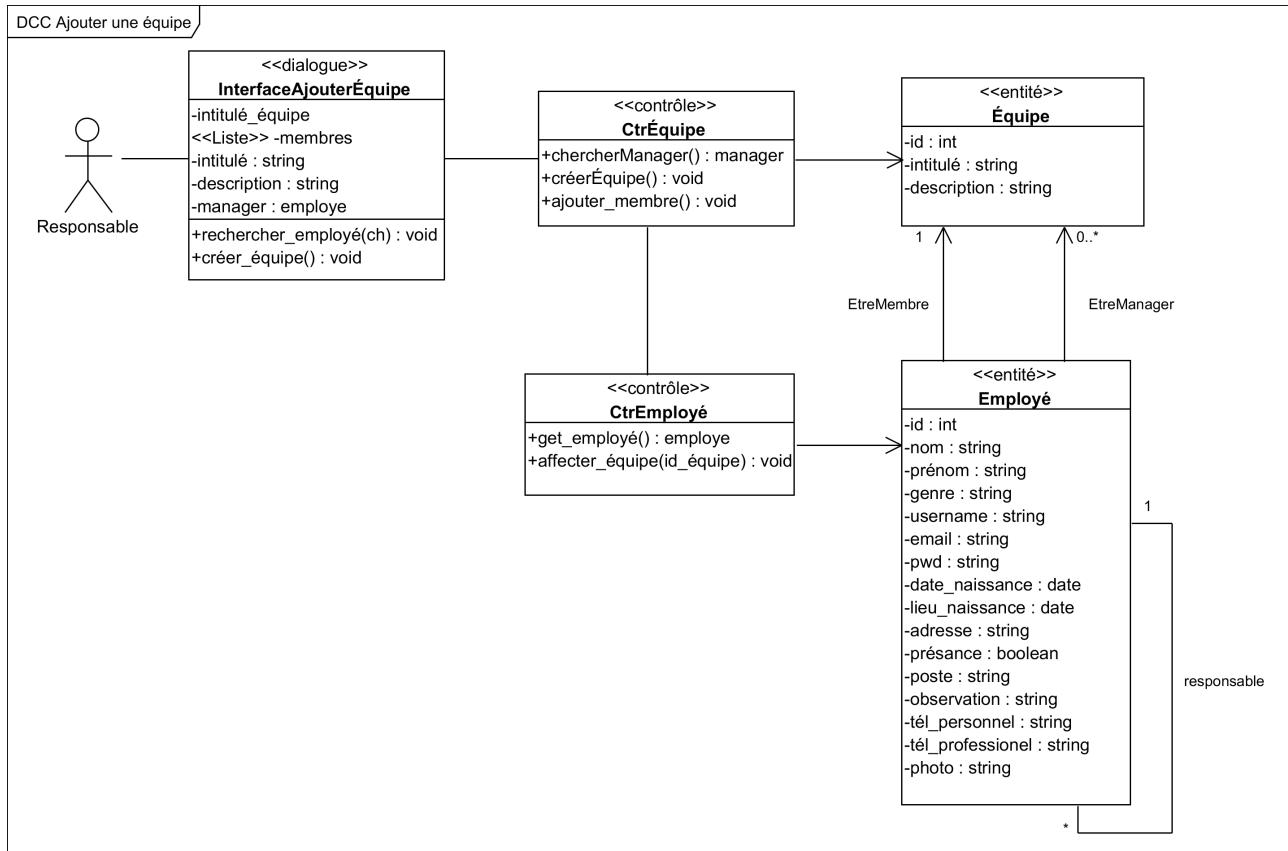


FIGURE 3.33 – Diagramme de classes conception préliminaire « Ajouter une équipe »

Diagramme de classes conception préliminaire du cas d'utilisation « Ajouter un planning »

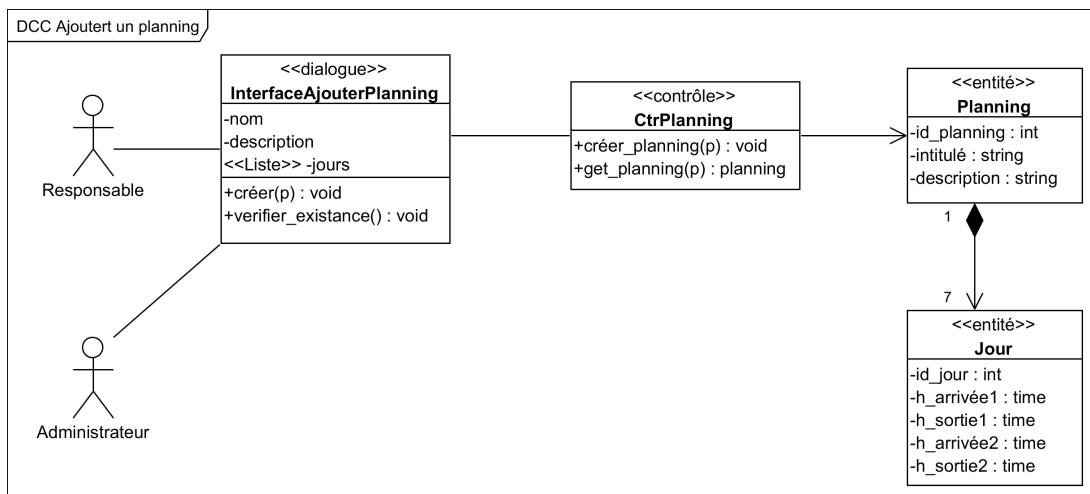


FIGURE 3.34 – Diagramme de classes conception préliminaire « Ajouter un planning»

Diagramme de classes conception préliminaire du cas d'utilisation « Ajouter un employé »

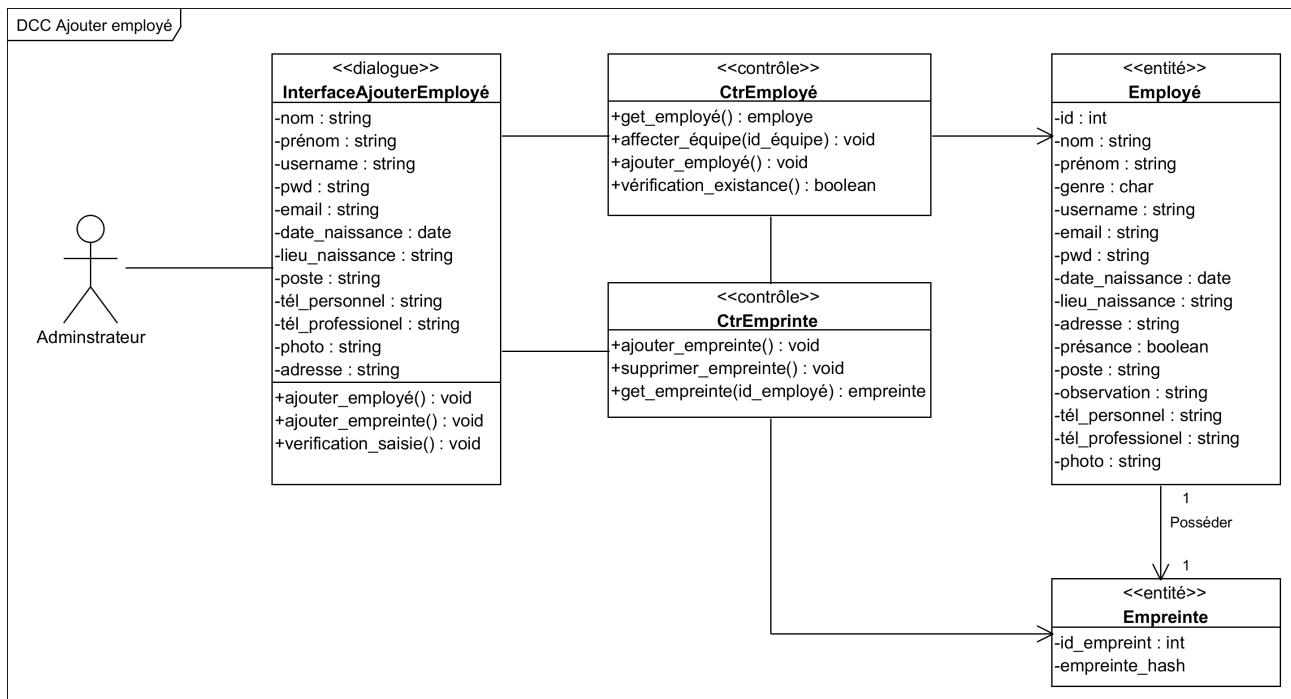


FIGURE 3.35 – Diagramme de classes conception préliminaire « Ajouter un employé»

Diagramme de classes conception préliminaire du cas d'utilisation « Consulter profil d'un employé »

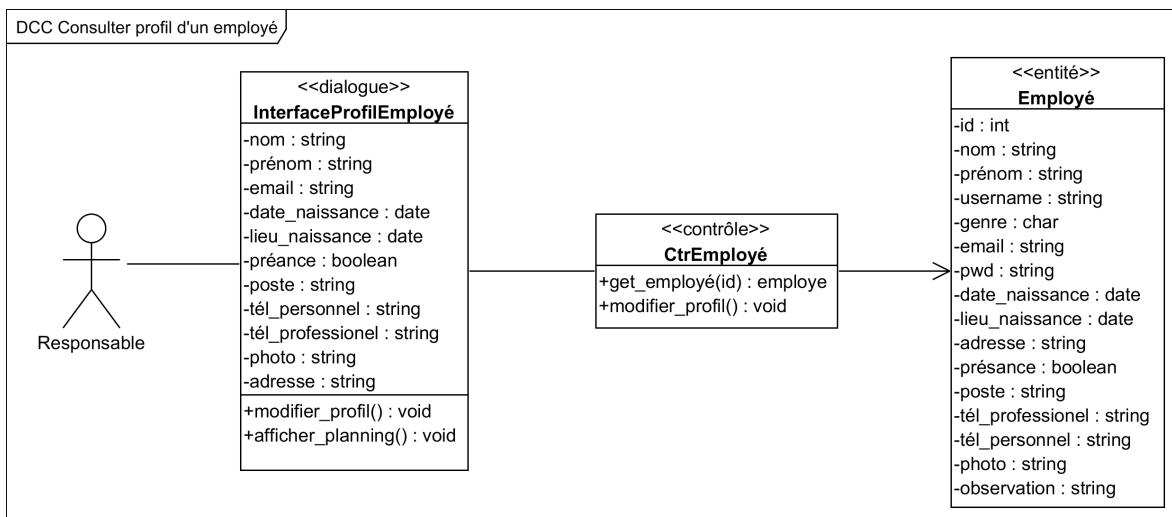


FIGURE 3.36 – Diagramme de classes conception préliminaire « Consulter profil d'un employé»

3.6 Diagramme de classes de conception

Les diagrammes de classes décrivent la structure ou plutôt l'architecture d'un système et sont donc la base de presque toutes les autres techniques de description. En conséquence, les diagrammes de classes, et en particulier les classes, sont un concept qui est utilisé universellement en modélisation et en programmation. Ils permettent l'encapsulation des attributs et des méthodes et la représentation des instances sous forme d'objets [10].

À l'aide des diagrammes de classes de conception établis dans la section précédente, nous avons modélisé le diagramme de classes ci-dessous.

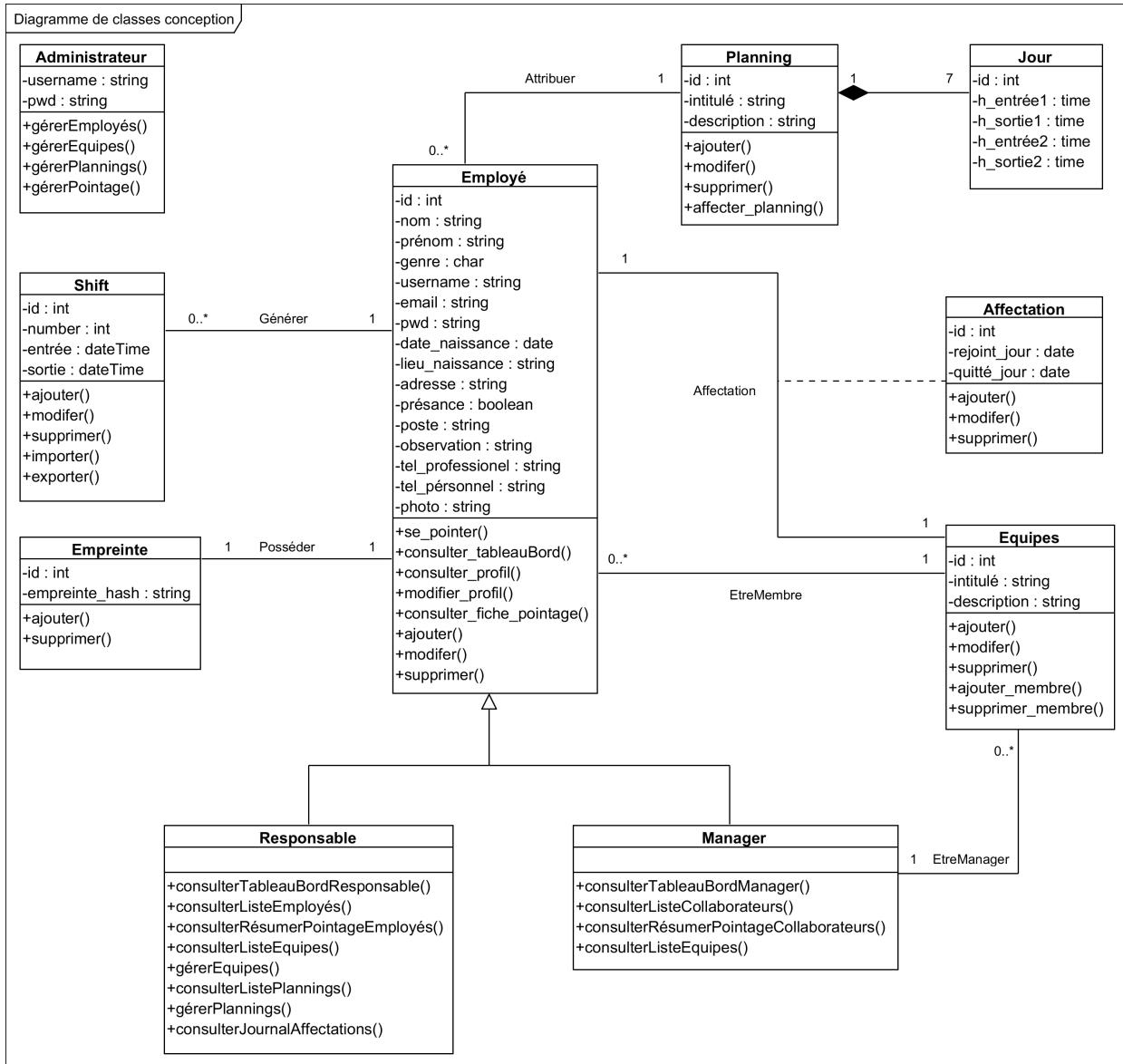


FIGURE 3.37 – Diagramme de classes de conception

3.7 Modèle relationnel

Le modèle relationnel a vu le jour en 1970 avec les travaux de Edgar Frank Codd. La principale publication est « A Relational Model for Large Shared Data Banks », Communications of the ACM, vol. 13, n° 6, 1970.

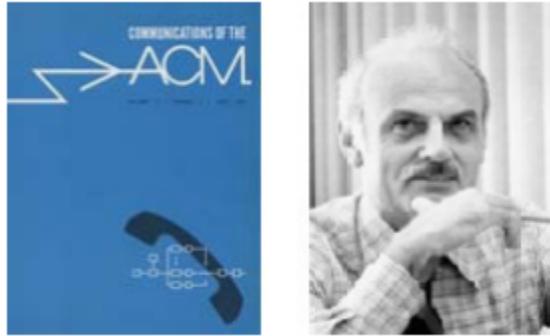


FIGURE 3.38 – Le père du modèle relationnel

Ce modèle de données est construit principalement autour des concepts suivants [11] :

- Relation : structure de données qui préfigure la table qui sera créée à l'aide du langage SQL.
- Attribut : représentation d'une information atomique qui préfigure une colonne d'une table. Un domaine de valeurs devrait être défini ainsi que d'éventuelles règles de validation (contraintes).
- Clé primaire : attribut(s) identifiant une relation qui préfigure(nt) la primary Key de la table.
- Clé étrangère : attribut(s) qui référence(nt) une tierce relation, préfigure(nt) une foreign Key de la table.

Afin de pouvoir implémenter une base de données, il faut pouvoir traduire le modèle conceptuel en modèle logique. Cela signifie qu'il faut pouvoir convertir un modèle UML en modèle relationnel. Les modèles conceptuels sont suffisamment formels pour que ce passage soit systématisé dans la plupart des cas. En appliquant les règles de passage[12], nous obtenons les relations suivantes :

- **Employé**(id_employé, nom, prénom, genre, username, email, pwd, date_naissance, lieu-naissance, adresse, présence, rôle, poste, observations, tél_professionnel, tél_personnel, photo, type, #ggid Équipe, #ggid_Planning, #id_empreinte).
- **Planning**(id_Planning, intitulé, description).
- **Jour**(id_jour, h_entrée1, h_sortie1, h_entrée2, h_sortie2, #ggid_Planning).
- **Équipe**(id_équipe, intitulé, description, #ggid_Employé).
- **Affectation**(id_affectation, rejoint_jour, quitté_jour, #ggid_équipe, #id_employé).
- **Shift**(id_shift, num_séquence, h_entrée, h_sortie, #ggid_employé).
- **Administrateur**(id_administrateur, username, email, pwd).

3.8 Conclusion

Au cours de ce chapitre, nous avons en premier lieu identifié les modèles du domaine pour avoir une vue globale sur les différentes entités de notre système, puis nous avons modélisé les diagrammes de classes participantes et d'interactions qui permettent l'enrichissement des diagrammes de séquences système. Par la suite, en modélisant les diagrammes de classes conception préliminaire, nous avons pu établir le diagramme de classes conception. Enfin, pour avoir une vue plus structurée et implémenter une base de données nous avons traduit ce dernier diagramme en un modèle relationnel.

Chapitre 4

Réalisations de la pointeuse

4.1 Introduction

Dans ce chapitre, nous aborderons l'aspect matériel du projet ; en l'occurrence la pointeuse biométrique à réaliser. En premier lieu, nous allons décrire Arduino en tant que plateforme, puis nous décrirons en détail le microcontrôleur choisi en justifiant le choix de ce dernier. Par la suite, nous citerons les caractéristiques du capteur d'empreinte. Une fois assemblés, ils formeront notre dispositif de pointage. Nous exposerons les différents schémas de branchement et réaliserons un prototype en 3D pour avoir un aperçu de l'équipement à concevoir.

La dernière section de ce chapitre concerne la communication entre la pointeuse et le système de gestions de pointage.

4.1.1 Logiciel

Le logiciel de programmation des modules Arduino, dont l'interface est appelée Arduino IDE, est une application Java libre et multiplateforme dérivée de Processing servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série (RS-232, Bluetooth ou USB selon le module). Il est également possible de se passer de l'interface Arduino, et de compiler et téléverser les programmes via l'interface en ligne de commande. Le langage de programmation utilisé est le C++, compilé avec avr.-g++ 8, et lié à la bibliothèque de développement Arduino, permettant d'utiliser la carte et ses entrées/sorties. La mise en place de ce langage standard rend aisément le développement de programmes sur les plateformes Arduino à toute personne maîtrisant le C ou le C++ [31].

4.2 ESP32

ESP32 est une série de microcontrôleurs de type système sur une puce (SoC) d'*Espressif Systems*, basé sur l'architecture Xtensa LX6 de Tensilica, intégrant la gestion du Wi-Fi et du Bluetooth (jusqu'à LE 5.0 et 5.11) en mode double, et un DSP. C'est une évolution d'ESP8266. Son support Wi-Fi et Bluetooth, en fait un système apprécié dans le domaine de l'internet des objets. Ce SoC

rencontre un certain succès depuis quelques années à la fois pour son coût, ses capacités et son intégration dans un nombre croissant de systèmes [32].



FIGURE 4.1 – ESP32 DevKit

4.2.1 Caractéristiques techniques

- **Processeurs :**

- CPU : Xtensa double-cœur (ou simple-cœur), microprocesseur LX 32 bits, fonctionnant à 160 ou 240 MHz et fournissant jusqu'à 600 DMIPS ;
- coprocesseur ultra basse consommation (ULP) ;

- **Mémoire :** 520 KiO SRAM ;

- **Connectivité sans-fil :**

- Wi-Fi : 802.11 b/g/n ;
- Bluetooth : v 4.2 BR/EDR et BLE jusqu'à v 5.0 et v 5.1 ;

- **Interfaces de périphériques**

- 12-bit Segmentation sur les ADC (SAR ADC) jusqu'à 18 canaux ;
- 2 × 8 bit DAC ;
- 10 × capteurs de touché (GPIO de capteur capacitif (en)) ;
- 4 × SPI ;
- 2 × interfacs I²S ;
- 2 × interfaces I²C ;
- 3 × UART ;
- contrôleur hôte SD/SDIO/CE-ATA (en)/MMC/eMMC ;

- contrôleur esclave SDIO/SPI ;
- interface MAC Ethernet avec DMA dédié et support du protocole de temps précis IEEE 1588 ;
- Bus de données CAN 2.0 ;
- contrôleur infrarouge distant (TX/RX, jusqu'à 8 canaux) ;
- Moteur PWM ;
- LED PWM (jusqu'à 16 canaux) ;
- Capteur à effet Hall ;
- préamplificateur analogique ultra-basse consommation ;

- **Sécurité**

- Standard de sécurité supportant complètement IEEE 802.11, incluant WPA/WPA2 et WAPI de WFA ;
- Secure boot (démarrage sécurisé) ;
- Chiffrement de la Flash ;
- 1024-bit OTP, jusqu'à 768 bits pour les clients
- Accélération matérielle du chiffrement : AES, SHA-2, RSA, elliptic curve cryptography (ECC), générateur de nombre aléatoire (en) (RNG) ;

- **Gestion de l'énergie**

- low-dropout regulator (en) interne ;
- Domaines d'alimentation individuels pour le RTC ;
- Alimentation en sommeil profond de $5 \mu A$;
- Réveil depuis des interruptions GPIO, timer, mesure ADC, interruption du capteur de touché capacitif.

Le choix du module ESP32 Devkit est justifié par sa petite taille (57mm*28mm*3mm), ainsi que ses caractéristiques techniques, notamment les connectivités sans-fil, la sécurité, et la gestion de l'énergie[33].

4.3 Capteur d'empreinte DY 50

Les techniques utilisées pour capter les images des empreintes digitales sont diverses, d'où l'existence de plusieurs types de capteurs. L'on peut citer :

- les capteurs thermique.
- les capteurs en silicium.

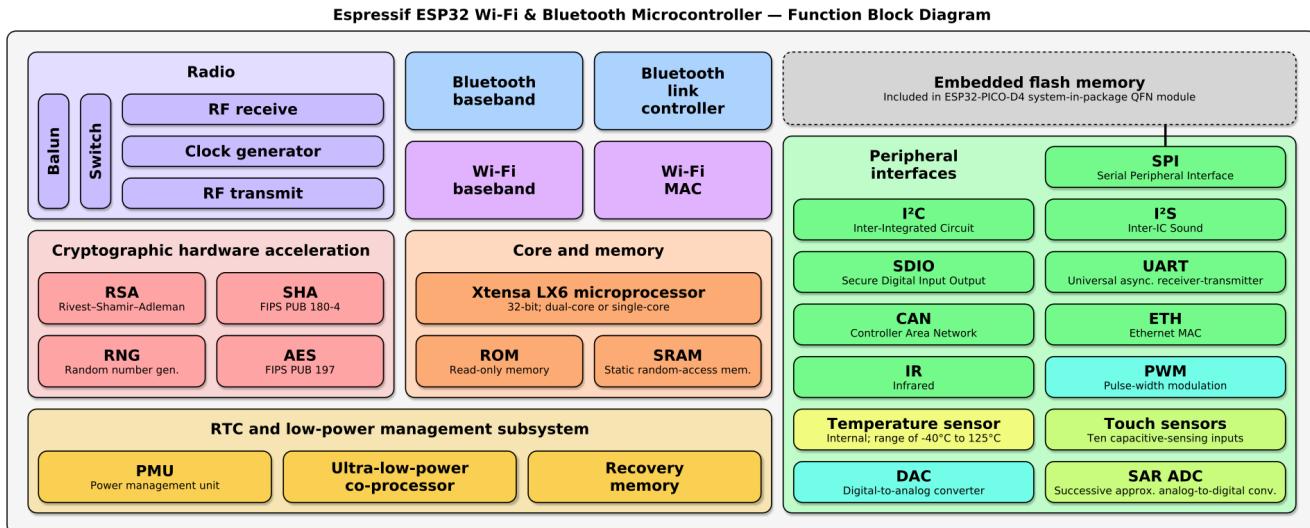


FIGURE 4.2 – Schéma fonctionnel

- les capteurs ultrasonique.
- les capteurs capacitif.
- les capteurs à champ électrique.
- les capteurs à pression.
- les capteurs optique.

Pour les besoins de notre projet, nous avons choisi le capteur DY50 de type optique en raison de son coût, sa taille, et du fait d'avoir une image précise de l'empreinte tout en étant intrinsèquement protégé contre les décharges électrostatiques.



FIGURE 4.3 – Capteur optique DY50

4.3.1 Principe de fonctionnement

Grâce au principe de l'imagerie optique, les lignes créées par l'irrégularité de la peau sur la face interne du doigt formeront diverses images d'empreintes digitales. La texture de la peau est différente par les motifs, les points de rupture et les intersections. Ils sont appelés dans le traitement de l'information « Points caractéristiques », les caractéristiques de chaque doigt sont différentes, c'est-à-dire uniques, en s'appuyant sur cette unicité, nous pouvons associer une personne à son empreinte digitale, en préstockant son empreinte digitale [34].

Le système d'identification d'empreintes digitales collecte, analyse et compare les empreintes digitales grâce à un équipement de conversion photoélectrique spécial et à une technologie de traitement d'image, et peut identifier automatiquement, rapidement et avec précision l'identité personnelle. Le système comprend principalement des processus tels que l'acquisition d'images d'empreintes digitales, le traitement d'images d'empreintes digitales, l'extraction de caractéristiques, ainsi que la comparaison et la correspondance des valeurs des caractéristiques.

Broche	Nom	Type	Description fonctionnelle
1	GND	Dans	Module d'alimentation l'apport positif
2	TXD	Sur	Sortie de données série TTL. Niveau logique TTL
3	RXD	Dans	Entrée de données série TTL. Niveau logique TTL
4	3.3V	-	Masse du Signal. Connexion à la terre interne et électrique
5	Tactile	Sur	Sortie de signal d'induction, niveau élevé par défaut efficace
6	TouchVin	Dans	Entrée d'alimentation à détection tactile, alimentation 3.3v

TABLE 4.1 – Interface de connexion [34]

4.3.2 Caractéristiques techniques du capteur optique

- Tension d'alimentation : cc 3.6 – 6.0V / 3.3V fournissant.
- Courant d'alimentation : courant : <120mA.
- Courant de crête : <140mA.
- Temps d'image d'empreinte digitale : <1.0 seconde.
- Taille de fenêtre : 14*18mm.
- Mode correspondant : mode Match (1 :1).
- Mode de recherche (1 : N).
- Fichier de Signature : 256 octets.

- Fichiers de modèle : 512 octets.
- Capacité de stockage : 127.
- Niveau de sécurité : cinq (min à max : 1,2,3,4,5).
- Taux de fausses acceptations : <0.001% (niveau de sécurité 3).
- Taux de faux rejet (FRR) : <1.0% (niveau de sécurité 3).
- Temps de recherche : <1.0 seconde (1 :500, la moyenne).
- Interface PC : UART (niveau logique TTL) ou USB2.0 / USB1.1.
- Débit en bauds de Communication (UART) : (9600 * N) bps où N = 1 – 12 (valeur par défaut N =6, soit 57600bps).
- Environnement de travail :
 - Température : -20 ° - + 50.
 - Humidité Relative : 40% RH-85 % hr (sans condensation).
- Environnement de stockage :
 - Température : -40 - + 85°.
 - Humidité Relative : <85% H (sans condensation).
- Dimensions (L * W * H) : 56*20*21.5mm .

4.4 Schémas et branchement

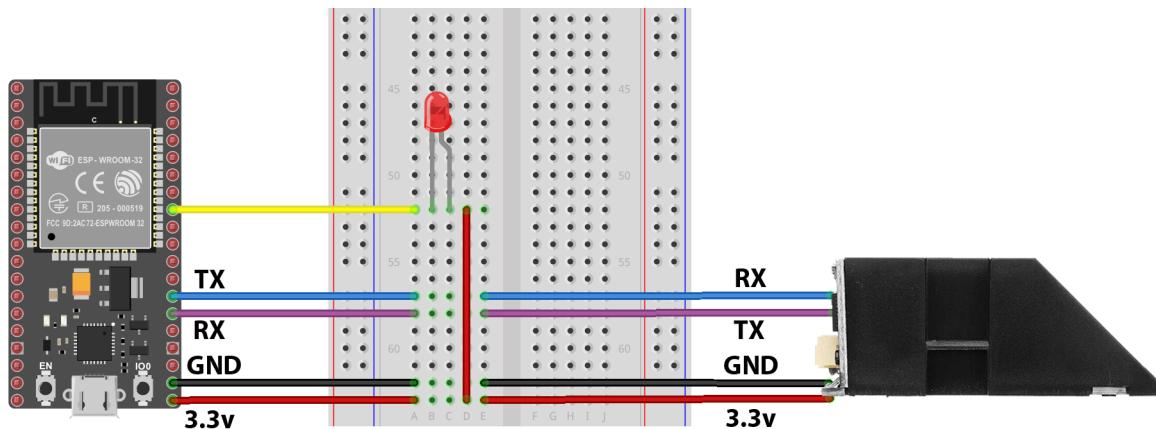


FIGURE 4.4 – Schéma expliquant le montage préliminaire des différents modules

Le schéma 4.4 révèle la manière dont le module ESP32 Devkit ainsi que le capteur d'empreinte DY50 ont été connectés, ainsi qu'une LED pour signaler à l'utilisateur la réussite ou l'échec des différentes opérations.

Branchement :

- TX (ESP32) -> RX (DY50)
- RX (ESP32) -> TX (DY50)
- GND (ESP32) -> GND (DY50)
- 3.3v (ESP32) -> vcc (DY50)

Afin d'améliorer l'expérience utilisateur, nous avons rajouté un écran qui permettra d'afficher les différents messages comme la réussite ou l'échec du pointage ou le nombre d'empreintes enregistrer.

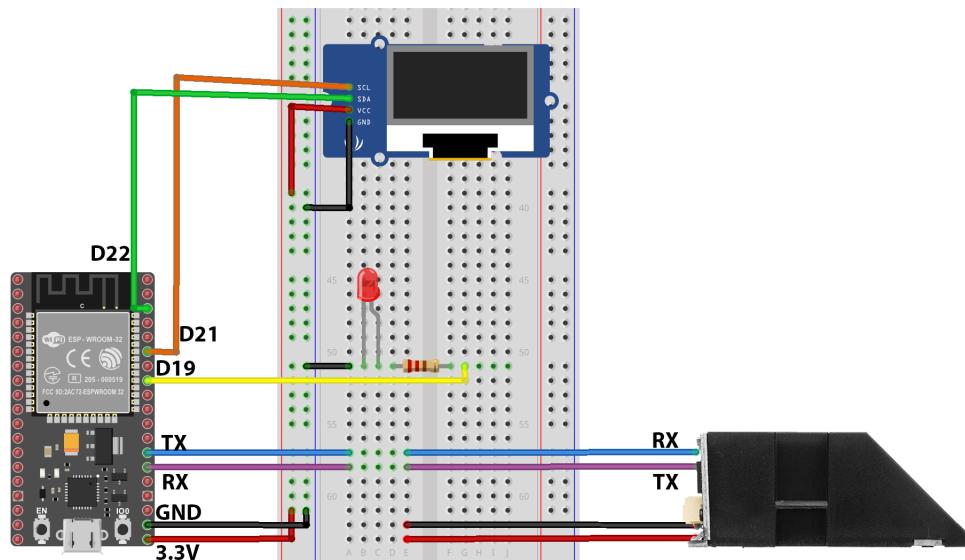


FIGURE 4.5 – Schéma expliquant le montage préliminaire des différents modules

Branchement :

- GPIO21 (ESP32) -> SLC (OLED Display)
- GPIO22 (ESP32) -> SDA (OLED Display)
- GND (ESP32) -> GND (OLED Display)
- 3.3v (ESP32) -> VCC (OLED Display)

Compte tenu du contexte actuel (crise sanitaire due au virus COVID-19) et afin de limiter la propagation du virus des gestes barrières sont recommandés. Par ailleurs, le fait de toucher une surface ayant été en contact avec plusieurs individus sans que cette dernière ne soit désinfectée au préalable constitue un risque sanitaire. Pour remédier à ce problème, nous avons eu l'idée d'équiper notre pointeuse d'une LED UV-C (lampe germicide). Néanmoins, ce composant doit faire l'objet

d'une recherche plus approfondie pour confirmer la possibilité de l'installer, car la longueur d'onde de ce type de lumière peut avoir des effets secondaires indésirables en cas de fortes expositions.

Pour garantir la disponibilité et l'accessibilité du service de pointage, la pointeuse doit être fonctionnelle même en cas d'absence de courant. Afin de remédier à cette contrainte, nous avons décidé d'inclure une source d'alimentation secondaire (batterie) qui alimentera tout le dispositif de pointage en cas de coupure de courant —ceci en supposant que le serveur hébergeant le système dispose lui aussi d'une alimentation d'urgence en cas de coupure.

4.5 Prototype

Le prototype matérialise une étape d'évolution d'un projet, souvent pour démontrer ou infirmer le bien-fondé d'un ou plusieurs concepts mis en jeu dans ce projet avant toute valorisation commerciale.

Nous avons décidé de réaliser un prototype en 3D du dispositif de pointage afin d'avoir une idée du résultat final.

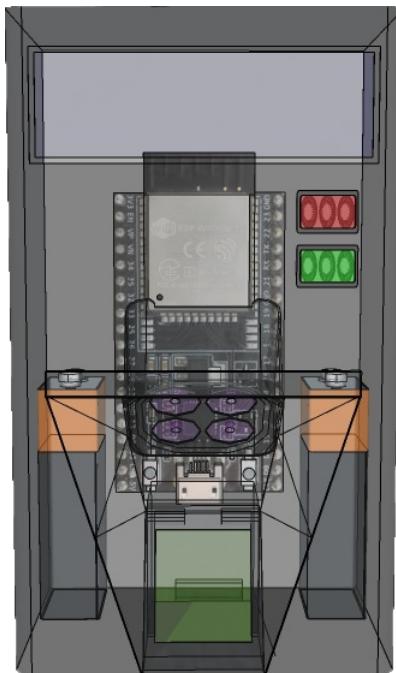


FIGURE 4.6 – Pointeuse vue de face

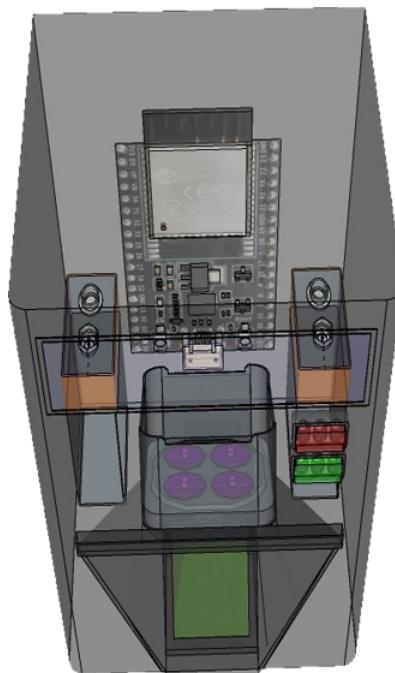


FIGURE 4.7 – Pointeuse vue de haut

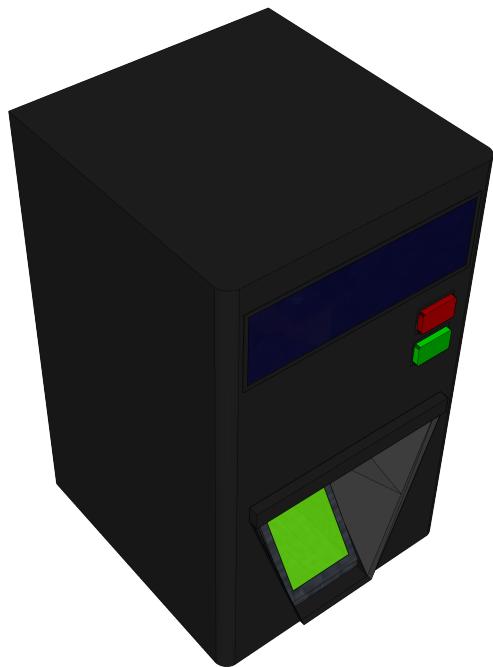


FIGURE 4.8 – Côté gauche extérieur

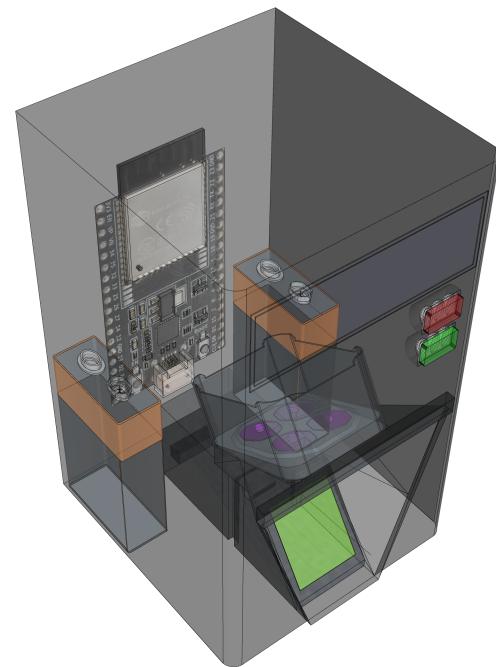


FIGURE 4.9 – Côté gauche intérieur



FIGURE 4.10 – Pointeuse ouverte

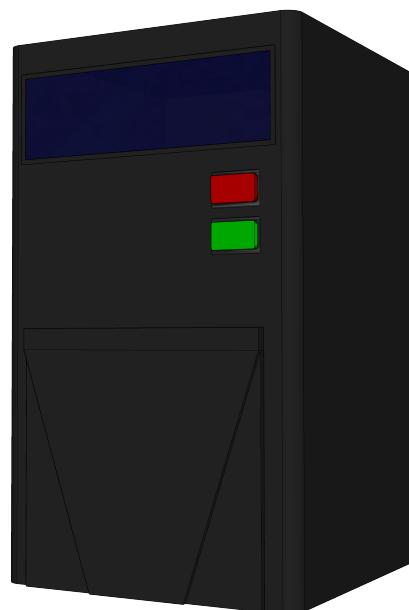


FIGURE 4.11 – Pointeuse fermée

Les images précédentes représentent la disposition des différents composants dans un boîtier. Elles nous donnent un aperçu sur l'aspect extérieur de la pointeuse et de sa forme. Il est utile de préciser que le lecteur dispose d'un cache rétractable, une manœuvre obligatoire lorsque :

- la LED UV-C est utilisée pour désinfecter la surface de lecture d'empreinte du capteur. Afin de protéger les utilisateurs des effets indésirables des rayonnements ultraviolets.
- Le dispositif est transporté ou déplacé afin de protéger la surface en verre de toute dégradation ou égratignure qui pourraient affecter les performances de la pointeuse ou la rendre complètement défectueuse.

4.6 Programmation

Une fois les parties matérielles correctement assemblées, nous allons passer à la partie programmation de la pointeuse. En premier lieu, nous allons définir le rôle de chaque composant ainsi que son champ d'action.

- **Le module ESP32 DevKit** : Ce dernier représente le cerveau de la pointeuse, car le programme de la pointeuse sera logé en lui. Il sera responsable de la communication entre la pointeuse et l'application. Il exécutera les différentes commandes exigées par l'administrateur (changement de mode enregistrement/suppression/pointage, suppression de la base de données des empreintes au niveau du capteur d'empreinte).
- **Le capteur d'empreinte DY50** : Ce dernier représente un périphérique d'entrée d'empreinte. À l'aide de sa mémoire interne, on pourra sauvegarder les images détaillées des empreintes, ce qui minimise drastiquement le temps de reconnaissance des différentes empreintes vu que leurs images sont déjà présentes dans la mémoire du capteur.

Comme expliqué précédemment la plateforme Arduino permet de travailler avec des modules compatibles, même s'ils sont issus d'un autre fabricant. Ce qui est le cas concernant le module ESP32 Devkit et notre capteur d'empreintes DY50. Néanmoins il est indispensable de télécharger les pilotes et les librairies des fabricants.

4.6.1 Environnement de travail

Arduino IDE

Le logiciel de programmation des cartes Arduino est une application Java libre et multiplate-forme, servant d'éditeur de code et de compilateur qui peut transférer le programme au travers de la liaison série (RS-232, Bluetooth ou USB selon le module). Il est également possible de se passer de l'interface Arduino, et de compiler les programmes via l'interface en ligne de commande.

```

Blink | Arduino 1.8.12
Fichier Edition Creer Outils Aide
Blink
modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
 */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}

```

Serial monitor window:

FIGURE 4.12 – interface Arduino IDE et un programme qui clignote une LED

Langage de programmation C

Le C est un langage de programmation impératif généraliste de bas niveau. Inventé au début des années 1970 pour réécrire UNIX. C'est devenu un des langages les plus utilisés, encore de nos jours. De nombreux langages plus modernes comme C++, C#

, Java et PHP ou JavaScript ont repris une syntaxe similaire et reprennent en partie sa logique. Il offre aux développeurs une marge de contrôle importante sur la machine (notamment sur la gestion de la mémoire) et est de ce fait utilisé pour réaliser les « fondations » (compilateurs, interpréteurs...) de ces langages plus modernes.

Ces caractéristiques en font un langage privilégié quand on cherche à maîtriser les ressources matérielles utilisées, le langage machine et les données binaires générées par les compilateurs étant relativement prévisibles. Ce langage est donc extrêmement utilisé dans des domaines comme la programmation embarquée sur microcontrôleurs, les calculs intensifs, l'écriture de systèmes d'exploitation et les modules où la rapidité de traitement est importante. Il constitue une bonne alternative au langage d'assemblage dans ces domaines, avec les avantages d'une syntaxe plus expressive et de la portabilité du code source. Le langage C a été inventé pour écrire le système d'exploitation UNIX, et reste utilisé pour la programmation système. Ainsi, le noyau de grands systèmes d'exploitation comme Windows et Linux sont développés en grande partie en C.

En contrepartie, la mise au point de programmes en C, surtout s'ils utilisent des structures de données complexes, est plus difficile qu'avec des langages de plus haut niveau. En effet, dans un souci de performance, le langage C impose à l'utilisateur de programmer certains traitements (libération de la mémoire, vérification de la validité des indices sur les tableaux...) qui sont pris en charge automatiquement dans les langages de haut niveau[35].

Esspresif/arduino-esp32 library

Une librairie écrite et maintenue par le constructeur Esspresif. Cette dernière doit être installée au niveau de l'IDE Arduino pour pouvoir téléverser les programmes écrits sur le module ESP32 Devkit. Elle est disponible sur : <<https://github.com/espressif/arduino-esp32>>

Adafruit fingerprint sensor library

Pour pouvoir interagir avec le capteur d'empreinte cette librairie est nécessaire, car elle implémente les différentes méthodes et fonctions qui régissent le fonctionnement de capteur d'empreinte DY50 on peut citer :

- check_model() : qui vérifie le statut du capteur d'empreinte. Retourne l'erreur ou OK.
- count_template() : qui demande au capteur de compter le nombre d'empreintes dans sa mémoire et stocker cette valeur dans *self.template_count*.
- creat_model() : qui demande au capteur de transformer l'empreinte en un modèle exploitable. Renvoie le code d'erreur du packet ou OK.
- delet_model(localisation) : qui demande au capteur de supprimer un modèle d'empreinte de la mémoire flash après l'avoir sélectionné grâce à l'argument 'localisation'. Retourne l'erreur ou OK.
- empty_library() : qui demande au capteur de réinitialiser toute la mémoire flash ainsi que de supprimer tous les modèles précédemment enregistrer.
- get_image() : qui demande au capteur de capturer une image et de l'enregistrer dans la mémoire. Retourne l'erreur ou OK.
- finger_search() : qui demande au capteur de chercher l'empreinte correspondante à partir du slot 1, enregistre la localisation et le taux de correspondance dans *in self.finger_id* et *self.confidence*. Renvois l'erreur ou Ok.[36]

4.6.2 Code et fonctionnement

Comme la plupart des programmes destinés ont des modules qui sont compatibles avec Arduino, le programme à téléverser est constitué de :

- une partie pour la déclaration des librairies utilisées et des variables globales

```
#include <Adafruit_Fingerprint.h>
#include <HardwareSerial.h>
#include <WiFi.h>
#include <HTTPClient.h>
```

```

const char *ssid = "TP-LINK_9C42F4";
const char *password = "*****";
// post array that will be send to the website
String postData;
//computer IP or the server domain
String linkee = "http://192.168.1.222:8000/pointage/getid";
// The Fingerprint ID from the scanner
uint8_t id;

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial2);
int FingerID = 0;

```

le code précédent représente la partie déclarative de notre programme où nous avons inclus les librairies nécessaires (<Adafruit_Fingerprint.h>, <HardwareSerial.h>, <WiFi.h>, <HTTP-Client.h>) au fonctionnement de la pointeuse ainsi que les variables globales utilisées (postData, postData, linkee, id).

- une partie void setup() : qui ne s'exécute qu'une seule fois est la partie où nous initialisons et configurons nos entrées/sorties et variables

```

void setup(){
    pinMode(2, OUTPUT);
    Serial.begin(115200);
    connectToWiFi();

    // set the data rate for the sensor serial port
    finger.begin(57600);
    Serial.println("\n\nAdafruit finger detect test");

    //-----*test the connection*-----
    if (finger.verifyPassword()){
        Serial.println("Found fingerprint sensor!");
    }else{
        Serial.println("Did not find fingerprint sensor:");
        while (1)
        {
            delay(1);
        }
    }
}

```

```

//-----
finger.getTemplateCount();
Serial.print("Sensor contains ");
Serial.print(finger.templateCount);
Serial.println(" templates");

//SendFingerprintID( FingerID );
//----------------* deleting all templates-----//
//emptyDataBase();

}


```

Dans cette section du code, nous avons initialisé la fréquence de transmission des différents modules. Ouvrir une connexion au point d'accès déclaré précédemment, tester la liaison entre l'ESP32 et le capteur, et afficher le nombre d'empreintes enregistrer dans le capteur (lors de la phase de test).

- une partie void loop() : est la partie principale où se déroule le programme. Tout ce qui va être écrit dans cette zone sera exécuté par la carte en boucle.

```

void loop(){
    //check if there's a connection to WiFi or not
    if (WiFi.status() != WL_CONNECTED){
        connectToWiFi();
    }
    //The ID start form 1 to 127
    // Get the Fingerprint ID from the Scanner
    FingerID = getFingerprintID();
    //don't need to run this at full speed.
    Serial.println(FingerID);
    delay(50);
    checkfinger();
    if (FingerID == 1){
        Serial.println("mode Enregistrement/Suppression ");
        checkToDelete();
        delay(5000);
        checkToAdd();
    }
}

```

Cette partie qui sera exécutée en boucle, englobe le comportement que va adopter la pointeuse. Après la connexion au point d'accès établie et le capteur correctement connecté.

- une partie pour les différentes procédures et méthodes utilisées

```

void checkToAdd(){
    HTTPClient http; //Declare object of class HTTPClient
    //Post Data
    // Add the Fingerprint ID to the Post array in order to send it
    postData = "Get_Fingerid=get_id";
    // Post methode
    //initiate HTTP request, put your Website URL or Your Computer IP
    http.begin(linkee);
    //Specify content-type header
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");

    int httpCode = http.POST(postData); //Send the request
    String payload = http.getString(); //Get the response payload

    if (payload.substring(0, 6) == "add-id"){
        // Serial.println("j'ai reçu une réponse qui contient add id");
        String add_id = payload.substring(6);
        Serial.println(add_id);
        id = add_id.toInt();
        if (id != 0){
            Serial.println("posez votre empreinte ");
            getFingerprintEnroll();
        }
    }
    //Close connection
    http.end();
}

```

Cette partie du code représente une des nombreuses fonctions déclarées au niveau de la quatrième partie de notre programme. La méthode checkToAdd() a pour rôle de vérifier si une empreinte est en attente d'enregistrement après avoir consulté la base de données du système. Dans le cas d'une réponse « add-idN°empreinte » le capteur enclenche le processus d'enregistrement d'une nouvelle empreinte.

Il existe plusieurs fonctions dans le programme qui sont destinées à la pointeuse (sendFingerPrintID, confirmAdding, checkFinger, etc.) nous avons décidé de présenter celle-ci à titre d'exemple.

4.7 Conclusion

Dans ce chapitre, nous avons présenté les différentes phases de la réalisation de la pointeuse ainsi que les différents composants la constituant et les outils utilisés. Après avoir brièvement présenté la plateforme Arduino, nous avons exposé les caractéristiques du module ESP32 Devkit ainsi que du capteur d'empreinte DY50. Ensuite, nous avons illustré le branchement des différentes parties. À l'aide du prototype, on a modélisé en 3D la pointeuse. Maintenant qu'on peut concrètement identifier chaque individu grâce à son empreinte, on peut passer à l'étape suivante qui est la réalisation du système responsable d'exploiter cette information.

Chapitre 5

Implémentations, mise en service et test

5.1 Introduction

Ce chapitre sera d'abord consacré à la présentation de l'environnement de développement et des librairies utilisées. Ensuite, nous expliquerons l'architecture globale du système ainsi que son principe de fonctionnement. La deuxième partie sera réservée à la présentation de l'application dans ses différents aspects en commençant par sa charte graphique et la hiérarchie des pages de l'application web. Enfin, nous illustrerons quelques interfaces issues de l'application et nous aborderons brièvement le côté sécurité de l'application réalisée.

5.2 Environnement de développement

Un environnement de développement se réfère à une suite d'applications et d'outils qui permettent un développement d'application facile. Par exemple, gérer des fichiers sources, déboguer du code, et enfin tester l'application avant de la lancer sur un environnement de production. Dans cette section nous allons présenter l'environnement de développement utilisé pour développer notre application web.

5.2.1 Plateformes utilisées

Slack

Slack est une plateforme de communication collaborative propriétaire, elle permet de faciliter l'organisation grâce à un système de canaux centralisant tout ce qui concerne un projet, un sujet ou une équipe. Elle permet également de conserver les fichiers et les messages partagés dans ces canaux[37].



Trello

Trello est un outil de gestion de projet en ligne, lancé en septembre 2011. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement. Il permet donc d'organiser les projets et de définir leur ordre de priorité de façon amusante, souple et enrichissante[38].



Github

GitHub offre un service d'hébergement qui permet aux développeurs de stocker et de partager, publiquement ou non, le code qu'ils créent. Elle fournit aussi plusieurs fonctionnalités de collaboration, telles que des wikis et des outils de gestion des tâches de base pour chaque projet[40].



Discord

Discord est une plateforme gratuite qui offre un service de communication par chat vidéo, vocal et textuel, il est donc utile et nécessaire de l'utiliser pour une communication en continu surtout en temps de pandémie[40].



5.2.2 Logiciels utilisés

Visual studio code

Visual Studio Code (VSCode) est un éditeur de code source, développé par Microsoft en tant que projet open source. Il s'agit de l'outil d'environnement de développement le plus populaire au monde selon un sondage auprès des développeurs réalisé par Stackoverflow[13]. En plus d'être un éditeur open source, il dispose d'un contrôle de gestion intégrée et d'un grand nombre d'extensions[41].



Git

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux en 2016. Il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes[43].



Conda

Conda est un système de gestion de paquets open source et un système de gestion d'environnement qui permet d'installer, exécuter et de mettre à jour rapidement les



packages et leurs dépendances ainsi que l'installation de multiples versions de logiciels au travers d'un mécanisme d'environnements virtuels[44].

Adobe XD

C'est une solution d'UX/UI design complète pour la conception de sites web, d'applications mobiles, etc.



C'est un outil performant pour créer des flux utilisateur, des wireframes, des maquettes extrêmement fidèles, des prototypes interactifs, des animations. Il permet aussi de modifier et de partager facilement des prototypes interactifs avec des collaborateurs sur l'ensemble des appareils et des plates-formes[45].

5.2.3 Technologie utilisées

Dans cette section, nous allons présenter les différentes technologies utilisées pour développer notre application web.

5.2.4 HTML

C'est un langage de balisage principal du World Wide Web. À l'origine, HTML était principalement conçu comme un langage pour décrire sémantiquement des documents scientifiques. Sa conception générale lui a toutefois permis de s'adapter, au cours des années suivantes, pour décrire un certain nombre d'autres types de documents et même d'applications[14].



5.2.5 CSS

Les feuilles de style en cascade sont un mécanisme simple pour ajouter du style (par exemple, polices, couleurs, espacement) aux documents Web. Les standards le définissant sont publiés par le World Wide Web Consortium (W3C)[15].



5.2.6 Sass

Sass est un langage de feuille de style compilé en CSS. Il permet d'utiliser des variables, des règles imbriquées, des mixins, des fonctions, etc., Le tout avec une syntaxe entièrement compatible CSS. Il permet aussi de garder les feuilles de style bien organisées et facilite le partage de la conception au sein et entre les projets[46].



5.2.7 JavaScript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web[16].



5.2.8 Python

Python est un langage de programmation puissant et facile à apprendre. Il a des structures de données de haut niveau efficaces et une approche simple, mais efficace de la programmation orientée objet. La syntaxe élégante et le typage dynamique de Python, ainsi que sa nature interprétée, en font un langage idéal pour la création de scripts et le développement rapide d'applications dans de nombreux domaines sur la plupart des plates-formes[17].



5.2.9 Django

Django est un framework Web Python de haut niveau open source qui encourage un développement rapide et une conception propre et pragmatique. Conçu par des développeurs expérimentés, il prend en charge une grande partie des tracas du développement Web. Il permet l'écriture des applications web sans avoir à réinventer la roue[18].



Il est basé sur l' architecture MVT, qui est un modèle de conception de logiciel composé des 3 parties suivantes :

Model : Un modèle est la source d'information unique et définitive à propos des données. Il contient les champs et le comportement essentiels des données stockées. Généralement, chaque modèle correspond à une seule table de base de données[19].

View : Une vue, pour faire simple, est une fonction Python acceptant une requête Web et renvoyant une réponse Web. Cette réponse peut contenir le contenu HTML d'une page Web, une redirection, une erreur 404, un document XML, une image... ou vraiment n'importe quoi d'autre. La vue elle-même contient la logique nécessaire pour renvoyer une réponse[20].

Template : Par sa nature liée au Web, Django a besoin d'un procédé agile de génération dynamique de HTML. L'approche la plus couramment utilisée est de se baser sur des templates. Un template contient la partie statique du résultat HTML souhaité ainsi qu'une certaine syntaxe particulière définissant comment insérer le contenu dynamique[21].

5.3 Persistance des données

Pour les données, Django comprend une couche ORM par défaut qui peut être utilisée pour interagir avec les données d'application de diverses bases de données relationnelles telles que SQLite, PostgreSQL et MySQL.

5.3.1 MySql

MySQL est un système de gestion de bases de données relationnelles SQL open source développé et supporté par Oracle. Son approche relationnelle permet d'organiser les données dans des tableaux à deux dimensions appelés des relations ou tables[47].



5.3.2 ORM

Django ORM est l'un des principaux piliers de Django. Il offre une couche d'abstraction qui permet de travailler avec des bases de données, de manière indépendante. Il combine la facilité d'utilisation en offrant une puissante API qui permet les opérations de base pour la persistance des données : créer, récupérer, mettre à jour, et supprimer facilement des objets. Ainsi, il garde les choses simples faciles et les choses difficiles possibles en offrant la possibilité d'exécuter des requêtes SQL brutes[48].



FIGURE 5.1 – Fonctionnement de l'ORM de Django [49]

5.4 Les librairies utilisées

5.4.1 Bootstrap

C'est le framework css le plus populaire au monde pour créer des sites responsifs et adaptés aux mobiles. Il dispose d'un système de grille réactif, de nombreux composants prédéfinis et de puissants plugins JavaScript[22].



5.4.2 JQuery

jQuery est une bibliothèque JavaScript rapide, petite, et riche en fonctionnalités. Il simplifie considérablement le parcours et la manipulation de documents HTML, la gestion des événements, l'animation grâce à une API facile à utiliser qui fonctionne sur une multitude de navigateurs[23].



5.4.3 Crispy forms

Django crispy-form permet de construire, personnaliser et réutiliser des formulaires en utilisant du css. Il permet d'éviter d'écrire beaucoup de code dans les templates et applique la philosophie DRY[24].

5.4.4 Pillow

C'est une bibliothèque de traitement d'images pour le langage de programmation Python. Elle permet d'ouvrir, de manipuler, et de sauvegarder différents formats de fichiers graphiques[25].



5.4.5 Django import/export

django-import-export est une application et une bibliothèque Django pour l'importation et l'exportation de données avec l'intégration d'administration incluse[26].

5.4.6 Popper

Popper facilite le positionnement des infobulles et des popovers, il positionnera tout élément d'interface utilisateur qui sort du flux d'un document et flotte près d'un élément cible. L'exemple le plus courant est une infobulle. Mais elle comprend également des popovers, des listes déroulantes, etc. Tous ces éléments peuvent être décrits de manière générique comme un élément popper[27].



5.4.7 Font Awesome

C'est la librairie d'icônes la plus populaire d'Internet, elle facilite l'utilisation d'icônes de différentes variantes (Solid, Regular, Light et Brand) en utilisant un préfixe de style et le nom de l'icône[28].



5.4.8 Chart.js

Chart.js est une bibliothèque JavaScript open-source gratuite créée par le développeur Web londonien Nick Downie en 2013. Elle est maintenant maintenue par la communauté, et permet la visualisation de données sur plusieurs types de graphiques tout en étant facile et flexible[29].



5.5 Architecture du système

La figure présente l'architecture de notre système qui est composée en deux parties. La première partie permet d'interagir directement avec la pointeuse biométrique à l'aide d'une api RESTful fournie par notre application Django qui est accessible à travers les endpoint suivants :

Resource	GET	POST	DELETE
pointage/bio	Récupère l'identifiant d'une empreinte.	Envoi l'identifiant d'une empreinte.	Supprime une empreinte.

TABLE 5.1 – Endpoint de l'api RESTful

La deuxième partie c'est l'application web, qui est accessible via un navigateur web, et présentée en détail dans la section suivante.

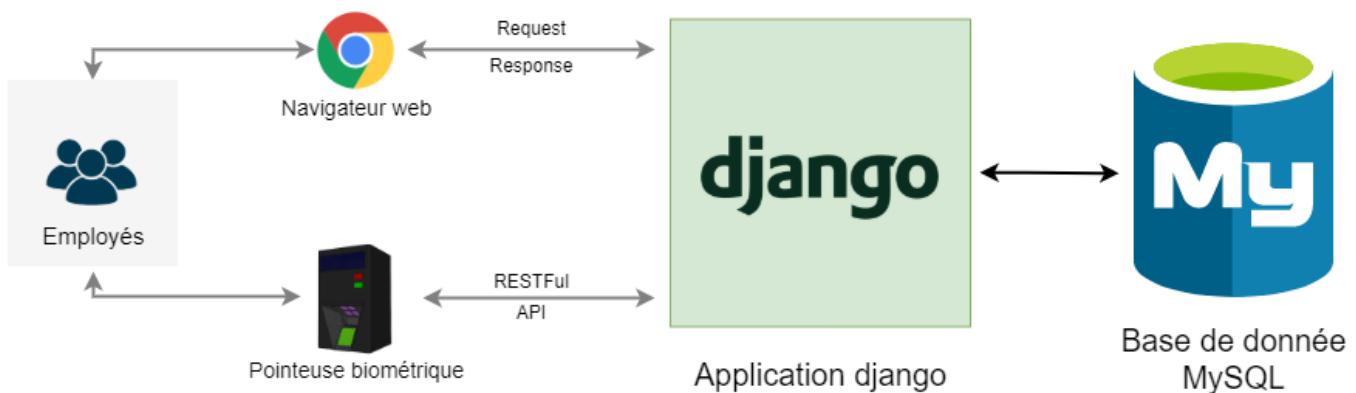


FIGURE 5.2 – Architecture du système

5.6 Présentation de l'application

5.6.1 Nom et logo de l'application

Nous avons décidé de nommer cette application '*ATime*' qu'on peut traduire par *à temps* ou par *au moment* pour faire référence au temps ainsi qu'au pointage et au fait d'arriver à l'heure.



FIGURE 5.3 – Nom de l'application

Pour le logo, nous avons pensé à une forme qui combine une empreinte et une horloge deux concepts primordiaux du projet.



FIGURE 5.4 – Logo avec couleur uniforme



FIGURE 5.5 – Logo sur fond bleu



FIGURE 5.6 – Logo sur fond blanc

5.6.2 Palette de couleurs

En raison de la nature du projet et du fait qu'il soit étroitement lié au domaine de l'entreprise, nous avons choisi des couleurs basiques et pas beaucoup nuancées afin de ne pas provoquer un sentiment de fatigue des yeux lors des longues sessions d'utilisations.

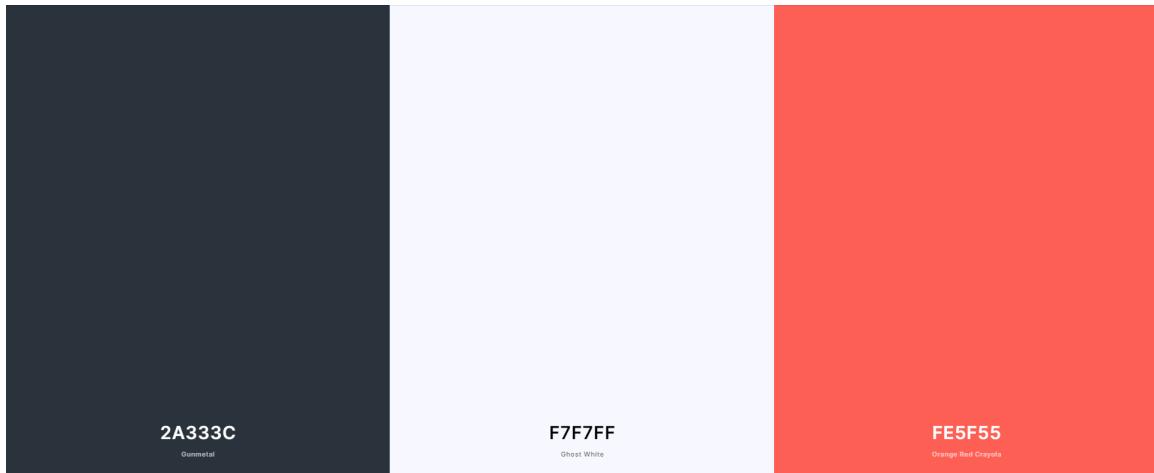


FIGURE 5.7 – Palette de couleur utilisée

5.6.3 Typographie

Le choix de la typographie de notre application repose sur le besoin d'avoir un rendu optimal du texte sur chaque appareil et système d'exploitation. Dans cette démarche, nous avons décidé de garder les polices web par défaut du framework bootstrap.

5.6.4 Squelette de l'application web

Le squelette d'une application web illustre la hiérarchie des différentes pages en les distinguant par niveau de profondeur. Il permet de structurer l'information pour que celle-ci soit claire et facilement accessible.

Dans notre application, le niveau 0 est représenté par la page d'authentification, les autres pages, elles aussi, sont réparties par niveaux en fonction du nombre de clique(s) nécessaire(s) pour se rendre dessus.

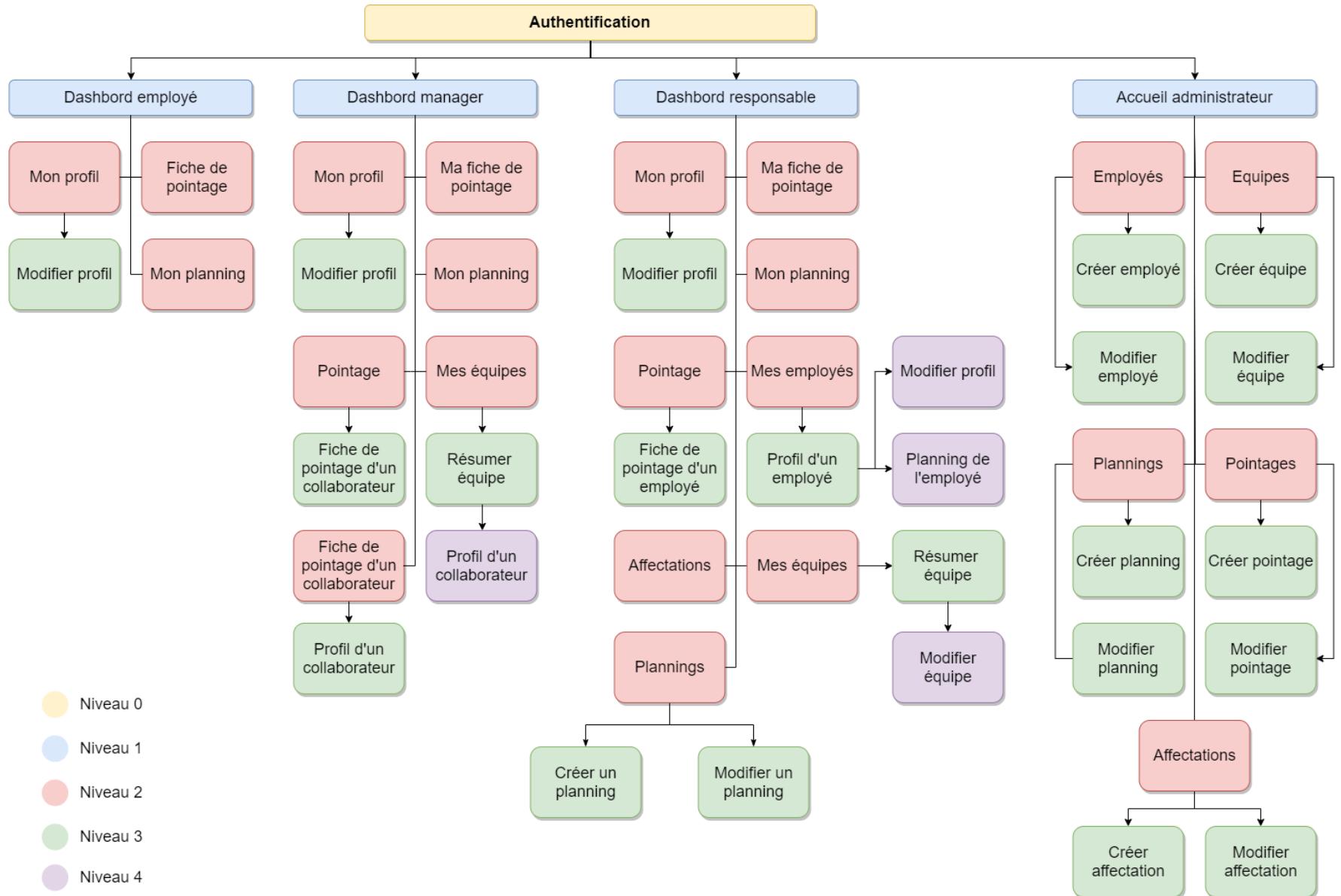


FIGURE 5.8 – Squelette de l’application web

5.7 Présentation des interfaces

Notre application comporte 4 types utilisateurs, afin de présenter les principales interfaces dans les conditions réelles d'utilisation de l'application, nous allons les présenter par espace de chaque acteur.

5.7.1 Espace employé

En plus de l'employé cette espace est partagée par plusieurs acteurs, car il comporte certaines fonctionnalités communes. Ci-dessous nous avons modélisé le parcours de ce dernier à travers les interfaces qui lui sont dédiées.

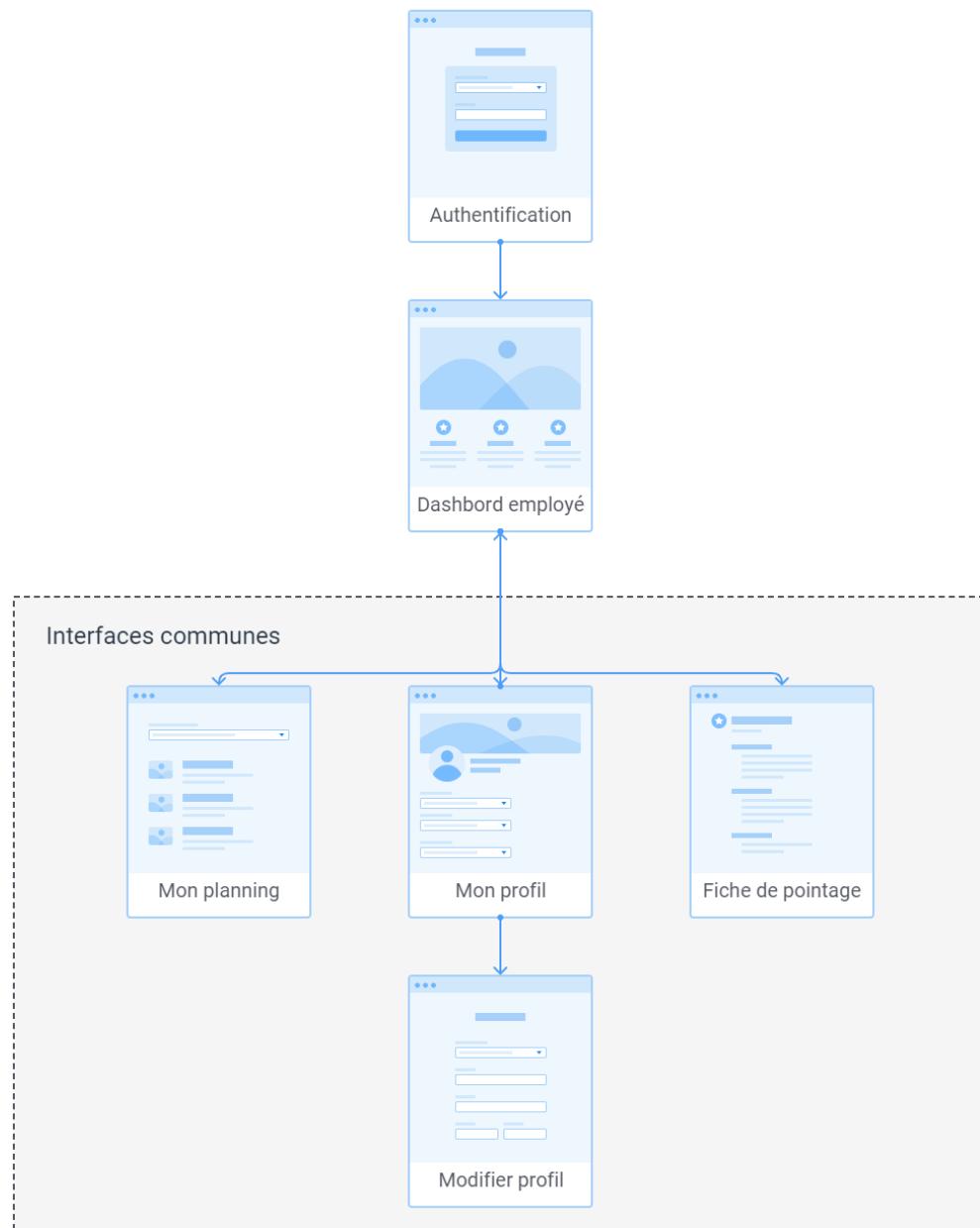


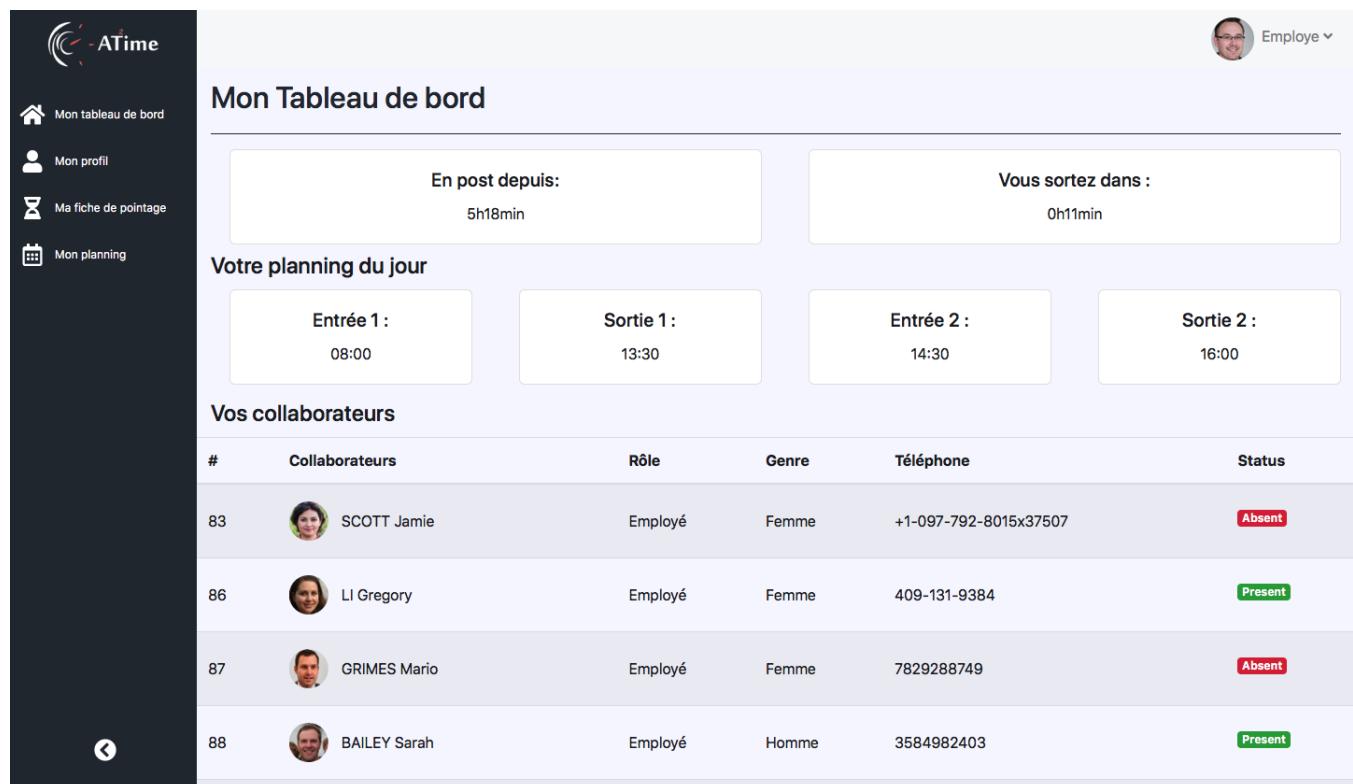
FIGURE 5.9 – User flow de l'employé

Interface dashbord employé

Une fois que l'employé s'authentifie, il sera redirigé vers l'interface *Dashbord employé* sur laquelle il retrouve :

- Le nombre d'heures en poste depuis son dernier pointage.
- Le nombre d'heures restant par rapport à son planning du jour.
- Son planning journalier dans lequel il a une vision précise de ces horaires de travaux.
- Une liste de ces collaborateurs avec les informations les plus utiles telle que le nom, prénom, le rôle au sein de l'entreprise, le genre, le numéro de téléphone professionnel et enfin un statut de présence/absence.

Après avoir consulté le dashbord il pourra utiliser le menu qui se trouve à gauche afin de naviguer à travers les différentes interfaces.



The screenshot shows the ATime employee dashboard. On the left, a dark sidebar contains icons for 'Mon tableau de bord', 'Mon profil', 'Ma fiche de pointage', and 'Mon planning'. At the bottom of the sidebar is a back arrow icon. The main content area has a light background. At the top right is a user profile picture and the text 'Employé'. Below this, the title 'Mon Tableau de bord' is displayed. Two boxes show 'En post depuis: 5h18min' and 'Vous sortez dans: 0h11min'. Under 'Votre planning du jour', four boxes show 'Entrée 1: 08:00', 'Sortie 1: 13:30', 'Entrée 2: 14:30', and 'Sortie 2: 16:00'. The section 'Vos collaborateurs' contains a table with five rows of data:

#	Collaborateurs	Rôle	Genre	Téléphone	Status
83	SCOTT Jamie	Employé	Femme	+1-097-792-8015x37507	Absent
86	LI Gregory	Employé	Femme	409-131-9384	Present
87	GRIMES Mario	Employé	Femme	7829288749	Absent
88	BAILEY Sarah	Employé	Homme	3584982403	Present

FIGURE 5.10 – Interface dashbord employé

Interface ma fiche de pointage

Depuis cette interface, l'employé aura la possibilité de consulter la liste de ses pointages effectués depuis son recrutement au sein de l'entreprise et qui seront regroupés par date. Afin de faciliter la recherche, il aura la possibilité de les filtrer par semaine/mois ainsi que de faire une recherche précise par période.

Date	Heure d'entrée	Heure de sortie	
8 septembre 2020	08:00	None	>
7 septembre 2020	08:00	11:20	>
	12:00	18:00	
6 septembre 2020	08:00	12:00	>
5 septembre 2020	14:00	18:00	>
2 septembre 2020	12:38	18:00	>
1 septembre 2020	10:23	12:38	>
31 août 2020	06:00	12:00	>
30 août 2020	12:40	18:00	>
27 août 2020	06:00	12:00	>

FIGURE 5.11 – Interface ma fiche de pointage

Interface consulter mon profil

Cette interface permet à l'employé de consulter son profil qui est constitué d'informations d'identification de base pour le bon fonctionnement du système, ainsi que de diverses informations le concernant et qui peuvent être renseignées par l'employé lui-même en accédant à l'interface de modification.

FIGURE 5.12 – Interface consulter mon profil

5.7.2 Espace manager

Cet espace qui est dédié au manager, lui permet non seulement de superviser ses collaborateurs et ses équipes, mais aussi de profiter des fonctionnalités partagées avec l'employé, ci-dessous nous avons modélisé le parcours du manager à travers son espace personnel. Pour ce qui est des interfaces communes elles sont modélisées dans la figure 5.13.

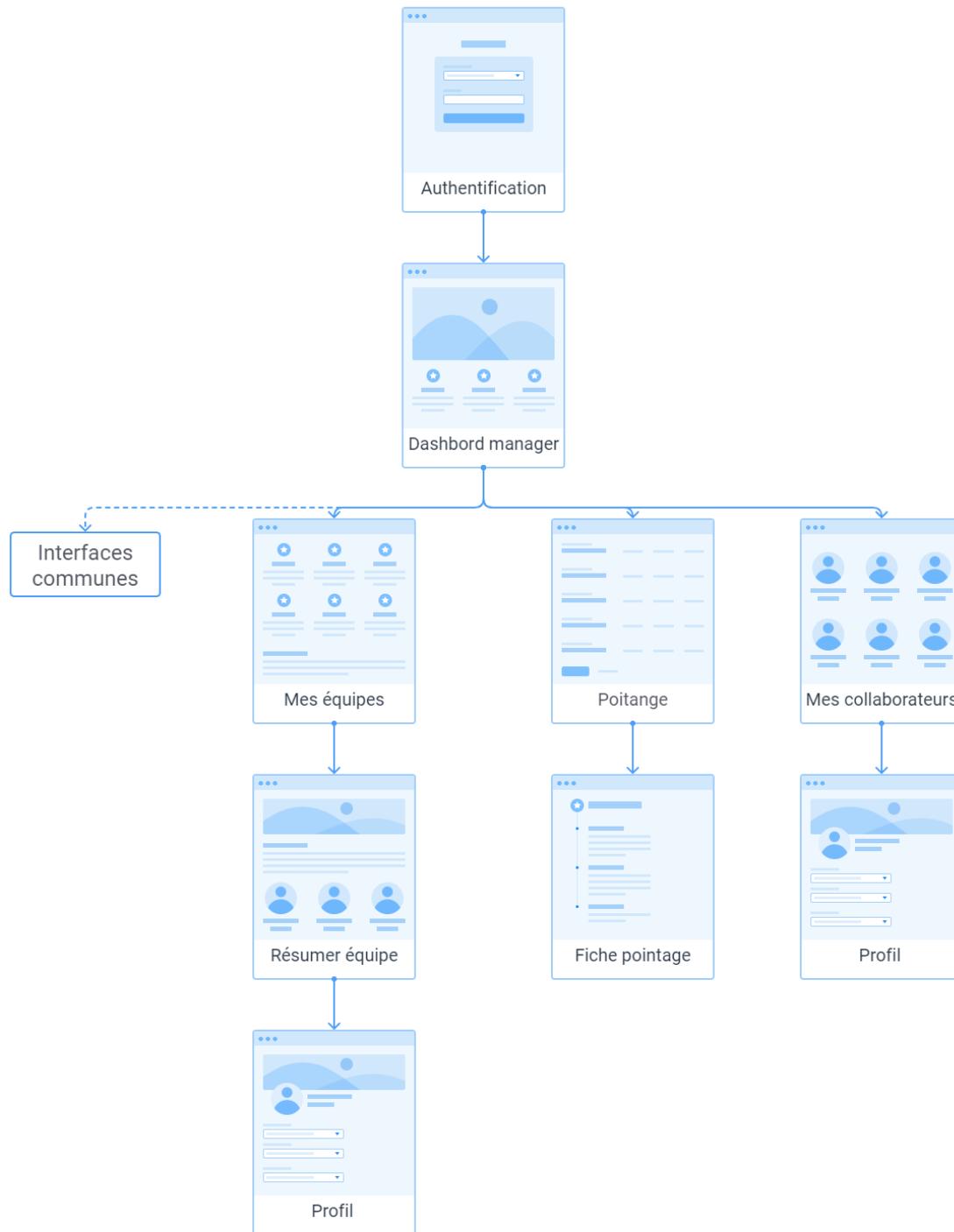


FIGURE 5.13 – User flow du manager

Interface consulter tableau de bord manager

Une fois que le manager s'authentifie il sera redirigé vers l'interface *dashbord manager* qui contient deux onglets, le 1er onglet est identique à celui de l'employé, le deuxième onglet «Mes équipes» résume les informations des collaborateurs de ses différentes équipes.

The screenshot displays the ATime dashboard manager interface, which is a continuation of the employee's dashboard. It features a sidebar on the left with navigation links: Mon tableau de bord, Mon profile, Ma fiche de pointage, Mes équipes, pointage, and Mes collaborateurs. The main content area is divided into two tabs: 'Mon tableau de bord' and 'Mes équipes'. The 'Mon tableau de bord' tab is active, showing the following information:

- En post depuis:** 4h50min
- Vous sortez dans :** 0h9min
- Votre planingue du jours:**
 - Entrée 1:** 08:40
 - Sortie 1:** 13:30
 - Entrée 2:** 14:30
 - Sortie 2:** 18:00
- Vos collaborateurs:** A table listing employees with their details and presence status (Present or Absent).

#	Collaborateurs	Genre	Téléphone	Présence
2	KASEEM Yanis	Homme	+213 38 65 04 55	Present
91	SCOTT Jamie	Femme	+1-097-792-8015x37507	Absent
94	LI Gregory	Femme	409-131-9384	Present
95	GRIMES Mario	Homme	7829288749	Absent
96	DALYX Scott	Homme	6504000100	Present

The 'Mes équipes' tab shows summary statistics for the number of teams (13), gender distribution (9 men, 6 women, total 16 employees), and overall attendance (4 employees in, 12 employees out, total 16 employees). Below this, a grid displays detailed profiles for each employee, including their name, role, and a small icon.

FIGURE 5.14 – Interface dashbord manager

Interface résumé d'une équipe

Après avoir sélectionné une équipe, le manager est redirigé vers l'interface ci-dessous qui lui permet d'avoir une vue d'ensemble, afin de superviser les membres de cette dernière.

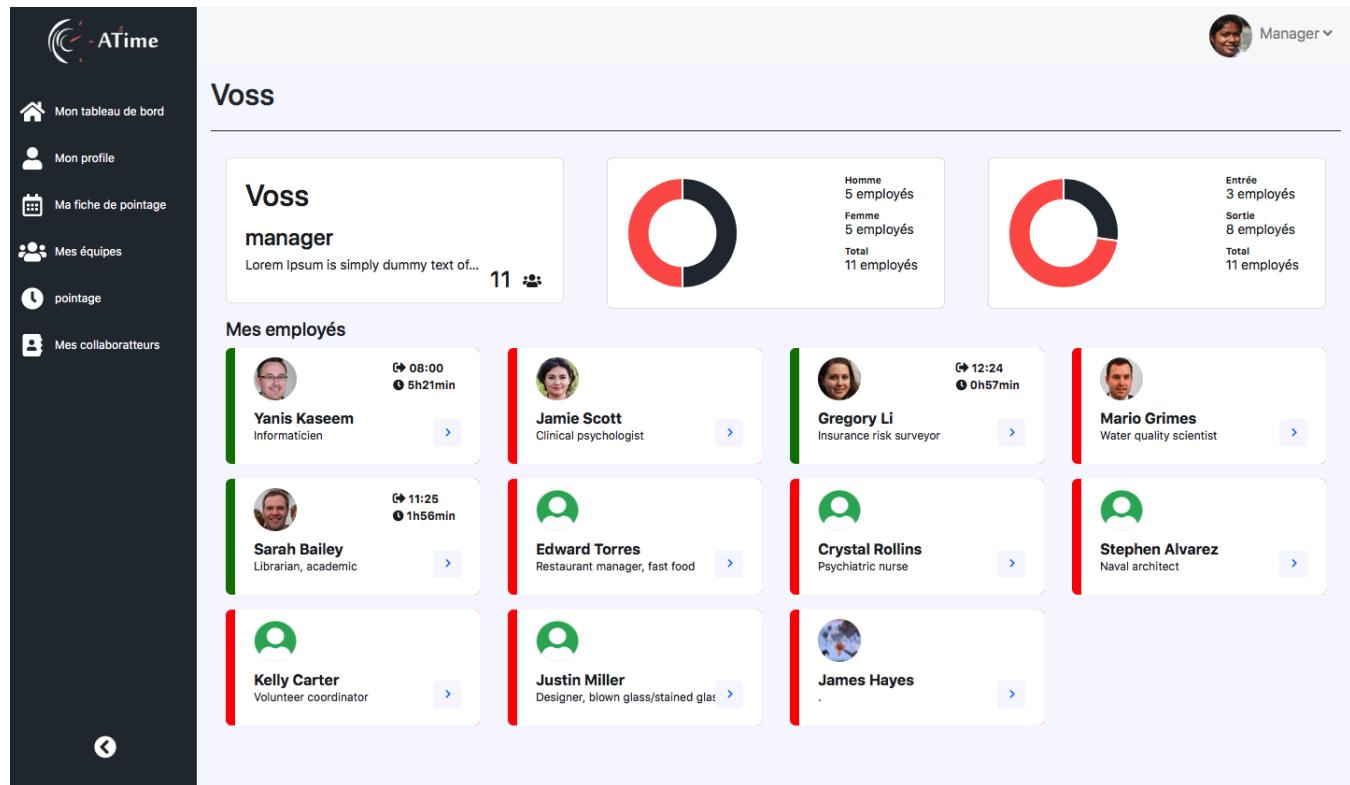


FIGURE 5.15 – Interface résumé d'une équipe

5.7.3 Espace responsable

Cet espace offre au responsable des fonctionnalités avancées avec un accès à la totalité des informations lui permettant de contrôler et d'analyser la situation dans laquelle se trouve l'entreprise, et ce afin de prendre des décisions en vue d'atteindre un ou plusieurs objectifs.

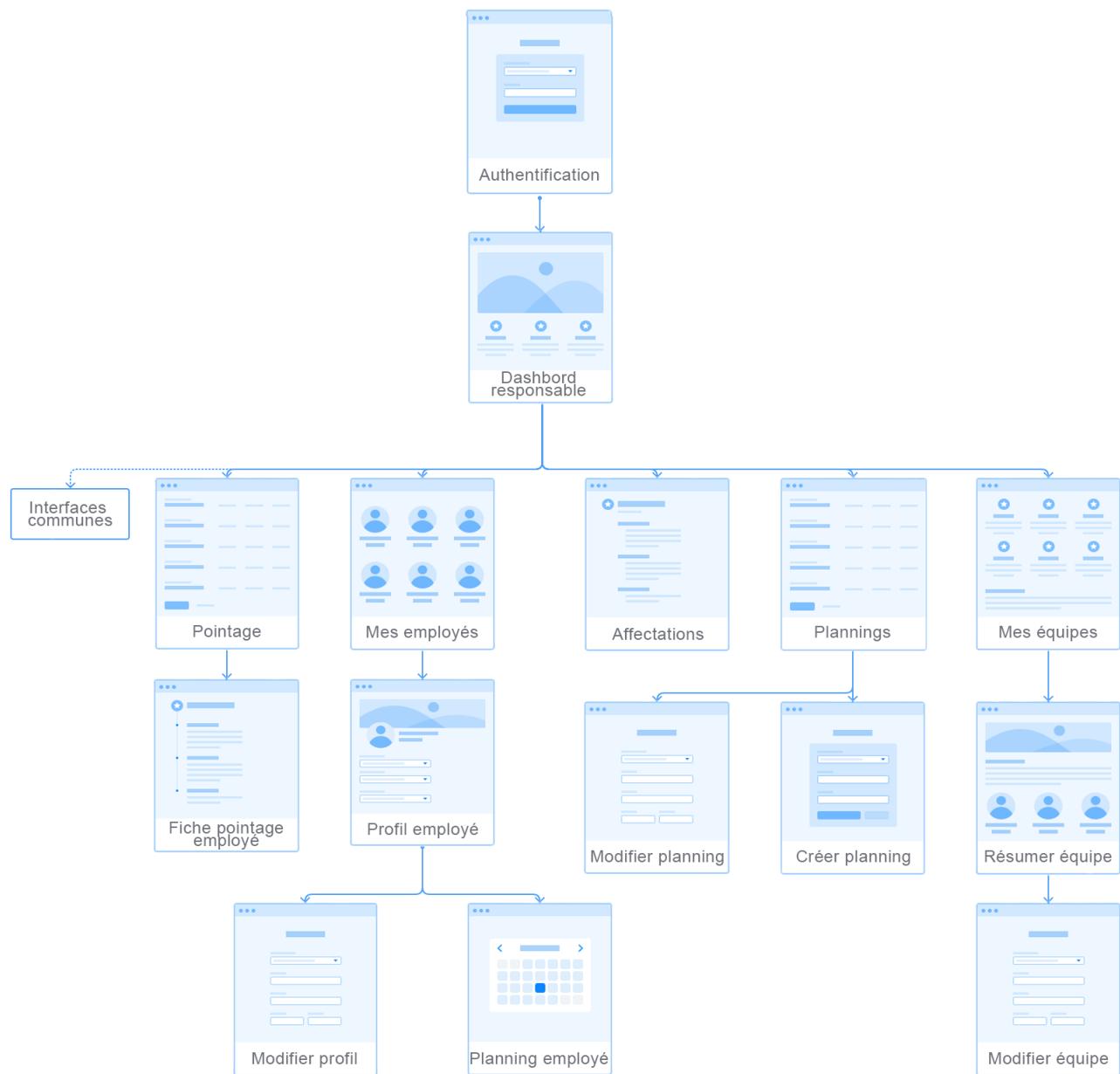


FIGURE 5.16 – User flow du responsable

Interface créer planning

La figure 5.17 présente l'interface qui permet au responsable de créer un planning. C'est un formulaire qui comporte le nom et une description du planning ainsi que les horaires de travail durant la semaine.

Crée un nouveau planning

Titre*

Description

	Parité 1	Parité 2		
Dimanche	Heure d'entrée	Heure de sortie	Heure d'entrée	Heure de sortie
Lundi	Heure d'entrée	Heure de sortie	Heure d'entrée	Heure de sortie
Mardi	Heure d'entrée	Heure de sortie	Heure d'entrée	Heure de sortie
Mercredi	Heure d'entrée	Heure de sortie	Heure d'entrée	Heure de sortie
Jeudi	Heure d'entrée	Heure de sortie	Heure d'entrée	Heure de sortie
Vendredi	Heure d'entrée	Heure de sortie	Heure d'entrée	Heure de sortie
Samedi				

FIGURE 5.17 – interface créer un planning

Interface affecter un employé

Une fois que le responsable sélectionne une équipe, il sera redirigé vers l’interface qui est présentée dans la figure 5.18, elle lui permet de consulter les informations de l’équipe ainsi que d’affecter un employé à cette dernière en effectuant une recherche avec son nom et prénom dans le champ correspondant.

Voss

Nom de l'équipe: Voss

Manager de l'équipe: manager

Déscription de l'équipe: Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s

Ajouter un employé: but

Butler Anthony (AnthoB)

Modifier l'équipe

Supprimer

#	Employé	Status	Genre	Téléphone
2	KASEEM Yanis	Present	Homme	+213 38 65 04 55
91	SCOTT Jamie	Sortie	Homme	+1-097-792-8015x37507
94	LI Gregory	Present	Homme	409-131-9384
95	GRIMES Mario	Sortie	Homme	7829288749
96	BAILEY Sarah	Present	Homme	3584982403

FIGURE 5.18 – Interface affecter un employé

5.7.4 Espace administrateur

Cet espace permet à l'administrateur d'avoir accès de manière structurée à la base de données. Ainsi, il peut assurer la cohérence des données de l'entreprise grâce aux quatre fonctions de base CRUD lui permettant de créer, lire, mettre à jour, et supprimer des données.

Interface ajouter employé

Une fois authentifié, l'administrateur pourra à travers cette interface qui se présente sous la forme d'un formulaire d'ajout d'un employé. Il doit renseigner les informations les plus sensibles pour le bon fonctionnement du système.

Administration de Django

Bienvenue, ADMIN Voir le site / modifier le mot de passe / Déconnexion

Accueil > Pointage > Utilisateurs > Ajouter utilisateur

Ajout de utilisateur

Saisissez tout d'abord un nom d'utilisateur et un mot de passe. Vous pourrez ensuite modifier plus d'options.

Informations du compte

Nom :

Prénom :

Role :

Email :

Nom d'utilisateur :
Requis. 150 caractères maximum. Uniquement des lettres, nombres et les caractères « @, ., _, - » et « _ ».

Mot de passe :
Votre mot de passe ne peut pas trop ressembler à vos autres informations personnelles.
Votre mot de passe doit contenir au minimum 8 caractères.
Votre mot de passe ne peut pas être un mot de passe couramment utilisé.
Votre mot de passe ne peut pas être entièrement numérique.

Confirmation du mot de passe :
Saisissez le même mot de passe que précédemment, pour vérification.

Options avancées (Afficher)

EMPLOYES
Employe: #1

FIGURE 5.19 – User flow de l'administrateur

5.8 Sécurité de l'application

La sécurité des applications Web est devenue un enjeu stratégique plus important que les fonctionnalités ou l'ergonomie. Pour cela, Django nous a permis de mettre en place des mesures de sécurité contre plusieurs types d'attaques.

5.8.1 Protection contre Cross Site Scripting

Le Cross Site Scripting, ou XSS, est la technique d'exploitation des applications Web pour inciter les navigateurs des utilisateurs à exécuter du JavaScript malveillant pour par exemple :

- Changer les mots de passe des utilisateurs à leur insu.
- Collecter des données.
- Exécuter des actions arbitraires.

Django suppose que toutes les données de contexte sont non sécurisées, sauf sur indication contraire. Cela signifie que la plupart des formes d'attaque XSS ne fonctionnent pas avec les modèles Django [18].

5.8.2 Protection contre Cross site request forgery

Les attaques CSRF permettent à un utilisateur malveillant d'exécuter des actions en utilisant les informations d'identification d'un autre utilisateur à son insu ou sans son consentement.

Le middleware CSRF et le template tag csrf token de Django permet une protection facile à mettre en place contre la plupart des types d'attaques CSRF [18].

5.8.3 Protection contre SQL injection

L'injection SQL est un type d'attaque où un utilisateur malveillant est capable d'exécuter du code SQL arbitraire sur une base de données. Cela peut entraîner la suppression d'enregistrements ou une fuite de données.

L'ORM de Django est une protection contre ce type d'attaque, il permet de protéger les requêtes en les construisant à l'aide du paramétrage des requêtes [18].

5.8.4 Protection contre le Clickjacking

Le détournement de clic est un type d'attaque où un site malveillant enveloppe un autre site. Ce type d'attaques se produit lorsqu'un site malveillant piège un utilisateur pour qu'il clique sur un élément caché d'un autre site que le site malveillant a chargé.

Les middlewares et décorateurs de Django fournissent une protection facile à utiliser contre le détournement de cliques [18].

5.9 Tests

La phase de test est une partie fondamentale du processus de développement d'application. Afin de nous assurer de la qualité et fiabilité de notre application, nous avons effectué des tests

tout au long du développement ce qui nous a permis d'avoir une détection précoce des erreurs et de les corriger dès que possible.

5.10 Conclusion

Dans ce chapitre nous avons énoncé les différents logiciels, plateformes, librairies utilisés pour la réalisation de notre application. À savoir le framework Django, et le langage de programmation Python et ses différents packages. Suite à cela, nous avons présenté notre application sous différentes facettes que ce soit le côté persistance de données, la charte graphique, la hiérarchie des pages, les interfaces, et la sécurité.

Conclusion générale et perspectives

Ce travail a été réalisé dans le cadre de notre projet de fin de cycle master en Génie logiciel. Il consiste en l'élaboration d'un système de pointage à empreinte biométrique constitué d'une application web et d'une pointeuse biométrique. Ce système est destiné aux PME voulant avoir un système de gestion de pointage automatisé afin de réduire le coût et le taux d'erreurs liés à la gestion des heures de travail des employés.

Grâce à un processus de développement, nous avons pu effectuer une spécification et une analyse des besoins des utilisateurs, et dégager les principaux acteurs et cas d'utilisation pour réaliser les différents diagrammes, à savoir les diagrammes de cas d'utilisation et de séquence système. Ensuite, nous avons conclu cette phase en produisant des maquettes du système à réaliser. La phase de conception une fois terminée, nous a permis de générer l'ensemble des diagrammes et la documentation nécessaire pour commencer la réalisation de l'application ; à savoir les modèles du domaine, les diagrammes de classes participantes, les diagrammes de séquence, les diagrammes de classes conception, ainsi que le modèle relationnel. Enfin, nous avons abordé la dernière phase aisément grâce au travail effectué dans les phases précédentes. Nous avons pu ainsi utiliser différents outils et plateformes (python, C, Django, MySQL, Bootstrap, SASS, Arduino, etc.) pour implémenter notre solution que nous avons dotée d'une identité graphique reflétant le domaine d'activité de celle-ci.

La réalisation de la partie matérielle s'est faite au fur et à mesure de l'avancement du projet. N'étant pas des professionnels du domaine électronique, nous avons consacré beaucoup de temps pour la maîtrise et l'apprentissage des bases de ce domaine pour élaborer une pointeuse biométrique en combinant différents composants (module ESP32, capteur d'empreinte DY50, écran OLED, etc.)

'AT²ime', est une application web qui permet non seulement au responsable d'avoir accès aux informations de pointages et de présences des employés en temps réel, mais elle permet aussi d'établir des plannings de façon dynamique et de les affecter aux collaborateurs concernés, de déléguer la supervision d'un groupe d'employés grâce au rôle de manager, et d'avoir une base de données contenant toutes les informations pertinentes à propos du personnel en plus de leurs informations de pointage. Tout ceci en identifiant de manière unique et infaillible chaque individu lors de son entrée ou sortie de l'entreprise grâce à la pointeuse.

Malgré tout le travail fourni, nous sommes conscients que plusieurs aspects de notre système peuvent et doivent être améliorés. On peut citer à titre d'exemple l'optimisation du code pour permettre une meilleure exploitation des ressources matérielles. En guise de perspectives, nous aspirons à enrichir l'application web avec d'autres fonctionnalités telles que la génération des bul-

letins de salaire en fonction des heures de travaux, et un système de ticket multicanal. Concernant la pointeuse biométrique, nous souhaitons pouvoir la rendre totalement mobile en y intégrant une batterie rechargeable, et un traqueur GPS dans le but de pouvoir enregistrer la position du pointage en plus de l'identifiant de l'employé ainsi que son horodatage. Ceci serait spécialement utile pour les entreprises ayant des employés mobiles ou des lieux de travail temporaires tels que les chantiers par exemple.

Bibliographie

- [1] IONOS, *Qu'est-ce qu'une application Web ? Définition et exemples*, 07 mars 2019, Disponible sur <<https://www.ionos.fr>>, (consulté le 2 juin 2020).
- [2] *The benefits of web-based applications*, Disponible sur <<https://www.magicwebsolutions.co.uk>>, (consulté le 2 juin 2020).
- [3] FABIEN PARRAIN, *Capteur intégré tactile d'empreintes digitales à microstructures piezorésistives* 2002/12/02, (consulté le 3 juin 2020).
- [4] BHUVAN UNHELKAR, *Software Engineering with UML*, 1^e Auerbach Publications, CRC PRESS, 2018.
- [5] P. ROQUES, *UML 2 Modéliser une Application Web*, 4^e édition, Eyrolles, Paris, 2008.
- [6] PASCAL ROQUES & FRANK VALLÉÉ, *UML 2 en action de l'analyse des besoins à la conception*, 4^e Eyrolles, 2007.
- [7] *UML 2 De l'apprentissage à la pratique*, Disponible sur : <https://laurent-audibert.developpez.com/Cours-UML/?page=mise-en-oeuvre-uml#L9-3-1>, (consulté le 19 juin 2020).
- [8] BOUCHLAGHEM S, CHERIFI F, KHANOUCHE F, MAOUCHÉ L, ZEBBOUDJ S, *Application Web JAVA EE pour la gestion d'un laboratoire de recherche scientifique*, licence en informatique générale, Béjaia, Université A/Mira, 2014, 200p.
- [9] JOSEPH GABAY, DAVID GABAY, *UML 2 Analyse et conception*, Dunod, Paris, 2008.
- [10] BERNHARD RUMPE, *Agile Modeling with UML*, Springer, Aachen, Germany, 2012.
- [11] CHRISTIAN SOUTOU, FRÉDÉRIC BROUARD, *UML 2 pour les bases de données*, 2^e édition, EYROLLES, Paris, France.
- [12] STÉPHANE CROZAT, *Modélisation avancée en UML et en relationnel*, 25 janvier 2018.
- [13] STACKOVERFLOW, *Résultats du sondage auprès des développeurs 2019*, Disponible sur : <<https://insights.stackoverflow.com/survey/2019>>, (consulter le 4 mars 2020).
- [14] WORLD WIDE WEB, *HTML 5.2*, Disponible sur <<https://www.w3.org/TR/2017/REC-html52-20171214/introduction.html#a-quick-introduction-to-html>>, (consulter le 6 mars 2020).
- [15] WORLD WIDE WEB, *WHAT IS CSS ?*, Disponible sur <<https://www.w3.org/Style/CSS/>>
- [16] *JavaScript*, Disponible sur <<https://developer.mozilla.org/fr/docs/Web/JavaScript>>, (consulté le 6 mars 2020).

- [17] *The Python Tutorial*, Disponible sur <<https://docs.python.org/3.9/tutorial/>>, (consulté le 6 mars 2020).
- [18] *Django*, Disponible sur <<https://www.djangoproject.com>>, (consulter le 6 mars 2020).
- [19] *Model*, Disponible sur <<https://docs.djangoproject.com/fr/3.1/topics/db/models/>>, (consulté le 6 mars 2020).
- [20] *View*, Disponible sur <<https://docs.djangoproject.com/fr/3.1/topics/http/views/>>, (consulter le 6 mars 2020).
- [21] *Template*, Disponible sur <<https://docs.djangoproject.com/fr/3.1/topics/templates/>>, (consulter le 6 mars 2020).
- [22] *Bootstrap*, Disponible sur <<https://getbootstrap.com>>, (consultr le 6 mars 2020).
- [23] *JQuery*, Disponible sur <<https://jquery.com>>, (consultr le 6 mars 2020).
- [24] *Crispy-form*, Disponible sur <<https://django-crispy-forms.readthedocs.io/en/latest/>>, (consulté le 6 mars 2020).
- [25] *Pillow*, Disponible sur <<https://pillow.readthedocs.io/en/stable/>>, (consultr le 6 mars 2020).
- [26] *Django import/export*, Disponible sur <<https://django-import-export.readthedocs.io/en/latest/>>, (consulter le 6 mars 2020).
- [27] *Popper*, Disponible sur <<https://popper.js.org>>, (consulter le 6 mars 2020).
- [28] *Font Awesome*, Disponible sur <<https://fontawesome.com>>, (consulter le 6 mars 2020).
- [29] *Chart.js*, Disponible sur <<https://www.chartjs.org>>, (consulter le 6 mars 2020).
- [30] PIERRICK ARLOT, *Les microcontrôleurs STM32 de ST sautent à pieds joints dans l'univers Arduino* [archive], Disponible sur <http://archive.wikiwix.com/cache/index2.php?url=https%3A%2F%2Fwww.lembarque.com%2Fles-microcontroleurs-stm32-de-st-sautent-a-pieds-joints-dans-lunivers-arduino_004896> (Consulté le 08/07/2020)
- [31] FABIEN DANIEAU, *Programmation Arduino en ligne de commande*, Disponible sur : <<http://www.francoistessier.info/blog/2011/07/06/programmation-arduino-en-ligne-de-commande/>>, (Consulté le 13/07/2020).
- [32] JEAN-LUC AUFRANC, *Espressif Systems ESP32 Gets Bluetooth LE 5.0/5.1 Certifications* CNX-Software, 03/01/2020, Disponible sur <<https://www.cnx-software.com/2020/01/03/espressif-systems-esp32-now-supports-bluetooth-le-5-0-5-1/>>, (Consulté le 13/07/2020).
- [33] *ESP32 Datasheet*[archive], Espressif Systems, Disponible sur <<https://www.espressif.com/sites/default/files/documentation/esp32/texttt.pdf>> (Consulté le 23/07/2020).

- [34] *DY50 Datasheet*, Adafruit Optical Fingerprint Sensor, Disponible sur <<https://cdn-learn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf>> (consulté le 10/07/2020).
- [35] BRIAN KERNIGHAN, DENNIS RITCHIE(trad.Thierry Buffenoir, *Le langage C /« The C Programming Language »*], Masson, Paris, 1983, 1er éd, 218 p.
- [36] LADYADA, *AdafruitFingerprint Library Documentation*, Release 1.0, Disponible sur <<https://readthedocs.org/projects/adafruit-circuitpython-fingerprint/downloads/pdf/latest/>> (consulté le 15/07/2020).
- [37] *Slack*, Disponible sur <<https://slack.com/intl/fr-dz/features>> (consulter le 12/04/2020).
- [38] *Trello*, Disponible sur <<https://trello.com/fr>> (consulté le 12/04/2020).
- [40] *Github*, Disponible sur <<https://github.com>> (consulté le 12/04/2020).
- [41] *Discord*, Disponible sur <<https://discord.com>> (consulté le 12/04/2020).
- [42] *Visual Code*, Disponible sur <<https://code.visualstudio.com>> (consulté le 12/04/2020).
- [43] *Git*, Disponible sur <<https://git-scm.com>> (consulté le 12/04/2020).
- [44] *Conda*, Disponible sur <<https://docs.conda.io/en/latest>> (consulté le 12/04/2020).
- [45] *AdobeXD*, Disponible sur <<https://www.adobe.com/fr/products/xd.html>> (consulté le 12/04/2020).
- [46] *SASS*, Disponible sur <<https://sass-lang.com>> (consulté le 12/04/2020).
- [47] *MySql*, Disponible sur <<https://www.mysql.com/fr/>> (consulté le 12/04/2020).
- [48] *ORM*, Disponible sur <<https://docs.djangoproject.com/fr/3.1/topics/db/queries/>> (consulté le 12/04/2020).
- [49] *Les bases de données et Django*, Disponible sur <https://zestedesavoir.com/tutoriels/598/developpez-votre-site-web-avec-le-framework-django/262_presentation-de-django/1519_les-bases-de donnees-et-django/> (consulté le 13/04/2020).

Annexe A

Cas d'utilisation « Consulter mon tableau de bord »

Sommaire d'identification	
Titre	Consulter mon tableau de bord
Acteur	Employé, Manager, Responsable
Résumer	L'acteur accède au tableau de bord qui affiche son pointage et le planning du jour, ainsi qu'une liste des collaborateurs faisant partie de la même équipe que lui.
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	1. Le système affiche le tableau de bord avec les informations correspondantes.
Enchaînement alternatif	
Postconditions	

TABLE 5.2 – Description du cas d'utilisation « Consulter mon tableau de bord »

Cas d'utilisation « Consulter le tableau de bord responsable »

Sommaire d'identification	
Titre	Consulter le tableau de bord responsable
Acteur	Responsable
Résumer	Permet d'afficher un tableau de bord récapitulatif des informations de pointage (du responsable lui-même, des équipes de l'entreprise, de tous les employés).
Description des scénarios	
Pré-conditions	Être authentifié
Scénario nominal	1. Le système affiche un résumé des informations de pointage concernant le responsable.

Enchainement alternatif	<ol style="list-style-type: none"> 1. Le responsable peut avoir une vue par équipe ou il a accès au résumé de pointage de chaque équipe. 2. Le responsable peut avoir une vue par collaborateur dans laquelle on affiche le résumé de pointage de chaque collaborateur.
Postconditions	

TABLE 5.3 – Description du cas d'utilisation « Consulter le tableau de bord responsable »**Cas d'utilisation « Rechercher période »**

Sommaire d'identification	
Titre	Rechercher période
Acteur	Employé, Manager, Responsable
Résumer	L'acteur peut faire une recherche sur les données qui sont liées à une période donnée, en sélectionnant la date du début et de fin de cette dernière.
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. l'acteur choisit une date de début 2. l'acteur choisit une date de fin 3. le système retourne les informations liées à cette période
Enchainement alternatif	<ol style="list-style-type: none"> 3a. Le système ne retrouve aucun résultat lié à cette période 1. le système informe l'acteur qu'aucun résultat n'a été trouvé.
Postconditions	

TABLE 5.4 – Description du cas d'utilisation « Rechercher période »**Cas d'utilisation « Rechercher employé »**

Sommaire d'identification	
Titre	Rechercher employé
Acteur	Manager, Responsable
Résumer	L'acteur peut faire une recherche pour trouver un employé (sous réserve de droit d'accès).
Description des scénarios	
Pré-conditions	Être authentifié.

Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur saisie une chaîne de caractère 2. Le système effectue une recherche dans la liste des employé enregistré en comparant la chaîne saisie avec le nom et le prénom de chaque employé 3. le système retourne le ou les employés dont la chaîne saisie apparaît dans leurs nom ou prénom
Enchainement alternatif	<p>3a. Le système ne retrouve aucun employé dont le nom ou prénom comporte la chaîne saisie.</p> <ol style="list-style-type: none"> 1. le système informe l'acteur qu'aucun employé n'a été trouvé.
Postconditions	

TABLE 5.5 – Description du cas d'utilisation « Rechercher employé »**Cas d'utilisation « Modifier mon profil »**

Sommaire d'identification	
Titre	Modifier mon profil
Acteur	Employé, Manager, Responsable
Résumer	L'acteur accède à une interface qui lui permet de modifier les informations présentes sur son profil (pas toutes les informations)
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur modifie les informations. 2. L'acteur valide les modifications. 3. Le système vérifie la conformité des informations saisie. 4. Le système confirme la modification des informations à l'acteur.
Enchainement alternatif	<p>2a. L'employé ne valide pas les modifications.</p> <ol style="list-style-type: none"> 1. Le système redirige l'employé vers le cas d'utilisations « Consulter mon profil ». <p>3a. L'employé a saisi des informations non conformes à celle exigée par le système.</p> <ol style="list-style-type: none"> 1. Le système met en évidence les champs non valides et demande à l'utilisateur de les modifier , le cas d'utilisations reprend de l'étape 1 du scénario nominale.
Postconditions	Les informations de l'employé sont mises à jour dans la base de données.

TABLE 5.6 – Description du cas d'utilisation « Modifier mon profil »

Cas d'utilisation « Consulter profil d'un collaborateur »

Sommaire d'identification	
Titre	Consulter profil d'un collaborateur
Acteur	Manager
Résumer	Le manager consulte le profil d'un collaborateur(un employé affecter à une équipe dont le manager est responsable.
Description des scénarios	
Pré-conditions	Le manager doit être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur sélectionne un collaborateur (cas d'utilisation <u>Consulter liste des collaborateurs</u>). 2. Le système affiche le profil d'un collaborateur.
Enchainement alternatif	
Postconditions	

TABLE 5.7 – Description du cas d'utilisation « Consulter profil d'un collaborateur »

Cas d'utilisation « Consulter le résumé de pointage des collaborateurs »

Sommaire d'identification	
Titre	Consulter le résumé de pointage des collaborateurs
Acteur	Manager
Résumer	Le manager accède a une interface qui affiche la liste des collaborateurs ainsi que l'heure du dernier pointage ,le temps en poste depuis ce dernier.
Description des scénarios	
Pré-conditions	Le manager doit être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche une liste des collaborateurs appartenant à l'équipe du manager avec leurs informations de pointage le plus récent .
Enchainement alternatif	<ol style="list-style-type: none"> 1a Le manager n'a pas été affecté à une équipe. <ol style="list-style-type: none"> 1. Le système signale au manager qu'il n'est pas responsable d'une équipe.
Postconditions	

TABLE 5.8 – Description du cas d'utilisation « Consulter le résumé de pointage des collaborateurs »

Cas d'utilisation « Consulter feuille de pointage d'un collaborateur »

Sommaire d'identification	
Titre	Consulter feuille de pointage d'un collaborateur
Acteur	Manager
Résumer	Le manager consulte la feuille de pointage d'un collaborateur appartenant à son équipe.
Description des scénarios	
Pré-conditions	Le manager doit être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. Le manager sélectionne un collaborateur (cas d'utilisation <u>Consulter le résumé de pointage des collaborateurs</u>). 2. Le système affiche les informations de pointage du jour actuel et des 6 derniers jours.
Enchainement alternatif	<ol style="list-style-type: none"> 2 Le manager souhaite afficher les informations de pointage par mois. <ol style="list-style-type: none"> 1. Le manager sélectionne le mode d'affichage par mois. 2. Le système affiche les informations de pointage du mois.
Postconditions	

TABLE 5.9 – Description du cas d'utilisation « Consulter feuille de pointage d'un collaborateur »

Cas d'utilisation « Consulter résumé de pointage des employés »

Sommaire d'identification	
Titre	Consulter résumé de pointage des employés
Acteur	Responsable
Résumer	Le responsable accède à un résumé de pointage du jour des tous les employés.
Description des scénarios	
Pré-conditions	Être authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche un tableau des employés ordonné par ordre alphabétique qui contient les informations de pointage du jour.
Enchainement alternatif	<ol style="list-style-type: none"> 1. le responsable peut cliquer sur une ligne du tableau pour afficher les informations de pointage relatives à un employé, voir cas d'utilisations « consulter feuille de pointage individuelle » .
Postconditions	

TABLE 5.10 – Description du cas d'utilisation « Consulter résumé de pointage des employés »

Cas d'utilisation « Consulter feuille de pointage d'un employé »

Sommaire d'identification	
Titre	Consulter feuille de pointage d'un employé
Acteur	Responsable
Résumer	Le responsable consulte la feuille de pointage d'un employé.
Description des scénarios	
Pré-conditions	Le responsable doit être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. Le responsable sélectionne un employé (Cas d'utilisation <u>Consulter résumé de pointage des employés</u>). 2. Le système affiche les informations de pointage du jour actuel et des 6 derniers jours.
Enchainement alternatif	<p>2a Le responsable souhaite afficher les informations de pointage par mois.</p> <ol style="list-style-type: none"> 1. Le manager sélectionne le mode d'affichage par mois. 2. Le système affiche les informations de pointage par mois.
Postconditions	

TABLE 5.11 – Description du cas d'utilisation « Consulter feuille de pointage d'un employé »

Cas d'utilisation « Consulter la liste des collaborateurs »

Sommaire d'identification	
Titre	Consulter la liste des collaborateurs
Acteur	Manager
Résumer	Le manager consulte la liste des collaborateurs appartenant à son/ses équipe(s) qui contient les informations les plus pertinentes(nom d'utilisateur, nom, prénom, statut de présence).
Description des scénarios	
Pré-conditions	Le manager doit être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche la liste des collaborateurs avec leurs informations.

Enchainement alternatif	<p>2a Le manager souhaite afficher les collaborateurs présents.</p> <ol style="list-style-type: none"> 1. Le manager sélectionne le filtre présent. 2. Le système affiche la liste des collaborateurs présents avec leurs informations. <p>2a Le manager souhaite afficher les collaborateurs sortis.</p> <ol style="list-style-type: none"> 1. Le manager sélectionne le filtre sortie. 2. Le système affiche la liste des collaborateurs sorties avec leurs informations.
Postconditions	

TABLE 5.12 – Description du cas d'utilisation « Consulter la liste des collaborateurs »**Cas d'utilisation « Consulter la liste des employés »**

Sommaire d'identification	
Titre	Consulter la liste des employés
Acteur	Responsable
Résumer	Le responsable consulte une liste de tous les employés qui contient les informations les plus pertinentes sur ces derniers.
Description des scénarios	
Pré-conditions	Être authentifié
Scénario nominal	<p>1. Le système affiche la liste de tous les employés avec leurs informations.</p>
Enchainement alternatif	<p>2a Le responsable souhaite afficher les employés présents.</p> <ol style="list-style-type: none"> 1. Le responsable sélectionne l'onglet présent. 2. Le système affiche la liste des employés avec leurs informations. <p>2b Le responsable souhaite afficher les employés sortis.</p> <ol style="list-style-type: none"> 1. Le responsable sélectionne l'onglet sortie. 2. Le système affiche la liste des employés avec leurs informations. <p>2c Le responsable décide d'afficher le profil d'un employé</p> <ol style="list-style-type: none"> 1. On déclenche le cas d'utilisations « Consulter profil d'un employé ».
Postconditions	

TABLE 5.13 – Description du cas d'utilisation « Consulter la liste des employés »

Cas d'utilisation « Modifier un employé »

Sommaire d'identification	
Titre	Modifier employé
Acteur	Responsable, Administrateur
Résumer	L'acteur accède à une interface qui lui permet de modifiée les informations d'un employé.
Description des scénarios	
Pré-conditions	Être authentifier.
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche le formulaire avec les informations de l'employé. 2. L'acteur modifie les informations. 3. Le système vérifie la conformité des informations saisie et demande une confirmation. 4. L'acteur confirme la modification. 5. Le système met à jours les informations de l'employé dans la base de données.
Enchaînement alternatif	<p>3a. Infoemations non conformes.</p> <p>4a. L'acteur annule la modification.</p> <ol style="list-style-type: none"> 1. Le système affiche la liste des employés.
Postconditions	Données mises à jour / ou modifiée.

TABLE 5.14 – Description du cas d'utilisation « Modifier employé »

Cas d'utilisation « Supprimer un employé »

Sommaire d'identification	
Titre	Supprimer un employé
Acteur	Administrateur
Résumer	L'administrateur supprime un employé.
Description des scénarios	
Pré-conditions	Être authentifié

Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur sélectionne un employé à supprimer. 2. L'administrateur demande la suppression. 3. Le système demande une confirmation pour la suppression. 4. L'administrateur confirme la suppression. 5. Le système supprime l'employé de la base de données. 6. le système supprime l'empreinte de l'employé ,voir cas d'utilisation Supprimer empreinte
Enchainement alternatif	<p>4a l'administrateur annule la suppression de l'employé.</p> <ol style="list-style-type: none"> 1. L'administrateur annule la suppression. 2. L'administrateur est redirigé vers la liste des employés, voir cas d'utilisations Consulter liste des employés .
Postconditions	Mise à jour des données présentes dans la base de données.

TABLE 5.15 – Description du cas d'utilisation « Supprimer un employé »**Cas d'utilisation « Ajouter une empreinte »**

Sommaire d'identification	
Titre	Ajouter une empreinte
Acteur	Administrateur, Employé, Pointeuse.
Résumer	L'administrateur ajoute l'empreinte d'un employé.
Description des scénarios	
Pré-conditions	Être authentifier/L'administrateur doit avoir saisi les informations de l'employé.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur active la pointeuse en mode enregistrement. 2. L'employé pose son empreinte sur la pointeuse, qui la scanne. 3. La pointeuse demande de scanner une 2ème fois l'empreinte. 4. L'employé pose son empreinte de nouveau . 5. La pointeuse enregistre l'empreinte. 6. La pointeuse envoie l'identifiant de l'empreinte au système. 7. Le système ajoute l'identifiant à l'employé.
Enchainement alternatif	<p>4a Échec du scan.</p> <ol style="list-style-type: none"> 1. La pointeuse affiche un message d'erreur. 2. Le cas d'utilisation reprend de l'étape 3 du scénario nominal.
Postconditions	Mise à jour des données présentes dans la base de données, ainsi que dans la pointeuse.

TABLE 5.16 – Description du cas d'utilisation « Ajouter une empreinte »

Cas d'utilisation « Supprimer une empreinte »

Sommaire d'identification	
Titre	Supprimer une empreinte
Acteur	Administrateur, Employé, Pointeuse.
Résumer	L'administrateur supprime l'empreinte d'un employé.
Description des scénarios	
Pré-conditions	Être authentifier/empreinte déjà ajouté.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur active la pointeuse en mode suppression. 2. L'administrateur sélectionne l'employé. 3. Le système envoie l'identifiant de l'empreinte. 4. La pointeuse supprime l'empreinte. 5. La pointeuse renvoie une confirmation. 6. Le système confirme la suppression.
Enchainement alternatif	<p>5a Le système renvoi un message d'erreur</p> <ol style="list-style-type: none"> 1. Le cas d'utilisation reprend de l'étape numéro 3 du scénario nominal.
Postconditions	Mise à jour des données présente dans la base de données, ainsi que dans la pointeuse.

TABLE 5.17 – Description du cas d'utilisation « Supprimer une empreinte »

Cas d'utilisation « Consulter la liste de mes équipes»

Sommaire d'identification	
Titre	Consulter la liste de mes équipes
Acteur	Manager
Résumer	Le manager consulte la liste des équipes dont il est responsable.
Description des scénarios	
Pré-conditions	Le manager doit être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche la liste des ses équipes avec pour chaque équipe le nombre des collaborateurs ainsi que le nombre des collaborateurs présents.

Enchainement alternatif	<p>1 Le manager décide de sélectionner une équipe afin d'avoir les informations détaillés grâce au cas d'utilisation "Consulter résumer de mon équipe" .</p> <p>1 Le manager n'est responsable d'aucune équipe pour le moment .</p> <p>1. Le système notifie le manager que il n'a aucune équipe sous sa responsabilité.</p>
Postconditions	

TABLE 5.18 – Description du cas d'utilisation « Consulter la liste de mes équipes »**Cas d'utilisation « Consulter le résumé de mon équipe»**

Sommaire d'identification	
Titre	Consulter le résumé de mon équipe
Acteur	Manager
Résumer	Le manager accède a une interface affichant des informations détaillés sur son équipe.
Description des scénarios	
Pré-conditions	Le manager doit être authentifié.
Scénario nominal	<p>1. Le système affiche le nom de l'équipe, sa description , le nombre de collaborateurs .</p> <p>2. Le système affiche la liste des membres de l'équipe.</p>
Enchainement alternatif	
Postconditions	

TABLE 5.19 – Description du cas d'utilisation « Consulter le résumé de mon équipe »**Cas d'utilisation « Consulter la liste des équipes »**

Sommaire d'identification	
Titre	Consulter la liste des équipes
Acteur	Responsable
Résumer	Le responsable consulte une liste de toutes les équipes existantes dans le système.
Description des scénarios	
Pré-conditions	Être authentifié

Scénario nominal	1. Le système affiche la liste des équipes existantes ainsi que les informations pertinentes de chaque équipe comme le titre de l'équipe et le manager ainsi que le nombre d'employés appartenant à cette équipe.
Enchainement alternatif	2a Le responsable souhaite afficher les informations d'une équipe en détail. 1. Le responsable sélectionne une équipe. 2. Le système affiche une interface contenant les informations détaillées de l'équipe sélectionnée, cas d'utilisations « Consulter résumé d'une équipe ».
Postconditions	

TABLE 5.20 – Description du cas d'utilisation « Consulter la liste des équipes »**Cas d'utilisation « Consulter le résumé d'une équipe »**

Sommaire d'identification	
Titre	Consulter le résumé d'une équipe
Acteur	Responsable
Résumer	Le responsable accède à une interface affichant des informations détaillés sur une équipe.
Description des scénarios	
Pré-conditions	Le manager doit être authentifié.
Scénario nominal	1. Le système affiche le nom de l'équipe, sa description , le nombre de collaborateurs . 2. Le système affiche la liste des membres de l'équipe.
Enchainement alternatif	3 Le responsable décide de supprimer l'équipe en question. 1. Voir le cas d'utilisation « supprimer une équipe »
Postconditions	

TABLE 5.21 – Description du cas d'utilisation « Consulter le résumé d'une équipe »**Cas d'utilisation « Modifier équipe »**

Sommaire d'identification	
Titre	Modifier équipe
Acteur	Responsable, Administrateur

Résumer	L'acteur accède à une interface qui lui permet de modifier les informations d'une équipe.
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche le formulaire avec les informations de l'équipe. 2. L'acteur modifie les informations. 3. Le système vérifie la conformité des informations saisies et demande une confirmation. 4. L'acteur confirme la modification. 5. Le système affiche l'interface qui permet d'ajouter des membres à l'équipe (voir le cas d'utilisations <u>Ajouter membre</u>).
Enchaînement alternatif	<p>2a. Le nom de l'équipe est déjà existant.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur pour signaler que le nom de l'équipe existe dans la base de données. 2. Le cas d'utilisation reprend de l'étape 1 du scénario nominal <p>4a. L'acteur annule la modification.</p> <ol style="list-style-type: none"> 1. Le système affiche l'interface liste des équipes (voir le cas d'utilisations <u>Consulter liste des équipes</u>).
Postconditions	

TABLE 5.22 – Description du cas d'utilisation « Modifier équipe »**Cas d'utilisation « Supprimer équipe »**

Sommaire d'identification	
Titre	Supprimer équipe
Acteur	Responsable, Administrateur
Résumer	L'acteur supprime une équipe.
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur sélectionne une équipe. 2. L'acteur supprime l'équipe sélectionnée. 3. Le système demande une confirmation de suppression. 4. L'acteur confirme la suppression. 5. Le système affiche l'interface liste des équipes (voir le cas d'utilisations <u>Consulter liste des équipes</u>).

Enchaînement alternatif	4a. L'acteur annule la suppression. 1. Le système affiche l'interface liste des équipes (voir le cas d'utilisations <u>Consulter liste des équipes</u>).
Postconditions	

TABLE 5.23 – Description du cas d'utilisation « Supprimer équipe »**Cas d'utilisation « Supprimer membre »**

Sommaire d'identification	
Titre	Supprimer membre
Acteur	Responsable, Administrateur
Résumer	L'acteur accède à une interface qui lui permet de supprimer des employés de l'équipe'.
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	1. L'acteur sélectionne une équipe. 2. L'acteur sélectionne un employé et le supprime. 3. le système demande un confirmation de suppression. 4. l'acteur confirme la suppression. 5. Le système affiche l'interface de suppression avec les données mis à jours.
Enchaînement alternatif	2a. Suppression annulée. 1. Le système affiche l'interface de suppression.
Postconditions	

TABLE 5.24 – Description du cas d'utilisation « Supprimer membre »**Cas d'utilisation « Consulter liste des plannings »**

Sommaire d'identification	
Titre	Consulter liste des plannings
Acteur	Responsable
Résumer	Le responsable consulte la liste des plannings, avec un résumé des informations pour chaque planning (Nom, nombre d'heures, nombre d'employés affectés).
Description des scénarios	

Pré-conditions	Être authentifié
Scénario nominal	1. Le système affiche la liste des différents plannings.
Enchainement alternatif	<p>1 Aucun planning.</p> <p>1. Le système affiche un message pour indiquer au responsable qu'il n'existe aucun planning.</p> <p>2a Le responsable souhaite afficher les informations d'un planning en détail.</p> <p>1. Le responsable sélectionne un planning.</p> <p>2. Le système affiche une interface contenant les informations détaillées du planning sélectionné, cas d'utilisations « Consulter un planning ».</p>
Postconditions	

TABLE 5.25 – Description du cas d'utilisation « Consulter liste des plannings »

Cas d'utilisation « Consulter mon planning »

Sommaire d'identification	
Titre	Consulter mon planning
Acteur	Employé, Manager, Responsable
Résumer	L'acteur accède aux informations relatives à son planning (Heures supposé d'arrivée et de sortie prévue par l'entreprise)
Description des scénarios	
Pré-conditions	Être authentifier.
Scénario nominal	1. Le système affiche le planning de l'acteur concerné pour une semaine, ce dernier comporte les jours de travail et de congé ainsi que les horaires d'entrée , sortie et pauses.
Enchainement alternatif	<p>1a. L'acteur ne possède pas un planning.</p> <p>1. Le système notifie l'acteur qu'aucun planning ne lui a été attribué.</p>
Postconditions	

TABLE 5.26 – Description du cas d'utilisation « Consulter mon planning »

Cas d'utilisation « Consulter planning d'un collaborateur »

Sommaire d'identification	
Titre	Consulter planning d'un collaborateur

Acteur	Manager
Résumer	Le manager consulte le planning d'un collaborateur.
Description des scénarios	
Pré-conditions	Le manager doit être authentifié.
Scénario nominal	<p>1. Le système affiche le planning d'un collaborateur (les jours de travail et de repos pendant une semaine ainsi que les horaires supposé d'arriver et de sortis .</p>
Enchainement alternatif	
Postconditions	

TABLE 5.27 – Description du cas d'utilisation « Consulter planning d'un collaborateur »**Cas d'utilisation « Consulter le planning d'un employé »**

Sommaire d'identification	
Titre	Consulter planning
Acteur	Responsable
Résumer	Le responsable consulte le planning.
Description des scénarios	
Pré-conditions	Le responsable doit être authentifié.
Scénario nominal	<p>1. Le système affiche le planning (les jours de travail et de repos pendant une semaine ainsi que les horaires supposé d'arriver et de sortis.</p> <p>2. L'acteur peut modifier le planning (voir le cas d'utilisations <u>Modifier planning</u>).</p>
Enchainement alternatif	
Postconditions	

TABLE 5.28 – Description du cas d'utilisation « Consulter planning d'un employé »**Cas d'utilisation « Modifier un planning »**

Sommaire d'identification	
Titre	Modifier un planning
Acteur	Responsable, Administrateur

Résumer	L'acteur accède à une interface qui lui permet de modifier le planning.
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche le formulaire avec les informations du planning. 2. L'acteur modifie les informations. 3. Le système vérifie la conformité des informations saisies et demande une confirmation. 4. L'acteur confirme la modification. 5. Le système affiche un message pour confirmer la modification. 6. Le système affiche la liste des plannings voir cas d'utilisation <u>Consulter liste des plannings</u>
Enchaînement alternatif	<ol style="list-style-type: none"> 2a. Le nom de l'équipe est déjà existant. <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur pour signaler que le nom de l'équipe existe dans la base de données. 2. Le cas d'utilisation reprend de l'étape 1 du scénario nominal 4a. L'acteur annule la modification. <ol style="list-style-type: none"> 1. Le système affiche l'interface liste des équipes (voir le cas d'utilisations <u>Consulter liste des équipes</u>).
Postconditions	

TABLE 5.29 – Description du cas d'utilisation « Modifier un planning »

Cas d'utilisation « Supprimer un planning »

Sommaire d'identification	
Titre	Supprimer un planning
Acteur	Responsable, Administrateur
Résumer	L'acteur supprime une équipe.
Description des scénarios	
Pré-conditions	Être authentifié.

Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur sélectionne une équipe. 2. L'acteur supprime l'équipe sélectionnée. 3. Le système demande une confirmation de suppression. 4. L'acteur confirme la suppression. 5. Le système affiche l'interface liste des équipes (voir le cas d'utilisations <u>Consulter liste des équipes</u>).
Enchaînement alternatif	<ol style="list-style-type: none"> 4a. L'acteur annule la suppression. 1. Le système affiche l'interface liste des équipes (voir le cas d'utilisations <u>Consulter liste des équipes</u>).
Postconditions	

TABLE 5.30 – Description du cas d'utilisation « Supprimer un planning»**Cas d'utilisation « Affectation d'un planning »**

Sommaire d'identification	
Titre	Affectation d'un planning
Acteur	Responsable, Administrateur
Résumer	L'acteur accède à une interface qui lui permet d'affecter le planning à un employé'.
Description des scénarios	
Pré-conditions	Être authentifié.
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur recherche un employé (voir le cas d'utilisations « Recherche employés »). 2. Le système retourne une liste d'employés. 3. L'acteur sélectionne un employé et l'ajoute. 4. Le système affiche l'interface d'affectation avec les données mis à jours.
Enchaînement alternatif	
Postconditions	

TABLE 5.31 – Description du cas d'utilisation « Affectation d'un planning »

Cas d'utilisation « Consulter le journal des affectations »

Sommaire d'identification	
Titre	Consulter le journal des affectations
Acteur	Responsable
Résumer	Permet d'afficher un tableau regroupant les changements d'équipe pour les employés.
Description des scénarios	
Pré-conditions	Être authentifier
Scénario nominal	1. Le système affiche les différentes affectations des employés selon les équipes, ainsi on peut savoir si un employé a changé d'équipe à travers sa carrière dans l'entreprise.
Enchainement alternatif	
Postconditions	

TABLE 5.32 – Description du cas d'utilisation « Consulter le journal des affectations »

Cas d'utilisation « Importer/Exporter »

Sommaire d'identification	
Titre	Importer/Exporter
Acteur	Responsable
Résumer	Permet au responsable d'importer ou d'exporter des informations de pointage.
Description des scénarios	
Pré-conditions	Être authentifier
Scénario nominal	1. Le responsable choisit d'importer/exporter 2. Le responsable choisit le format. 3. Importation ou exportations des données selon le format choisi.
Enchainement alternatif	1 Erreur dans les données sélectionnées lors de l'importation 1. Le système signale l'erreur. 2. Le cas d'utilisation redémarre de l'étape numéro 2.
Postconditions	Mise à jour de la BDD en cas d'importations / Création d'un fichier contenant les données exporter dans le format voulu.

TABLE 5.33 – Description du cas d'utilisation « Importer/Exporter »

Annexe B

Cas d'utilisation « Consulter résumé de pointage des collaborateur »

Ce cas est utilisé dans le bute d'avoir accès au informations de pointage des collaborateur, à partir d'une liste qui contient le nom, prénom et le nom d'utilisateur ainsi que l'heure du dernier pointage et le statu de présence.

Le manager pourras ensuite sélectionner un collaborateur pour avoir accès a la totalité de ses informations de pointage de manière détaillée.

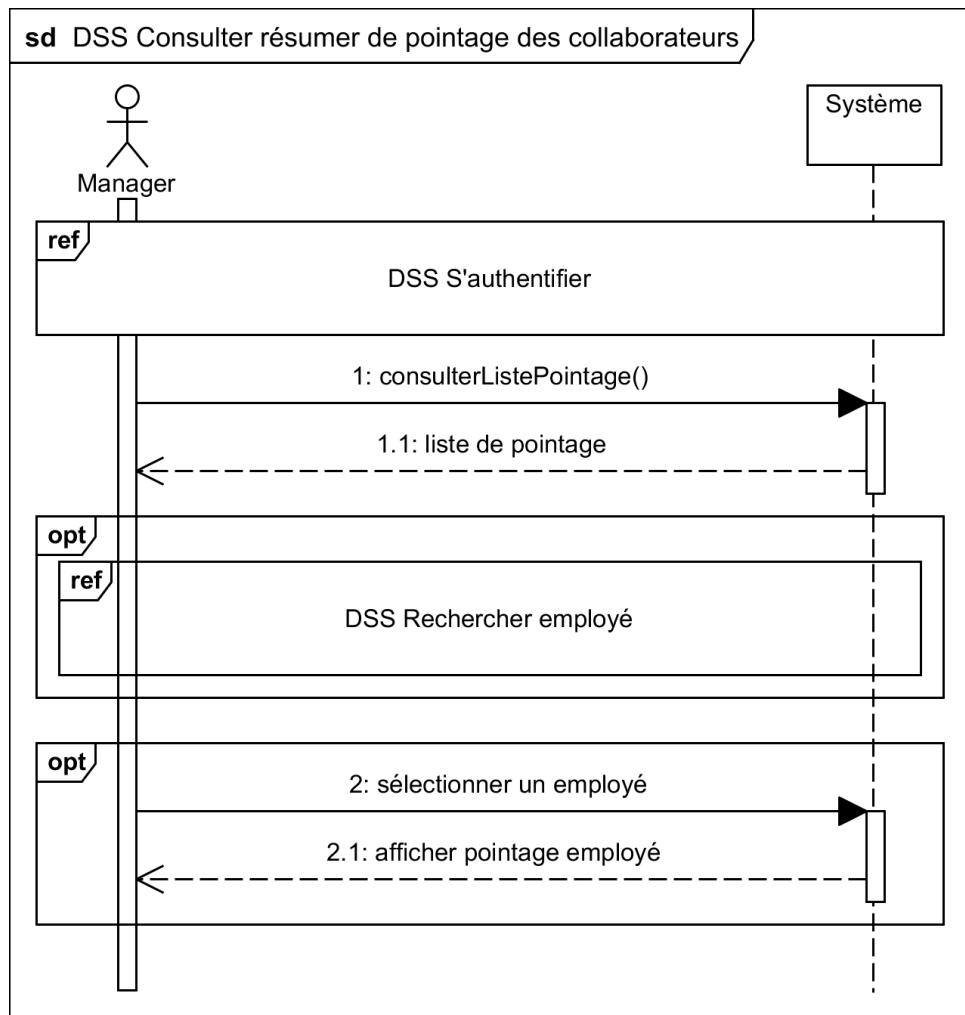


FIGURE 5.20 – Diagramme de séquence système « Consulter résumé de pointage des collaborateur »

Cas d'utilisation « Consulter la liste des collaborateurs »

Un manager peut consulter la liste de ses collaborateurs (employés faisant partie d'une équipe dont il est responsable). A partir de cette liste il peut consulter le profil d'un collaborateur et avoir plus d'informations en cas de besoins.

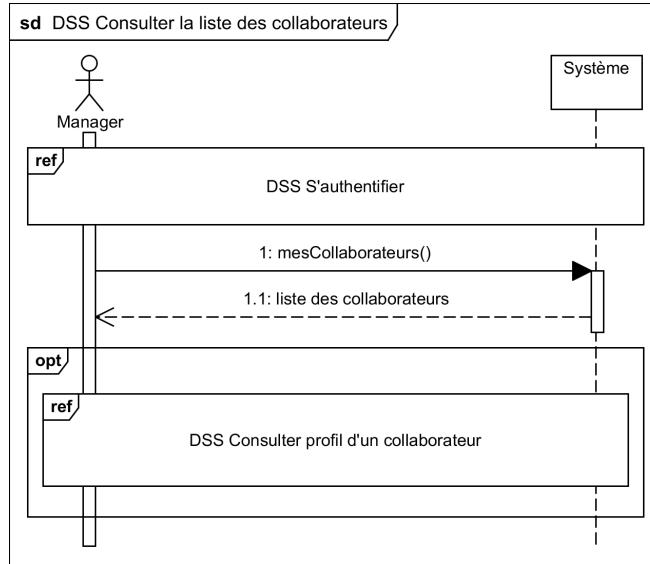


FIGURE 5.21 – Diagramme de séquence système « Consulter la liste des collaborateurs »

Cas d'utilisation « Consulter mes équipes »

Un manager peut avoir une ou plusieurs équipes, donc il aura besoin d'accéder à une liste de celles-ci et accéder aux détails d'une équipe après l'avoir sélectionnée.

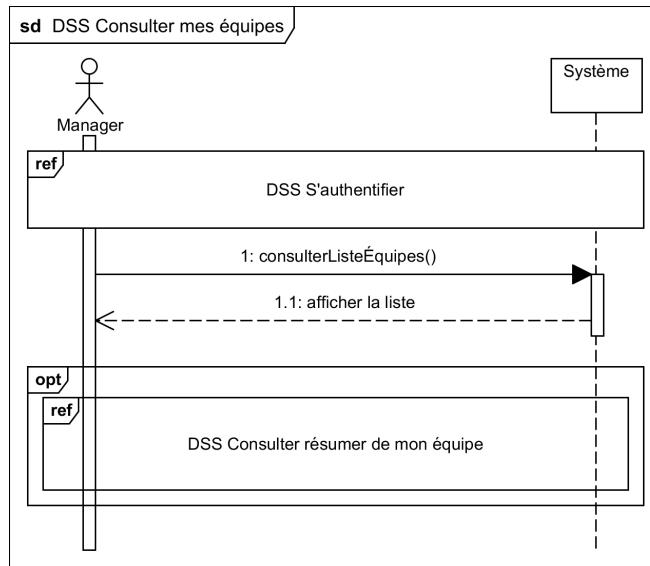


FIGURE 5.22 – Diagramme de séquence système « Consulter mes équipes »

Cas d'utilisation « Consulter mon profil »

L'employé peut décider de consulter son profil et avoir accès au différentes informations qui les constitue en modifier certaines ou accéder à son planning.

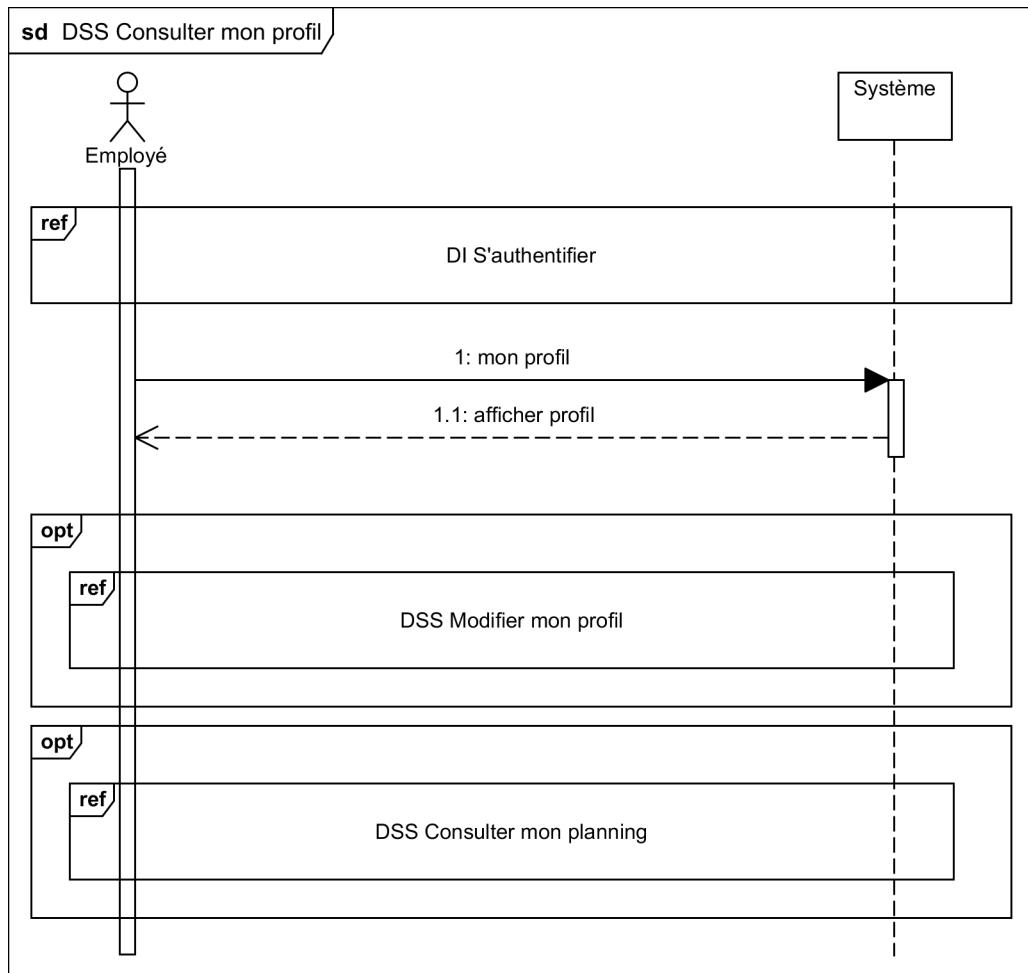


FIGURE 5.23 – Diagramme de séquence système « Consulter mon profil »

Cas d'utilisation « Modifier mon profil »

Un employé peut modifier des informations de son profil sauf le nom, prénom et le nom d'utilisateur, qui sont définie par l'administrateur lors de l'enregistrement de l'employé en question dans le système.

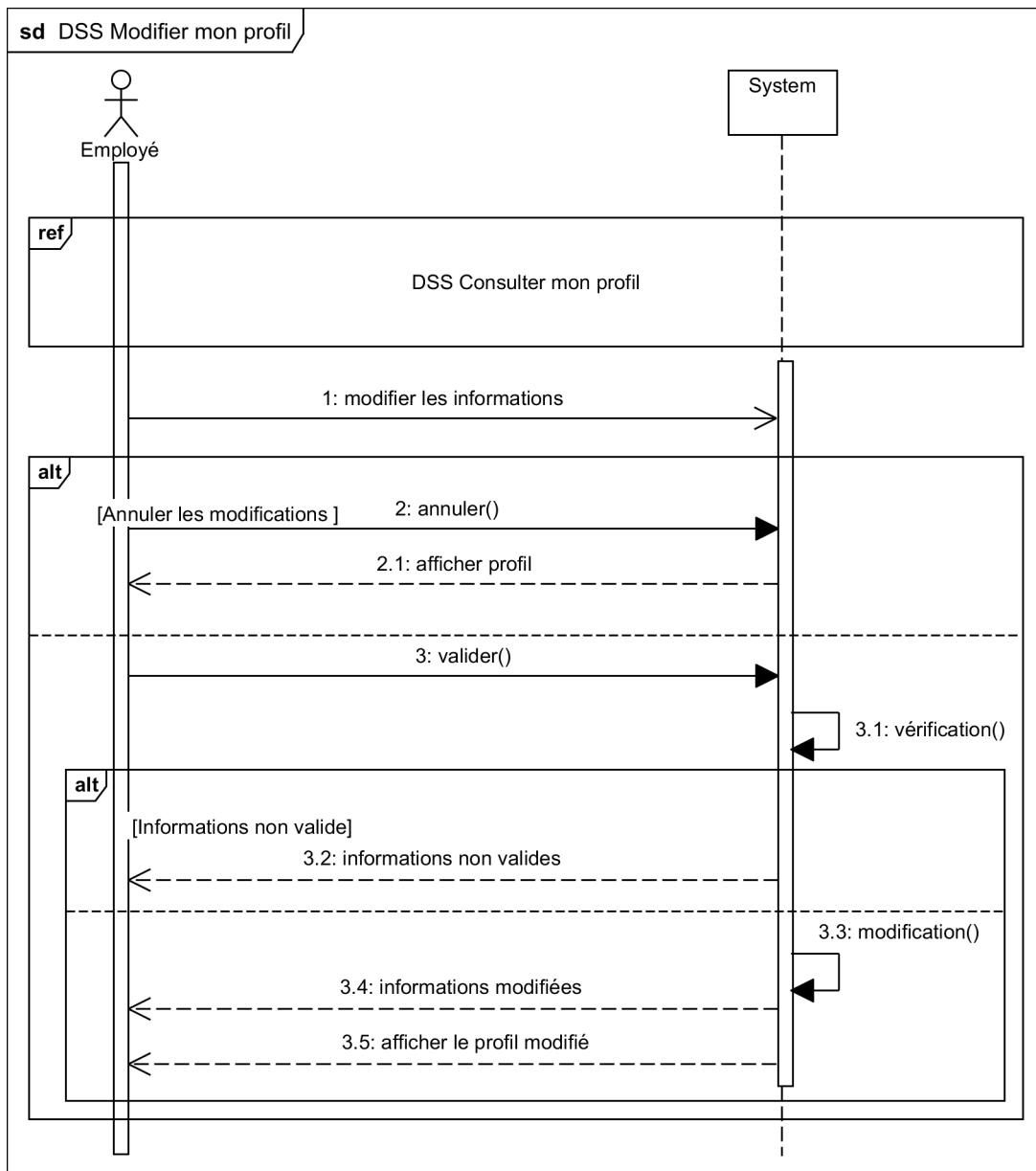


FIGURE 5.24 – Diagramme de séquence système « Modifier mon profil »

Cas d'utilisation « Affectation d'un planning »

Le responsable a la possibilité d'affecter des plannings a tous les employés. Une fois le planning créé il peut l'attribuer a des employés pour que ils soient au courant des horaires de travail.

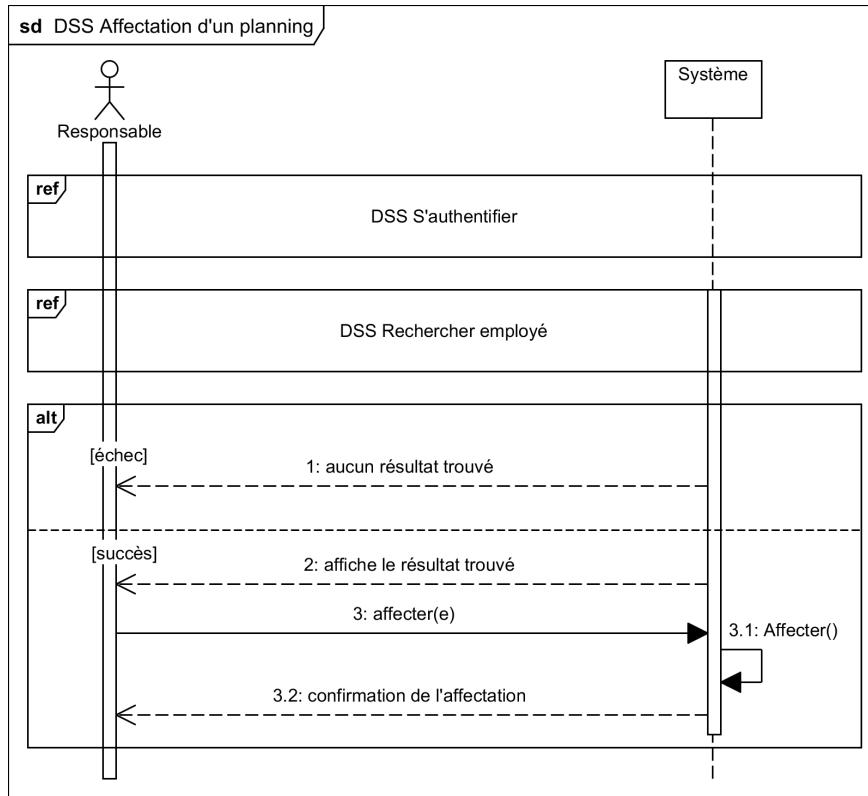


FIGURE 5.25 – Diagramme de séquence système « Affectation d'un planning »

Cas d'utilisation « Consulter liste des équipes »

Grâce à ce cas une responsable consulte la liste des équipes existantes avec l'intitulé et le nom, prénom du manager de chaque équipe. A partir de cette liste il peut gérer les équipes ou consulter les informations détaillées à-propos d'une équipe une fois sélectionnée.

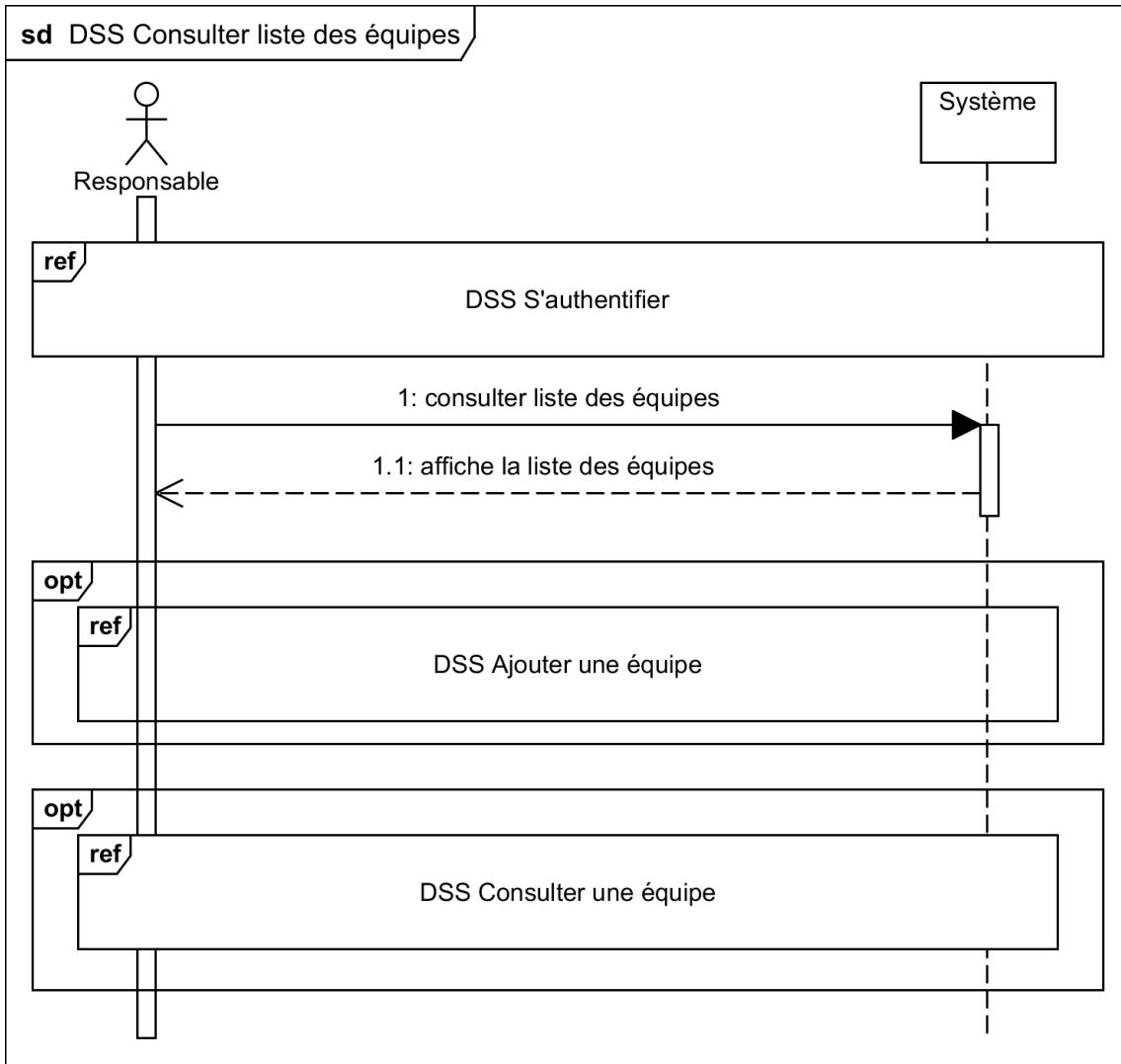


FIGURE 5.26 – Diagramme de séquence système « Consulter liste des équipes »

Cas d'utilisation « Ajouter empreinte »

Lorsque un employé intègre l'entreprise.l'administrateur lui crée un compte dans le système et saisit les informations principales pour ensuite ajouter son empreinte dans la pointeuse.Une opération décrite dans ce cas d'utilisation.En premier lieu il active le mode enregistrement de la pointeuse et demande a l'employé de posé son index droit une premier fois le retire et le place de nouveau pour confirmation.Une fois l'empreinte enregistrée l'administrateur est notifier du bon déroulement de l'opération.

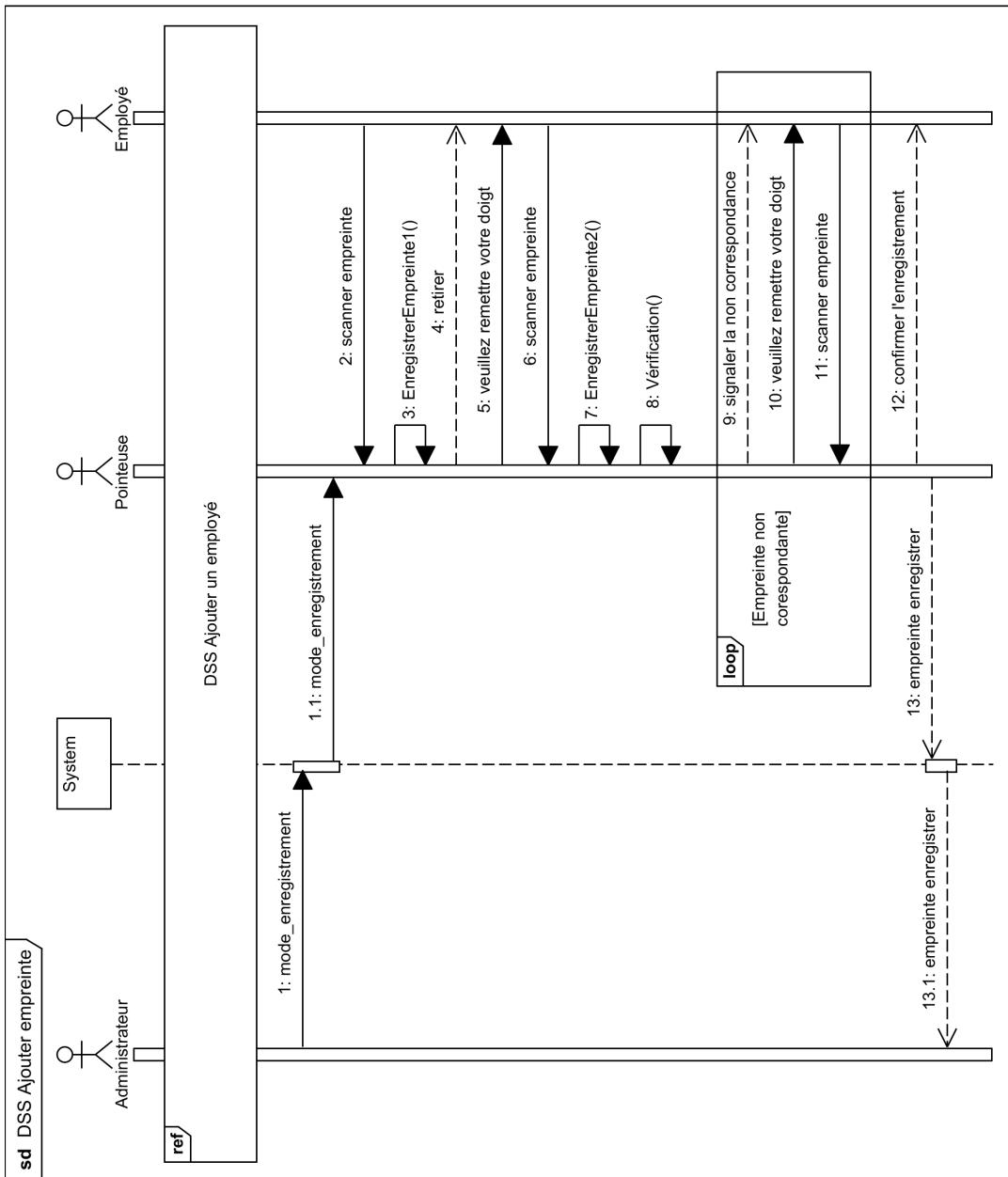


FIGURE 5.27 – Diagramme de séquence système « Ajouter empreinte »

Cas d'utilisation « Supprimer empreinte »

Ce cas d'utilisation est déclenché lorsque l'administrateur supprime un employé. La pointeuse passe en mode suppression et récupère l'identifiant de l'empreinte à supprimé puis la supprime en confirmant la suppression au système qui confirme la suppression au responsable à son tour.

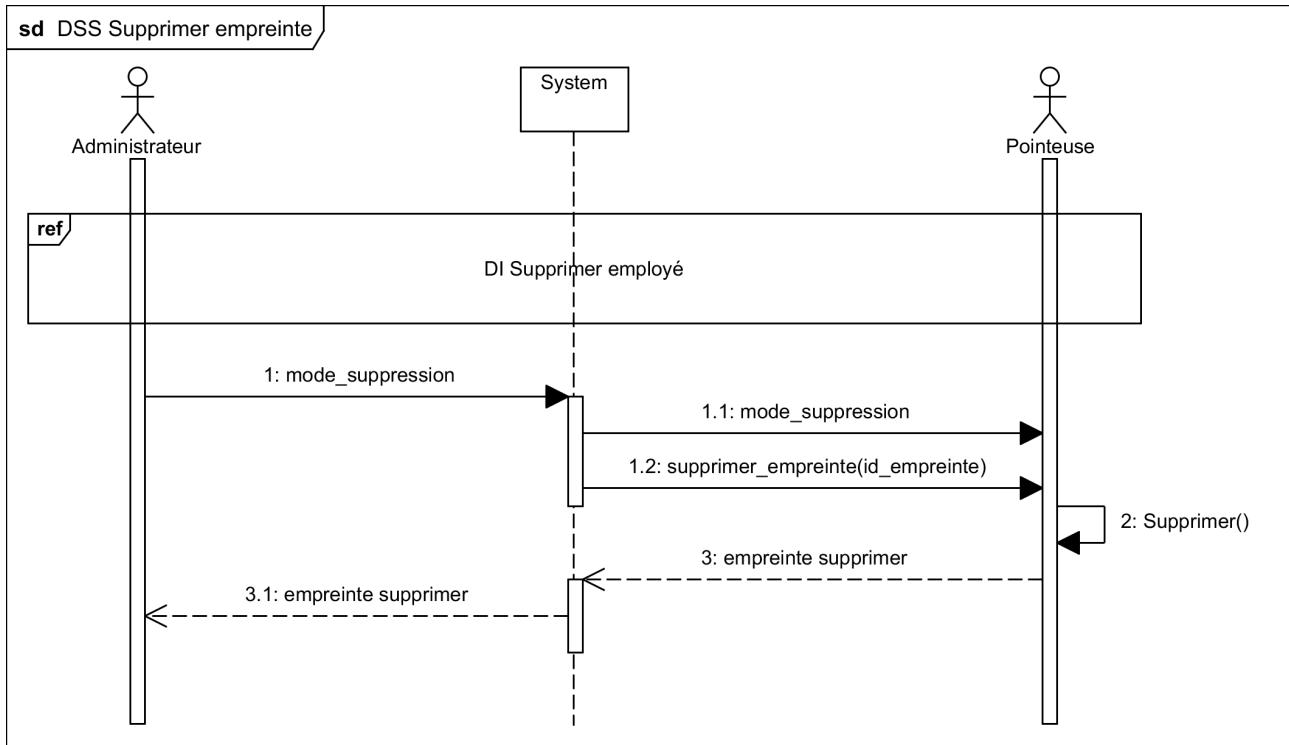


FIGURE 5.28 – Diagramme de séquence système « Supprimer empreinte »

Résumé

Ce document a été rédigé en vue de l'obtention du diplôme de Master en Génie Logiciel. La problématique traitée est la gestion des heures de travail des employés ainsi que leurs pointages au sein d'une entreprise. À la suite d'une analyse de la problématique, une solution a été proposée. Composée d'une application web et d'une pointeuse biométrique, cette alternative offrira aux PMEs la possibilité de répondre à ce besoin de gestion de manière plus fiable et plus efficace. Pour concrétiser cela, nous avons eu recours à un processus de développement et au langage de modélisations UML pour les phases d'analyse et de conceptions. Pour ce qui est de la phase de réalisation, elle s'est caractérisée par l'utilisation de Django, qui est un framework Python, ainsi que d'autres librairies pour l'implémentation de l'application web. Quant à la pointeuse, elle est composée du module ESP32 (dans lequel nous avons injecté un programme en C tel que pratiqué sur la plateforme Arduino) et d'un capteur d'empreinte biométrique DY50.

Mots clés : *empreinte biométrique, pointage, entreprise, UP, UML, agile, web, Python Django, ESP32*

Abstract

This document was written for a Master's degree in Software Engineering. It revolves around the management of working hours and employees clocking within a company. After this issue was analysed, the realization of a timekeeping and working hours management system was proposed. This system, consisting of a web application and a biometric timekeeper, could fulfil SMEs' needs for a better and more reliable management system. For this project, our methodology consisted of the usage of the Unified Modelling Language (UML) for the analysis and the design activities. Moreover, the web application was developed using Python and the Django framework, as well as other libraries. The timekeeping device, for its part, consists of an ESP32 module (in which we imported a program coded in C using the Arduino platform) and a DY50 biometric fingerprint sensor.

Keywords : *biometric fingerprint, attendance, business, UP, UML, agile, web, Python Django, ESP32*