

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



*Mémoire de fin de cycle en vue de l'obtention d'un Master professionnel
Option : Génie Logiciel*

Thème

Conception et réalisation d'une pointeuse numérique
Cas d'étude : RH Partner - Business platforms

Réalisé par :

M. SABRI Mehani

M. TARHLISSIA Aymen

Soutenu le 14 October 2021 devant le jury composé de :

Présidente	M ^{me} ALLOUI Soraya	M.C.A	U. A/Mira Béjaïa.
Examineur	M AISSANI Sofiane	M.C.A	U. A/Mira Béjaïa.
Examineur	M BELHOCINE Sidali	Doctorant	U. A/Mira Béjaïa.
Encadrante	M ^{me} KHOULALENE Nadjat	M.C.B	U. A/Mira Béjaïa.
Co-encadrante	M ^{me} KHENNOUCHE Ferial	M.O.A	Business Platforms.

Promotion 2020-2021.

- *Remerciements* -

Au terme de ce projet de fin de cycle, nous tenons à remercier dans un premier temps le Gérant fondateur de *Business-Platforms* M. BELATTAF Youcef et tout le personnel de l'entreprise, pour leurs confiances et pour nous avoir fait découvrir le monde de l'informatique.

Nous remercions notre encadrant de mémoire Mme. KHOULALENE Nadjat, de nous avoir encadré.

Nous tenons à exprimer nos remerciements également à notre co-encadrant de mémoire Mme KHENNOUCHE Feriel, responsable MOA et Vice Président à *Business-Platforms*, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter nos réflexions.

Nos remerciements s'adressent aussi à tous les membres de *Business-Platforms* qui ont su se rendre disponibles pour répondre à nos questions, suivre notre travail et aussi pour leurs encouragements

Nos remerciements vont pareillement aux membres du jury pour avoir accepté d'examiner notre travail et de l'enrichir par leurs propositions.

Nous remercions toute l'équipe pédagogique du département d'informatique de l'université ABDERRAHMANE MIRA.

Ainsi qu'à toutes les personnes qui ont contribué au succès de notre stage et qui nous ont aidés lors de la rédaction de ce mémoire.

- Dédicaces -

A nos parents, pour leurs sacrifices déployés à notre égard, pour leur patience, leur amour et leur confiance. Qu'ils trouvent dans ce modeste travail, le témoignage de notre profonde affection et de notre attachement indéfectible, nulle dédicace ne puisse exprimer ce qu'on leur doit.

A nos frères et soeurs et tous nos amis pour chaque mot reçu, chaque geste d'amitié, à chaque main tendue et pour toute attention témoignée.

Table des matières

Table des matières	i
Table des figures	iii
Liste des tableaux	v
Liste des abréviations	vi
Introduction générale	1
1 Contexte du projet et méthodologie de conception	2
1.1 Introduction	2
1.2 Contexte du projet	2
1.3 Organisme d'accueil	3
1.4 Problématique	3
1.5 Objectif du projet	4
1.6 Processus du développement	4
1.6.1 Les méthodes agiles	5
1.6.2 Processus Scrum	5
1.7 Conclusion	7
2 Spécification et analyse des besoins	8
2.1 Introduction	8
2.2 Spécification des besoins	8
2.2.1 Besoins fonctionnels	8
2.2.2 Besoins non fonctionnels	9
2.3 Identification des acteurs	9
2.4 Identification des cas d'utilisation	10
2.5 Diagramme de cas d'utilisation	10
2.6 Identifications des cas d'utilisation	11
2.7 Diagramme de séquence système	12
2.7.1 Formalisme et concepts de base	12
2.7.2 Fragment d'interaction	14

2.7.3	Diagrammes de séquence système des cas d'utilisation	14
2.8	Diagramme de sequence	20
2.8.1	Diagrammes de séquence des cas d'utilisation de notre application	20
2.9	Diagramme de classe	25
2.9.1	Définition d'un diagramme de classe	25
2.9.2	Diagramme de classe de notre système	25
2.10	Modèle relationnel	26
2.11	Conclusion	26
3	Réalisation de la pointeuse	27
3.1	Introduction	27
3.2	Raspberry (PI 3B PLUS)	27
3.3	Capteur d'empreinte digitale	29
3.3.1	Caractéristique d'une empreinte digitale	29
3.3.2	Caractéristique du capteur d'empreinte digitale DY50	29
3.3.3	Module USB CP2102	30
3.4	Schémas et branchement de la pointeuse réalisée	31
3.5	Environnement de développement	33
3.5.1	Plateformes utilisées	33
3.5.2	Logiciels utilisés	33
3.5.3	Technologies utilisées	34
3.5.4	Node JS	34
3.5.5	NestJS	35
3.5.6	Angular 11	35
3.5.7	Python	35
3.5.8	HTML	35
3.5.9	CSS	36
3.5.10	Sass	36
3.5.11	TypeScript	36
3.5.12	SQLite	36
3.5.13	MySQL	36
3.6	Architecture du système à réaliser	37
3.7	Conclusion	37
4	Implémentations, mise en service et test	38
4.1	Introduction	38
4.2	Présentation de l'application	38
4.2.1	Nom de l'application	38
4.2.2	Logo :	39
4.2.3	Palette de couleurs	39
4.3	Présentation des interfaces graphiques de notre application	40

4.3.1	Espace employé	40
4.3.2	Application web de gestion	43
4.4	Sécurité de l'application	49
4.4.1	Protection contre Cross Site Scripting	49
4.4.2	RECAPTCHA V3	49
4.4.3	Protection contre SQL injection	49
4.5	Tests	49
4.6	Conclusion	50
	Conclusion générale et perspectives	51
	Bibliographie	53

Table des figures

1.1	Problèmes rencontrés lors de l'utilisation d'une pointeuse numérique selon 12 responsables RH algériens	4
1.2	Récapitulatif du processus de développement «Scrum»	7
2.1	La relation d'inclusion.	10
2.2	La relation d'extension.	11
2.3	La relation de généralisation/spécialisation.	11
2.4	Diagramme de cas d'utilisation global du système.	12
2.5	Formalisme général d'un diagramme de séquence système.	13
2.6	Diagramme de séquence système « Authentification ».	15
2.7	Diagramme de séquence système « Pointer ».	16
2.8	Diagramme de séquence système « Gestion des plannings ».	17
2.9	Diagramme de séquence système « Gestion des employées ».	19
2.10	Formalisme du diagramme de séquence.	20
2.11	Diagramme de séquence du cas d'utilisation Pointer.	21
2.12	Diagramme de séquence du cas d'utilisation S'authentifier.	22
2.13	Diagramme de séquence du cas d'utilisation Ajouter Employé.	23
2.14	Diagramme de séquence du cas d'utilisation Ajouter un planning.	24
2.15	Diagramme de classe de Système.	25
3.1	Raspberry PI B Plus	28
3.2	Capteur d'empreinte digitale DY50	30
3.3	Module USB CP2102	31
3.4	Schéma expliquant le montage préliminaire des différents modules	32
3.5	Architecture du système	37
4.1	Nom de l'application	38
4.2	Logo sur fond blanc	39
4.3	Logo sur fond noir	39
4.4	Palette de couleur RH Partner	39
4.5	Écran de démarrage	40
4.6	Interface d'initialisation	40
4.7	En attente de pointage	41

4.8	Pointage réussie : Entrée	41
4.9	Pointage réussie : Sortie	41
4.10	Échec du pointage	42
4.11	Mise à jour en cours	42
4.12	Interface authentification	43
4.13	Interface menu de navigation	44
4.14	Interface liste d'employés	44
4.15	Interface d'ajout employé	45
4.16	Interface gestion des pointages	45
4.17	Interface ajouter un pointage	46
4.18	Interface liste des plannings	46
4.19	Interface ajouter un planning	47
4.20	Interface Affecter un planning à un employé	48
4.21	Interface supprimer un planning	48

Liste des tableaux

2.1	Description des cas d'utilisation	11
-----	---	----

Liste des abréviations

API	A pplication P rogramming I nterface
CSS	C ascading S tyle S heets
HTML	H yper T ext M arkup L anguage
IDE	I ntegrated D evelopment E nvironment
ORM	M appage O bjet R elationnel
PDG	P résident- D irecteur G énéral
SAAS	S oftware A s A S ervice
SASS	S yntactically A wesome S tyle S heets
SQL	S tructured Q uery L anguage
UML	U nified M odeling L anguage
USB	U niversal S erial B us

Introduction générale

Une entreprise est un ensemble de moyens réunis et organisés. Aujourd'hui, l'entreprise la plus performante est sans doute l'entreprise la plus réactive. C'est celle qui, non seulement, définit des stratégies appropriées à son contexte, à ses moyens et à ses ambitions, mais encore mobilisent l'ensemble des énergies humaines et des compétences dont elle dispose efficacement pour réaliser ses objectifs.

Avec leurs anciennes méthodes de gestion, les entreprises sont obligées de déployer des moyens humains et financiers pour faire face à des tâches répétitives, mais nécessaires. Dans cette optique, ce projet a pour but de réduire les dépenses inutiles d'une entreprise en offrant un système de gestion de pointage qui sera composé d'une application web disposant de fonctionnalités très étendues, et d'une pointeuse biométrique afin d'automatiser l'une des tâches centrales de toute entreprise, et ce en gardant la trace temporaire et l'heure de pointage de chaque employé.

Ce manuscrit sera organisé comme suite :

- Le premier chapitre introduit le contexte du projet, sa problématique, et l'organisme d'accueil. 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 Méthodologie de conception et analyse, permettra d'identifier les différents besoins fonctionnels et non fonctionnels, afin de modéliser les diagrammes et de les décrire de façon détaillée.
- Dans le troisième chapitre, nous aborderons l'aspect matériel et logiciel qui nous permettra de réaliser notre système de pointage, puis nous présenterons le schéma de branchement et l'architecture du système.
- Quant au dernier chapitre, il sera consacré à la présentation de l'application dans ses différents aspects, et il décrira brièvement la phase de tests et le côté sécurité.

Chapitre 1

Contexte du projet et méthodologie de conception

1.1 Introduction

Dans ce premier chapitre, nous allons dans un premier lieu exposer le contexte de notre projet, et nous présenterons l'organisme d'accueil de notre stage. Ensuite, nous aborderons la problématique à résoudre. Pour enfin définir le processus de développement entrepris afin de faciliter l'élaboration du projet.

1.2 Contexte du projet

L'entreprise est une unité économique, juridiquement autonome dont la fonction principale est de produire des biens ou des services pour le marché [1]. Autrement dit, il y a entreprise dès que des personnes mobilisent leur talent et leur énergie, rassemblent des moyens matériels et de l'argent pour apporter un produit ou un service à des clients. Pour une meilleure organisation du travail au sein de l'entreprise, les services RH (Ressources Humaines) utilisent différents mécanismes de pointage pour monitorer le temps de travail de ses collaborateurs.

Dans l'optique de minimiser les dépenses et de mieux exploiter leurs moyens humains et matériels, les entreprises ont 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 efficients. Parmi ces tâches, nous avons choisi de traiter la gestion du pointage des employés ainsi que leur temps de travail en créant une pointeuse qui répond aux besoins du marché algérien.

1.3 Organisme d'accueil

Fraîchement créée, *Business Platforms* est une jeune startup basée à Bejaia, qui réunit une équipe pluridisciplinaire de passionnés du digital : Web, Cloud, Mobile, Marketing, etc. autour d'une idée : la création de plateformes modernes en SAAS, au service des métiers de l'entreprise [2]. Parmi les nombreuses solutions editées, developpées et mise sur le marché par l'entreprise *Business Platforms*, on trouve *RH Partner*.

RH Partner est une solution SIRH en mode SAAS (cloud) , 100% Algérienne. Elle permet de simplifier la gestion des ressources humaines et s'adresse tant aux professionnels du métier, qu'aux gérants, assistants administratifs, gestionnaires des ressources humaines et DRH [3].

1.4 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.

RH Partner dispose d'une solution de pointage *light* qui permet à un salarié de pointer en utilisant son téléphone mobile [4]. Cependant, dans certains cas, elle est insuffisante. Il existe sur le marché pléthores de pointeuses qui peuvent être intégrées dans *RH Partner*, mais l'hétérogénéité des solutions et le peu de visibilité sur l'avenir de ces solutions (dans notre pays, toutes les solutions de pointages/pointeuses sont importées), nous ont poussés à nous lancer dans un projet de création d'une pointeuse Algérienne, conçue et fabriquée en Algérie.

Il existe plusieurs types de pointeuses biométriques. Chaque type diffère selon la technologie d'identification utilisée. Le type d'appareil le plus communément répandu est la pointeuse à empreinte digitale. Avant l'utilisation de l'appareil, les empreintes digitales des salariés sont enregistrées dans la pointeuse. Ainsi, lorsqu'une personne souhaite pointer, elle place son doigt sur le lecteur d'empreinte et la badgeuse biométrique comparera l'empreinte scannée avec celle enregistrée dans sa base de données. S'il y a concordance, le pointage sera enregistré.

Après avoir interviewé plus de 12 Responsables RH dans des petites et moyennes entreprises algériennes, plus de 80% d'entre eux affirment avoir des problèmes réguliers liés à l'utilisation de ces horodateurs dont le blocage, la difficulté de reconnaissance d'empreintes digitales dans certains cas, et la difficulté d'exporter et d'utiliser les données enregistrées par la pointeuse dans d'autres cas.

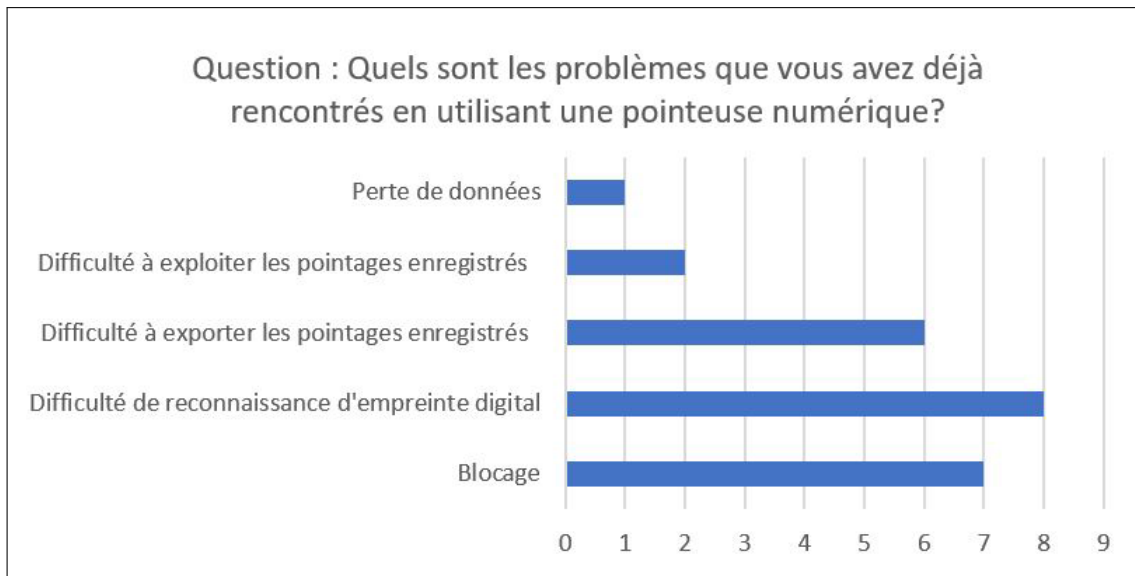


FIGURE 1.1 – Problèmes rencontrés lors de l’utilisation d’une pointeuse numérique selon 12 responsables RH algériens

1.5 Objectif du projet

Le but étant de créer une pointeuse biométrique et de concevoir et réaliser son application de gestion, qui permet une gestion précise du pointage des employés et de leur temps de travail au sein des différentes entreprises de toutes catégories. Notre projet va permettre de résoudre certains problèmes majeurs liés à l’utilisation des pointeuses existantes sur le marché algérien tel que :

- Difficulté du transfert et du traitement des données, et notamment les risques de perte ou endommagement de ces dernières en cas de panne (inconvenient liée à la centralisation des données).
- Problèmes techniques liés à l’utilisation de la pointeuse (par exemple le blocage, la difficulté de reconnaissance, lenteur etc.).
- Le coût élevé des pointeuses existantes sur le marché algérien.

Nous souhaitons une fois ce projet à terme, inciter les entreprises à abandonner leurs anciennes méthodes de pointage pour gagner en efficacité et réduire les ressources allouées à cette tâche. Ceci étant possible, en offrant un outil de supervision simple et ergonomique et en collectant les informations primordiales permettant de faciliter la supervision aux responsables.

1.6 Processus du développement

Un processus décrit 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 «Scrum», qui s’inspire des bonnes pratiques prônées par les tenants de la modélisation agile.

1.6.1 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. Ce manifeste prône quatre valeurs fondamentales [9] :

- « 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.
- « 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.

1.6.2 Processus Scrum

A ce jour, Scrum est la méthode Agile la plus populaire. L’approche Scrum suit les principes de la méthodologie Agile, c’est-à-dire l’implication et la participation active du client tout au long du projet. Le terme Scrum signifie « mêlée » au rugby, le principe de base étant que l’équipe avance ensemble et soit toujours prête à réorienter le projet au fur-et-à-mesure de sa progression, tel un ballon de rugby qui doit passer de main en main jusqu’à marquer un essai. La méthode Scrum s’appuie sur des Sprints qui sont des espaces temps assez courts, généralement entre 2 et 4 semaines. À la fin de chaque Sprint, l’équipe présente ce qu’elle a ajouté au produit [8].

a) Acteurs de la méthode Scrum

La méthode Scrum regroupe trois acteurs : le Product Owner, le Scrum Master et l'équipe de développement. Dans cette partie, nous allons voir plus en détail le rôle de chacun de ces acteurs.

- **Product Owner** : le Product Owner, ou directeur de produit, est le représentant des clients et utilisateurs dans le cadre du projet. Le Product Owner a une très bonne vision produit, connaît les évolutions souhaitables et définit le roadmap (plan) à suivre afin de faire en sorte que le produit s'adapte au mieux aux besoins des clients.
- **Scrum Master** : membre de l'équipe, le Scrum Master a pour objectif de faciliter l'organisation de l'équipe et d'améliorer la capacité de production de l'équipe. Le rôle du Scrum Master est de s'assurer de l'implication de chaque membre et de les aider à franchir les différents obstacles qu'ils pourraient rencontrer. Il doit également s'assurer de l'auto-organisation de l'équipe afin de respecter au mieux le cadre méthodologique Scrum.
- **Équipe de développement** : l'équipe de développement a une responsabilité : délivrer à la fin de chaque Sprint les items qui ont été priorisés pour ce Sprint.

b) Événements de la méthode Scrum

La vie d'un projet Scrum est rythmée par un ensemble de réunions définies avec précision et limitées dans le temps. Nous allons voir ici les différents événements de la méthode Scrum.

- **Sprint** : Pour simplifier le terme, c'est une itération. Il s'agit d'une période généralement entre 2 et 4 semaines maximum pendant lesquelles une version terminée et utilisable du produit est réalisée. Un nouveau Sprint commence dès la fin du précédent. Chaque Sprint a un objectif et une liste de fonctionnalités à réaliser.
- **Planification d'un Sprint** : c'est une réunion qui a pour but de déterminer les éléments prioritaires du Product Backlog (liste ordonnancée des exigences fonctionnelles et non fonctionnelles du projet) qu'elle pense pouvoir réaliser au cours du Sprint, en accord avec le Product Owner.
- **Mêlée quotidienne (Daily)** : c'est une réunion quotidienne qui traite de la progression quotidienne du Sprint. Elle permet à l'équipe de synchroniser ses activités et de faire un plan pour les prochaines 24 heures.
- **Revue du Sprint** : il s'agit du bilan du Sprint réalisé. L'équipe de développement présente les fonctionnalités terminées au cours du Sprint et recueille les feedbacks du Product Owner.
- **Rétrospective du Sprint** : après la revue du Sprint, cette réunion est l'occasion de déterminer ce qui peut être amélioré suite au Sprint écoulé (productivité, qualité, efficacité, conditions de travail, etc.). Cette réunion repose sur le principe d'amélioration continue.

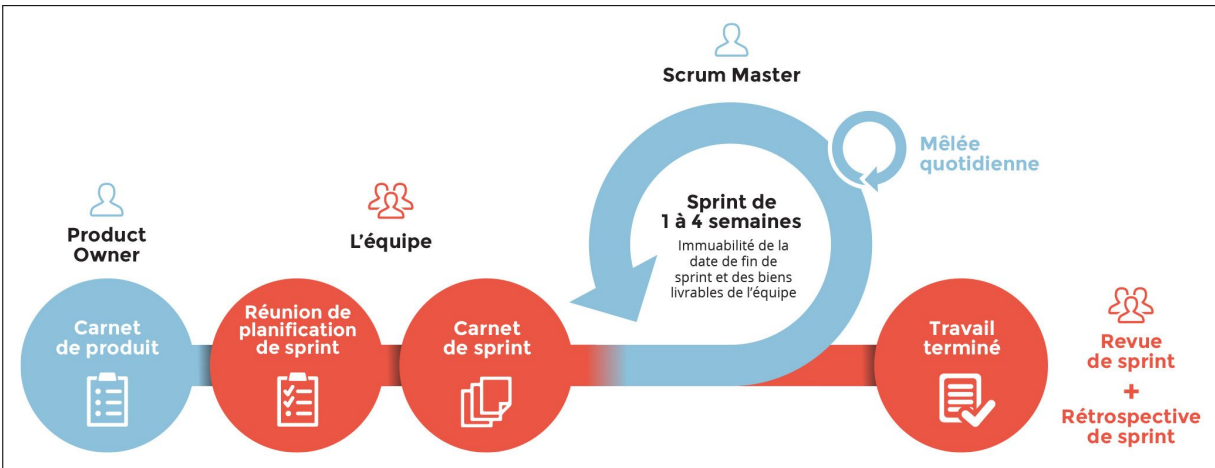


FIGURE 1.2 – Récapitulatif du processus de développement «Scrum»

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 décrit le processus de développement à suivre tout au long du projet. Le chapitre suivant sera consacré à la phase de spécification et d'analyse des besoins.

Chapitre 2

Spécification et analyse des besoins

2.1 Introduction

Ce chapitre sera consacré à la phase de spécification et d'analyse des besoins. Durant cette étape, nous allons tout d'abord identifier les différents acteurs du système, puis nous procéderons à la description des exigences fonctionnelles et non fonctionnelles attendues. Cela nous permettra de modéliser les diagrammes qui nous aideront tout au long du processus de développement.

2.2 Spécification des besoins

2.2.1 Besoins fonctionnels

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). Nous allons séparer nos besoins fonctionnels, car ils ne sont pas les mêmes pour la pointeuse et l'application de gestion associée.

a) Pointeuse (Raspberry + Capteurs) :

Les besoins fonctionnels de la pointeuse sont :

- Enregistrement des pointages (Entrées/Sorties) des employées en utilisant un capteur d'empreintes digitales.
- Envoi des informations relatives aux pointages à l'application de gestion.

b) Application de gestion de la pointeuse :

Les besoins fonctionnels de l'application de gestion de la pointeuse sont

- Authentification.
- Enregistrement et modification des empreintes digitales des nouveaux et anciens employés.
- Ajout, consultation et mise à jour des informations relatives aux employés de l'entreprise.

- Ajout, consultation et mise à jour des plannings de pointage des employés.
- Ajout des pointages manuellement si nécessaire (Disfonctionnement matériel par exemple).
- Consultation, modification et exportation personnalisée des pointages enregistrées.
- Synchronisation des données (Employées, Plannings, Pointages) avec l'application RH PARTNER.

2.2.2 Besoins non fonctionnels

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** : la pointeuse et l'application associée doivent répondre aux besoins des utilisateurs avec un temps de réponse minimal.
- **Sécurité** : aucune opération ne doit être possible sans authentification préalable. Une protection contre diverses vulnérabilités est impérative (Injection SQL, attaques DDOS, etc). Les informations doivent être accessibles aux utilisateurs possédants les bons droits d'accès seulement. Le système doit identifier la pointeuse afin d'accepter les informations reçues.
- **Ergonomie attractive et efficace** : le design des interfaces doit permettre une identification immédiate de ses différents éléments pour permettre à l'utilisateur d'accéder de manière intuitive à ce qu'il cherche, dès la première utilisation.
- **Formulaire intelligible** : les formulaires ne doivent contenir qu'un minimum de champs à renseigner et devront donc être particulièrement soignés.
- **Aide en ligne puissante** : a tout moment, l'internaute peut consulter une page d'aide contextuelle permettant de faciliter sa navigation. Son objectif est d'expliquer à l'internaute le contenu d'un service et comment manipuler l'interface.
- **Modularité du code** : 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.

2.3 Identification des acteurs

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

- **L'employé** : Joue le rôle d'une personne qui occupe un emploi dans l'entreprise et qui interagit avec la pointeuse, pour marquer le pointage des horaires de travail réalisé.
- **Le gestionnaire de la pointeuse** : il s'occupe de l'organisation et du contrôle des équipes et des collaborateurs, ainsi que leurs données de pointage enregistrées.

2.4 Identification des cas d'utilisation

Un cas d'utilisation (use case) représente un ensemble de séquences d'action qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera [10]. On obtient un diagramme de cas d'utilisation en représentant sur un schéma les cas d'utilisation (ovales) reliés par des associations (lignes) à leurs acteurs (personnage) [11]. Depuis UML 2.0, un diagramme peut être inclus dans un cadre accueillant tout le contenu graphique. C'est un rectangle avec un petit pentagone appelé "tag de nom" placé dans l'angle supérieur gauche, qui contient le type du diagramme et son nom [10]. La spécification UML définit les tags de chaque type de diagramme, mais chaque outil a fait ses propres choix. L'outil que nous utiliserons, "Enterprise Architect" de la société Sparx Systems, a pris le parti de définir tous les tags avec deux lettres ; `uc` qui signifie use case diagram.

2.5 Diagramme de cas d'utilisation

On obtient un diagramme de cas d'utilisation en représentant sur un schéma les cas d'utilisation (ovales) reliés par des associations (lignes) à leurs acteurs (personnage). Afin de bien organiser les cas d'utilisation et d'affiner les diagrammes et descriptions, UML définit trois (03) types de relations standardisées entre les cas d'utilisation, détaillées ci-après [11] :

- **La relation d'inclusion** : le cas de base en incorpore explicitement un autre, à un endroit spécifié dans ses enchaînements. Le cas d'utilisation "inclus" n'est jamais exécuté seul, mais seulement en tant que partie d'un cas de base plus vaste. On utilise fréquemment cette relation pour éviter de décrire plusieurs fois le même enchaînement, en factorisant le comportement commun dans un cas d'utilisation à part.

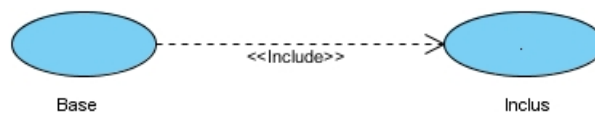


FIGURE 2.1 – La relation d'inclusion.

- **La relation d'extension** : le cas de base en incorpore implicitement un autre, à un endroit spécifié indirectement dans celui qui étend. Le cas de base peut fonctionner tout seul, mais il peut également être complété par un autre, sous certaines conditions, et uniquement à certains points particuliers de son flot d'événements appelés points d'extension. On utilise principalement cette relation pour séparer le comportement optionnel (les variantes) du comportement obligatoire.

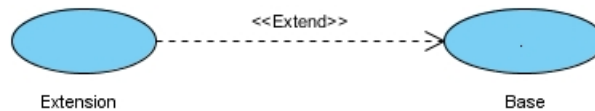


FIGURE 2.2 – La relation d’extension.

- **La relation de généralisation/spécialisation** : les cas d’utilisation peuvent être hiérarchisés par généralisation/spécialisation. Les cas d’utilisation descendants héritent de la sémantique de leur parent. Ils peuvent comprendre des interactions spécifiques supplémentaires, ou modifier les interactions héritées.

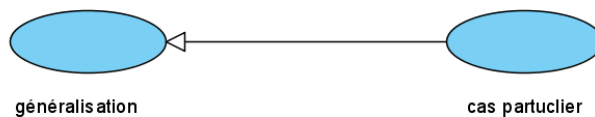


FIGURE 2.3 – La relation de généralisation/spécialisation.

2.6 Identifications des cas d’utilisation

Le tableau ci-dessous présente les différents cas d’utilisations avec leurs acteurs respectifs.

N	Cas d’utilisation		Acteur
1	Pointer		Employé
2	S’authentifier		Gestionnaire pointeuse
4	Gérer les plannings	Ajouter un planning	
		Modifier un planning	
		Supprimer un planning	
		Affecter un planning	
		Configurer un planning	
5	Gérer les employées	Ajouter un employée	
		Modifier un employée	
		Supprimer un employée	
6	Gérer les pointages	Ajouter un pointage	
		Modifier un pointage	
		Supprimer un pointage	
		Exporter les pointages	

TABLE 2.1 – Description des cas d’utilisation

Le diagramme de cas d’utilisation global est illustré dans la figure ci-dessus :

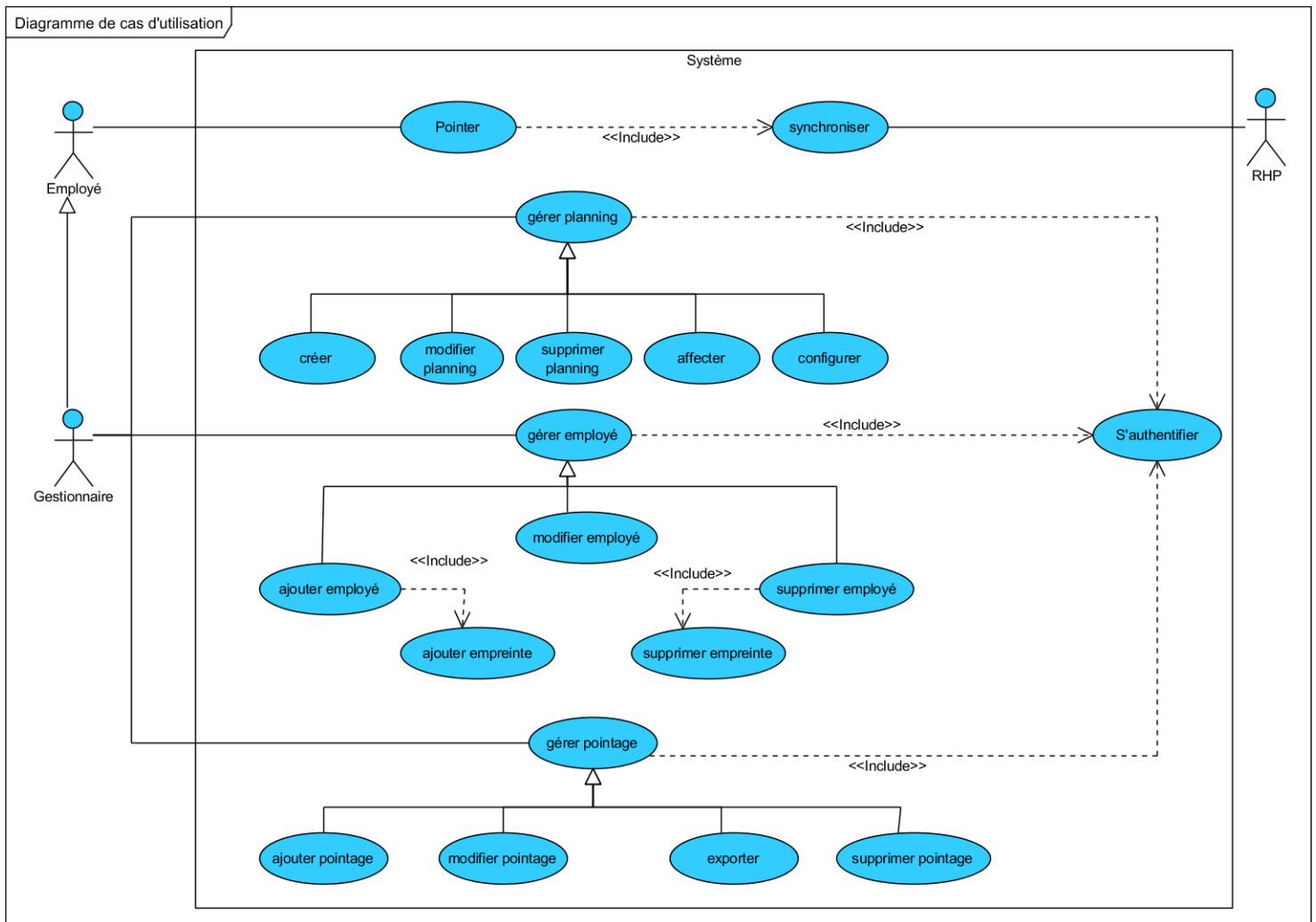


FIGURE 2.4 – Diagramme de cas d'utilisation global du système.

2.7 Diagramme de séquence système

On utilise le terme diagramme de séquence système pour souligner le fait de considérer le système informatique comme une boîte noire. Le comportement du système est décrit, vu de l'extérieur, sans préciser comment il le réalisera. L'objectif des diagrammes de séquence est de représenter les interactions entre les objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés[10].

2.7.1 Formalisme et concepts de base

Dans un diagramme de séquence système, les objets et les acteurs sont énumérés en colonnes avec leurs lignes de vie verticales indiquant la durée de vie de l'objet.

- Un objet est une instance de classe rangée horizontalement et représenté par un rectangle.

- Une ligne de vie (life-line) représente l'ensemble des opérations exécutées par un objet.
- La notation utilisée pour une ligne de vie est une ligne pointillée verticale partant de l'objet.
- Une activation est modélisée par des boîtes rectangulaires sur la ligne de vie. Elles indiquent quand l'objet effectue une action. Un message est modélisé par une flèche horizontale entre les activations. Il indique les communications entre les objets.

Dans un diagramme de séquence système, deux types de messages peuvent être distingués :

- **Message synchrone** : l'émetteur reste en attente d'une réponse à son message avant de poursuivre ses actions. La flèche avec extrémité pleine symbolise ce type de message.
- **Message asynchrone** : dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. Une flèche avec une extrémité qui n'est pas pleine symbolise ce type de message.

La figure ci-après illustre le formalisme général suivi pour représenter un diagramme de séquence système

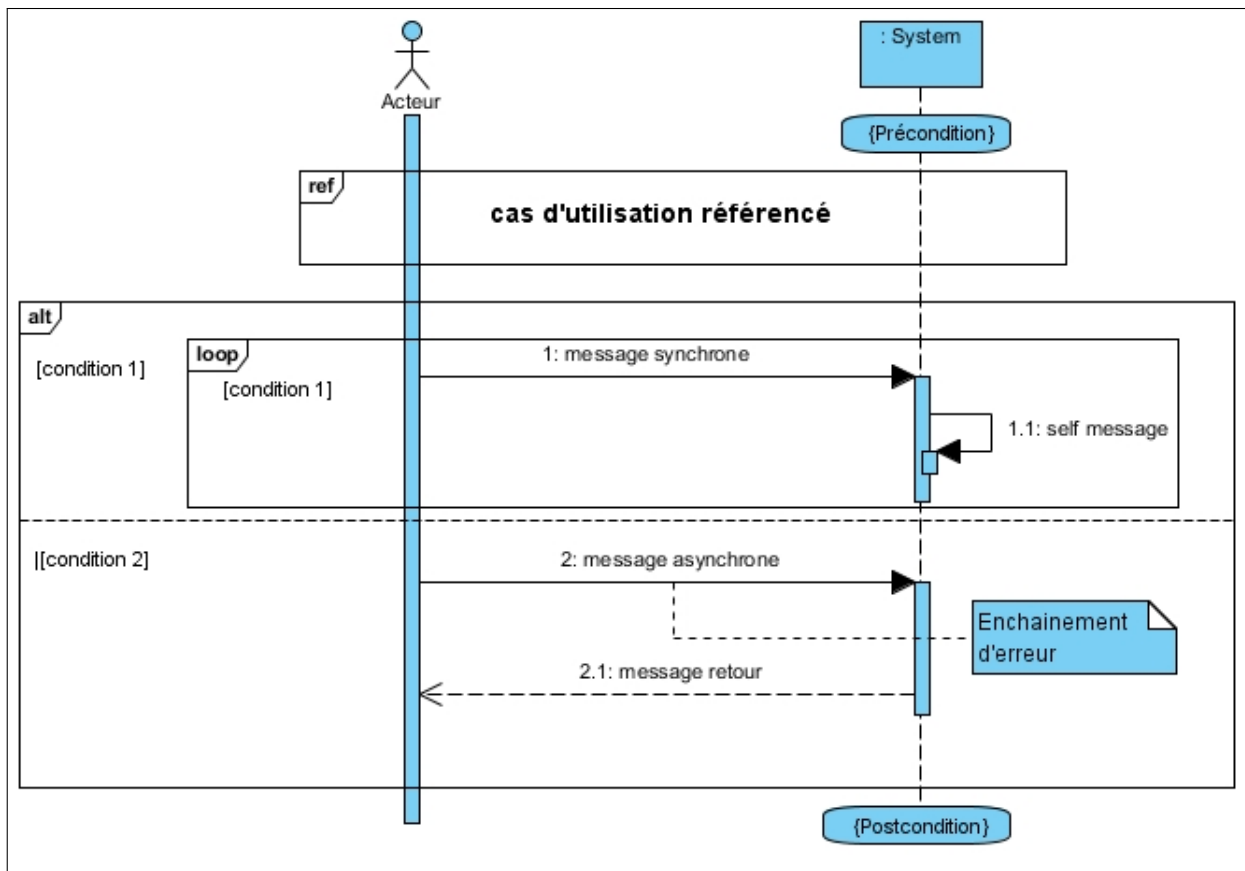


FIGURE 2.5 – Formalisme général d'un diagramme de séquence système.

2.7.2 Fragment d'interaction

Dans un diagramme de séquence système, il est possible de distinguer des sous-ensembles d'interactions qui constituent des fragments.

Un fragment d'interaction correspond à un ensemble d'interactions auxquelles on applique un opérateur. On le représente globalement, comme un diagramme de séquence, dans un rectangle avec indication dans le coin gauche du nom de l'opérateur.

Treize (13) opérateurs ont été définis dans UML : alt, opt, loop, par, strict/weak, break, ignore/consider, critical, negative, assertion et ref. Nous décrivons dans ce qui suit les opérateurs utilisés pour l'analyse de notre projet : alt, opt, loop et ref.

- L'opérateur alternatif ou « alt » correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il permet aussi d'utiliser les clauses de type sinon et se représente dans un fragment possédant au moins deux parties séparées par des pointillés.
- L'opérateur optional ou « opt » correspond à une instruction de test sans alternative. Il se représente dans un fragment possédant une seule partie.
- L'opérateur « ref » permet d'appeler une séquence d'interactions décrite par ailleurs, constituant ainsi une sorte de sous-diagramme de séquence.
- L'opérateur « loop » correspond à une instruction de boucle qui permet d'exécuter une séquence d'interactions tant qu'une condition est satisfaite.[12]

2.7.3 Diagrammes de séquence système des cas d'utilisation

Dans ce qui suit, nous représentons le diagramme de séquence système d'un scénario représentatif de chacun des cas d'utilisation décrits précédemment.

a) Diagramme de séquence système du cas d'utilisation « Authentification »

Pour s'authentifier, un utilisateur doit remplir le formulaire d'authentification en introduisant son login et son mot de passe puis valide son envoi. Le système vérifie si les deux (02) champs ont bien été saisis. Tant que le gestionnaire en oublie un, le système lui demande de ressaisir les informations (utilisation de loop). Enfin, après avoir vérifié, le système renvoie à la page d'accueil sinon il affiche un message d'erreur.

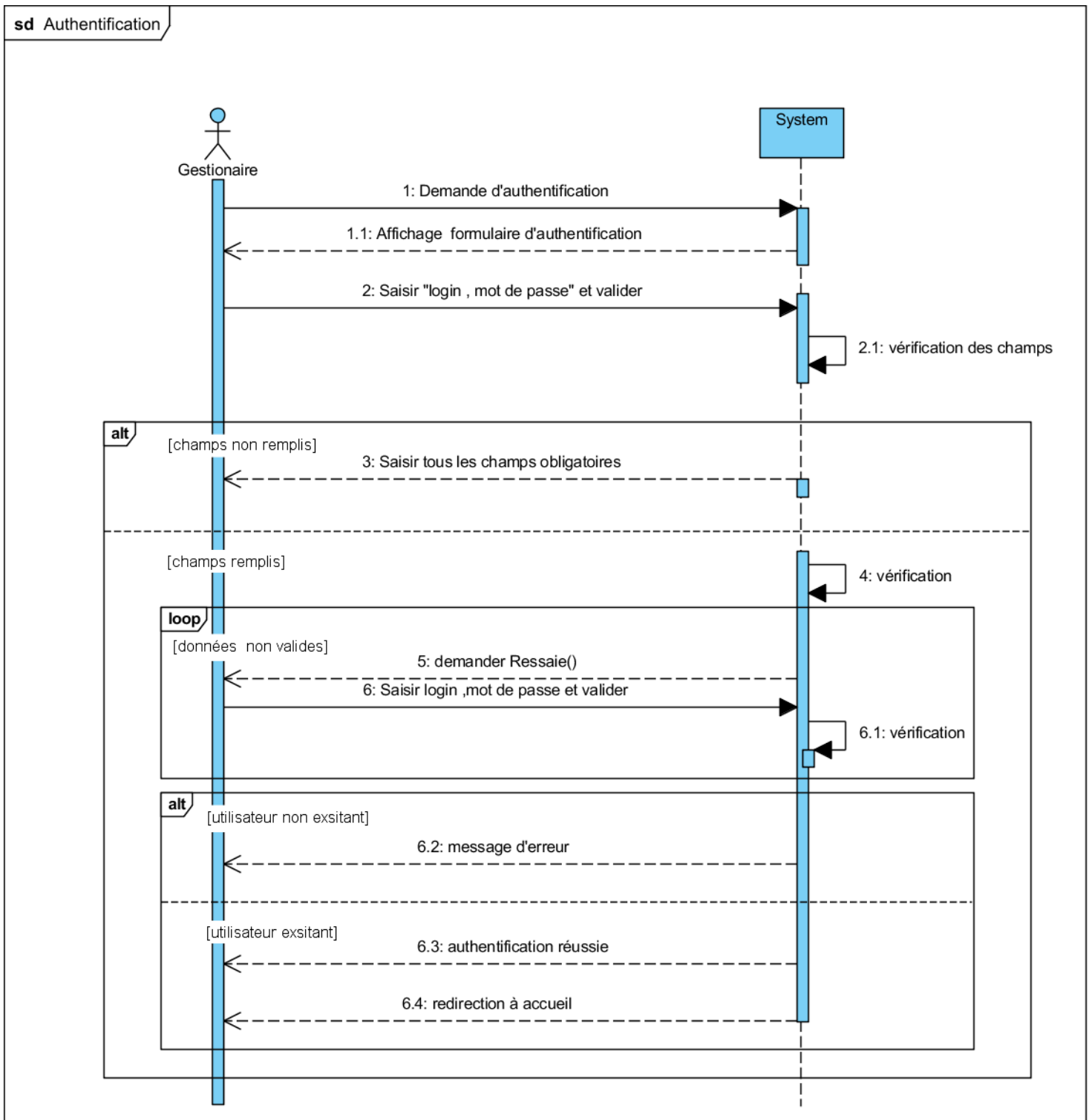


FIGURE 2.6 – Diagramme de séquence système « Authentication ».

b) Diagramme de séquence système du Cas d'utilisation « Pointer »

L'employé doit renseigner ses heures de travail via la pointeuse biométrique, qui vérifie son identité puis enregistre son pointage et lui signale le bon déroulement de l'opération.

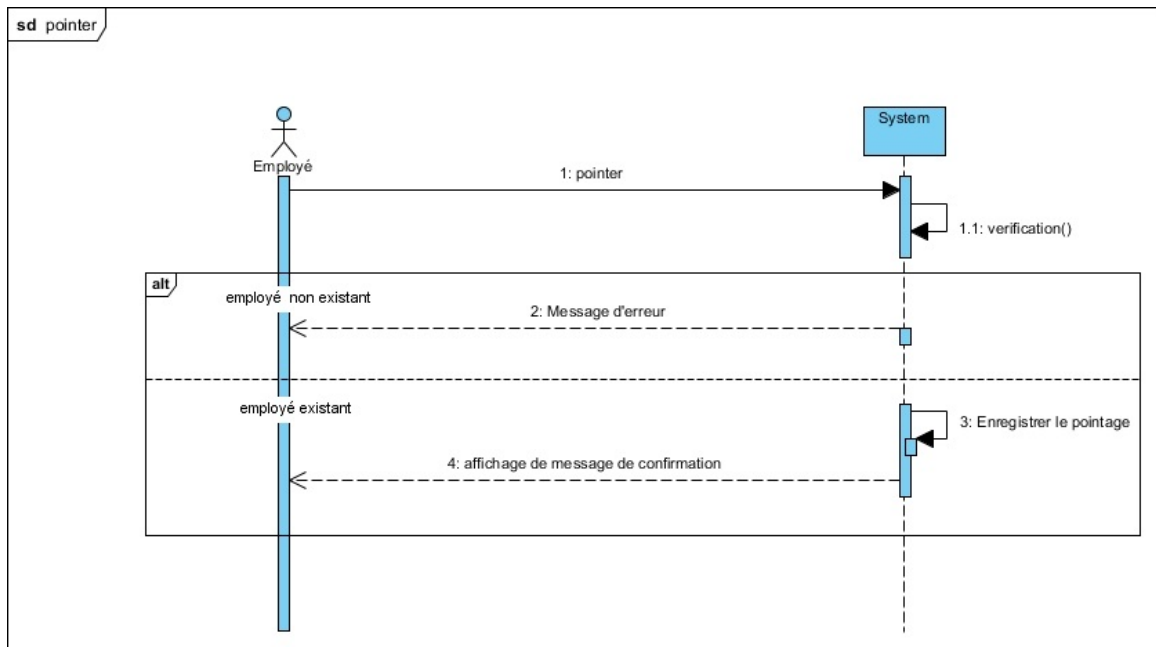


FIGURE 2.7 – Diagramme de séquence système « Pointer ».

c) Diagramme de séquence système du Cas d'utilisation « Gestion des plannings »

La figure suivante décrit l'enchaînement séquentiel des échanges entre le système et le gestionnaire lorsque ce dernier consulte la liste des plannings et éventuellement ajoute, modifie ou supprime un planning.

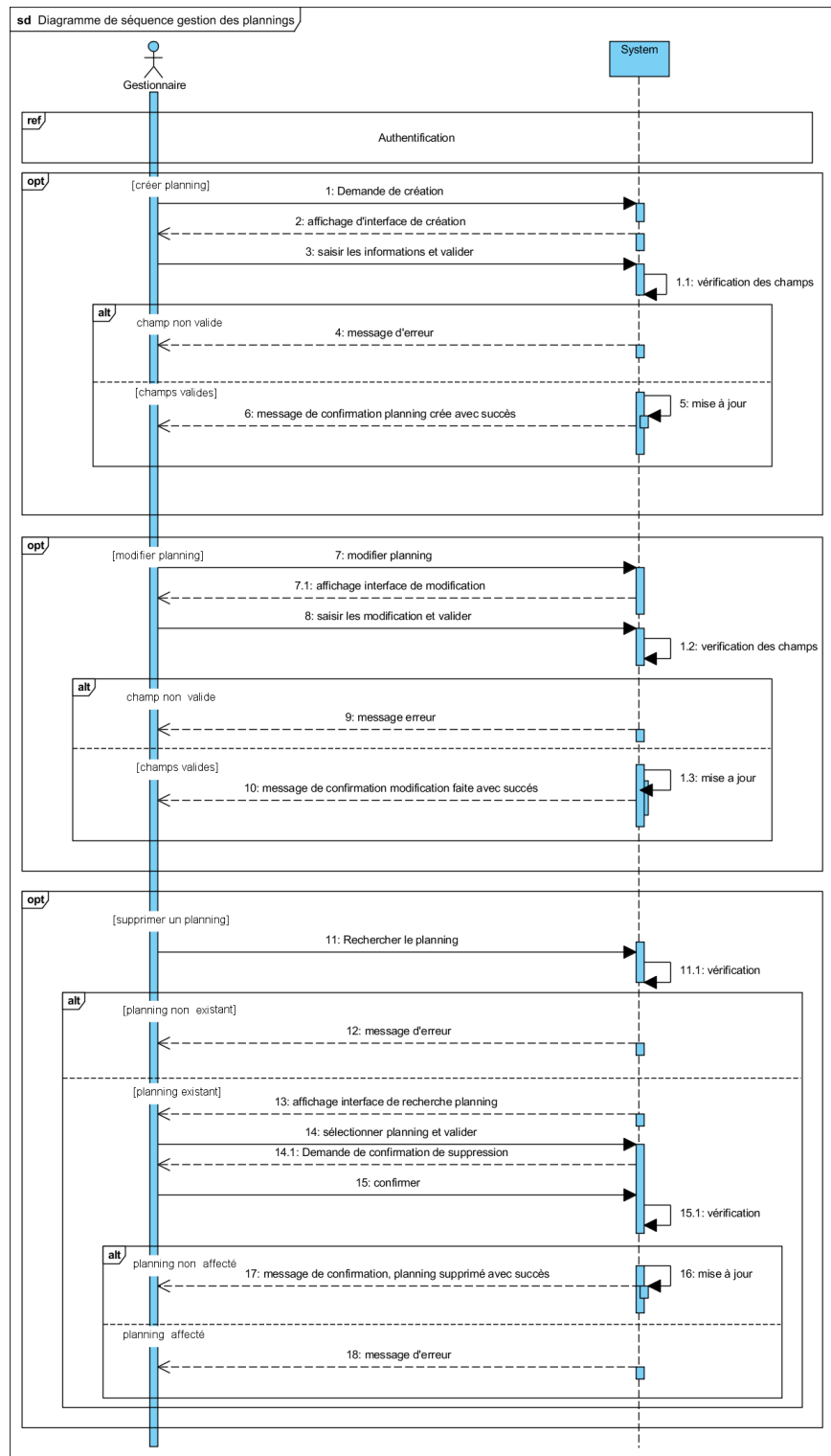


FIGURE 2.8 – Diagramme de séquence système « Gestion des plannings ».

d) Diagramme de séquence système du Cas d'utilisation « Gestion des employées »

La figure suivante décrit l'enchaînement séquentiel des échanges entre le système et le gestionnaire lorsque ce dernier consulte la liste des employés et éventuellement ajoute, modifier ou supprime un employé

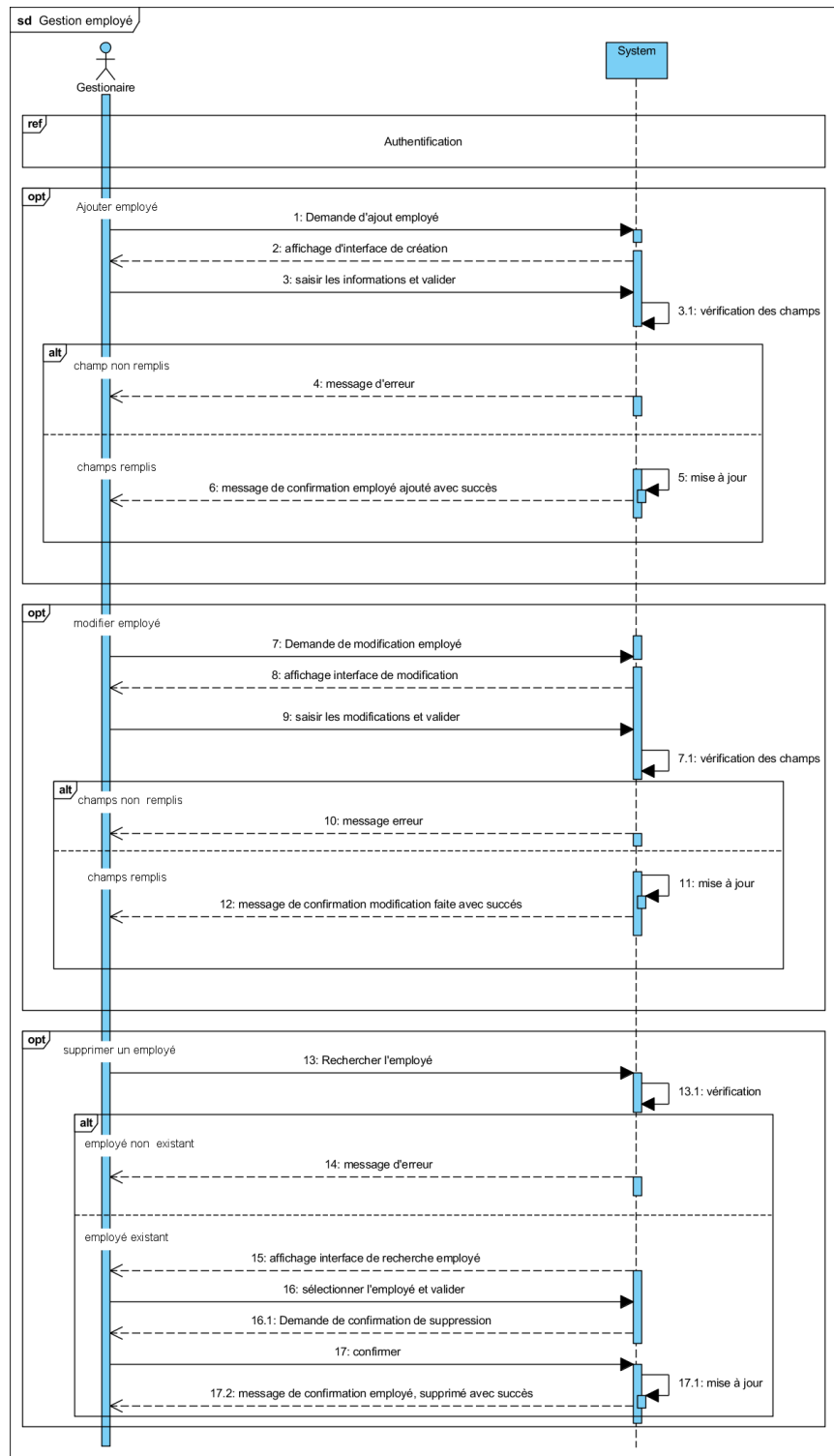


FIGURE 2.9 – Diagramme de séquence système « Gestion des employés ».

2.8 Diagramme de sequence

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. Dans cette partie, nous allons détailler les diagrammes de séquence système élaboré, 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 2.9

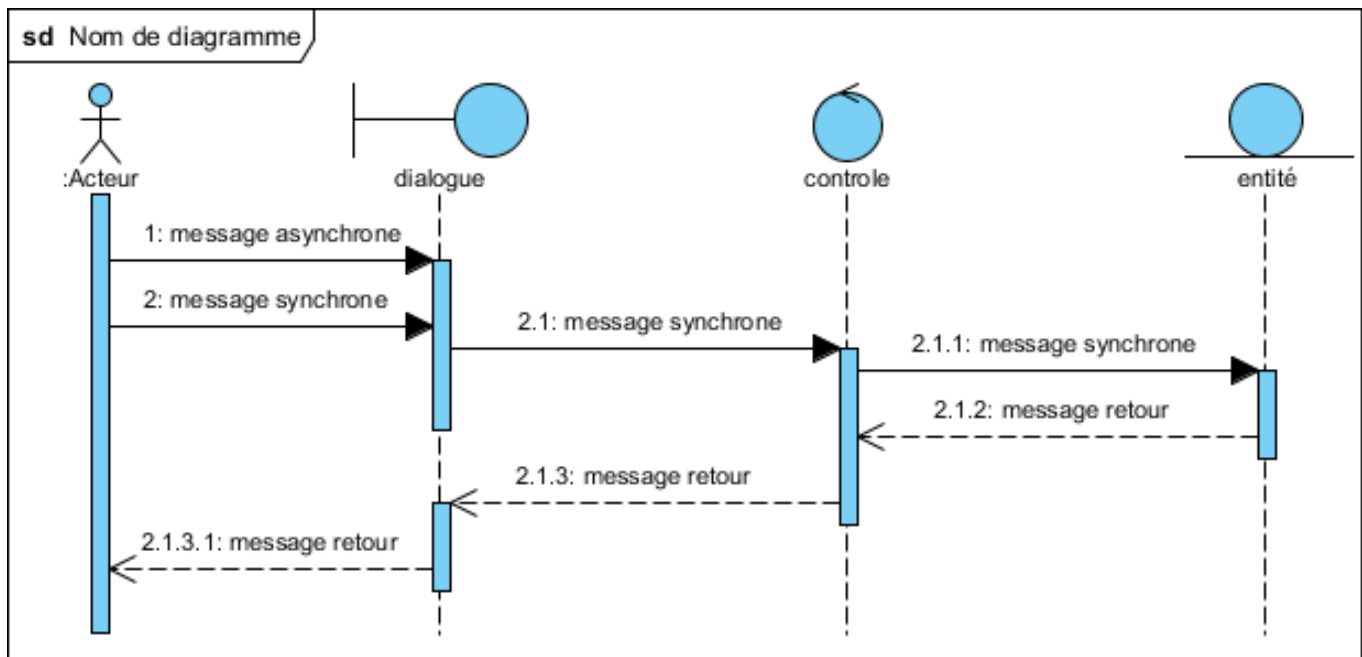


FIGURE 2.10 – Formalisme du diagramme de séquence.

2.8.1 Diagrammes de séquence des cas d'utilisation de notre application

Pour illustrer notre démarche, nous allons étudier les cas d'utilisation que nous avons jugé majeurs pour notre system, à savoir :

- Pointer
- S'authentifier
- Ajouter employé
- Ajouter planning

a) Diagrammes de séquence du cas d'utilisation Pointer

La figure suivante représente le diagramme de séquence détaillé du cas d'utilisation "Pointer".

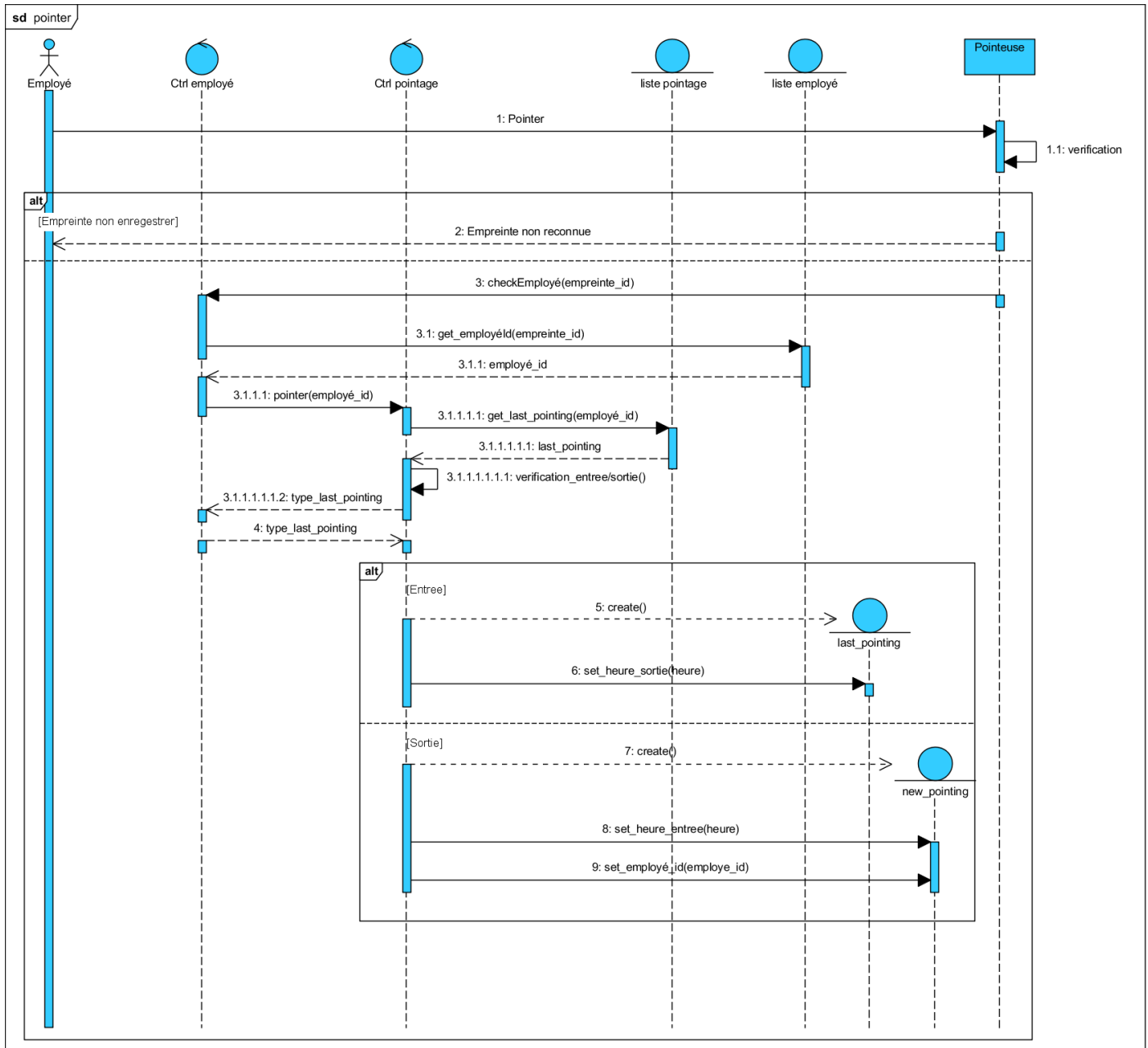


FIGURE 2.11 – Diagramme de séquence du cas d'utilisation Pointer.

b) Diagrammes de séquence du cas d'utilisation S'authentifier

La figure suivante représente le diagramme de séquence détaillé du cas d'utilisation "S'authentifier".

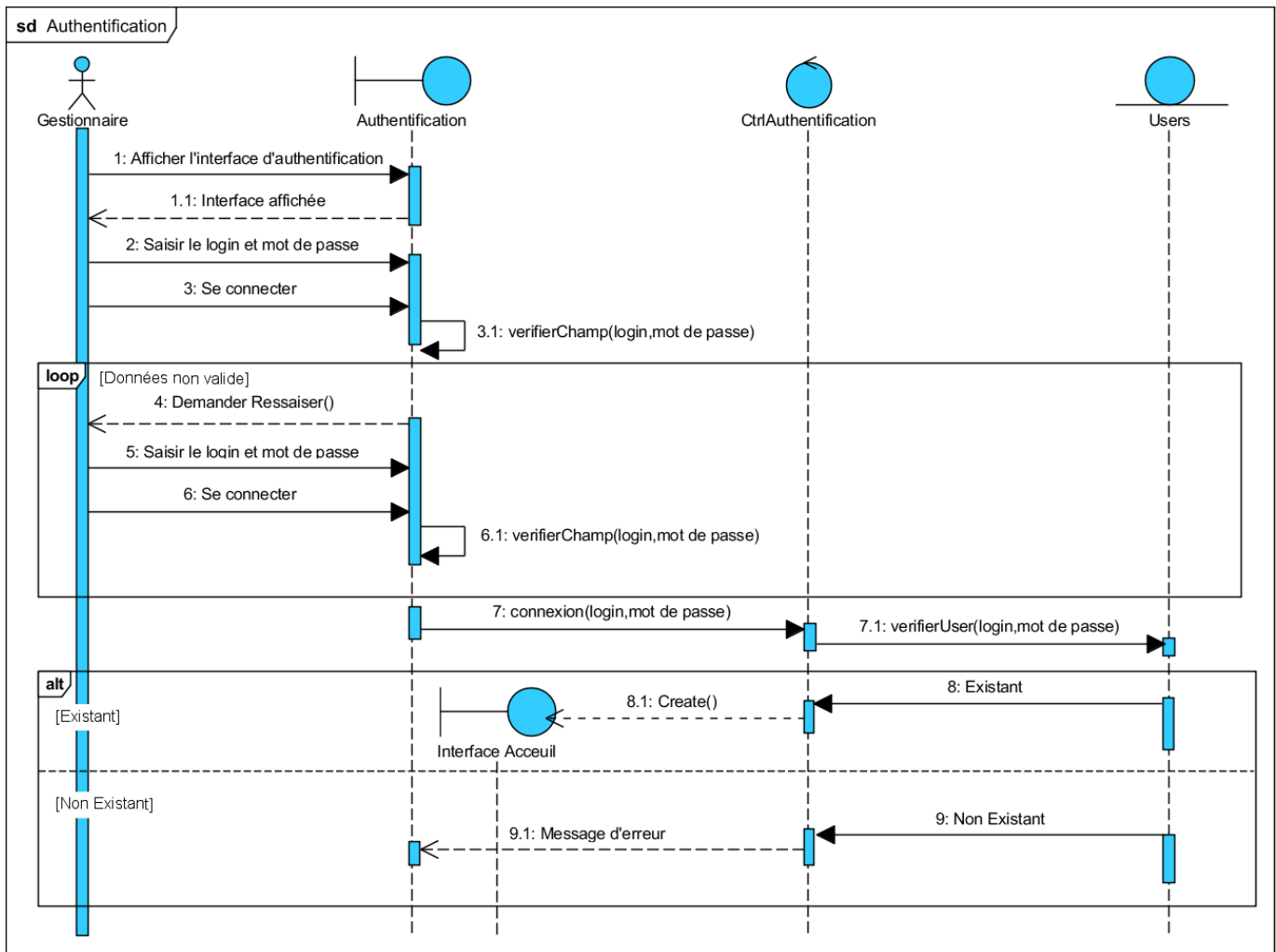


FIGURE 2.12 – Diagramme de séquence du cas d'utilisation S'authentifier.

c) Diagramme de séquence du cas d'utilisation Ajouter employé

La figure suivante présente le diagramme de séquence du cas d'utilisation « Ajouter un employé ». Comme le diagramme le décrit, après l'authentification du gestionnaire, ce dernier peut ajouter un nouveau employé en accédant à la page appropriée.

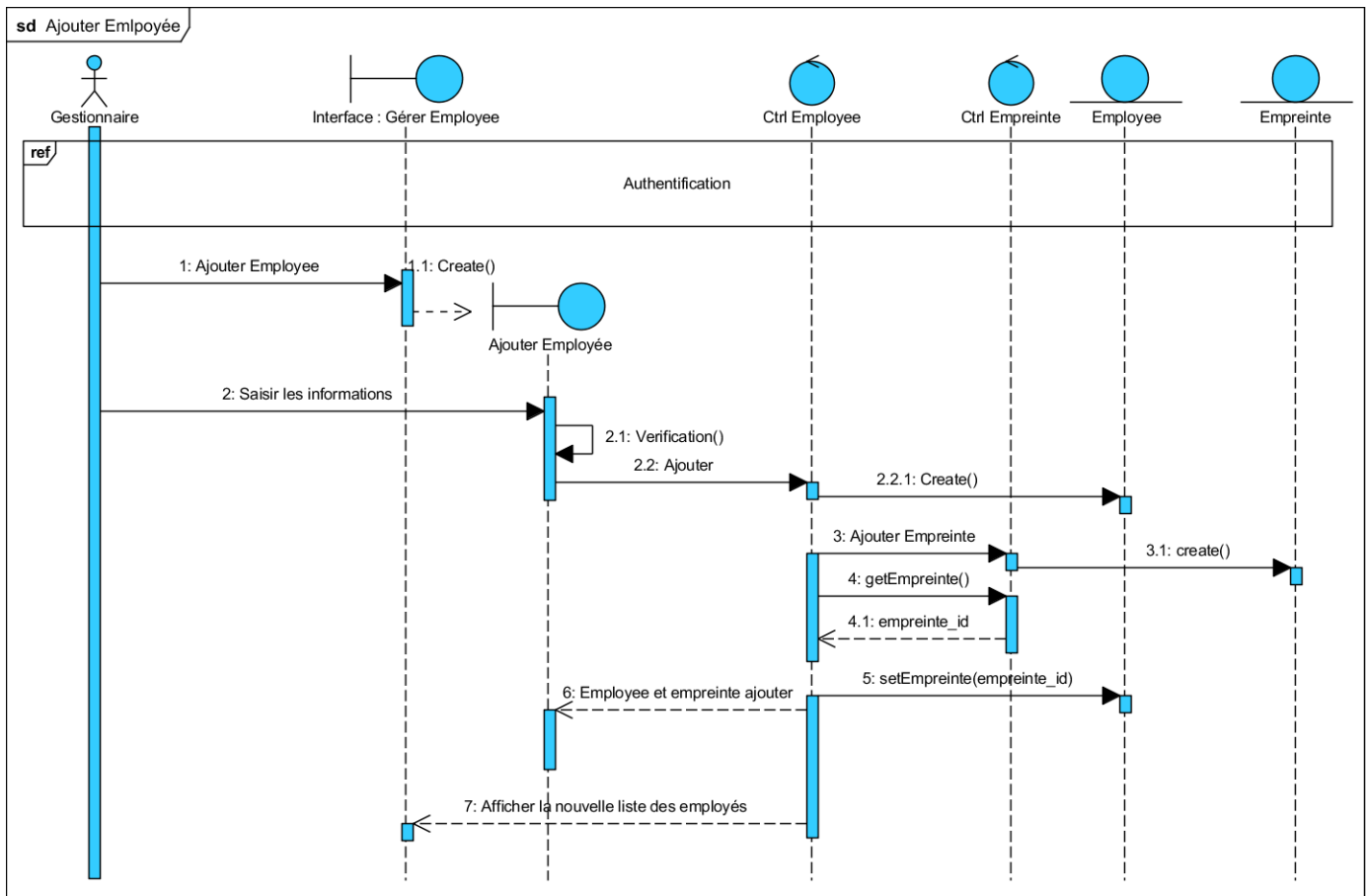


FIGURE 2.13 – Diagramme de séquence du cas d'utilisation Ajouter Employé.

d) Diagrammes de séquence du cas d'utilisation Ajouter un planning

Le gestionnaire 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 le gestionnaire devra les corriger. Si les informations sont valides, le contrôle CtrlPlanning vérifiera si aucun planning n'existe avec l'intitulé saisi afin de valider l'ajout du planning. Une fois terminé, le gestionnaire pourra affecter un planning aux employés.

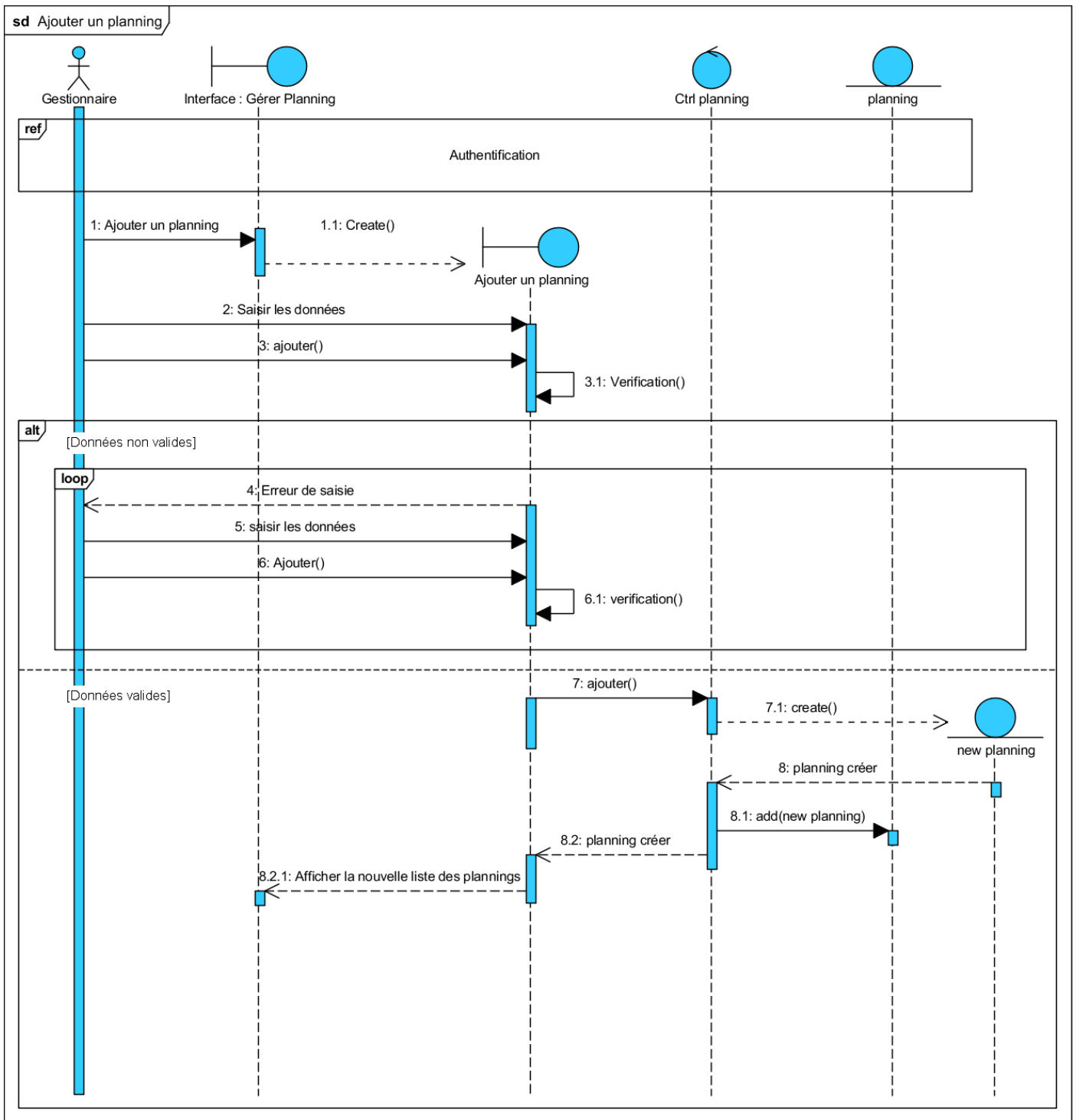


FIGURE 2.14 – Diagramme de séquence du cas d’utilisation Ajouter un planning.

2.9 Diagramme de classe

2.9.1 Définition d'un diagramme de classe

Le diagramme de classe constitue l'un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations. Le détail des traitements n'est pas représenté directement dans le diagramme de classe ; seul l'algorithme général et le pseudocode correspondant peuvent être associés à la modélisation. La description du diagramme de classe est fondée sur [13] :

- le concept d'objet,
- le concept de classe comprenant les attributs et les opérations,
- les différents types d'association entre classes.

2.9.2 Diagramme de classe de notre système

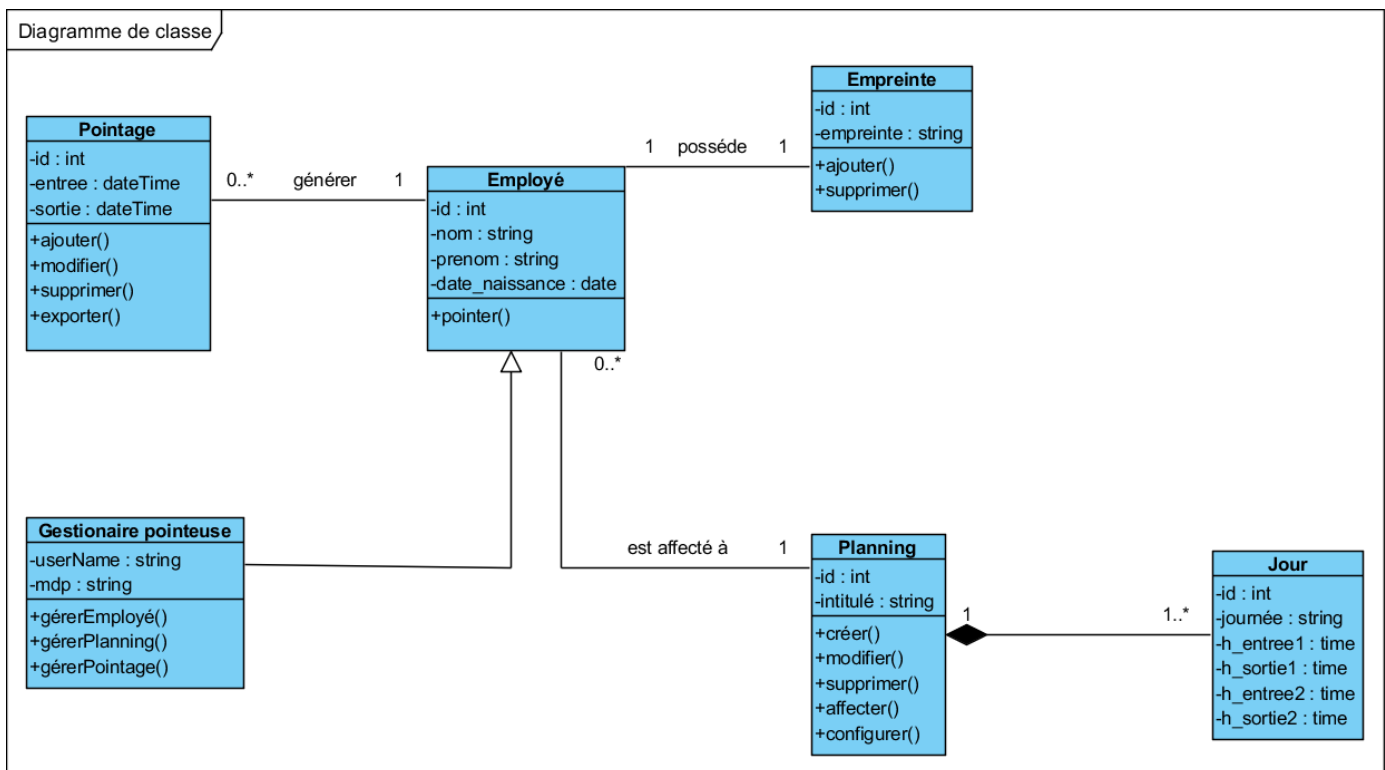


FIGURE 2.15 – Diagramme de classe de Système.

2.10 Modèle relationnel

Le modèle relationnel est construit principalement autour des concepts suivants [12] :

- 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 identifiant une relation qui préfigure la primary Key de la table.
- Clé étrangère : attribut qui référence une tierce relation, préfigure 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, date_naissance, type, #id_planning, #id_empreinte).
- **Planning**(id_planning, intitulé).
- **Jour**(id_jour, h_entrée1, h_sortie1, h_entrée2, h_sortie2, #id_planning).
- **Pointage**(id_pointage, h_entrée, h_sortie, durée #id_employé).
- **Empreinte**(id_empreinte, empreinte)
- **Gestionnaire**(#id_employé, username, pwd).

2.11 Conclusion

Au cours de ce chapitre, nous avons en premier lieu identifié les besoins fonctionnels et non fonctionnels de notre système. Puis nous avons modélisé le diagramme de séquence et le diagramme de classes du système. Enfin, pour avoir une vue plus structurée et implémenter une base de données nous avons traduit ce dernier diagramme en un modèle relationnel. Ceci nous permettra de réaliser notre projet

Chapitre 3

Réalisation de la pointeuse

3.1 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 Raspberry en tant que plateforme, par la suite, nous citerons les caractéristiques du capteur d'empreinte. Une fois assemblés, ils formeront notre dispositif de pointage. Dans la deuxième partie de ce chapitre nous allons présenter l'environnement de développement et les bibliothèques utilisées. Et enfin, nous expliquerons l'architecture globale du système.

3.2 Raspberry (PI 3B PLUS)

À l'origine le Raspberry Pi est un PC miniature destiné à fonctionner sous Linux. Il est développé par la Foundation Raspberry et destiné à favoriser et encourager l'apprentissage de l'informatique et du développement en proposant sur le marché une machine simple, universelle, peu gourmande en énergie et très attractive en terme de prix. La façon de l'utiliser n'a pas de limites, que ce soit comme plateforme de développement (puisque'il est compatible avec de très nombreux langages Perl, Ruby, Python, Java, Lua), comme mini serveur web ou comme véritable "media center" efficace et tout à fait apte à décoder de la HD en 1080p (distribution OpenELEC optimisée pour Kodi XBMC recommandée) [31].

Le Raspberry Pi 3 Model B s'appuie sur les fonctionnalités de ses prédécesseurs avec un nouveau processeur plus rapide à bord pour augmenter sa vitesse. Il dispose également de capacités Wi-Fi et Bluetooth Low Energy pour améliorer la fonctionnalité et la capacité d'alimenter des appareils plus puissants via les ports USB. Parmi les spécifications techniques du Raspberry Pi 3 on peut citer [31] :

- Processeur Broadcom BCM2837B0 64 bit ARMv8 QUAD Core A53 64 bit 1.4GHz.
- 1 go de RAM - BCM43143 WiFi à bord.
- Bluetooth basse énergie (BLE) à bord.
- GPIO étendu à 40 broches-4 ports USB2.

- Sortie stéréo 4 pôles et port vidéo Composite.
 - HDMI pleine taille-compatible.
 - Port de caméra CSI pour connecter la Raspberry.
 - Caméra Pi-port d'affichage DSI pour connecter la Raspberry.
 - Écran tactile Pi-port MicroSD pour charger le système d'exploitation et stocker des données.
 - Adaptateur d'alimentation Micro USB commuté amélioré (supporte jusqu'à 2.5 ampères).
- Pour les besoins de notre projet, nous avons choisi un raspberry de type Pi3 B plus en raison de son prix, sa taille et ces performances.

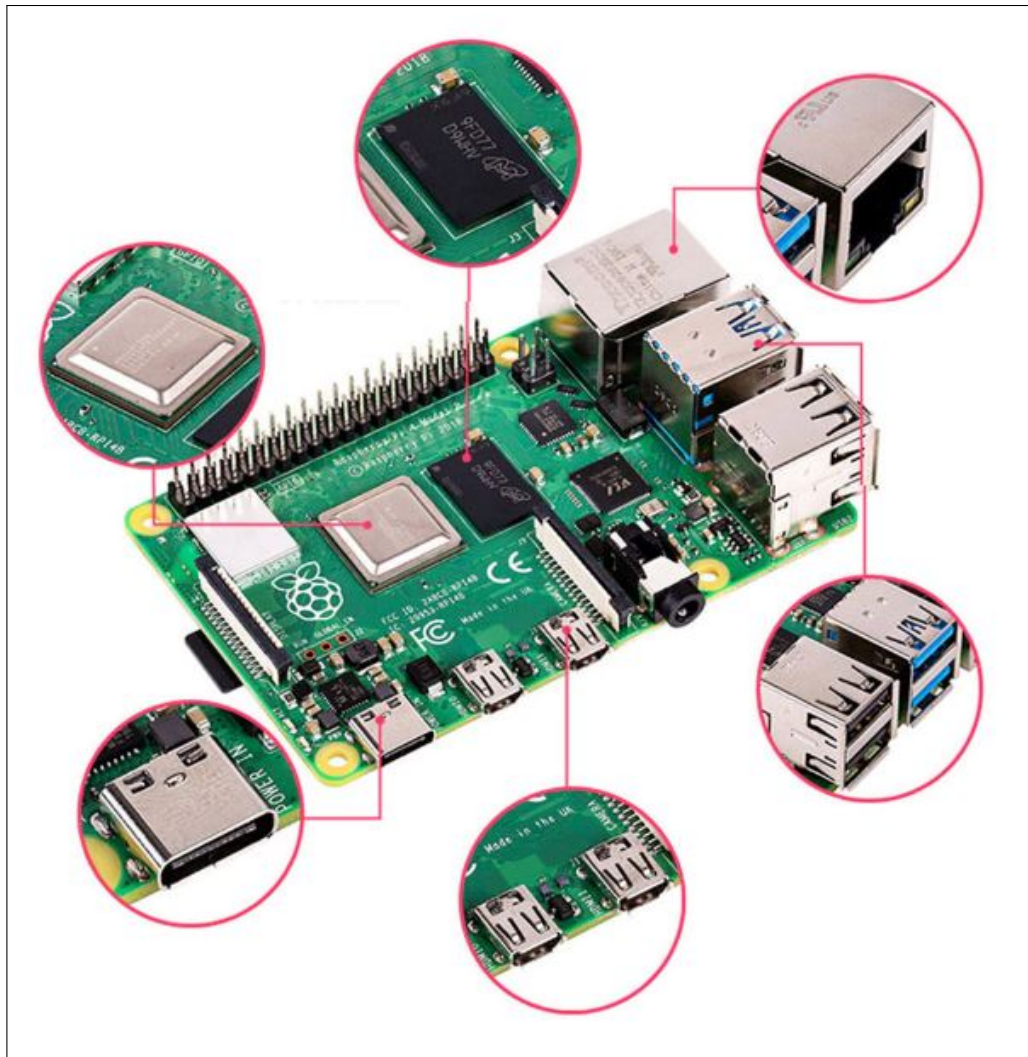


FIGURE 3.1 – Raspberry PI B Plus

3.3 Capteur d’empreinte digitale

3.3.1 Caractéristique d’une empreinte digitale

Une empreinte digitale se compose d’un ensemble de stries (ici définies comme étant les reliefs positifs qui entrent 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 d’identifier un individu de manière extrêmement fiable [6].

Dans la perspective de réaliser un système de pointage fiable pour mesurer le temps de travail et gérer la présence des employés, ainsi que de maximiser et de motiver la productivité de ces derniers 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, car elle permet d’identifier et d’authentifier les personnes de manière fiable et rapide, en fonction de caractéristiques biologiques uniques. Pour ce faire, nous allons utiliser le capteur DY50 de type optique en raison de son coût, sa taille, et du fait d’avoir une image précise de l’empreinte.

3.3.2 Caractéristique du capteur d’empreinte digitale DY50

Les principales caractéristiques du capteur d’empreintes digitales que nous utiliserons dans notre projet sont :

- Faible taux de faux accepter avec un maximum de 0.001%, rend la reconnaissance plus sûre et fiable.
- La fenêtre d’empreintes digitales utilise une technologie étanche avancée, sans craindre qu’elle ne soit influencée par l’eau.
- Temps d’image d’empreinte digitale : <1.0 secondes
- Niveau de sécurité : cinq (de bas en haut : 1,2,3,4,5)
- Temps de recherche : <1.0 secondes (1 :500, la moyenne)
- Tension d’alimentation : alimentation cc 3.6 6.0V / 3.3V
- Courant d’alimentation : courant : <120mA
- Mode de correspondance : mode de correspondance (1 :1)
- Mode de recherche (1 : N)



FIGURE 3.2 – Capteur d’empreinte digitale DY50

3.3.3 Module USB CP2102

Le pilote CP2102 est requis pour le fonctionnement du lecteur d’empreintes digitales DY50 en tant qu’USB virtuel afin de faciliter la communication avec le Raspberry.

a) Caractéristique du Module CP2102 [33] :

- Taille du PCB : 26.5mm * 15.6mm.
- Poids : 4g
- 3.3V et 5V avec deux sorties d’alimentation
- Broches de plomb
- Compatible avec tous les systèmes d’exploitation
- Avec fusible réinitialisable. En cas de court-circuit accidentel peut protéger efficacement le lecteur d’empreinte et le port USB de Raspberry.
- Facilite le travail avec trois LEDs : indicateur d’alimentation, indicateur de réception de données et indicateur de transmission de données

b) Branchement CP2102 avec le capteur DY50 :

- Sortie de données : TX (CP2102) -> RX (DY50)

- Entrée de données : RX (CP2102) -> TX (DY50)
- GND (CP2102) -> GND (DY50)
- 3.3v (CP2102) -> vcc (DY50)

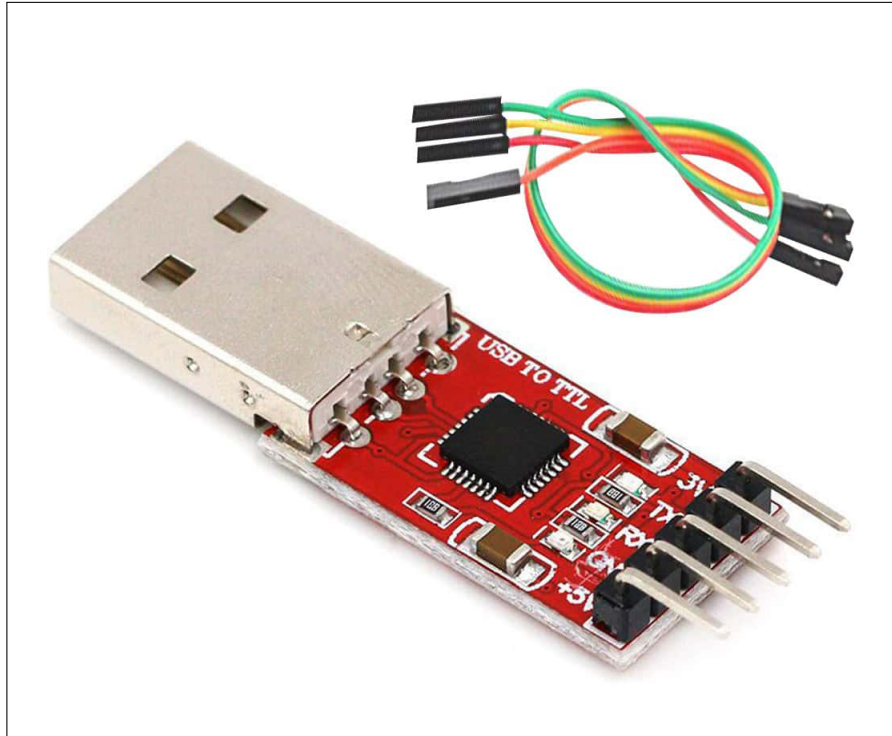


FIGURE 3.3 – Module USB CP2102

3.4 Schémas et branchement de la pointeuse réalisée

La pointeuse que nous avons réalisé est composée de plusieurs modules, un d'entrée et un autre de sortie; tous deux connectées au noyau qui est le raspberry. Le premier qui est le lecteur d'empreinte DY50 est connecté au module de conversion CP2102, qui lui aussi est connecté au noyau via un port USB. Le deuxième, qui est l'écran d'affichage LCD 3.5", est connecté au raspberry via un port appelé GPIO. Il occupe les 28 premières têtes (Headers).

Le schéma suivant révèle la manière dont les modules de la pointeuse ont été connectés.

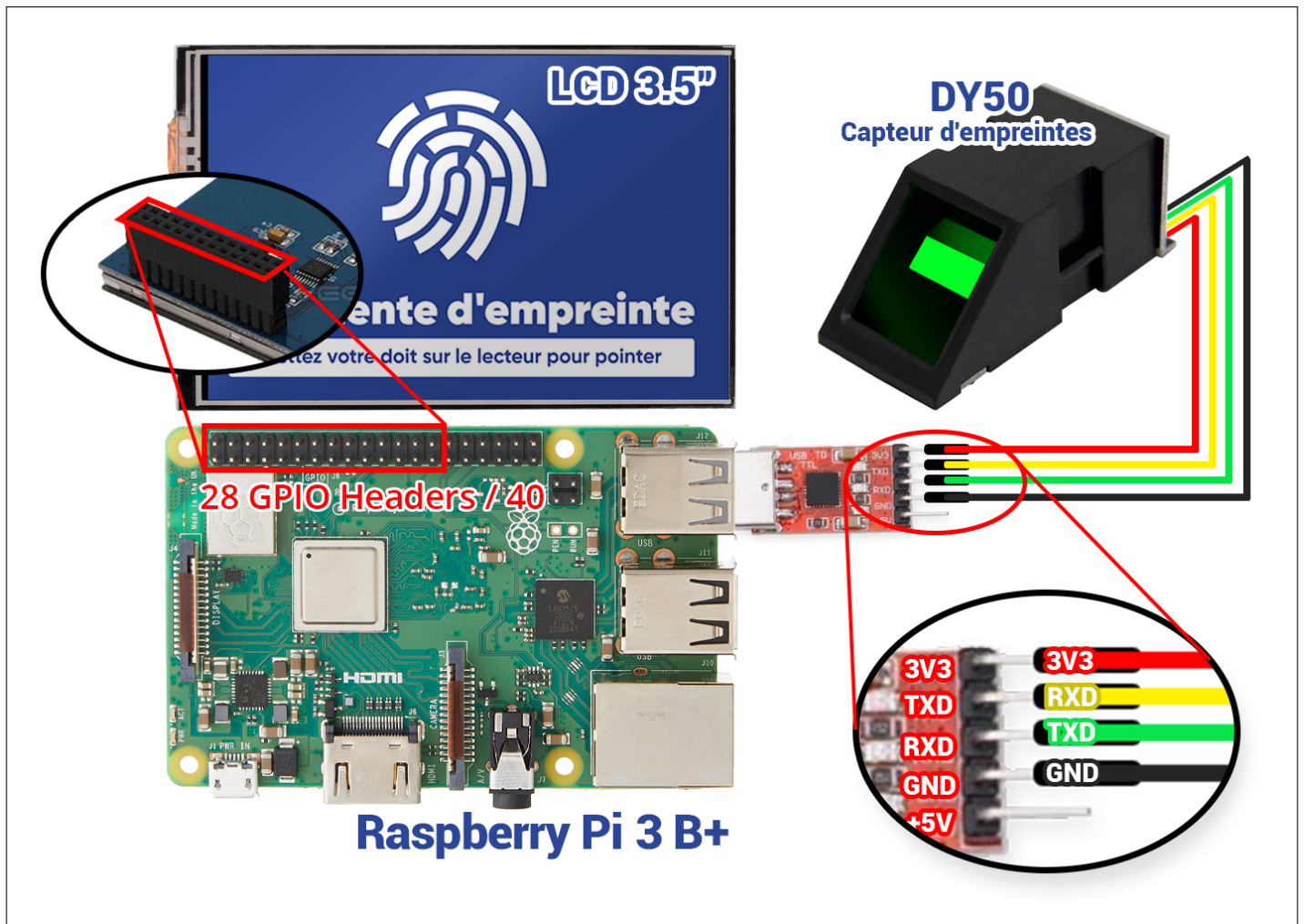


FIGURE 3.4 – Schéma expliquant le montage préliminaire des différents modules

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 pré-stockant son empreinte digitale.

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.

3.5 Environnement de développement

Un environnement de développement se réfère à une suite d'applications et d'outils qui permettent un développement d'application facile. Dans cette section nous allons présenter l'environnement de développement utilisé pour développer notre application web.

3.5.1 Plateformes utilisées

a) 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[16].



b) GitLab

GitLab est une plateforme de développement collaborative open source. Elle 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[17].



c) Asana

Créée par Dustin Moskovitz, co-fondateur de Facebook, et le programmeur Justin Rosenstein, Asana est un outil de gestion de projets, disponible sur mobile et web. L'application ASANA assure la gestion des tâches principales et accessoires d'un projet, de leur ajout jusqu'à leur archivage, ce qui inclut leur ordre de priorité, temps de traitement et suivi[18].



3.5.2 Logiciels utilisés

a) 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. En plus d'être un éditeur open source, il dispose d'un contrôle de gestion intégrée et d'un grand nombre d'extensions[15].



b) Git

Git est un système de contrôle de version distribué gratuit et open source conçu pour tout gérer, des petits aux très grands projets, avec rapidité et efficacité. 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[14].



c) Adobe Photoshop

Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur, Édité par Adobe (depuis 1992), il est principalement utilisé pour le traitement des photographies numériques, mais sert également à la création d'images et la conception graphique[19].



3.5.3 Technologies utilisées

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

3.5.4 Node JS

Node.JS est une plateforme de développement JavaScript intégrant un serveur HTTP. Son fonctionnement se base sur une boucle événementielle qui lui permet le support de fortes montées en charge. Caractérisée comme étant une bibliothèque de ce langage, elle permet la réalisation d'actions comme créer un fichier ou bien ouvrir et fermer des connexions réseau. Un point important qu'il faut noter est qu'elle n'est ni un framework ni un serveur [20]

Node JS est caractérisé comme :

- un logiciel libre sous licence MIT (licence pour logiciels libres et open source)
- un système non bloquant
- une performance du moteur de JavaScript V8 de Google qui est aussi focalisée sur la sécurité.



Cette plateforme logicielle libre conçoit des applications réseau rapides et évolutives. Elle doit élaborer ces applications en temps réel pendant que le serveur doit être capable de donner des informations au client. Elle a été conçue pour le développement d'applications côté serveur tout en disposant d'un environnement d'exécution ainsi que d'une bibliothèque JavaScript. Elle est aussi très souvent utilisée pour écrire des services côté serveur que l'on appelle API qui signifie Application Programming Interface.

3.5.5 NestJS

Nest (NestJS) est un framework permettant de créer des applications côté serveur Node.js efficaces et évolutives . Il utilise du JavaScript progressif, est construit avec et prend entièrement en charge TypeScript (mais permet toujours aux développeurs de coder en JavaScript pur) et combine des éléments de POO (programmation orientée objet), FP (programmation fonctionnelle) et FRP (programmation réactive fonctionnelle) [21].



3.5.6 Angular 11

Développé par Google, Angular est un Framework open source écrit en JavaScript qui permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications » : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action. Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités. Un type d'architecture qui a largement fait ses preuves et qui permet une forte maintenabilité et une amélioration du travail collaboratif [22].



3.5.7 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 [23]



Tkinter (Tool kit interface) est la bibliothèque graphique libre d'origine pour le langage Python, permettant la création d'interfaces graphiques [24].

3.5.8 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



3.5.9 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). Le but de CSS est séparer la structure d'un document HTML et sa présentation [26].



3.5.10 Sass

Sass est un langage de feuilles 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[27].



3.5.11 TypeScript

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Il s'agit d'un sur-ensemble syntaxique strict de JavaScript (c'est-à-dire que tout code JavaScript correct peut être utilisé avec TypeScript). Le code TypeScript est trans- compilé en JavaScript, et peut ainsi être interprété par n'importe quel navigateur web ou moteur JavaScript. TypeScript a été cocréé par Anders Hejlsberg, principal inventeur de C# [28].



3.5.12 SQLite

SQLite est une bibliothèque écrite en langage C qui propose un moteur de base de données relationnelles accessibles par le langage SQL. Contrairement aux serveurs de bases de données traditionnels, sa particularité est de ne pas reproduire le schéma habituel client-serveur mais d'être directement intégrée aux programmes. L'intégralité de la base de données (déclarations, tables, index et données) est stockée dans un fichier indépendant de la plateforme [29].



3.5.13 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[30].



3.6 Architecture du système à réaliser

Le système à à réaliser fonctionne comme suit :

- L'appareil de pointage est constitué d'un capteur d'empreinte numérique scannant l'empreinte de l'employé. Les données recueillies sont traitées avec le langage python, puis envoyées vers la base de données locale en SQLite, qui est dans le raspberry, en cas d'absence de connexion, ou synchronisé directement vers l'application de gestion (Backend). La pointeuse est aussi capable de mettre a jours les données automatiquement, si il y a eu modification dans le système de gestion de données.
- L'application frontend, conçue avec le framework Angular, communique avec l'application Backed, programmée avec le framework Nest JS, avec des requêtes HTTP. Les données sont stockées sur une base de données MySQL. C'est ce qui constitue le Système de gestion.

La figure ci-dessous représente l'architecture de notre système.

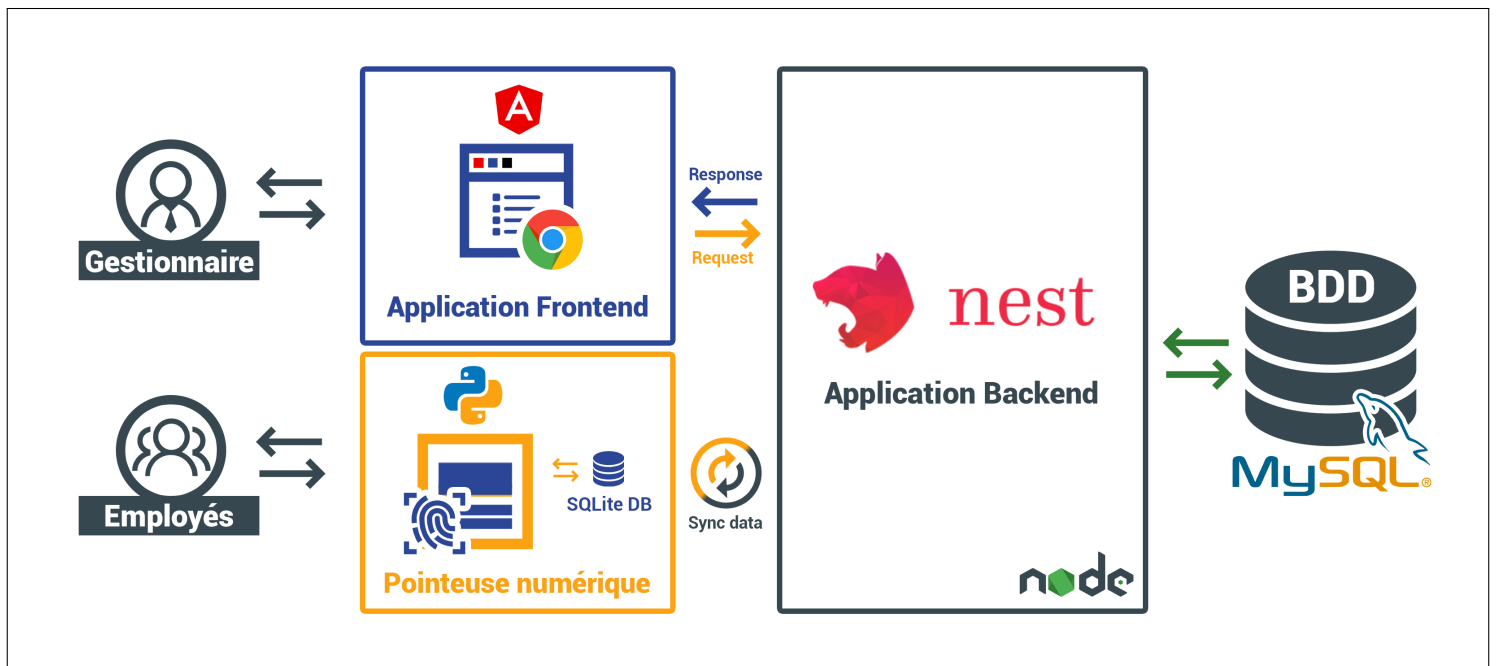


FIGURE 3.5 – Architecture du système

3.7 Conclusion

Dans ce chapitre, nous avons présenté les différents composants qui constituent notre système de pointage et les outils utilisés afin d'illustrer le branchement de ces composants. Suite à cela, nous avons cité les différents logiciels, plateformes, libraires utilisés pour la réalisation de notre application. Enfin, nous avons présenté un schéma qui explique l'architecture globale du système.

Dans le chapitre suivant, nous passerons à la phase d'implémentations , mise en service et tests.

Chapitre 4

Implémentations, mise en service et test

4.1 Introduction

Ce chapitre sera consacré à la présentation de l'application dans ces différents aspects en commençant par sa charte graphique. Nous illustrerons quelques interfaces issues de l'application. Enfin, nous aborderons brièvement la phase de tests et le côté sécurité de l'application réalisée.

4.2 Présentation de l'application

4.2.1 Nom de l'application

L'application est nommée 'RH Partner' qu'on peut traduire par "Partenaire Ressources Humaines", pour faire référence à l'accompagnement des responsables des ressources humaines dans leurs tâches quotidiens.



FIGURE 4.1 – Nom de l'application

4.2.2 Logo :

Pour le logo, c'est une forme qui associe la troisième lettre du nom et des petits pictogrammes qui représentent les ressources humaines.



FIGURE 4.2 – Logo sur fond blanc



FIGURE 4.3 – Logo sur fond noir

4.2.3 Palette de couleurs

Les couleurs choisies sont des couleurs peu nuancées afin de ne pas provoquer de sensation de fatigue au niveau des yeux lors de longues séances d'utilisation. La couleur bleue représente la fiabilité, la confiance et la responsabilité. La couleur motard inspire la réflexion et stimule l'enthousiasme et la confiance, c'est la couleur des mathématiques et de la science en général. Dans un domaine qui tourne autour des chiffres, c'est la couleur la plus appropriée pour la représenter.

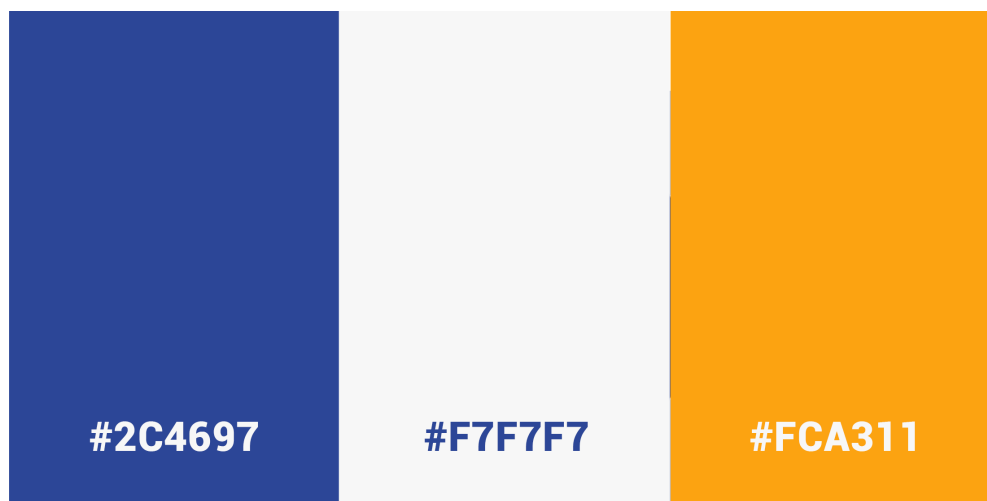


FIGURE 4.4 – Palette de couleur RH Partner

4.3 Présentation des interfaces graphiques de notre application

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

4.3.1 Espace employé

a) Écran de démarrage

Une fois la pointeuse allumée, l'interface d'initialisation s'affiche sur l'écran. Dès que tout est prêt pour enregistrer les pointages, cet écran disparaîtra.



FIGURE 4.5 – Écran de démarrage



FIGURE 4.6 – Interface d'initialisation

b) Attente de pointage

Lorsque cet écran est affiché, l'employé peut marquer son entrée ou sa sortie en scannant son empreinte digitale à l'aide du capteur d'empreintes digitales intégré.



FIGURE 4.7 – En attente de pointage

c) Pointage réussie

Cet écran montre à l'utilisateur que son entrée/sortie a été correctement marquée.



FIGURE 4.8 – Pointage réussie : Entrée



FIGURE 4.9 – Pointage réussie : Sortie

d) Échec du pointage

Lorsqu'un employé non inscrit sur l'application essaie de marquer son pointage, cet écran s'affichera.



FIGURE 4.10 – Échec du pointage

e) Mise à jour des données des employés

Cet écran est affiché ponctuellement aux utilisateurs, pour indiquer qu'il y a une synchronisation en cours avec l'application de gestion associée.

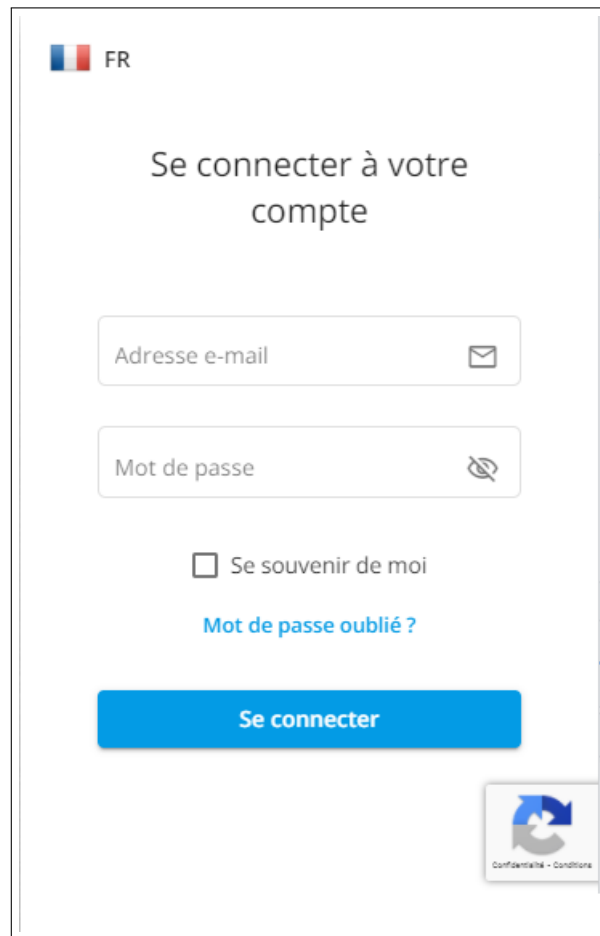


FIGURE 4.11 – Mise à jour en cours

4.3.2 Application web de gestion

a) Interface authentification

Cette interface permet aux gestionnaires possédant un compte au sein de notre application de s'authentifier



The screenshot shows a login form with the following elements:

- Top left: French flag icon and "FR".
- Center: Title "Se connecter à votre compte".
- Input fields: "Adresse e-mail" with an envelope icon and "Mot de passe" with an eye icon.
- Checkbox: "Se souvenir de moi".
- Link: "Mot de passe oublié ?" in blue text.
- Button: "Se connecter" in a blue box.
- Bottom right: "Confidentialité - Conditions" link with a circular arrow icon.

FIGURE 4.12 – Interface authentification

b) Interface menu de navigation

Cette interface représente la vitrine de notre application, elle englobe les principales fonctionnalités de notre application et joue le rôle de portail de cette dernière.

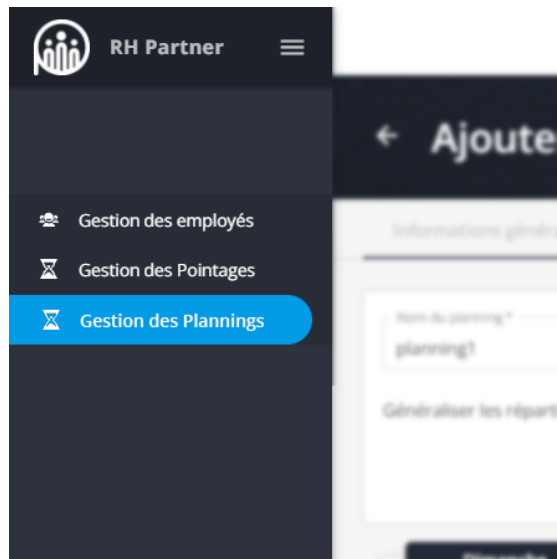


FIGURE 4.13 – Interface menu de navigation

c) Interface gestion des employés

Une fois authentifié, le gestionnaire pourra à travers cette interface lister tous les employés.

Photo	Nom prénom	Code	Profession	N° SS	Date d'entr...	Statut	Actions
<input type="checkbox"/>	TARHLISSIA Aymen	2	/	/	05/10/2021	En poste	⋮
<input type="checkbox"/>	SABRI Mehani	1	/	434343434344	05/10/2021	En poste	⋮

2 Employé(s)

RH Partner 2021. Powered by BP

FIGURE 4.14 – Interface liste d'employés

d) Interface ajouter un employé

Une fois authentifié, le gestionnaire pourra à travers cette interface qui se présente sous la forme d'un formulaire, ajouter un employé. Pour ce faire il doit renseigner les informations obligatoires.

FIGURE 4.15 – Interface d’ajout employé

e) Interface Gestion des pointages

Une fois authentifié, le gestionnaire pourra à travers cette interface lister tous les pointages des employés.

Code employé	Nom prénom	Durée	Actions
2	TARHLISSIA Aymen	Pas de pointage	⋮
1	SABRI Mehani	Pas de pointage	⋮
2 Pointage(s)			

FIGURE 4.16 – Interface gestion des pointages

f) Interface ajouter un pointage

Cette interface, qui se présente sous la forme d'un tableau, permet d'ajouter des pointages à un employé.

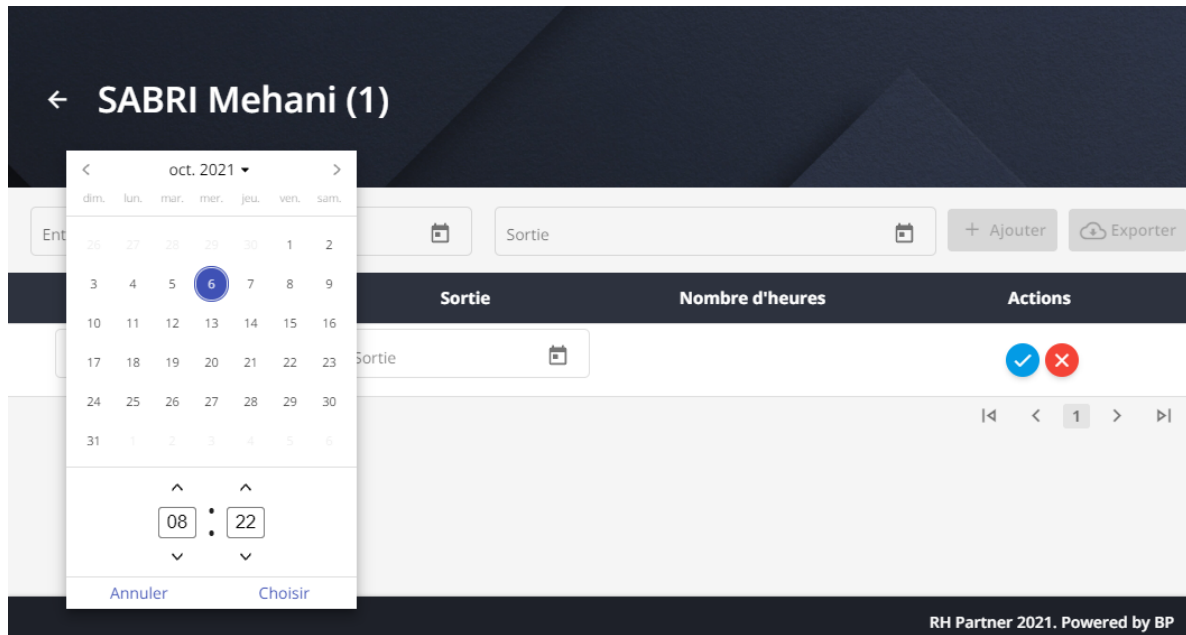


FIGURE 4.17 – Interface ajouter un pointage

g) Interface Gestion des planning

Une fois authentifié, le gestionnaire pourra à travers cette interface lister tous les plannings.

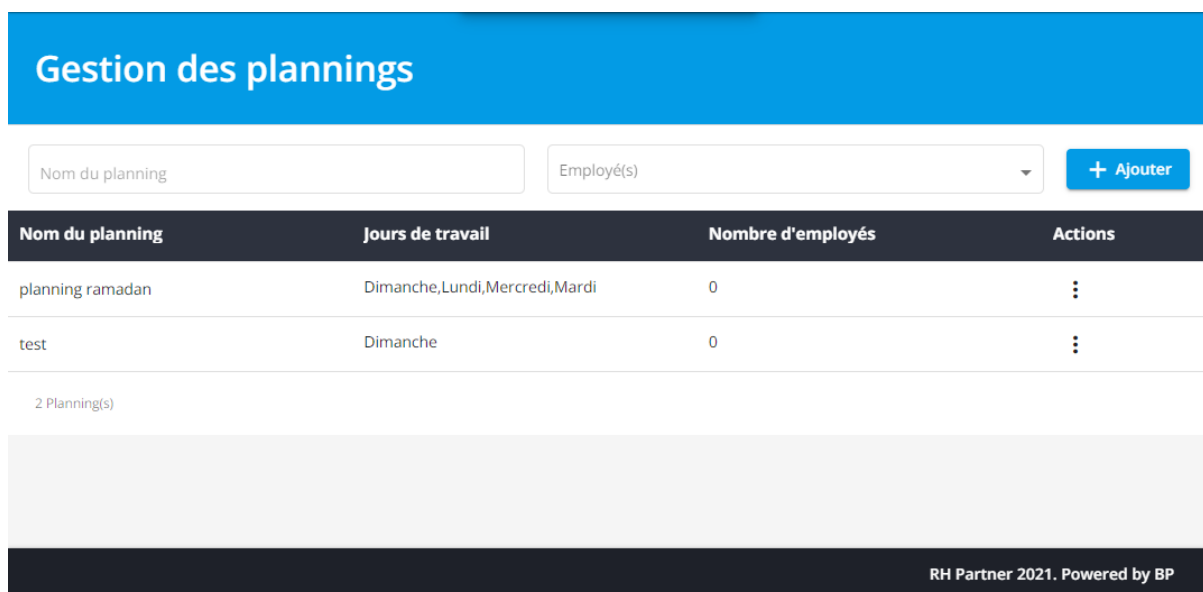


FIGURE 4.18 – Interface liste des plannings

h) Interface ajouter un planning

Les figure 4.19 et 4.20 représentent les interfaces qui permettent au gestionnaire de créer un planning. La première étape consiste à remplir un formulaire qui comporte le nom et les horaires de travail d'un employé durant la semaine. Et la deuxième consiste à affecter un employé au planning.

← **Ajouter un planning**

Informations générales Employés concernés

Nom du planning*
planning1

Veuillez sélectionner les jours de travail

Dimanche Lundi Mardi Mercredi

Jeudi Vendredi Samedi

Généraliser les répartitions sur le reste des jours

Dimanche		
Heure d'ouverture	Heure de fermeture	Actions
08:00	12:00	
13:00	17:00	
Ajouter		

Lundi		
Heure d'ouverture	Heure de fermeture	Actions
Aucune donnée		
Ajouter		

Mardi		
Heure d'ouverture	Heure de fermeture	Actions
Aucune donnée		
Ajouter		

Mercredi		
Heure d'ouverture	Heure de fermeture	Actions
Aucune donnée		
Ajouter		

Jeudi		
Heure d'ouverture	Heure de fermeture	Actions
Aucune donnée		
Ajouter		

← Précédent Étape 1 / 2 Suivant >

FIGURE 4.19 – Interface ajouter un planning

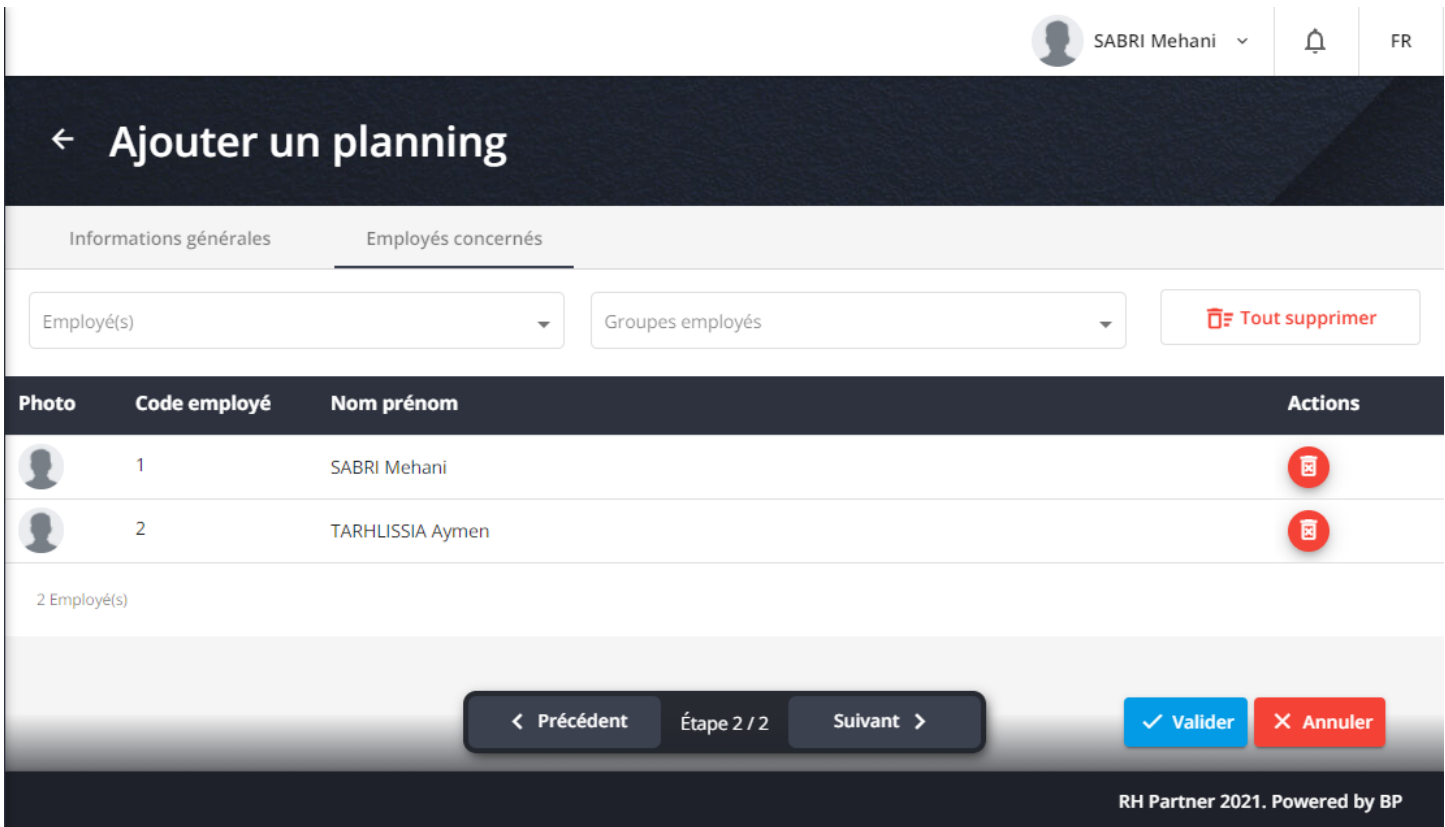


FIGURE 4.20 – Interface Affecter un planning à un employé

i) Interface supprimer un planning

C'est cette boîte de dialogue de confirmation qui s'affiche lorsque vous souhaitez supprimer un planning.

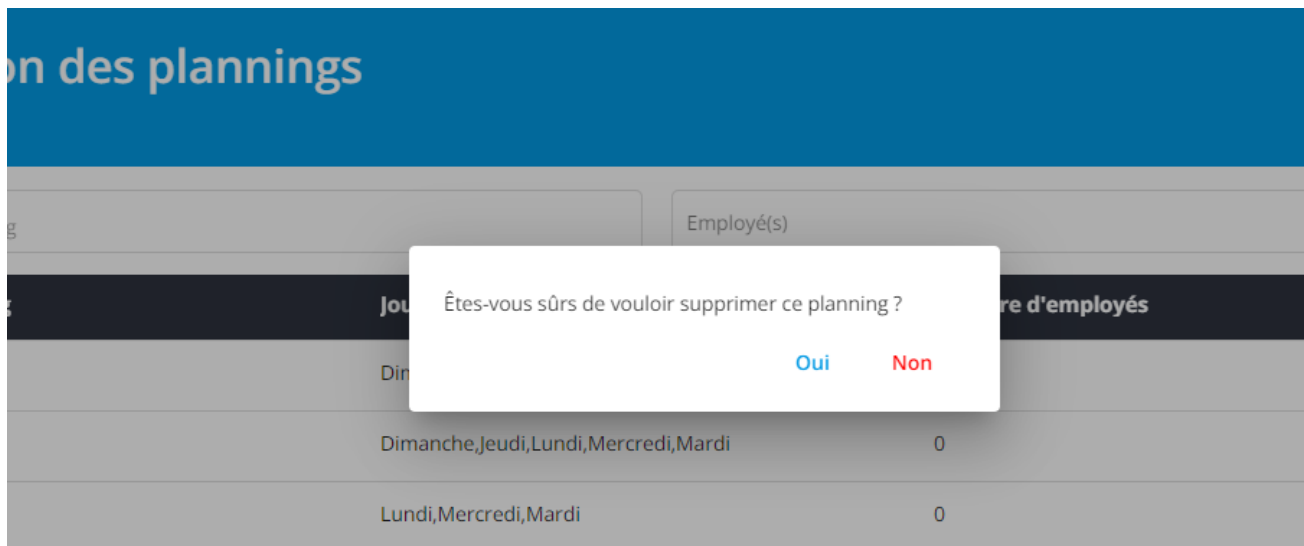


FIGURE 4.21 – Interface supprimer un planning

4.4 Sécurité de l'application

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

4.4.1 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 de données.
- Exécuter des actions arbitraires.

Node JS 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 Node JS.

4.4.2 RECAPTCHA V3

C'est un système de détection automatisée d'utilisateurs, appartenant à Google et mettant à profit les capacités de reconnaissance de ces derniers, mobilisées par les tests CAPTCHA, pour améliorer par la même occasion le processus de reconnaissance des formes par les robots. Jusqu'en 2014, il était présenté sous forme de CAPTCHA de reconnaissance de texte, et servait à la numérisation de livres, là où échouaient les systèmes de reconnaissance optique de caractères (OCR). Aujourd'hui il s'agit d'une case à cocher, suivie, si jugé nécessaire par le robot, par un CAPTCHA de reconnaissance d'images.

Nous avons utilisé RECAPTCHA V3 sur les formulaires d'authentification pour protéger l'application des utilisateurs frauduleux.

4.4.3 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.

Node JS fournit une protection contre ce type d'attaque, il permet de protéger les requêtes en les construisant à l'aide du paramétrage des requêtes.

4.5 Tests

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

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

En plus de cela, ce qui nous a le plus aidé avec les tests, ce sont les tests automatisés, que ce soit pour la partie frontend avec Angular, ou la partie backend avec le framework Nest JS. L'automatisation permet de répéter facilement et rapidement des tests individuels ou des suites de tests pendant le développement. Cela permet de garantir que les versions répondent aux objectifs de qualité et de performances.

4.6 Conclusion

Dans ce dernier chapitre, nous avons présenté notre application sous différentes facettes. Nous nous sommes ensuite intéressés à la sécurité de notre application définissant les différents aspects sécuritaires établis pour rendre notre système plus sûr lors de sa manipulation.

La dernière partie de ce chapitre présente les tests mis en place pour valider le bon fonctionnement de l'application.

Conclusion générale et perspectives

Ce travail a été réalisé dans le cadre de notre projet de fin de cycle Master en Génie logiciel. Il consisté en l'élaboration d'une pointeuse numérique à empreinte digitale constitué d'une application web et d'une pointeuse numérique. Ce système est destiné aux entreprises 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. 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. Nous avons pu ainsi utiliser différents outils et plateformes (python, Nest JS, Angular, MySQL, 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 (Raspberry Pi3 B+, capteur d'empreinte DY50, module CP2102, écran LCD 3.5", etc.).

L'application web crée permet non seulement au responsable d'avoir accès aux informations de pointages et de présences des employés en temps réel, mais 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 la pointeuse avec d'autres fonctionnalités telles que la reconnaissance faciale, comme 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] JEANTET, Aurélie et SAVIGNAC, Emmanuelle. C'est quoi l'entreprise ? Le Medef et l'initiative pédagogique. L'orientation scolaire et professionnelle, 2010, no 39/3.
- [2] Business Platforms. «Qui sommes-nous?», Disponible sur <<https://business-platforms.net/about-us/>> (consulté le 17/03/2021).
- [3] RH-Partner. «A propos», Disponible sur <<https://rh-partner.com/about-us/>> (consulté le 19/03/2021).
- [4] RH-Partner. «UNE SOLUTION INCLUSIVE POUR UNE COMMUNICATION FLUIDE», Disponible sur <<https://rh-partner.com/portail-employe/>> (consulté le 27/03/2021).
- [5] admin4876. «Pourquoi choisir un logiciel en SaaS ?» Demain Un Autre Jour (blog), 1 février 2019. Consulté le 25 mars 2021. <https://www.demainunautrejour.com/pourquoi-choisir-un-logiciel-en-saas/>.
- [6] PARRAIN, Fabien. Capteur intégré tactile d'empreintes digitales à microstructures piezorésistives. Thèse de doctorat, Institut National du Pétrole et du Gaz, 2002.
- [7] Reconnaissance faciale automatisée dans les secteurs public et privé, édition mars 2013.
- [8] AUBRY, Claude. Scrum. Le guide pratique de la méthode Agile la plus populaire. Malakoff, France : Dunod Editions, 2011., 2ème édition.
- [9] Roques, Pascal. Les Cahiers du programmeur UML : modéliser une application web. Paris : Eyrolles, 2002.
- [10] Roques, Pascal. Les Cahiers du programmeur UML : modéliser une application web. Paris : Eyrolles, 2008, 4ème édition.
- [11] Pascal, Roques, Franck Vallée. UML 2 en action de l'analyse des besoins à la conception. Paris : Eyrolles, 2007, 4ème édition.
- [12] Bouchlaghem S, Cherifi F, Khanouche F, Maouche L, Zebboudj s, Application Web JAVA EE pour la gestion d'un laboratoire de recherche scientifique, licence en informatique générale, Université A/Mira Béjaia, 2014.
- [13] Joseph Gabay, J. G., David Gabay, D. G. UML 2 Analyse et conception. Dunod, 2008.
- [14] Git, Disponible sur <<https://git-scm.com/>> (consulté le 05/08/2021)
- [15] Visual Code, Disponible sur <<https://code.visualstudio.com>> (consulté le 12/08/2021).
- [16] slack, Disponible sur <<https://slack.com/intl/fr-dz/features>> (consulté le 12/04/2021).

-
- [17] GitLab, Disponible sur <<https://about.gitlab.com/fr-fr/>> (consulté le 12/04/2021).
 - [18] Asana, Disponible sur <<https://asana.com/fr/features>> (consulté le 12/04/2021).
 - [19] Adobe photoshop, Disponible sur <<https://www.adobe.com/fr/products/photoshop.html>> (consulté le 1/10/2021)
 - [20] Node JS, Disponible sur <<https://nodejs.org/fr>> (consulté le 29/09/2021)
 - [21] NesJS, Disponible sur <<https://docs.nestjs.com/>> (consulté le 29/09/2021)
 - [22] Angular, Disponible sur <<https://angular.io/features>> (consulté le 1/10/2021)
 - [23] Python, Disponible sur <<https://docs.python.org/3.9/tutorial/>>, (consulté le 6/07/2021).
 - [24] Tkinter, Disponible sur <<https://docs.python.org/fr/3/library/tkinter.html>>, (consulté le 6/07/2021).
 - [25] world wide web, HTML 5.2, Disponible sur <<https://www.w3.org/TR/2017/REC-html52-20171214/introduction.html>>, (consulté le 3/10/2021).
 - [26] CSS3, Disponible sur <<https://www.w3.org/Style/CSS/Overview.en.html>>, (consulté le 3/10/2021).
 - [27] Sass, Disponible sur <<https://sass-lang.com/>>, (consulté le 3/10/2021).
 - [28] TypeScript, Disponible sur <<https://www.typescriptlang.org/>>, (consulté le 3/10/2021).
 - [29] Sqlite, Disponible sur <<https://www.sqlite.org/index.html>>, (consulté le 3/10/2021).
 - [30] MySQL, Disponible sur <<https://www.mysql.com/fr/>> (consulté le 3/10/2021).
 - [31] Ltd, Raspberry Pi, Disponible sur <<https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>>, (consulté le 3/10/2021).
 - [32] Capture DY50, Disponible sur <<https://forums.raspberrypi.com/viewtopic.php?t=254890/>>, (consulté le 3/10/2021).
 - [33] Module CP2102, Disponible sur <<https://amzn.to/3mNbdEs/>>, (consulté le 3/10/2021).

RÉSUMÉ

Ce document a été rédigé en vue de l'obtention d'un diplôme de Master en Génie Logiciel. La thématique traitée est la conception et réalisation d'une pointeuse numérique. Suite à une analyse des pointeuses déjà existantes sur le marché, une solution a été proposée. Composée d'une application web et d'une pointeuse numérique, cette alternative offrira aux entreprises la possibilité de répondre à leurs besoins 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 Nest JS et Angular, ainsi que d'autres bibliothèques pour l'implémentation de l'application web. Quant à la pointeuse, elle est composée du raspberry Pi3 B+ dans lequel nous avons injecté un programme en Python, d'un capteur d'empreinte biométrique et d'un écran LCD.

Mots clés : *UML, web, Python, Angular, Nest JS, raspberry, pointeuse, LCD, biométrique*

ABSTRACT

This document was written for a Master's degree in Software Engineering. The theme dealt with is the design and production of a digital attendance system. Following an analysis of the existing products on the market, a solution was proposed. Comprised of a web application and a digital attendance system, this alternative will provide companies with the ability to meet their needs more reliably and efficiently. To make this happen, we used a development process and the UML modeling language for the analysis and design phases. Regarding the phase of achievement, it was characterized by the use of Nest JS and Angular, as well as other libraries for the implementation of the web application. As for the digital attendance system, it is composed of a raspberry Pi3 B + (in which we have injected a python program), a biometric fingerprint sensor and an LCD screen.

Key words : *UML, web, Python, Angular, Nest JS, raspberry, pointing, LCD, biometric*