Table des matières

I. INTRODUCTION GENERALE 3

I.1 CONTEXTE 4

I.2 ORGANISME D'ACCUEIL 5

I.3 PROBLEMATIQUE 5

I.4 OBJECTIFS 5

I.5 PLAN DU MEMOIRE 6

II. CHAPITRE I L'ERP ODOO 7

II.1 INTRODUCTION 8

II.2 LES ERP 8

II.2.1 Différents ERP 9

II.2.2 Evolution des systèmes ERP 9

II.3 ODOO 10

II.3.1 Architecture modulaire d'Odoo 11

II.3.2 Architecture technique 12

II.3.3 Modèle Vue Contrôleur 13

II.3.4 Structure d'un module Odoo 13

II.3.5 Une communauté dynamique 14

II.4 EXEMPLE D'UN MODULE 15

II.4.1 Règles salariales 15

II.4.2 Structure des salaires 16

II.4.3 Bulletins de l'employé 17

II.4.4 Inconvénients 18

II.5 CONCLUSION 18

III. CHAPITRE 2 LA TECHNOLOGIE RFID 19

III.1 INTRODUCTION 20

III.2 LES DIFFERENTES TECHNOLOGIES D'IDENTIFICATION 20

III.3 RFID 23

III.4 TYPES DE RFID 23

III.4.1 RFID active 23

III.4.2 RFID passive *23*
III.4.3 Les fréquences de transfert de données *23*
III.5 COMPOSANTS DES SYSTEMES RFID 24
III.5.1 Tags *25*
III.5.2 Lecteurs *25*
III.6 FONCTIONNEMENT DE LA RFID 26
III.7 DOMAINES D'APPLICATION DE LA RFID 27
III.8 LES ASPECTS JURIDIQUES DE LA TECHNOLOGIE RFID 29
III.9 AVANTAGES 29
III.10 INCONVENIENT 29
III.11 CONCLUSION 29
IV. CHAPITRE III REALISATION DU PROJET 30
IV.1 INTRODUCTION 31
IV.2 LES LANGAGES ET OUTILS UTILISES 31
IV.3 COMPOSANTS ELECTRONIQUES UTILISES 33
IV.4 ADAPTATION DES MODULES ODOO 34
IV.5 GESTION DU PROJET 34
IV.6 APPRENTISSAGE DE L'ASPECT FONCTIONNEL (SPRINT 1) 38
IV.7 APPRENTISSAGE TECHNIQUE (SPRINT 2) 39
IV.7.1 Analyse des exigences (Sprint 3) *39*
IV.7.2 Diagramme de composants *39*
IV.8 REALISATION 40
IV.8.1 Choix du lecteur RFID (sprint 3) *40*
IV.8.2 Connexion du lecteur RFID avec arduino (sprint 4) *42*
IV.8.3 Utilisation d'Arduino (sprint 4) *44*
IV.8.4 Exploitation conjointe de l'outil Processing avec Arduino *45*
IV.8.4.1 Connexion avec arduino (sprint 5) 45
IV.8.4.2 Récupération des données d'arduino avec processing 46
IV.8.4.3 Enregistrement dans un fichier csv (sprint 6) 47
IV.8.4.3.1 Programme dans processing 47
IV.8.4.3.2 Programme dans python 47
IV.8.4.4 Enregistrement dans la base de données (sprint 7) 49
IV.8.4.5 Calcul de la paie à partir des heures travaillées (Sprint 8) 54
IV.9 CONCLUSION 54

V. CONCLUSION GENERALE 55

VI. RÉFÉRENCES 57

VII. LISTE DES FIGURES 60

VIII. LISTE DES TABLEAUX 62

Introduction Générale

I.1 Contexte

Depuis quelques décennies, nous assistons à une grande révolution du monde informatique. Cette dernière a envahie tous les domaines de la vie quotidienne. Aussi, les systèmes informatiques jouent un rôle de plus en plus important dans notre société. Les ordinateurs se sont rapidement améliorés et démocratisés. Aujourd'hui, notre société est de plus en plus dépendante de ces systèmes informatiques.

À l'aide de ces systèmes et des nouvelles technologies, de nombreuses entreprises ont pu s'affirmer par des gains de compétitivité, au sein de marchés de plus en plus exigeants. L'investissement dans ces nouvelles technologies apparaît clairement comme l'un des principaux moteurs de développement au sein des entreprises quelle que soit leur taille.

Un gain de productivité pour ces entreprises est synonyme d'un gain de temps. De ce fait, ces derniers ont eu besoin de piloter les entrées sorties de leurs employés d'une manière facile et efficace. Ils étaient donc contraints de chercher une technologie qui les aidera à relever ce défi. Parmi les technologies qui assurent le pointage on trouve la reconnaissance faciale, les empreintes digitales, la RFID, etc.

Ces derniers temps, nous constatons que la technologie RFID est la plus répandue dans différents domaines en raison de son efficacité et de son usage plus ou moins facile. C'est pourquoi nous nous sommes intéressées à cette technologie.

C'est dans ce cadre que s'inscrit notre projet de fin d'étude de master en informatique option génie logiciel. Ce dernier consiste à intégrer une solution RFID connectée au serveur Odoo pour la gestion du pointage. Odoo anciennement nommé OpenERP est une plateforme informatique qui permet de gérer et suivre au quotidien les différents services de l'entreprise.

Pour ce faire, nous nous sommes rendues à la société SOGESI où nous avons été très bien accueillies. Cette société permet à ses clients d'avoir des solutions informatiques adéquates à leurs besoins.

I.2 Organisme d'accueil

SOGESI est une société basée en Algérie qui a vu le jour en 2013. Elle est spécialisée dans les services informatiques et d'ingénierie de proximité, la gestion des entreprises et formations.

C'est une Société de Services en Ingénierie Informatique (SSII) Algérienne, qui offre des solutions de la gestion des applicatifs et des infrastructures ainsi que le conseil en technologie pour les PME/PMI publics et privés.

L'approche sectorielle permet à la société de proposer de nouvelles offres structurées pour chaque secteur, et de capitaliser sur l'expérience forte de ses collaborateurs en France pour se développer en Algérie et proposer des solutions à des prix très compétitifs.

Elle permet également de choisir la prestation qui convient le plus grâce à une flexibilité due à l'organisation des métiers :

- Intégration et déploiement de solutions ERP
- Audit, conseil et gestion de projets
- Formation et délégation de personnel [1].

Elle a apporté ses services à plusieurs entreprises. Parmi eux nous retrouvons : Socope, Mobilis, Promedal, TECLIB GROUP etc.

I.3 Problématique

Un grand nombre de sociétés rencontrent des problèmes liés à la gestion de la paie en fonction du pointage des employés et SOGESI en fait partie.

De nombreuses sociétés utilisant Odoo, n'ont pas un dispositif qui peut signaler les entrées et les sorties de leurs employés. Il faut que chaque employé se dirige vers la plateforme Odoo et se connecte pour pointer son entrée ou sortie manuellement. Ceci peut générer beaucoup de mal entendus entre les employés et les responsables, par exemple un employé qui a oublié de pointer alors qu'il était présent.

I.4 Objectifs

Notre projet a pour objectif d'intégrer au monde des ERP une nouvelle technologie qui se nomme RFID. Ce qui va permettre d'assurer l'identification et la traçabilité des employés pour pouvoir calculer la paie en fonction des heures travaillées. Il s'agit d'une technologie largement déployée actuellement et qui ne cesse de croître dans de multiples domaines. Elle va nous permettre de leur faciliter la tâche de pointage et d'être traitée automatiquement.

À partir des enregistrements du pointage que nous allons récupérer, on va calculer la paie des employés en se basant sur les horaires réellement travaillées par chacun d'entre eux.

Pour réaliser cette solution nous devons assurer les objectifs suivants :

- Trouver le dispositif qui propose une meilleure solution par rapport à notre besoin.
- Chercher une solution qui permettra de connecter ce dispositif au serveur Odoo.
- Enregistrer les données récupérées du dispositif dans la base de données.
- Récupérer les informations de la base de données et calculer la paie sur la base de ces dernières.

I.5 Plan du mémoire

Nous avons structuré notre manuscrit comme suit :

- Une introduction générale qui présente le contexte, la problématique et les objectifs de notre travail.
- Nous décrivons dans le premier chapitre le monde des ERP ensuite Odoo en particulier.
- Dans le second chapitre, nous exposons les différentes technologies d'identification, la RFID en particulier, ses composants, son fonctionnement, les différents types de la RFID et enfin ses avantages ainsi que ses inconvénients.
- Nous entamons dans le troisième chapitre la partie réalisation de notre projet en passant par sa planification.
- Nous terminons ce rapport par une conclusion générale suivie de quelques perspectives.

Rapport-Gratuit.com

Chapitre I *L'ERP Odoo*

II.1 Introduction

De nos jours, les entreprises rencontrent beaucoup de problèmes et de difficultés qui concernent la gestion quotidienne et le maintien de ses différents services. L'entreprise voit aujourd'hui de nombreux avantages à adopter un progiciel de gestion intégré.

Pour faire face à ces différents problèmes et améliorer leurs performances et leur productivité, les entreprises font appel aux ERP qui offrent des solutions à ces problèmes et permettent d'avoir une bonne gestion des ressources internes de l'entreprise.

II.2 Les ERP

« On définit par ERP "Entreprise Ressource Planning" traduit en français par "Progiciel de Gestion Intégré ou PGI", un sous-ensemble du système d'information capable de prendre en charge la gestion intégrale de l'entreprise incluant la gestion comptable et financière, la gestion de la production et de la logistique, la gestion des ressources humaines, la gestion administrative ainsi la gestion des ventes et des achats » [2].

La planification des ressources d'entreprise (ERP) est un logiciel de gestion des processus métiers qui permet à une organisation d'utiliser un système d'applications intégrées pour gérer l'entreprise et automatiser de nombreuses fonctions liées à la technologie, aux services et aux ressources humaines. Le logiciel ERP intègre toutes les facettes d'une opération y compris la planification, le développement, la fabrication, la vente et le marketing de produits dans une seule base de données [3].

Un ERP répond aux caractéristiques suivantes :

- **Intégration :** les fournisseurs ERP promettent une intégration complète de l'ensemble de l'information.
- **Flexibilité :** elle est importante dans le cas des systèmes ERP, compte tenu de la taille de l'investissement dont ils ont besoin. Si un système ERP n'était pas adaptable, il limiterait le potentiel de développement de l'organisation dans un environnement changeant.
- **Evolutivité :** L'ERP n'est pas une solution constante. Il change au fil du temps en termes de services et d'architecture comme toute autre technologie de l'information [4].

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Les domaines de compétence liés aux ERP sont les suivantes :

- Gestion de la production.
- Gestion des stocks, des approvisionnements et des achats.
- Gestion commerciale.
- Gestion des ressources humaines.
- Gestion comptable et financière.
- La gestion de production (planification, …).
- La gestion du personnel et de la paie (GRH).

II.2.1 Différents ERP

On distingue deux types d'ERP :

- Le premier type concerne les logiciels propriétaires qui impliquent l'achat d'une licence.

Figure 1 : logiciels propriétaires

- Le deuxième type comprend les logiciels gratuits et open-source.

Figure 2 : logiciels gratuits

Nous nous intéressons à la catégorie des ERP open-source et plus particulièrement à Odoo.

II.2.2 Evolution des systèmes ERP

L'évolution des systèmes ERP a suivi de près les développements spectaculaires dans le domaine du matériel informatique et des systèmes logiciels.

Au cours des années 1960, la plupart des organisations ont conçu, développé et mis en place des systèmes informatiques.

Les systèmes de planification des besoins (MRP) ont été développés dans les années 1970 qui impliquaient principalement la planification des besoins en produits en fonction du programme de production principal.

En suivant cette voie, de nouveaux logiciels appelés planification des ressources de fabrication (MRP II) ont été introduites dans les années 1980, en mettant l'accent sur l'optimisation des processus de fabrication. MRP II a inclus des zones telles que la gestion de la distribution, la gestion de projet, finance, ressources humaines et ingénierie.

Les systèmes ERP sont apparue à la fin des années 1980 et au début des années 1990 avec le pouvoir de l'entreprise de coordination et d'intégration inter fonctionnelles.

Au cours des années 1990, les fournisseurs ERP ont ajouté plus de modules et de fonctions aux modules de base donnant naissance aux "ERP étendus" [5].

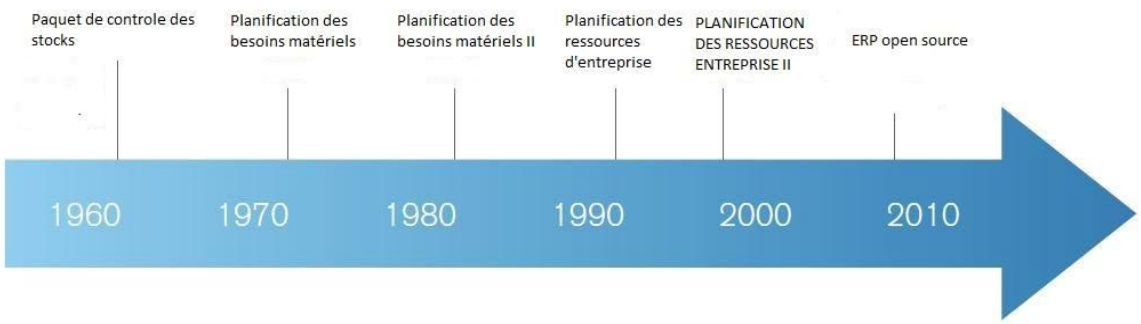


Figure 3 : L'évolution des ERP

II.3 Odoo

Odoo, anciennement connu sous le nom d'OpenERP, est un éditeur de logiciels open source fondé en 2004. Qui propose une suite complète de modules de gestion d'entreprise entièrement intégrés [6].

Odoo représente la nouvelle génération des ERP avec sa modularité extrême qui fait sa force, sa facilité de prise en main, sa compatibilité avec les technologies les plus modernes ou encore son modèle de développement communautaire qui compte 1500 contributeurs.

Odoo est utilisé par les grandes entreprises, les petites entreprises, les associations et autres types d'organisations. La structure d'Oddo est unique par les fondations techniques solides qu'elle présente [7].
Odoo est caractérisé par son architecture modulaire et technique. Cette architecture facilite l'intégration transparente de modules, fournissant un flux d'informations entre toutes les fonctions de l'entreprise de manière visible. De cette manière, Odoo évolue plus rapidement que n'importe quelle autre solution [8].

II.3.1 Architecture modulaire d'Odoo

Avec plus de 3000 modules, Odoo couvre de multiples domaines de gestion et offre la flexibilité de s'organiser en fonction des besoins tout en partageant une base de données unique. Ce qui élimine les saisies multiples et évite l'ambiguïté des données de même nature.

Odoo permet aussi de développer facilement de nouveaux modules ou encore de connecter l'application à n'importe quel portail web ou logiciel grâce à ses API prêtes à l'emploie.

L'architecture modulaire des ERP lui permet de couvrir plusieurs domaines illustrés dans la figure ci-dessous :

Figure 4 : Les différentes domaines des ERP

II.3.2 Architecture technique

Architecture client/serveur

Odoo est formé de trois composants principaux :

- *Le serveur de base de données PostgreSQL*, qui contient toutes les bases de données, dont chacune contient toutes les données et la plupart des éléments de la configuration du système.
- *Le serveur d'applications Open ERP*, qui contient toute la logique d'entreprise et assure un fonctionnement optimal.
- *Le serveur Web*, une application séparée appelée le client Web Open Object, ce qui nous permet de nous connecter à OpenERP grâce à des navigateurs Web standard [9].

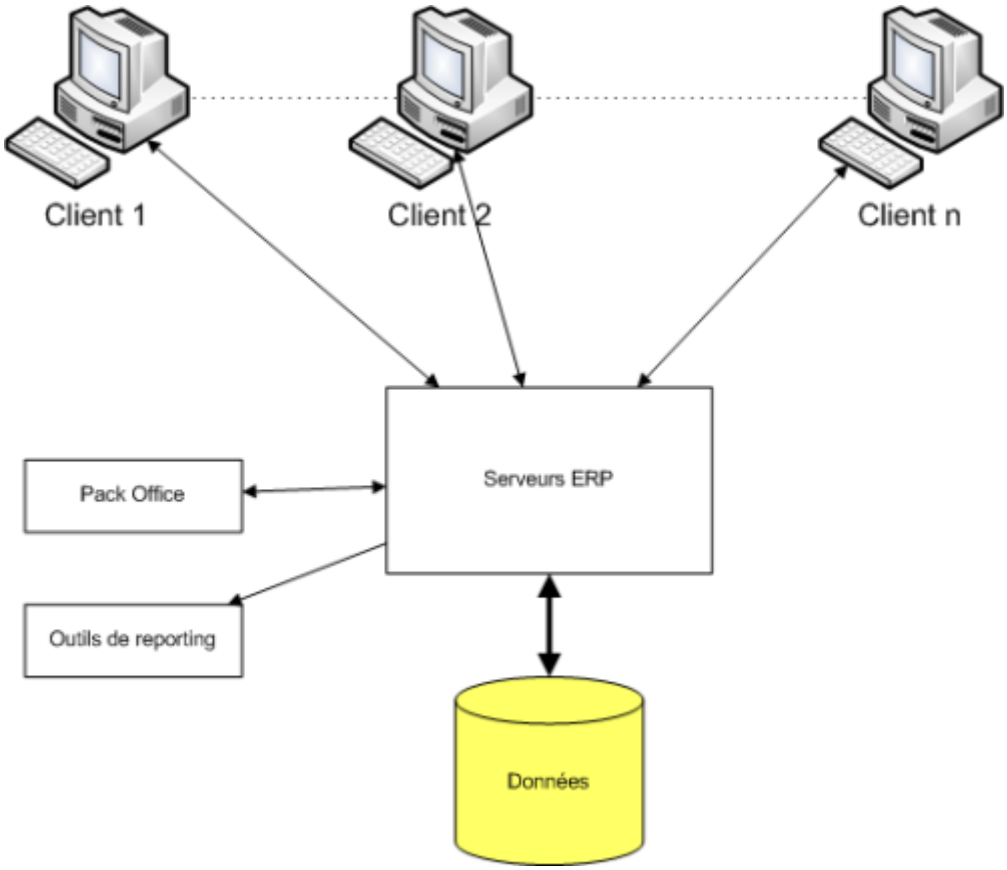
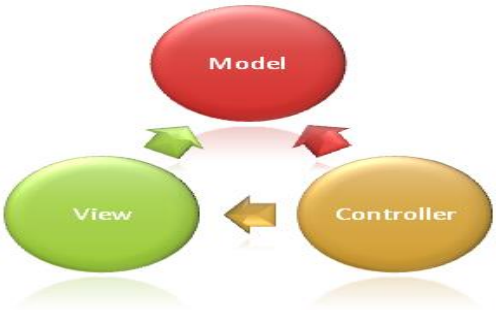


Figure 5 : Architecture clients/serveur

II.3.3 Modèle Vue Contrôleur



En génie logiciel, le MVC (Model Vue Controller) est un modèle de conception fondamental pour la séparation de la logique de l'interface utilisateur de la logique métier.

Quand nous avons des applications qui présentent des lots de données, nous souhaitons séparer l'interface utilisateur (vue) et les données (modèle). Le MVC résout ce problème de sorte que les changements à l'interface utilisateur n'affectent pas le traitement des données. Cela se fait en découplant l'accès des données et la logique des applications de la présentation des données et de l'interaction utilisateur avec l'intermédiaire d'un contrôleur [10].

Odoo suit le modèle-vue-contrôleur (MVC) comme modèle architectural :

- Modèle : ce sont les tables PostgreSQL.
- Vue : des vues sont définies dans des fichiers XML dans OpenERP.
- Contrôleur : Ce sont les objets d'OpenERP [11].

II.3.4 Structure d'un module Odoo

Pour créer un module Odoo, il est nécessaire de créer des fichiers obligatoires, qui permettent à Odoo de lancer le module et d'autres répertoires facultatif comme suit :

- Un répertoire **i18n** contient les fichiers de traduction du module.
- Un répertoire **security** contient les fichiers de contrôle d'accès et les règles pour les enregistrements.
- Un répertoire **static** contient la partie « Web » du module.
- À la racine du module, nous trouvons les fichiers Python ainsi que les vues XML.
- Il y a trois fichiers obligatoires lorsque nous créons un module.

__init__.py

__openerp__.py

module.py

La figure ci-dessous montre la structure d'un module Odoo :

Obligatoires	__init__.py	Initialisation du package Python
	__openerp__.py	La fiche descriptive du module
	module.py	Les classes Python, les objets du module
	module_view.xml	Les vues (Formulaires, listes, graph,...), menus et actions
	demo	Données de test et de démonstration
Optionnels	i18n	Fichiers de traduction – Internationalisation
	report	Définition des rapports
	security	Déclaration des groupes et des droits d'accès
	workflow	Définition des workflow
	wizard	Définition des assistants

Tableau 1 : La structure d'un module Odoo

II.3.5 Une communauté dynamique

Odoo Community Association (OCA), un organisme à but non lucratif dans le but de promouvoir l'utilisation répandue d'Odoo et de soutenir le développement collaboratif des fonctionnalités. Il fournit également un soutien financier, organisationnel et juridique à la communauté Odoo Open Source.

Cette communauté offre des démos en ligne, des téléchargements, un moyen rapide de comparer les différentes éditions, un service d'assistance, un forum, des guides d'utilisation et plus pour ceux qui utilisent les applications. Il existe également des sections spécifiques pour les développeurs et les partenaires engagés avec l'entreprise.

La dynamique est très forte, portée par la virilité de l'open source, mais aussi par un marketing efficace autour du produit et une diffusion internationale sans limite. Sans compter bien sur les capacités du produit. Odoo s'appuie sur un réseau mondial de partenaires présents dans 110 pays.

Figure 6 : la communauté d'Odoo

II.4 Exemple d'un module

Parmi les modules qui existent dans Odoo, nous trouvons le module de la paie. C'est un module qui permet une gestion de la paie comprenant les fonctionnalités suivantes :

II.4.1 Règles salariales

Elles sont utilisées pour calculer des données liées au calcul des salaires telles que les déductions, les primes, etc.

Pour créer une règle de salaire, il faut aller au menu : **Ressources humaines > Configuration > Paye > Règles Salariales.**

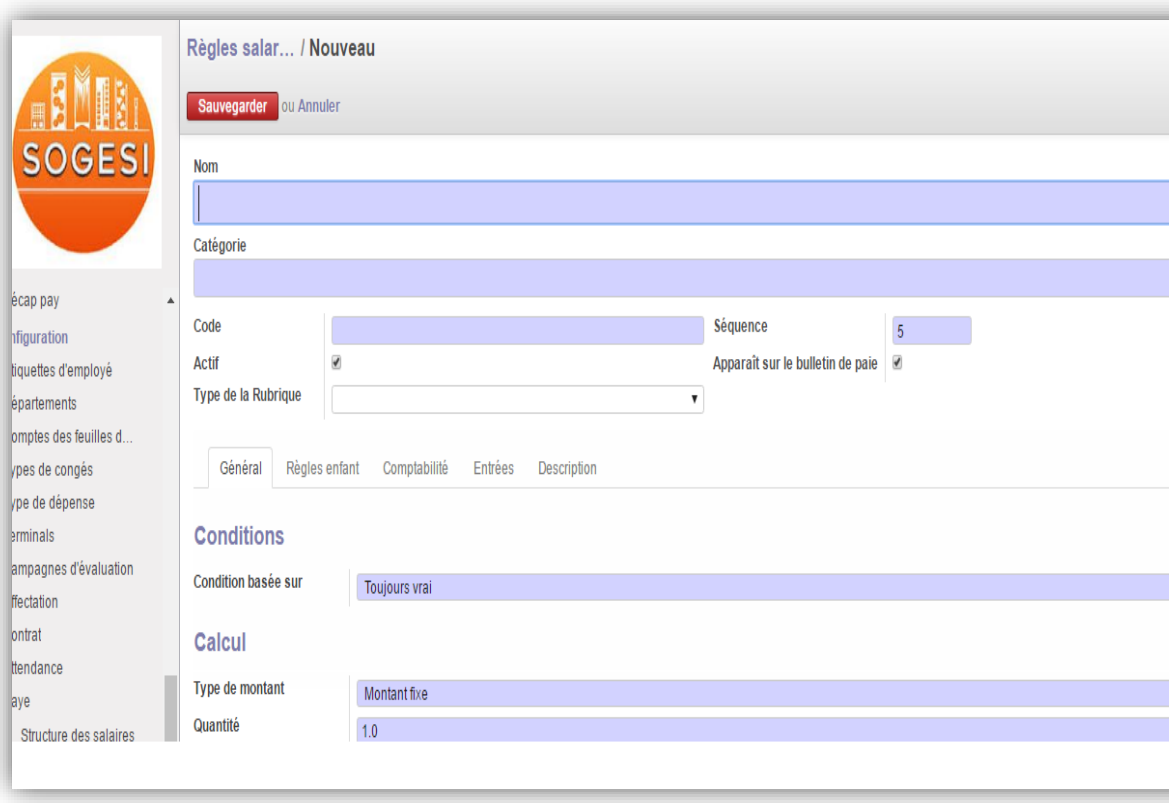


Figure 7 : Création d'une règle salariale

Pour définir une règle salariale nous avons entré les informations suivantes :

- Nom : le nom de la règle salariale.
- Code : le code de la règle salariale.
- Catégorie : choisir la catégorie à laquelle la règle salariale appartient.
- Condition basée sur : la règle sera prise en compte si cette condition est vérifiée.

- Type de montant : Il existe 3 types de calcul pour déterminer le montant de la règle :
  - Montant fixe : le montant est fixé dès le départ.
  - Pourcentage : il fait le calcul des montants par pourcentage. Il faut donner la valeur de base et le pourcentage.
  - Code python : le montant peut être exprimé par un code python. Par exemple : result=contract.wage.

II.4.2 Structure des salaires

Elle permet de choisir l'ensemble des règles salariales qui vont entrer en jeu, pour calculer les bulletins de salaire.

Nous pouvons définir une structure salariale en se rendant sur : **Ressources humaines > Configuration > Paye > Structure des salaires.**

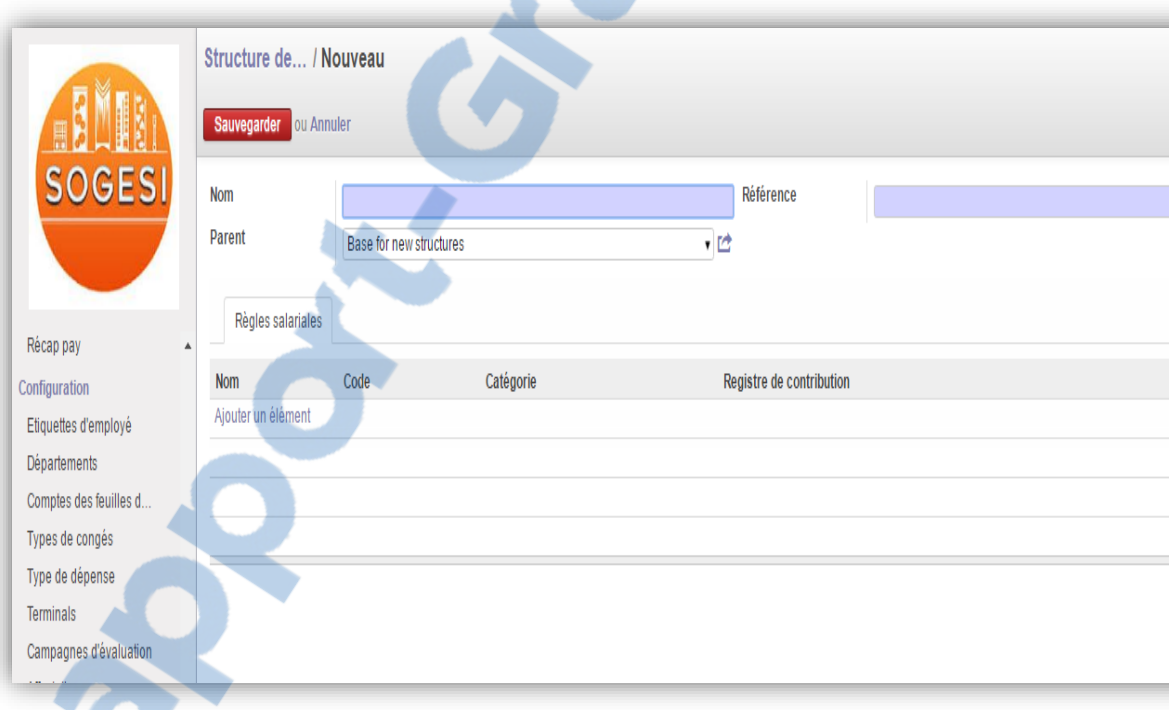


Figure 8 : Structure salariale

Nous enregistrons les données suivantes :

- ➔ Nom : le nom de la structure salariale.
- ➔ Référence : le code de cette structure.
- ➔ Règles salariales : Ajouter les règles salariales qui vont constituer votre structure.

Rapport-Gratuit.com

II.4.3 Bulletins de l'employé

Nous pouvons définir un bulletin de l'employé en se rendant sur : **Ressources humaines > Paye > Bulletins de l'employé.**

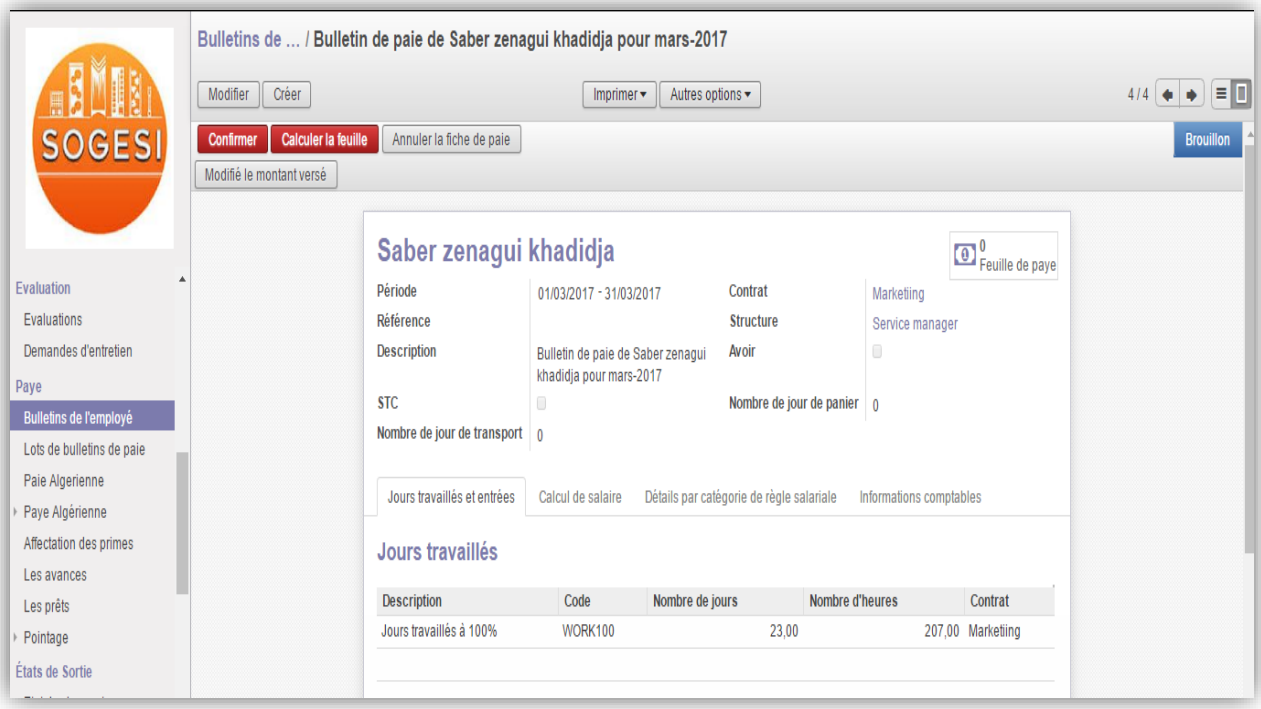


Figure 9 : Bulletin de paie de l'employé

- ➔ Employé : choisir un employé.
- ➔ Référence : le numéro du bulletin.
- ➔ Contrat : sélectionnez le contrat pour chaque employé (un contrat pour chaque employé).

Quand nous choisissons un employé, les champs de référence, structure, contrat, jours travaillés seront remplis automatiquement à partir du contrat et non pas des heures travaillées récupérées de la feuille de temps.

En cliquant sur le bouton calculer la feuille, les lignes du bulletin de salaire seront remplies. L'onglet jours travaillés et entrées affiche le nombre de jours travaillés durant la période indiquée dans le bulletin, ainsi que le nombre des absences.

II.4.4 Inconvénients

Même si le module "hr_payroll" existe déjà dans Odoo. Ce module à lui seul ne permet pas de satisfaire convenablement les besoins des entreprises. En effet, il ne pourra pas calculer la paie en fonction des heures travaillées.

II.5 Conclusion

Dans ce chapitre, nous avons présenté les ERP en général, puis Odoo en particulier en présentant son architecture, mais aussi la structure d'un module. Par la suite, nous avons réalisé une démonstration d'un exemple de module de gestion de la paie que nous exploiterons plus tard dans le chapitre 3.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Chapitre 2 La technologie RFID

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

III.1 Introduction

Depuis quelques dizaines d'années, nous assistons à une révolution des méthodes de travail et à des usages et pratiques principalement liées aux nouvelles technologies.

Ces dernières ont forgé une place indispensable dans nos vies et dans le fonctionnement de l'entreprise. En effet, actuellement très peu, voire même aucune entreprise ne peut se passer des nouvelles technologies pour se simplifier le travail en utilisant, par exemple, un simple traitement de texte, des logiciels de gestion du temps de travail, etc.

Parmi ces nouvelles technologies on trouve les pointeuses. Elles permettent d'assurer la gestion des entrées sorties des employés. Chaque entreprise possède son propre moyen qui lui permet de gérer le nombre d'heures travaillées en fonction des E/S de chacun d'entre eux. Ce travail peut être réalisé par des logiciels, des lecteurs d'empreintes, de reconnaissance faciale ou encore des lecteurs RFID.

Dans le cadre de ce projet nous nous intéressons à la gestion du pointage afin de la rendre automatique et permettre un contrôle rigoureux de la gestion du temps de travail de chaque employé.

III.2 Les différentes technologies d'identification

L'identification automatique se fait par plusieurs types de technologies. Dans ce présent travail, nous n'allons pas nous étendre sur tous les différents types de technologies, mais nous allons présenter quelque-uns à l'instar des codes à barres, des cartes intelligentes, des RFID passives et actives ainsi que les lecteurs d'empreintes.

- **Les Codes à barres** : sont destinés à automatiser l'acquisition d'une information généralement numérique. Ils trouvent leurs applications dans différents domaines comme la gestion des prêts d'une bibliothèque, les caisses enregistreuses, etc.

- **Les Etiquette RFID passive** : comme son nom l'indique, les tags passifs attendent un signal d'un lecteur RFID. Le lecteur envoie de l'énergie à une antenne qui convertit cette énergie en une onde RF qui est envoyée dans la zone de lecture. (Nous allons le détailler dans le titre qui suit) [12].

- **Les Etiquette RFID active** : comme les systèmes RFID passifs, les systèmes RFID actifs sont composés de lecteurs, d'étiquettes et d'antennes. Cependant, alors que les systèmes passifs exigent que les étiquettes soient activées par le lecteur, les systèmes RFID actifs utilisent des étiquettes RFID alimentées par batterie qui ne nécessitent pas d'énergie pour envoyer un signal [13].

- **Le lecteur d'empreinte** : est parfait pour les gestionnaires RH (Ressource Humaine) de toute entreprise petite ou moyenne. Ce lecteur d'empreintes digitales fera des merveilles et facilitera le travail en permettant de suivre de plus près le temps des employés [14].

Nous avons fait une étude sur les différentes technologies d'identification automatique en faisant une comparaison sur leurs avantages et leurs inconvénients.

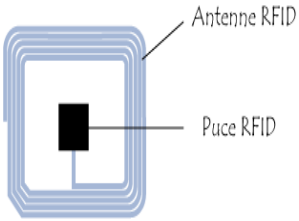
Nous comparons dans le tableau ci-dessous les codes à barres, les cartes intelligentes, les RFID passives et les RFID actives et les empreintes [15].

Technologies	Modification des données	Sécurité de données	Volume de données	Couts	Durée de vie
Codes a barres	Non modifiable	Minimale	Codes a barre linéaire contient 8-30 bytes. Certains codes a barre a 2D contient jusqu'a 7200 bytes.	Bas	Court
Étiquettes RFID active	Modifiable	Haute	Jusqu'à 8 MB	Très élevé	3-5 ans
Etiquettes RFID passive	Modifiable	Moyenne	Jusqu'à 64 KB	Moyen	Indéfini
Lecteurs d'empreintes	Non modifiable	Moyenne	certains lecteurs peuvent mémoriser jusqu'à 9000 empreintes.	Moyen (ça diffère selon les caractérist iques)	—
Carte intelligente	Modifiable	Haute	Jusqu'à 8 MB	Elevé	Long

Tableau 2 : Comparaison des différents types de technologie d'identification automatique

Après avoir fait cette comparaison nous nous sommes penchées sur la technologie RFID. Une technologie qui nous semble être la mieux adaptée par rapport aux avantages qu'elle présente et à sa facilité d'utilisation.

III.3 RFID



L'abréviation RFID signifie « Radio Frequency IDentification », en français, « Identification par Radio Fréquence ». Cette technologie permet d'identifier un objet, d'en suivre le cheminement et d'en connaître les caractéristiques à distance grâce à une étiquette émettant des ondes radio, attachée ou incorporée à l'objet. La technologie RFID permet la lecture des étiquettes même sans ligne de vue directe et peut traverser de fines couches de matériaux (peinture, neige, etc.).

L'étiquette radiofréquence (transpondeur, étiquette RFID), est composée d'une puce (en anglais « chip ») reliée à une antenne, encapsulée dans un support (RFID Tag ou RFID Label). Elle est lue par un lecteur qui capte et transmet l'information [16].

III.4 Types de RFID

III.4.1 RFID active

La radio identification active est une forme de technologie d'identification caractérisée par l'usage de tags actifs également appelés étiquettes actives c'est-à-dire qu'ils sont alimentés par une source d'énergie embarquée : batterie, pile... etc. Une source d'énergie qui a la capacité de diffuser un signal vers le lecteur RFID.

Les tags sont de petits objets qui peuvent être collés sur des objets ou insérés dans ces mêmes objets, ils sont composés :

- D'une puce électronique.
- D'une antenne.

III.4.2 RFID passive

Contrairement aux tags actifs, les tags passifs ne disposent pas d'une source d'énergie. Ils puisent leur énergie à travers le signal électromagnétique du lecteur qui permet d'activer le tag et lui permet ainsi d'émettre les informations.

Les tags passifs utilisent différentes bandes de fréquences radio selon :

- Leur capacité à transmettre les données à des distances plus ou moins grandes.
- Les substances différentes que les données doivent traverser (air, eau, métal...).

III.4.3 Les fréquences de transfert de données

Nous pouvons également classifier les RFID en fonction des fréquences utilisées par chaque technologie. Les échanges de données entre les étiquettes, les lecteurs et le serveur se propagent à

travers des ondes électromagnétiques à de fréquences données. L'antenne du transpondeur émet des signaux radio pour activer, lire et écrire les données.

Plusieurs paramètres sont importants dans l'échange d'information entre les étiquettes et le lecteur, mais le plus important est la « fréquence radio d'échange ». Actuellement, la technologie RFID utilise les gammes de fréquence suivantes [15] :

> 125 à 150 KHz (Basse fréquence).

> 13,56 MHz (Haute fréquence).

> 863 à 915 MHz (Ultra haute fréquence).

> 2,45 GHz (Super haute fréquence).

Types de fréquence	Bande de fréquences	Portée de lecture	Domaines d'application
Basse fréquence (LF)	30 – 300 KHz	En général : 20 cm Maximale : 1.2 m	Contrôle d'accès et de parcours, identification des animaux, antivol …
Haute fréquence (HF)	3 – 30 Mhz	En général : 20 cm Maximale : 1.2 m	Gestion des ressources, billetterie, gestion des bibliothèques, suivi et traçabilité
Ultra-haute fréquence (UHF)	300 Mhz – 3 GHz	433 MHz : 100 m maxi. 865 - 956 MHz : 0.5 à 5 m	Suivi de palettes et de conteneurs
Micro-onde	2 – 30 GHz	En général : 3 m Maximale : 10 m	Péage routier, suivi des conteneurs, contrôle de production

Tableau 3 : les différents types de fréquence de la RFID

III.5 Composants des systèmes RFID

Pour pouvoir mettre en place un système de traçabilité RFID, nous devons avoir un équipement spécifique composé de :

III.5.1 Tags

Les tags RFID, également appelés transpondeurs, peuvent être classés en plusieurs catégories selon toute une série de caractéristiques. Ainsi, on fait généralement une distinction entre les puces actives et passives. La capacité de mémoire et les fonctions de lecture-écriture sont d'autres critères de distinction.

On le retrouve souvent intégré dans un boîtier en plastique, ou à l'intérieure d'une étiquette d'un emballage dit « intelligent ». Le transpondeur comprend une antenne associée à une puce électronique qui peut répondre aux requêtes émises depuis un émetteur-récepteur [17].

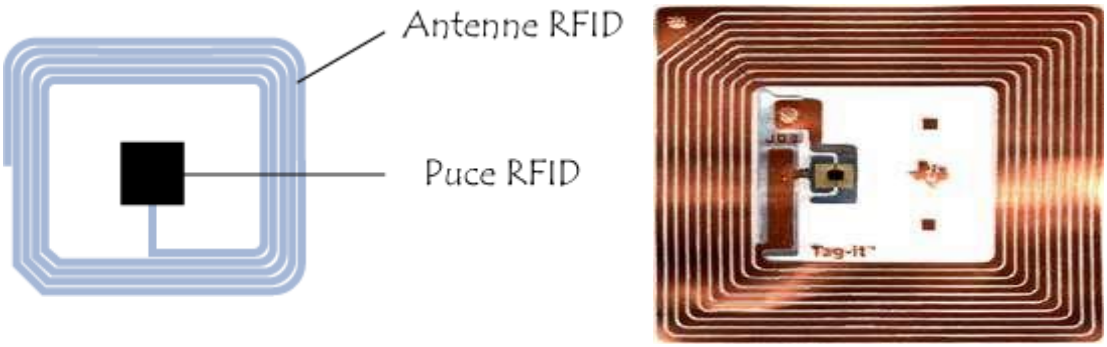


Figure 10 : Les tags RFID

Il existe plusieurs types de fonctionnement et de communication possibles pour les étiquettes :

- Les étiquettes « lecture seule », c'est-à-dire non modifiables ce mode permet seulement de lire le contenu du tag.
- Les étiquettes « écriture une fois, lecture multiple », ce mode de fonctionnement permet la réutilisation, le tag est réinscriptible.
- Les étiquettes en « écriture une fois et lecture plusieurs fois » [18].

III.5.2 Lecteurs

Les lecteurs, souvent appelés « interrogateurs », sont des équipements actifs, portables ou fixes, constitués d'un circuit qui émet une énergie sous forme de champ magnétique ou d'onde radio. Dans un scénario type, le lecteur envoie un signal à la puce et attend sa réponse. La puce détecte le signal et envoie une réponse qui contient un numéro de série ainsi qu'éventuellement d'autres informations au lecteur. Cette communication se fait grâce à chaque antenne RFID intégrée dans chacun d'entre eux.

Dans les systèmes plus sophistiqués, le signal radio du lecteur peut contenir des commandes destinées à la puce, des instructions pour effectuer des opérations de lecture/d'écriture dans la mémoire de la puce, voire des mots de passe. La taille du lecteur, dépend de nombreux paramètres. Il peut varier de la taille d'une pièce de monnaie à celle d'un ordinateur de poche. Un lecteur peut être doté de fonctionnalités GPS et de dispositifs de connexion à des systèmes et des réseaux d'information [17].

Figure 11 : Les différents types de RFID

III.6 Fonctionnement de la RFID

Un système RFID se compose d'étiquettes et de lecteurs. L'étiquette contient l'identité à transmettre, tandis que le lecteur émet des signaux radio afin de lire ou d'inscrire des données sur l'étiquette. Lorsqu'une étiquette RFID détecte le signal entrant d'un lecteur (c'est-à-dire lorsqu'elle passe à distance de lecture de celui-ci), elle répond par un signal sortant qui contient l'identité. Le lecteur reçoit alors cette identité et la transmet pour traitement à l'ordinateur auquel il est relié [19].

Rapport-Gratuit.com

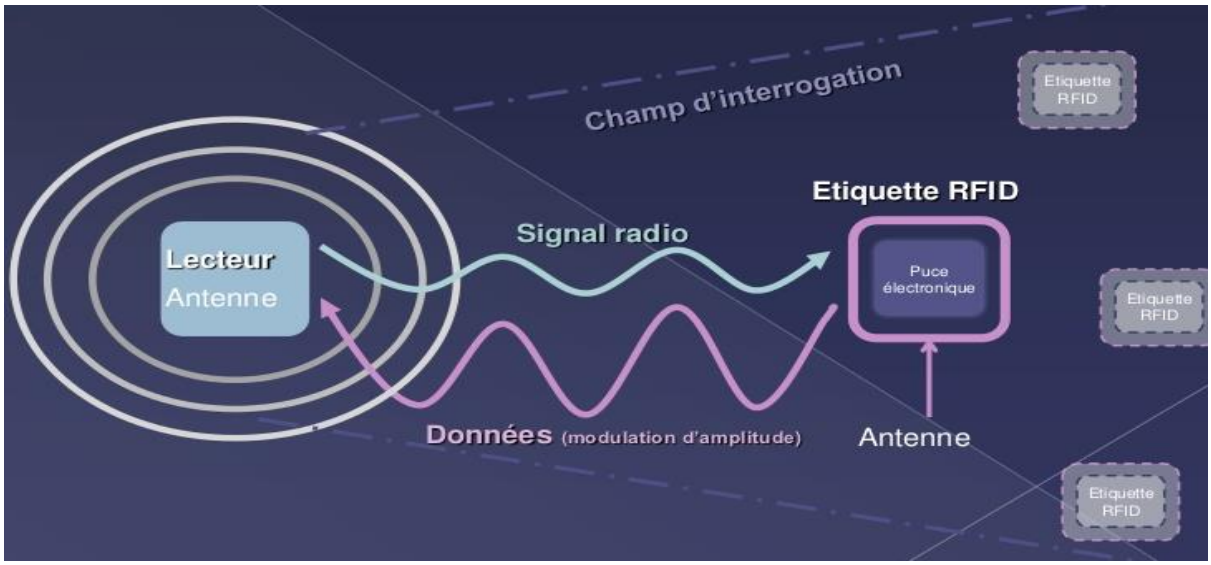


Figure 12 : Fonctionnement du RFID

Le principe de fonctionnement est simple. Lorsque le lecteur est alimenté en tension, il génère un champ magnétique et lorsqu'un tag s'en approche par un effet électromagnétique cela génère un courant électrique et donc une différence de potentiel. C'est cette différence de potentiel qui permet à une puce électronique dans le tag d'être alimentée en tension.

À partir de cet instant, le lecteur et le tag utilisent leur antenne pour échanger des donnés a une courte distance dont le numéro d'identification du tag.

Les données captées par le lecteur sont transmises et traitées par un système informatique comportant un logiciel, tel qu'un système de contrôle d'inventaire, d'un système de contrôle d'accès ou d'un système de contrôle de production [20].

III.7 Domaines d'application de la RFID

Dans cette section nous présentons quelques applications spécifiques de la technologie RFID à travers le monde.

- **L'accès aux transports publics**

  Les péages sur les autoroutes, les transports publics sont utilisés dans la plupart des villes asiatiques (Japon, Corée, Chine...) et aux Etats-Unis. Le péage est fait via un système RFID et les étiquettes incorporées aux téléphones mobiles.

- **L'accès sans formalité aux bibliothèques**

Les codes à barres ont été beaucoup utilisés dans le passé dans plusieurs bibliothèques pour les opérations de prêt et de retour des ouvrages. Les bibliothèques choisissent de plus en plus la technologie RFID pour l'automatisation de ces opérations et peuvent vérifier les entrées et sorties à l'aide de lecteurs à balayage optique.

- **La sécurité des personnes**

Les bracelets pourvus de puces RFID sont disponibles pour la location auprès de l'administration du parc pour les parents soucieux de suivre les allées et venues de leurs enfants. Les parents peuvent à tout moment se questionner sur la localisation de leurs enfants en envoyant par leur téléphone mobile un message texte à une application. Cette dernière indique par un message texte réponse la localisation de l'enfant.

- **La santé**

Le domaine de la santé est parmi les domaines dans lesquels la technologie RFID est utilisée. La technologie RFID trouve d'innombrables applications, nous avons répertorié quatre axes d'applications dans les centres hospitaliers. Ces applications couvrent la gestion des équipements, le suivi des dossiers médicaux, le suivi et l'identification de patients.

- **Les applications de logistique (pour suivre l'acheminement de marchandises)**

L'usage de la technologie RFID dans une chaîne logistique augmente la visibilité de produits tout le long de la chaîne au cours de leurs cycles de vie. Avec les étiquettes RFID les responsables de production peuvent identifier à tout moment les différents produits le long de la chaîne. Ils peuvent même saisir la vitesse à laquelle chaque item est produit à chaque étape de la chaîne logistique. Aussi, la technologie permettra de prendre les décisions de production en fonction de données de vente en temps réel, au lieu de considérer les informations prévisionnelles.

Les autres avantages que la technologie RFID pourrait avoir dans une chaîne logistique est de fiabiliser en temps record les procédures de vérification des entrées et sorties de produits, la localisation de produits dans les entrepôts.

La technologie RFID combinée à d'autres paramètres physiques comme la température, l'humidité et autres offrira de multiples applications dans les chaînes logistiques [15].

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

III.8 Les aspects juridiques de la technologie RFID

Les recherches sur la technologie RFID laissent entrevoir à ce stade de préoccupations constituant de défis juridiques à relever. Notamment la traçabilité de personnes, de biens, le contrôle des accès, le partage et la collecte des données personnelles, la sécurisation des contenus, le croisement des informations RFID (associées aux objets) et de données personnelles (cartes de banques, cartes de fidélité), la création de profil marketing.

Pour les organisations de protection de la vie privée, toutes ces préoccupations devraient faire contrôler la vulgarisation de l'usage de la technologie RFID [15].

III.9 Avantages

La technologie RFID est souvent présentée comme une prochaine grande révolution dans le monde informatique où elle peut être utilisée dans de nombreux domaines : billetterie, transport et logistique, soins de santé, transports publics, lutte contre la contrefaçon, fabrication et production, paiement électronique, sûreté…etc.

Elle est efficace pour tirer le meilleur parti possible des processus existants, améliorer la fiabilité, créer de nouveaux services et aussi augmenter la productivité [21].

III.10 Inconvénient

Comme les systèmes RFID deviennent moins coûteux et plus sophistiqués, les questions de confidentialité ont remonté à la surface. Les préoccupations sont liées aux confidentialités des individus qui se retrouvent liés à des informations spécifiques sur l'histoire de leur achat et ça devient gênant [22].

III.11 Conclusion

Ce chapitre décrit les technologies de pointages automatiques existants et en particulier la RFID. Nous avons présenté le principe de fonctionnement d'un système RFID, ses composants, les types de RFID, ses avantages ainsi que ses inconvénients.

Chapitre III *Réalisation du projet*

IV.1 Introduction

Dans ce chapitre, nous présentons la réalisation de notre projet en commençant par lister les outils et les technologies utilisées, puis nous parlerons de la méthode de gestion de projets que nous avons adoptée. Enfin nous montrerons les solutions que nous avons proposées pour intégrer la technologie RFID dans la gestion du pointage au niveau du serveur Odoo.

IV.2 Les langages et outils utilisés

Pour développer des modules Odoo, nous avons utilisé plusieurs langages et technologies comme le python, XML, PostgreSQL, etc.

Python

Python est un langage de programmation de haut niveau, orienté objet, totalement libre et très efficace, conçu pour produire du code de qualité, portable et facile à intégrer. Ainsi la conception d'un programme Python est très rapide et offre au développeur une bonne productivité. En tant que langage dynamique, il est très souple d'utilisation et constitue un complément idéal à des langages compilés. Odoo utilise Python comme langage de programmation [23].

PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD. Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MySQL et Firebird), ou propriétaires (comme Oracle, Sybase, DB2 et Microsoft SQL Server) [24].

Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, mais il est fondé sur une communauté mondiale de développeurs et d'entreprises.

PostgreSQL peut stocker plus de types de données que les types traditionnels : entiers, caractères, etc. Odoo utilise le système de gestion de base de données relationnelles PostgreSQL pour l'enregistrement de ses données.

XML

XML est l'abréviation d'Extensible Markup Language. Il n'est pas un langage de programmation. On ne peut pas faire de tests, ni inclure un fichier dans un autre. De manière générale, XML sert uniquement à stocker des données. XML est simplement une méthode pour représenter les données. Celles-ci sont écrites entre des balises ou sous forme d'attributs, et l'ensemble est écrit sous forme d'un arbre. [25].

Odoo utilise le langage de balisage XML pour la description des données, des interfaces et des rapports.

Arduino IDE

Le logiciel arduino permet d'écrire facilement un code et de le téléverser sur l'arduino. Il fonctionne sous Windows, Mac OS et Linux. L'environnement est écrit en java et basé sur le traitement [26].

Processing

Processing est un langage et un environnement de programmation open-source (c'est aussi le nom du compilateur). Il est employé par des étudiants, des artistes, des créateurs, des chercheurs, et des amateurs. Ce langage basique offre de très nombreuses possibilités, et laisse libre cours à votre imagination, notamment grâce à sa centaine de librairies disponibles. Le logiciel fonctionne sur Macintosh, sous Windows et sous Linux, en effet il est basé sur la plate-forme Java [27].

Java

Le langage Java est un langage de programmation informatique orienté objet. Par nature, un programme s'y trouvera formé d'une classe ou de la réunion de plusieurs classes et il instanciera des objets. Il sera impossible de créer un programme n'utilisant aucune classe [28].

IV.3 Composants électroniques utilisés

Arduino

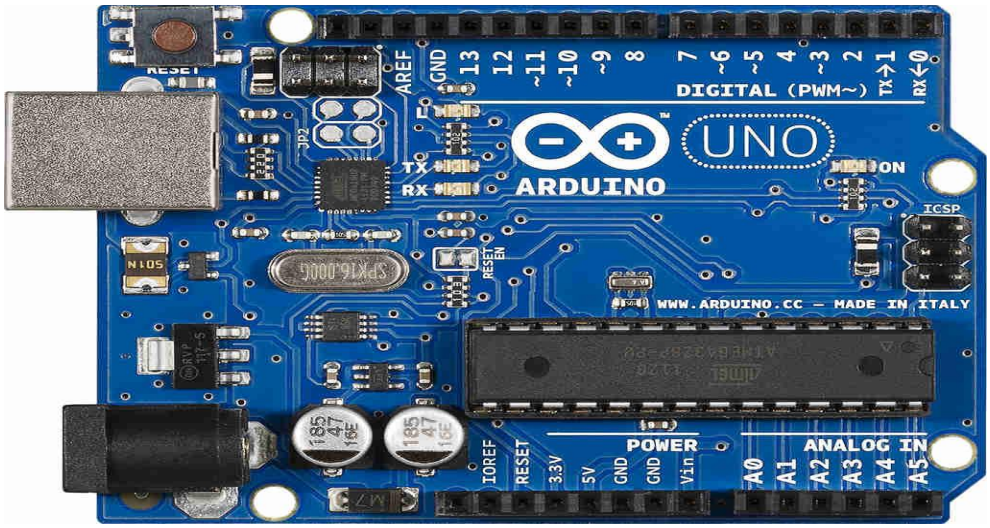
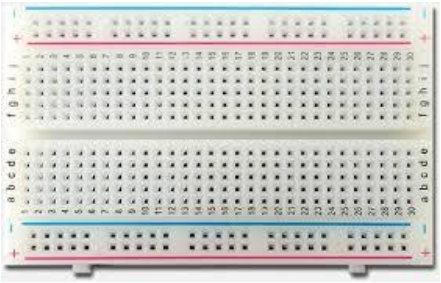


Figure 13 : Arduino UNO

Arduino / Genuino Uno est un microcontrôleur basé sur l'ATmega328P. Il dispose de 14 broches d'entrée / sortie numériques, 6 entrées analogiques, un cristal à quartz 16 MHz, une connexion USB, une prise de courant, un en-tête ICSP et un bouton de réinitialisation. Il contient tout le nécessaire pour supporter le microcontrôleur. Il suffit de le connecter à un ordinateur avec un câble USB ou de l'alimenter avec un adaptateur AC-DC ou une batterie pour commencer [29].

Breaboard



Une breadboard est une planche en plastique rectangulaire avec un tas de trous minuscules. Ces trous vous permettent d'insérer facilement des composants électroniques en prototype (c'est-à-dire pour construire et tester une version antérieure) d'un circuit électronique [30].

Les connexions ne sont pas permanentes, il est donc facile d'enlever un composant si vous faites une erreur, ou tout recommencer et faire un nouveau projet. Vous pouvez utiliser la breadboard pour créer toutes sortes de projets électroniques.

IV.4 Adaptation des modules Odoo

Souvent, les entreprises ont des besoins spécifiques qui ne sont pas implémentés dans Odoo, c'est pourquoi ce dernier propose des solutions standards pour la majorité des entreprises et offre la possibilité d'adapter ses modules de base aux besoins de chaque entreprise. Dans le but de satisfaire les besoins exprimés par SOGESI, nous avons adapté les modules suivants :

- Module de la paie : ce module est destiné à gérer la paie des employés. Nous avons adapté le module selon le besoin de l'entreprise. Ceci dans le but de pouvoir calculer la paie directement et automatiquement à partir des heures réellement travaillées.
- Module des employés : nous avons réutilisé ce module en lui ajoutant un champ pour le numéro d'identification de la carte RFID que nous allons détailler un peu plus loin dans ce chapitre. Chaque employé aura donc un numéro d'identification qui lui sera affecté. Lors du passage devant le lecteur RFID, l'employé sera détecté.

IV.5 Gestion du projet

Avant de démarrer ce projet, nous avons jugé qu'il était nécessaire de le définir et de le planifier afin d'atteindre avec succès les objectifs fixés au départ. En concertation avec l'entreprise, nous nous sommes mis d'accord sur une réalisation progressive en adoptant un processus itératif et incrémental. Nous avons donc utilisé une méthode agile appelée « Scrum ». Le principe de la méthode SCRUM est de découper les taches en sprints à un rythme constant avec des livraisons très fréquentes. Nous avons choisi des sprints de 2 semaines. La figure ci-dessous montre une illustration de la méthode SCRUM.

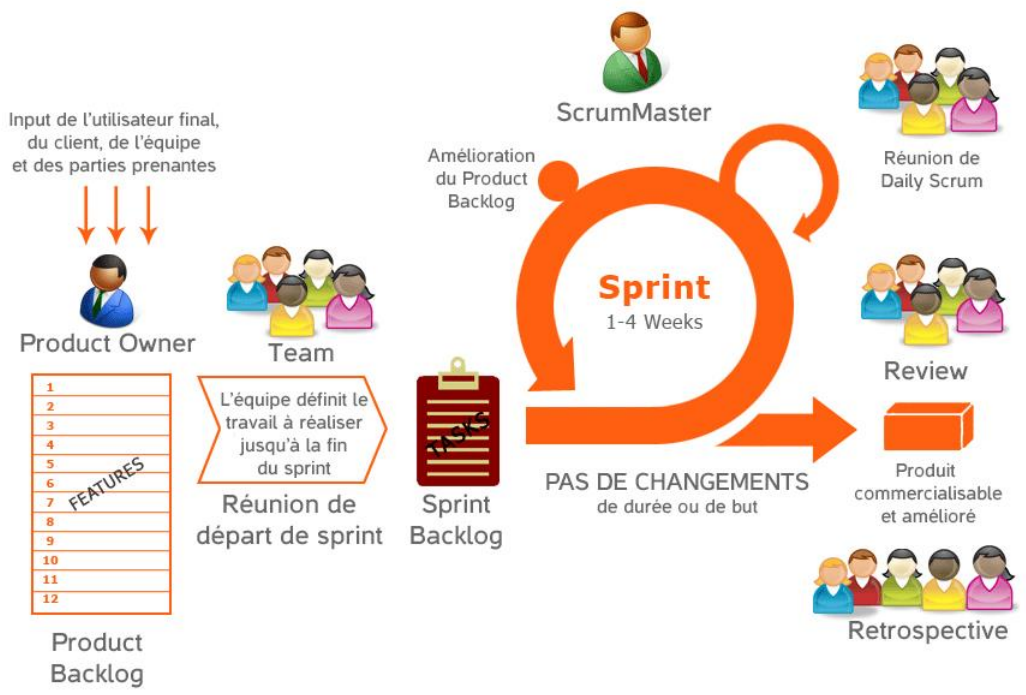


Figure 14 : Cycle de vie de la méthode scrum

Afin de mieux nous organiser, nous avons choisi l'outil Trello en ligne. Cet outil est ergonomique, gratuit et suit la méthode Scrum pour organiser nos tâches, consigner les informations essentielles, estimer les délais de leur réalisation, les ordonnancer selon leur priorité et tenir un planning en commun avec les membres de notre équipe.

Ce service en ligne nous aide à consigner nos tâches sur des post-it « cards » que nous accrochons sur un panneau « board » en les rangeant dans des listes de gauche à droite :

- Les tâches à faire « To Do ».
- Les tâches en cours de réalisation « Doing ».
- Les tâches à tester « To Test ».
- Les taches réalisées « Done ».

Au fur et à mesure de l'avancement de nos tâches, nous déplaçons les post-it d'une colonne à l'autre. La figure 15 donne un aperçu de quelques taches de notre projet dans Trello :

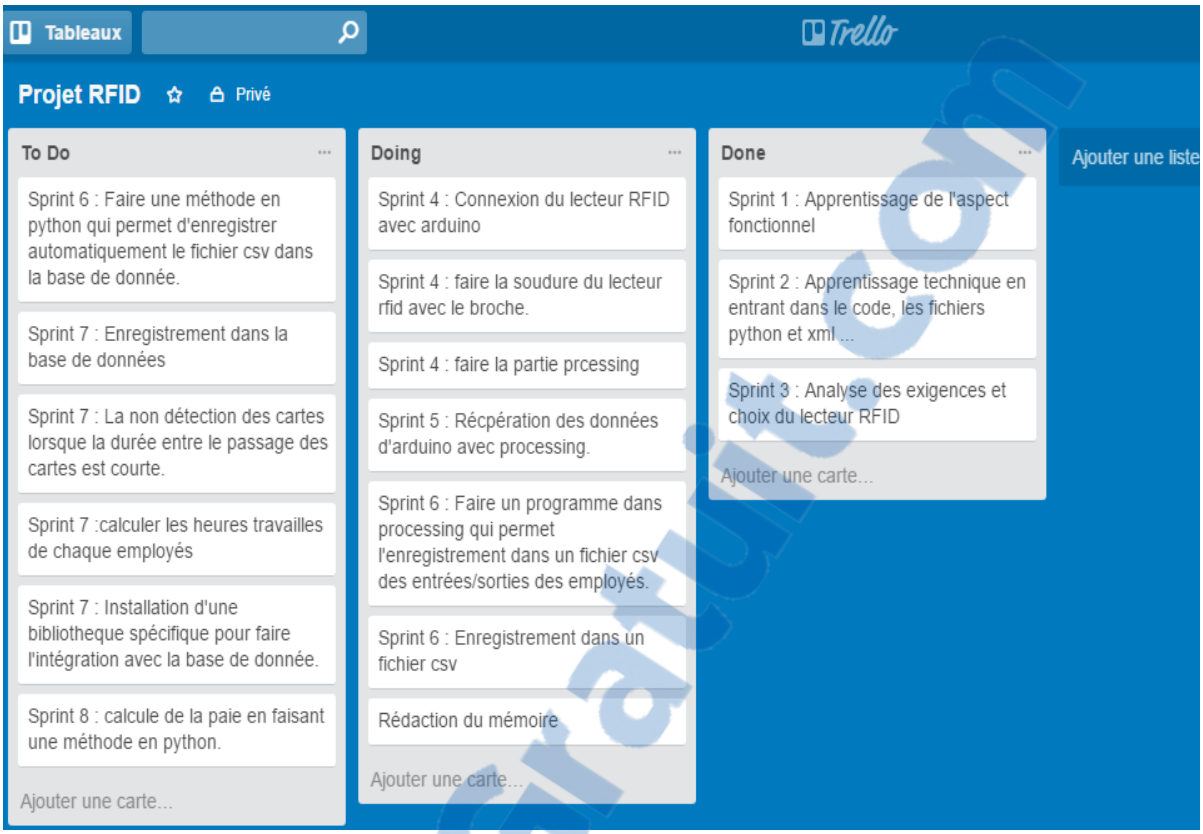


Figure 15 : Les taches de Trello

Nous montrons dans cette partie un exemple d'un déroulement d'un sprint. Nous avons créé le sprint 7 le 01/05/2017 dans la colonne « To Do » en lui affectant la date limite comme montré dans la figure ci-dessous :

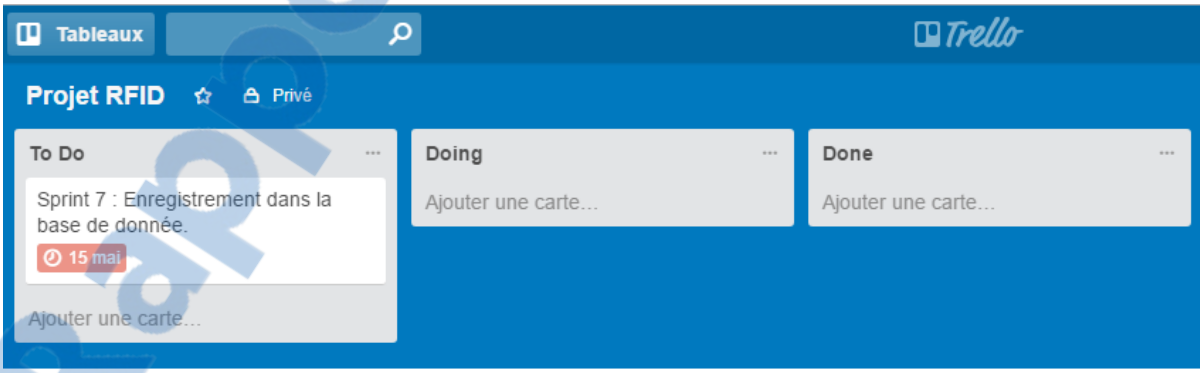


Figure 16 : Création du sprint 7

Nous avons déplacé le post-it dans la colonne « Doing » pour indiquer le commencement de la réalisation de la tache comme le montre la figure 17.

Rapport-Gratuit.com

Figure 17 : Le développement du sprint 7

Après avoir fini le sprint 7 dans le temps estimé, nous l'avons déplacé vers la colonne « Done » pour que les autres membres du groupe sachent que cette tâche a été terminée. Voici une capture d'écran qui illustre cette étape :

Figure 18 : la fin du sprint 7

Nous terminons cette partie de la gestion du projet par un schéma qui englobe tous les sprints que nous avons réalisés dans la figure 19 :

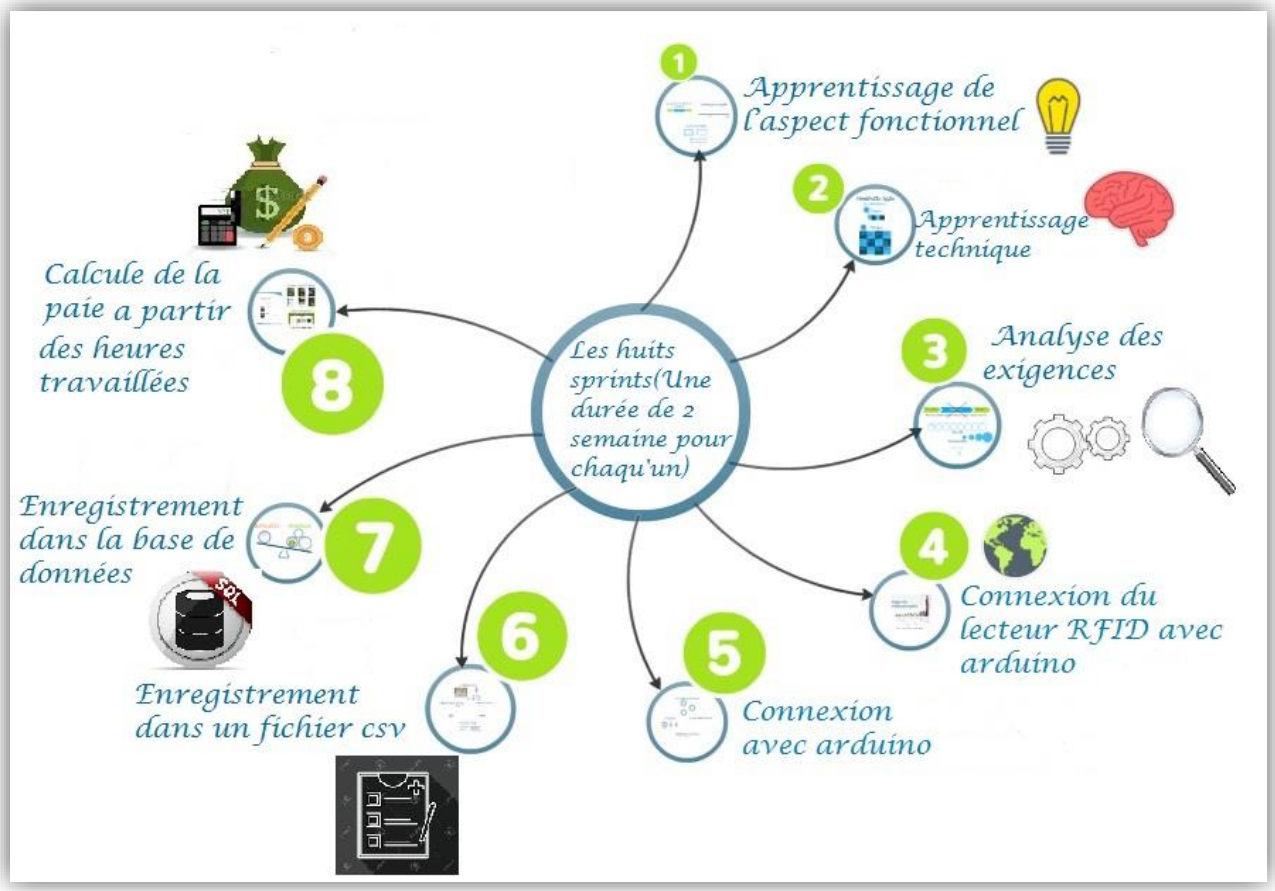


Figure 19 : les huit sprints du projet

IV.6 Apprentissage de l'aspect fonctionnel (Sprint 1)

Avant de commencer la mise en œuvre du projet, il a fallu améliorer nos compétences dans le domaine de l'ERP Odoo. La difficulté résidait dans la documentation, parce qu'elle est peu fournie, cependant nous avons trouvé les réponses à nos questions en tirant profit des connaissances de l'équipe de la société SOGESI.

L'ERP Odoo offre une interface riche à l'utilisateur, caractérisée par plusieurs modes d'affichage, les 3 modes principaux sont :

- Vue kanban : affiche les enregistrements de manière élégante.
- Vue liste : affiche les enregistrements sous forme d'une liste ordonnée.
- Vue formulaire : permet d'afficher et d'éditer les informations d'un enregistrement.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

À travers cette interface nous avons pu effectuer des configurations techniques sans changer le code source.

IV.7 Apprentissage technique (Sprint 2)

La configuration à travers l'interface n'est appliquée que dans la base de données courante. Il faudra générer du code pour développer une application réutilisable dans plusieurs bases de données. Lors de la formation faite au sein de la société, nous avons appris à :

- Créer un objet (model).
- Hériter d'un objet et ajouter des champs
- Créer des menus et des vues en XML.
- Faire des actions planifiées.

IV.7.1 Analyse des exigences (Sprint 3)

Cette phase consistait à recueillir les besoins en faisant des recherches et en collectant les informations nécessaires pour répondre aux questions que nous nous sommes posées à l'instar du matériel qu'il faut utiliser, d'une étude comparative entre les différentes technologies, de se renseigner sur le système de gestion de base de données, de la communication avec Odoo, de la communication série avec python, du fonctionnement du lecteur RFID, etc. Cette étude a déjà été détaillée dans le chapitre 2.

IV.7.2 Diagramme de composants

Ce diagramme montre les unités logicielles et matérielles à partir desquelles nous avons construit notre système. Ça nous permet d'avoir une vue plus générale et plus claire. Nous avons décrit l'organisation de notre système en connectant les éléments matériels et logiciels entre eux.

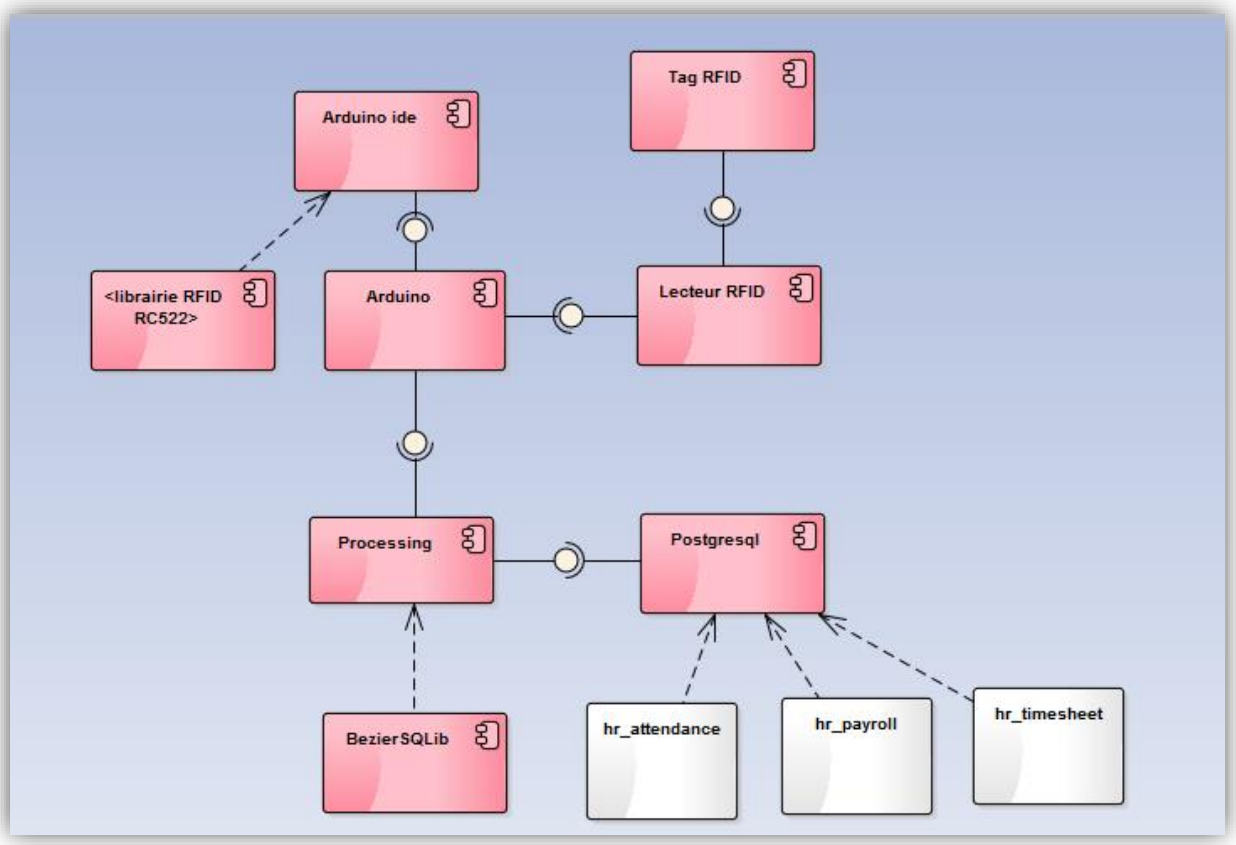


Figure 20 : Diagramme de composants

IV.8 Réalisation

Notre projet consiste à faire un système de pointage qui sert à faciliter la gestion de pointage des employés en offrant une bonne gestion des heures de travail. Notre projet se divise en deux parties :

- Partie électronique.
- Partie informatique

IV.8.1 Choix du lecteur RFID (sprint 3)

Pour choisir une telle technologie, il nous a fallu faire une étude très détaillée sur l'environnement où on va l'installer afin de bien savoir quel lecteur utilisé. Pour cela, tout au début, nous avons pensé à faire des recherches sur les différents lecteurs RFID, puis rassembler toutes les informations essentielles en matière de matériels.

Nous avons choisi le lecteur RFID RC522 qui est une interface qui permet l'identification sans contact à partir d'un badge ou d'une clé RFID. Il est basé sur le circuit intégré RC522 en

communiquant avec Arduino via l'interface SPI. Il utilise la bande ISM 13.56MHz, la distance de communication peut aller jusqu'à 6 cm.

Le mécanisme RFID/NFC se base sur une radio communication de courte distance, ils utilisent la norme ECM-A340 et ISO/IEC 18092. Ce module est idéal pour notre projet qui est basé sur l'identification des employés avec leur badge [31].

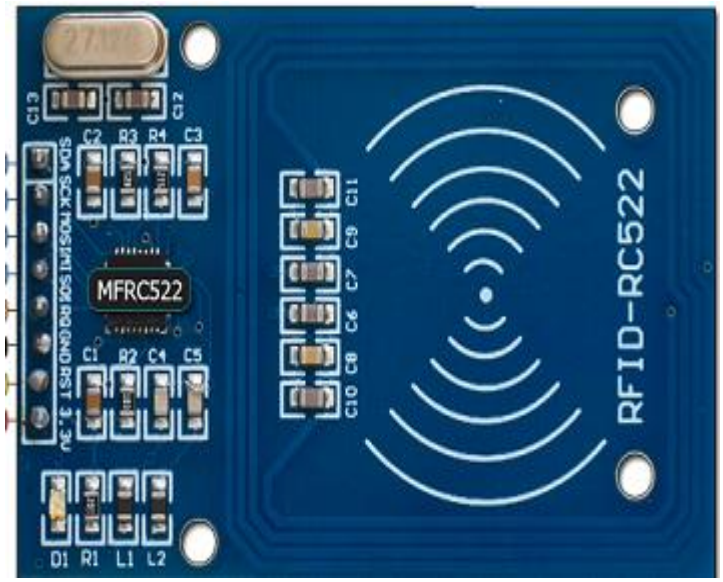


Figure 21 : RFID RC522

Caractéristiques

- Basée sur la puce Philips MFRC522.
- Power Voltage : 3.3V.
- Current :13-26mA.
- Fréquence d'utilisation : 13.56MHz.
- Distance opérationnelle : 0 ~ 60mm.
- Interface : SPI.
- Dimensions : 40mm × 60mm.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

IV.8.2 Connexion du lecteur RFID avec arduino (sprint 4)

Nous avons choisi le microcontrôleur "Arduino" pour faire le traitement des signaux électriques issus du lecteur RFID. Au début, nous avons branché le lecteur RFID aux broches de l'Arduino en suivant la table de la figure 20. Il faut juste faire attention de ne pas connecter l'alimentation sur 5v mais sur 3.3 v comme représenté ci-dessous :

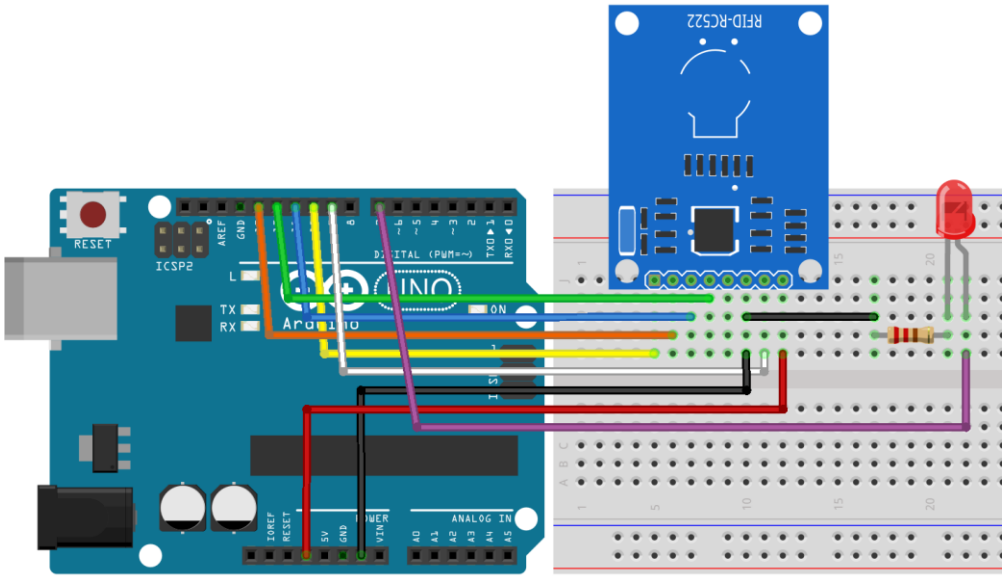


Figure 22 : Branchement du lecteur Arduino avec RFID

- Le GND de l'arduino UNO est connecté à la broche GND du stick LED et à la broche GND du module RFID.
- Le pin 3.3V de l'arduino UNO est connecté au pin VCC du LED et au 3.3V du module RFID.
- Les broches 9-10-11-12-13 de l'UNO sont connectées successivement aux broches RST-SDA-MOSI-MISO-SCK du module RFID.
- La broche 5 de l'UNO est reliée à la broche DIN du stick LED.

Table de correspondance :

RFID RC522	Arduino
VCC	+5V
RST	9
GND	GND
MISO	12
MOSI	11
SCK	13
NSS	10
IRC	/

Figure 23 : Tableau de correspondance entre RFID et Arduino

Concernant le programme, nous avons utilisé l'IDE Arduino en intégrant deux librairies : RFID et SPI.

- Serial Peripheral Interface (SPI) : est un protocole de données série synchrone utilisé par les microcontrôleurs pour communiquer avec un ou plusieurs périphériques rapidement sur de courtes distances. Il peut également être utilisé pour la communication entre deux microcontrôleurs.
- Nous avons intégré la librairie RFID dans l'ide d'arduino pour communiquer avec le module RFID.

Il est intéressant de noter ici que nous avons eu de gros problèmes dans l'utilisation du lecteur RFID pour récupérer le code des carte RFID, car au début il ne détectait ni la carte ni le tag. Après plusieurs tentatives nous nous sommes rendu compte qu'il s'agissait de problème de soudure. Il fallait donc souder la broche avec le lecteur RFID et le problème était résolu.

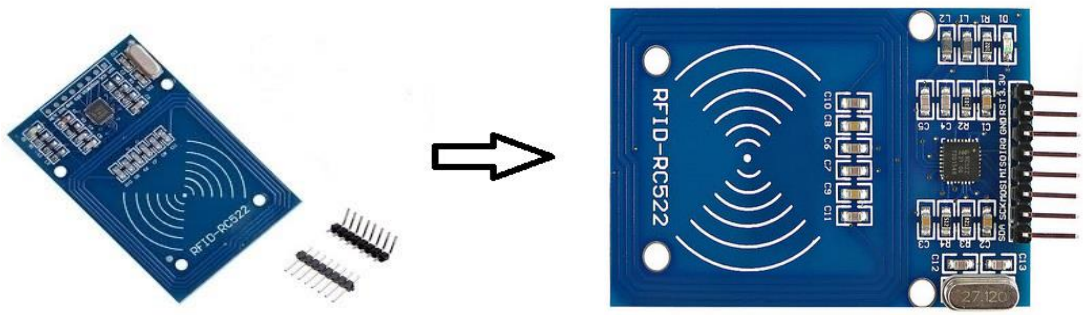


Figure 24 : RFID et sa Broche

Nous avons tenté de développer deux solutions. La première est basée sur un fichier CSV que nous créons et qui devrait être importé par Odoo par la suite. La seconde est beaucoup plus intéressante puisqu'elle consiste à enregistrer directement les informations du pointage dans la base de données Odoo en temps réel.

IV.8.3 Utilisation d'Arduino (sprint 4)

Après le branchement de l'arduino avec le lecteur RFID. Nous avons utilisé le logiciel arduino, pour pouvoir récupérer les informations de la carte RFID. Pour cela il nous a fallu inclure la bibliothèque MFRC522 et la bibliothèque SPI pour avoir une communication en port série comme suite :

```
#include "SPI.h"
#include "MFRC522.h"

#define SS_PIN 10
#define RST_PIN 9
#define LED_PIN 2

MFRC522 rfid(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

void setup() {
    Serial.begin(9600);
    SPI.begin();
    rfid.PCD_Init();
    pinMode(LED_PIN, OUTPUT);
}
```

Figure 25 : Code arduino

Les instructions disponibles dans arduino sont limitées, car adaptées aux possibilités du microcontrôleur. C'est pour cela que nous avons utilisé un autre logiciel appelé "Processing" pour la réalisation de notre solution que nous détaillerons dans la section suivante. Après avoir fait la lecture des tags nous avons envoyé les résultats à ce dernier pour les traiter. Voici le code correspondant :

```
if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI &&
  piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
  piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
  Serial.println(F("La carte n'est pas une carte mifaire."));
  return;
}

String strID = "";
for (byte i = 0; i < 4; i++) {
  strID +=
  (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
  String(rfid.uid.uidByte[i], HEX) +
  (i!=3 ? ":" : "");
}
strID.toUpperCase();


sensorVals[0] = strID;



Serial.println(sensorVals[0]);
//Serial.print(",");
delay(100);
```

Figure 26 : La récupération et l'envoi du numéro de carte à Processing

IV.8.4 Exploitation conjointe de l'outil *Processing avec Arduino*

Pour faire l'enregistrement des données soit dans le fichier CSV ou dans la base de données, il nous a fallu utiliser un autre logiciel parce que l'utilisation d'arduino seul ne suffisait pas. Après de nombreuses recherches et plusieurs tentatives, nous avons trouvé des déclarations de personnes dans des forums ayant travaillé avec un logiciel nommé "Processing" pour faire le lien avec arduino. C'est de cette manière que nous avons pensé à travailler avec ce logiciel.

IV.8.4.1 Connexion avec arduino (sprint 5)

Pour pouvoir faire la relation entre arduino et processing nous avons importé la bibliothèque Serial. Ce qui va nous permettre d'écouter toute communication en série, nous devons obtenir un objet *Serial*, qui est myPort qui assure l'écoute sur le port série pour toute donnée entrante. Nous avons également besoin d'une variable pour recevoir les données réelles.

```
1
2  import processing.serial.*;
3  import de.bezier.data.sql.*;
4  PostgreSQL pgsql;
5
6  Serial myPort;
7  Table dataTable;
8  int numReadings = 500;
9  int readingCounter = 0;
10 String fileName, val;
11
12 void setup()
13 {
14   String user     = "openpg";
15 String pass     = "openpgpwd";
16 String database = "RFID";
17 pgsql = new PostgreSQL( this, "localhost", database, user, pass );
18 String portName = Serial.list()[0];
19
20 myPort = new Serial(this, portName, 9600);
21 dataTable = new Table();
```

Figure 27 : Code processing pour la connexion entre arduino et processing

Dans la méthode setup() nous avons cherché le port série auquel notre Arduino est connecté et configuré notre Serial pour écouter ce port. Le dernier argument (9600) dans l'objet serial et le même qui définit dans arduino dans Serial.begin(9600).

IV.8.4.2 Récupération des données d'arduino avec processing

```
53 void serialEvent(Serial myPort){
54 try {
55 val = myPort.readStringUntil('\n');
56 if (val!= null) {
57 val = trim(val);
58 println(val);
59 //tring sensorVals[] = float(split(val, ','));
60 String sensor1Pin = "0";
61 String sensorVals[] = {val};
62 TableRow newRow = dataTable.addRow();
63 Date jour = new Date();
64 Date heure=new Date();
```

Figure 28 : Code pour récupérer les données d'arduino

La méthode serialEvent() que nous avons utilisée est appelée à chaque fois qu'il y a un évènement dans le tampon. Notre fonction myPort.readStringUntil('\n') va nous permettre de stocker les données entrantes dans un tampon, jusqu'à ce que nous voyons le séparateur (\n) qui va séparer chaque boucle arduino. Ce qui nous permettra d'analyser les données pour chaque séparateur. Dans cette méthode nous proposons deux solutions la première qui consiste à enregistrer dans un fichier CSV et une autre qui fait un enregistrement dans la base de données en temps réelle.

Rapport-Gratuit.com

IV.8.4.3 Enregistrement dans un fichier csv (sprint 6)

Pour faire l'enregistrement des informations dans le fichier CSV, il nous a fallu faire deux programmes un dans processing et l'autre dans python.

IV.8.4.3.1 Programme dans processing

```
53 println(val);
54 //String sensor1Pin = "0";
55 String sensorVals[] = {val};
56 TableRow newRow = dataTable.addRow();
57 Date jour = new Date();
58 Date heure=new Date();
59 newRow.setInt("id", dataTable.lastRowIndex());
60 newRow.setInt("year", year());
61 newRow.setInt("month", month());
62 newRow.setInt("day", day());
63 newRow.setInt("hour", hour());
64 newRow.setInt("minute", minute());
65 newRow.setInt("second", second());
66 newRow.setString("IDCarte" , sensorVals[0]);
67 println( sensorVals[0]);
68 Date dates = new Date();
69 Date dt = new Date();
70 System.out.println(formater.format(dates));
71 println("le fichier est enregistre ");
72 fileName = str(year()) + str(month()) + str(day()) +str(second())+ str(dataTable.lastRowIndex())+".csv";
73 saveTable(dataTable,fileName);
```

Figure 29 : Programme pour l'enregistrement du fichier csv

Dans la méthode setup() nous avons créé la première ligne dans le fichier CSV ou nous donnons à chaque colonne son nom. Ensuite à chaque fois que l'utilisateur passe sa carte, une nouvelle ligne sera créée avec les colonnes suivantes :

- La première colonne pour l'id de la table ou il s'incrémente à chaque fois que nous ajoutons une nouvelle ligne.
- Une colonne pour l'année courante que nous avons récupéré à l'aide de la fonction year().
- Les autres colonnes de la même manière pour le mois, le jour, l'heure, la minute, et la seconde.
- La dernière colonne, nous enregistrons l'id de la carte qui est récupéré à partir de l'arduino.

Et à la fin, nous enregistrons le fichier CSV avec la fonction saveTable() qui prend en paramètre le nom de la table et la table.

IV.8.4.3.2 Programme dans python

Nous avons également programmé en python une méthode qui s'exécute automatiquement tous les jours. Voici le code python de l'action planifiée :

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import datetime
import math
from datetime import date
import csv
import psycopg2
from openerp.osv import fields, osv, orm
from openerp import api

class obj_pointage(osv.osv):
    _name = "obj.pointage"
    _inherit = ['base_import.import']


    _description = "Objet Pointage"

    _columns = {
        'heure': fields.datetime( "Date Heure"),
        'employe': fields.char( "Employé", required=True ),
        'action': fields.char( "Entrée/Sortie" ),
        'heures_travailles': fields.float( compute='attendance_calcul', string = "Heures travailles" ),
        'image': fields.binary( "Image" ),

    }

    @api.model
    @api.multi
    def importer_csv(self):
        conn = psycopg2.connect("host='localhost' dbname='RFID' user='root' password='amina'")
        cur = conn.cursor()
        sqlqtr= "COPY obj_pointage (employe, action, heure, pointeuse) FROM STDIN DELIMITER ',' NULL as '' CSV HEADER"
        with open('E:/amina/pointage2.csv') as f:
            cur.copy_expert(sqlqtr, f)
        conn.commit()
```

Figure 30 : Programme python pour l'enregistrement du fichier csv dans la base de données

La méthode importer CSV établie une connexion avec la base de données “RFID” pour qu’elle puisse insérer le fichier CSV dans la table obj_pointage.

```xml
<?xml version="1.0" encoding="utf-8"?>
<openerp>
    <data noupdate="1">
        <record id="ir_cron_scheduler_demo_action" model="ir.cron">
            <field name="name">Import pointage RFID</field>
            <field name="user_id" ref="base.user_root"/>
            <field name="interval_number">1</field>
            <field name="interval_type">days</field>
            <field name="numbercall">-1</field>
            <field eval="False" name="doall"/>
            <field eval="'obj.pointage'" name="model"/>
            <field eval="'importer_csv'" name="function"/>
        </record>
    </data>

</openerp>
```

Figure 31 : Programme XML pour l'enregistrement du fichier csv

Dans le fichier xml, nous avons spécifié que la méthode “importer_CSV” est une action planifiée et nous lui avons affecté l’intervalle d’une journée pour qu’elle puisse être exécutée. Ci-dessous un

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

exemple de quelques pointages des employés insérés dans la base de données grâce à l'import de notre fichier CSV.

Pointage RFID

Créer ou Importer 1-14 sur 14

Employé	Date Heure	Entrée/Sortie
Belhadj ibtissem	04/12/2017 08:00:00	entrée
Ouadah amina	04/12/2017 09:15:00	entrée
Saber zenagui khadidja	04/12/2017 09:30:00	entrée
Aissaoui Fatéma zohra	04/12/2017 10:00:00	entrée
Remaci zineb	04/12/2017 10:30:00	entrée
Belhadj ibtissem	04/12/2017 12:00:00	sortie
Saber zenagui khadidja	04/12/2017 12:00:00	sortie
Ouadah amina	04/12/2017 12:15:00	sortie
Aissaoui Fatéma zohra	04/12/2017 12:20:00	sortie
Remaci zineb	04/12/2017 14:00:00	sortie
Ouadah amina	13/04/2017 09:15:00	entrée
Saber zenagui khadidja	13/04/2017 10:00:00	entrée
Ouadah amina	13/04/2017 14:15:00	sortie
Saber zenagui khadidja	13/04/2017 15:00:00	sortie

Figure 32 : Table de pointage des employés remplie depuis le fichier CSV

IV.8.4.4 Enregistrement dans la base de données (sprint 7)

Les concepteurs de processing ont choisi de garder le logiciel simple. Il est donc relativement pauvre en termes de fonctions et d'ergonomie, mais parallèlement, ils l'ont conçu pour que toute personne puisse venir à étendre facilement ses fonctionnalités de base, en rajoutant des « bibliothèques externes ». Ce qui nous a amené à chercher s'il y a une bibliothèque externe pour faire la connexion avec une base de données.

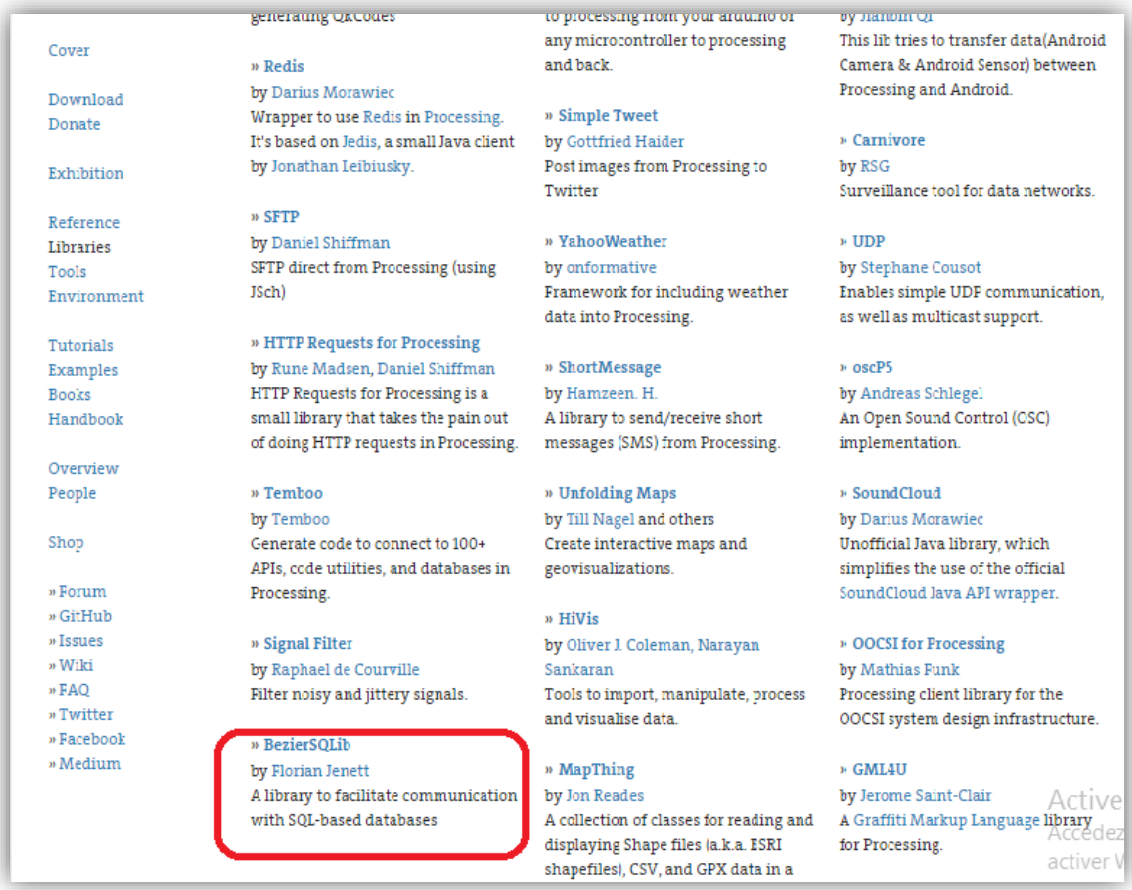


Figure 33 : Interface de processing pour les bibliothèques

C'est la librairie externe BezierSql qui nous a permis de faire la connexion avec la base de données pour faire l'enregistrement des données récupérées d'arduino dans les tables d'Odoo correspondantes.

Pour faire la connexion nous avons précisé le nom de la base de données, le mot de passe et l'utilisateur. Ensuite nous avons établi la connexion avec la base.

```
21 void setup()
22 {
23   String user     = "openpg";
24 String pass     = "openpgpwd";
25 String database = "RFIDJ";
26 pgsql = new PostgreSQL( this, "localhost", database, user, pass );
```

Figure 34 : Connexion avec la base de données

À chaque fois où l'utilisateur passe sa carte RFID devant le lecteur nous enregistrons dans la table hr.attendance les données suivantes : l'id de l'employée, l'id de l'utilisateur, l'action de l'utilisateur

entrée ou sortie, l'heure exacte ou l'utilisateur a passé sa carte, le nombre d'heures travaillées entre l'entrée et sortie, la clé étrangère entre hr.attandance et hr.timesheet.sheet.

Pour enregistrer l'id de l'employé correspondant à la carte passé nous avons sélectionné l'id de l'employé de la table hr.employe ou le champ RFID_id (que nous avons rajouté auparavant dans cette table avec un programme python) qui est égale à l'id de la carte récupérée d'arduino. Ensuite nous enregistrons l'action qui est soit une entrée ou une sortie. Voici le code correspondant :

```
86        pgsql.query("SELECT id,write_uid  FROM hr_employee WHERE rfid_id='"+sensorVals[0]+"'" );
87        while(pgsql.next())
88        {
89          //println(pgsql.getString("action"));
90          id=(pgsql.getInt( "id" ));
91          uid=(pgsql.getInt( "write_uid" ));
92              println(id);
93        }
94         pgsql.query("SELECT id  FROM hr_timesheet_sheet_sheet WHERE employee_id="+id+"" );
95        while(pgsql.next())
96        {
97
98          Time_sheet=(pgsql.getInt( "id" ));
99              println(Time_sheet);
100       }
101       pgsql.query("SELECT hr_attendance.action AS action FROM hr_attendance WHERE name=(SELECT MAX(name) FROM hr_attendance WHERE employee_id="+id+") GROUP BY employee_id,id" );
102       while(pgsql.next())
103       {
104         //println(pgsql.getString("action"));
105       String  Act=(pgsql.getString( "action" ));
106
107       if (Act.equals("sign_out"))  A="sign_in";
108
109       if(Act.equals("sign_in")) A="sign_out";
110
111       println(A);
112       }
```

Figure 35 : L'enregistrement dans la base de données

Des fois, les utilisateurs ne sont pas sûrs que leur carte est passée correctement, alors ils repassent à nouveau leur carte pour s'assurer qu'ils ont bien pointé (sortie ou entrée). Cela peut engendrer des anomalies du fait que l'utilisateur va enregistrer son entrée et sa sortie en pensant qu'il a enregistré que son entrée ou le contraire. C'est pourquoi nous avons effectué des contrôles permettant d'enregistrer l'entrée ou la sortie seulement si le temps séparant les deux actions dépasse 60 secs.

```
124
125         if((jour.getYear()==dates.getYear())&&(jour.getDate()==dates.getDate())&&((jour.getMonth()==dates.getMonth())&&(HR==dA))&&((heure.getMinutes()==dates.getMinutes()))){
126           if((d-H)<=60){
127
128         println("déja entrée");}
129         }
130        else{
131          if(A.equals("sign_in")){
132          w=0;
133        }
```

Figure 36 : Vérification de la carte si elle est passée plus d'une fois

Les enregistrements sont visibles depuis le menu Présences dans Odoo comme l'illustre la figure37.

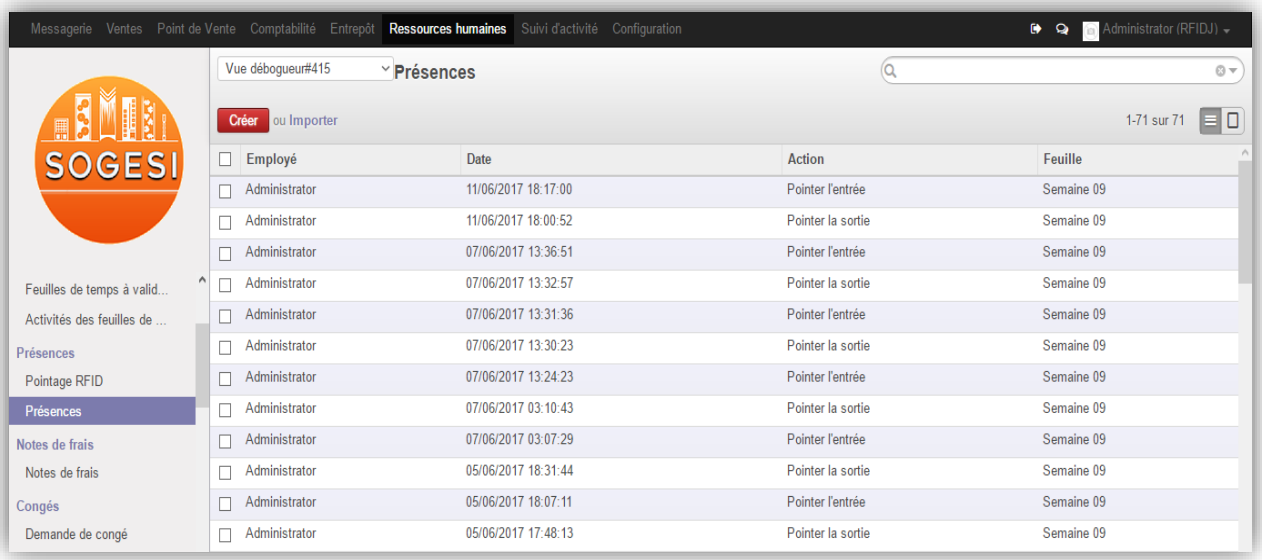


Figure 37 : Table dans Odoo des enregistrements de présence des employés

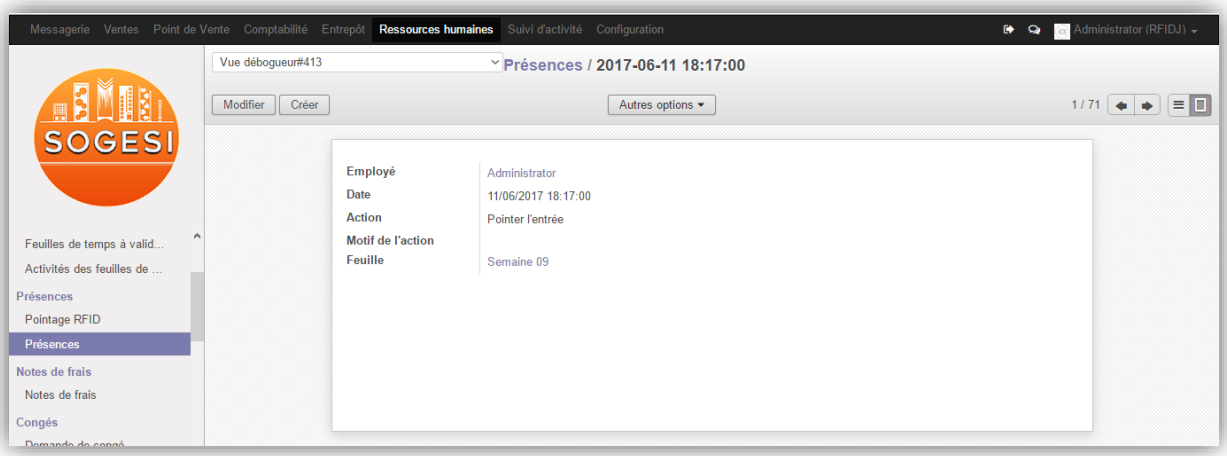


Figure 38 : Detail d'une présence d'un employé sélectionné

Le total de pointage d'un employé ainsi que son détail se trouve dans le menu Gestion du temps -> Feuille de temps à valider.

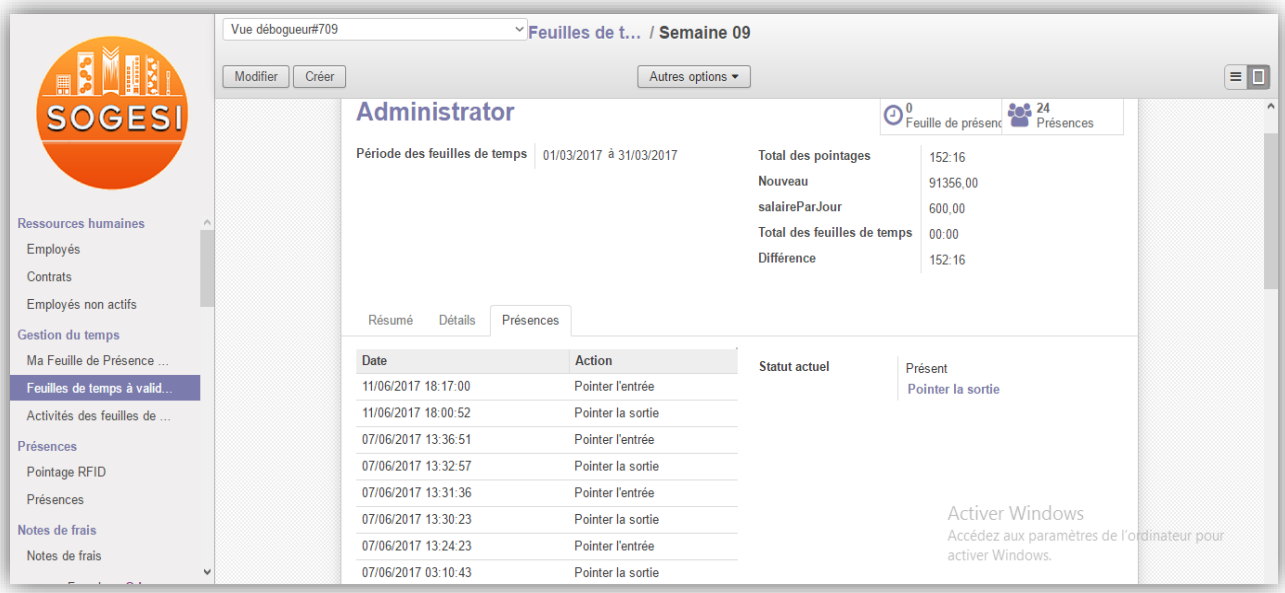


Figure 39 : Feuilles de temps à valider d'un employé

Le total de pointage qui se trouve dans la table timesheet.sheet se calcule à partir de la somme des heures travaillées de l'employé qui est dans la table hr.attendance et la relation se fait à l'aide du champ sheet_id. Nous avons calculé les heures de travail dans notre programme processing.

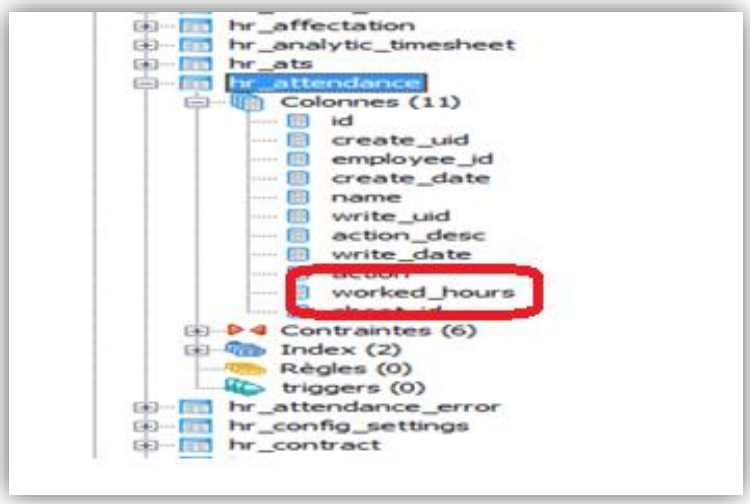


Figure 40 : La table hr_attendance

IV.8.4.5 Calcul de la paie à partir des heures travaillées (Sprint 8)

Après le calcul des heures travaillées pour chaque employé nous avons calculé la paie à partir de ces heures et cela après que nous avons précisé le coût d'une heure de travail. Comme le montre la figure 38.

Figure 41 : Calcule de la paie

IV.9 Conclusion

Dans ce chapitre nous avons présenté notre solution pour la gestion des entrées et sorties dans Odoo avec la technologie RFID qui n'était pas assurée auparavant par Odoo. Nous avons montré comment nous avons utilisé la méthode agile Scrum pour mieux gérer notre projet.

Pour ce faire, nous avons décomposé notre travail en deux parties : électronique et informatique. Dans la partie informatique nous avons présenté deux solutions. La première consistait à construire un fichier CSV et l'importer automatiquement dans la base de données Odoo. La seconde solution est plus sophistiquée et plus intéressante. Elle consiste à enregistrer les informations du pointage RFD directement dans la base de données Odoo en temps réel.

Conclusion générale

Dans ce manuscrit nous avons eu le plaisir de présenter notre projet de fin d'étude qui consiste à intégrer une solution RFID dans la gestion du pointage Odoo.

Ce projet a été une l'occasion pour nous de côtoyer le monde professionnel, le monde des ERP en particulier et de découvrir une nouvelle technologie qui est la RFID. Le travail a été réalisé au sein de la société SOGESI.

Nous avons commencé ce projet en nous familiarisant avec l'environnement d'Odoo et en effectuant des recherches et en appliquant ce que nous avons appris. Il faut souligner que le fait d'avoir travaillé dans le monde de Odoo nous a permis d'acquérir une large partie des technologies notamment le langage python, xml, Qweb, sql, et de rejoindre une communauté mondiale de plus de 1500 de développeur Odoo.

Ensuite, il a fallu intégrer la technologie RFID dans Odoo. Comme nous avons pu le constater, la technologie RFID n'est pas une technologie qui vient de voir le jour. Cependant ses applications dans les domaines industriels et commerciaux sont récentes et les potentialités pour l'avenir sont sans limites. Ce qui nous a amenés à exploiter cette technologie dans le cadre d'Odoo afin de simplifier la gestion du pointage des employés.

Au final donc, il est important de mentionner que ce projet nous permis d'acquérir de nouvelles connaissances qui viennent compléter celles que nous avons acquises tout au long de notre formation.

Néanmoins, nous pouvons toujours apporter des améliorations à ce travail en ouvrant la voie vers de nouvelles perspectives dans des travaux de recherches ultérieurs. En effet, nous proposons d'enrichir notre application et cela en intégrant la RFID avec la gestion de la production et la gestion de stocks d'Odoo. Ceci pourrait permettre une gestion plus intelligente et plus efficace des produits et augmenter considérablement la productivité des entreprises. Nous souhaitons également intégrer Odoo et la RFID dans d'autres domaines sensibles et vitaux, à l'exemple du domaine médical de celui des transports, etc.

Rapport-Gratuit.com

RÉFÉRENCES

[1] http://www.sogesi-dz.com/ consulté le 10/03/2017

[2] Lequeux, Jean-Louis. "Manager avec les ERP." *Paris: Ed. d'Organisation* (2002).

[3] http://www.webopedia.com/TERM/E/ERP.html consulté le 25/03/2017

[4] Barki, Henri, and Alain Pinsonneault. "Explaining ERP implementation effort and benefits with organizational integration." *Cahier du GReSI no* 2.01 (2002).

[5] Rashid, Mohammad A., Liaquat Hossain, and Jon David Patrick. "The evolution of ERP systems: A historical perspective." (2002): 2-6.

[6] http://www.celge.fr/editeurs/Odoo-logiciel-de-gestion-dentreprise-crm-erp-facturation-comptabilite-gpao-cms-ecommerce consulté le 11/04/2017

[7] http://www.enterpriseappstoday.com/erp/what-is-openerp.html consulté le 18/04/2017

[8] https://www.Odoo.com/fr_FR/ consulté le 03/04/2017

[9] https://doc.Odoo.com/6.0/book/1/1_1_Inst_Config/1_1_Inst_Config_architecture/

consulté le 01/05/2017

[10] Moss, Greg. *Working with Odoo*. Packt Publishing Ltd, 2015

[11] https://doc.Odoo.com/trunk/server/02_architecture consulté le 22/03/2017

[12] http://blog.atlasRFIDstore.com/active-RFID-vs-passive-RFID consulté le 06/05/2017

[13] https://www.atlasRFIDstore.com/active-RFID/ consulté le 27/04/2017

[14] https://www.transplanet.fr/biometrie/lecteur_d_empreintes_digitales.html consulté le 15/03/2017

[15] Katamba, Paulin Ilunga. *Technologie RFID (Radio Frequency Identification): concepts et stratégie de mise en oeuvre*. Diss. Université Laval, 2007.

[16]http://www.commentcamarche.net/contents/1028-RFID-radio-frequency-identification

consulté 19/05/2017

[17] L'IDENTIFICATION, PAR RADIOFRÉQUENCE RFID, and SÉCURITÉ DE. "Ce document réunis les trois rapports de l'OCDE sur l'identification par radiofréquence (RFID)." (2008).

[18] Paret, Dominique. *RFID en ultra et super hautes fréquences: UHF-SHF: Théorie et mise en oeuvre*. Dunod, 2008.

[19] http://europa.eu/rapid/press-release_MEMO-09-232_en.htm?locale=en consulté 25/05/2017

[20] http://RFID.comprendrechoisir.com/comprendre/systeme-RFID consulté 19/05/2017

[21] http://www.paragon-rfid.com/avantages-rfid/ consulté 20/05/2017

[22] http://apprendrez.info/article/les-inconvnients-de-la-technologie-rfid consulté 20/05/2017

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

[23] Ziadé, Tarek. *Programmation Python: conception et optimisation*. Editions Eyrolles, 2009.

[24] https://doc.ubuntu-fr.org/postgresql consulté 13/03/2017

[25] http://haypo.developpez.com/tutoriel/xml/introduction/ consulté 27/03/2017

[26] https://www.arduino.cc/en/main/software consulté 06/05/2017

[27]https://openclassrooms.com/courses/processing-1/avant-de-commencer-22#/id/r-1397867 consulté 20/05/2017

[28] Delannoy, Claude. *Programmer en JAVA*. Eyrolles, 2006.

[29] https://www.arduino.cc/en/main/arduinoBoardUno consulté 17/05/2017

[30]http://www.sciencebuddies.org/science-fair-projects/how-to-use-a-breadboard.shtml#what-breadboard consulté 25/05/2017

[31] https://www.moussasoft.com/product/module-RFID-rc522-lecteur-RFID

consulté 22/05/2017

Liste des figures

FIGURE 1 : LOGICIELS PROPRIETAIRES 9
FIGURE 2 : LOGICIELS GRATUITS 9
FIGURE 3 : L'EVOLUTION DES ERP 10
FIGURE 4 : LES DIFFERENTES DOMAINES DES ERP 11
FIGURE 5 : ARCHITECTURE CLIENTS/SERVEUR 12
FIGURE 6 : LA COMMUNAUTE D'ODOO 14
FIGURE 7 : CREATION D'UNE REGLE SALARIALE 15
FIGURE 8 : STRUCTURE SALARIALE 16
FIGURE 9 : BULLETIN DE PAIE DE L'EMPLOYE 17
FIGURE 10 : LES TAGS RFID 25
FIGURE 11 : LES DIFFERENTS TYPES DE RFID 26
FIGURE 12 : FONCTIONNEMENT DU RFID 27
FIGURE 13 : ARDUINO UNO 33
FIGURE 14 : CYCLE DE VIE DE LA METHODE SCRUM 35
FIGURE 15 : LES TACHES DE TRELLO 36
FIGURE 16 : CREATION DU SPRINT 7 36
FIGURE 17 : LE DEVELOPPEMENT DU SPRINT 7 37
FIGURE 18 : LA FIN DU SPRINT 7 37
FIGURE 19 : LES HUIT SPRINTS DU PROJET 38
FIGURE 20 : DIAGRAMME DE COMPOSANTS 40
FIGURE 21 : RFID RC522 41
FIGURE 22 : BRANCHEMENT DU LECTEUR ARDUINO AVEC RFID 42
FIGURE 23 : TABLEAU DE CORRESPONDANCE ENTRE RFID ET ARDUINO 43
FIGURE 24 : RFID ET SA BROCHE 43
FIGURE 25 : CODE ARDUINO 44
FIGURE 26 : LA RECUPERATION ET L'ENVOI DU NUMERO DE CARTE A PROCESSING 45
FIGURE 27 : CODE PROCESSING POUR LA CONNEXION ENTRE ARDUINO ET PROCESSING 46
FIGURE 28 : CODE POUR RECUPERER LES DONNEES D'ARDUINO 46
FIGURE 29 : PROGRAMME POUR L'ENREGISTREMENT DU FICHIER CSV 47
FIGURE 30 : PROGRAMME PYTHON POUR L'ENREGISTREMENT DU FICHIER CSV DANS LA BASE DE DONNEES 48
FIGURE 31 : PROGRAMME XML POUR L'ENREGISTREMENT DU FICHIER CSV 48
FIGURE 32 : TABLE DE POINTAGE DES EMPLOYES REMPLIE DEPUIS LE FICHIER CSV 49
FIGURE 33 : INTERFACE DE PROCESSING POUR LES BIBLIOTHEQUES 50
FIGURE 34 : CONNEXION AVEC LA BASE DE DONNEES 50
FIGURE 35 : L'ENREGISTREMENT DANS LA BASE DE DONNEES 51
FIGURE 36 : VERIFICATION DE LA CARTE SI ELLE EST PASSEE PLUS D'UNE FOIS 51
FIGURE 37 : TABLE DANS ODOO DES ENREGISTREMENTS DE PRESENCE DES EMPLOYES 52
FIGURE 38 : DETAIL D'UNE PRESENCE D'UN EMPLOYE SELECTIONNE 52

FIGURE 39 : FEUILLES DE TEMPS A VALIDER D'UN EMPLOYE 53
FIGURE 40 : LA TABLE HR_ATTENDANCE 53
FIGURE 41 : CALCULE DE LA PAIE 54

Liste des tableaux

TABLEAU 1 : LA STRUCTURE D'UN MODULE ODOO 14
TABLEAU 2 : COMPARAISON DES DIFFERENTS TYPES DE TECHNOLOGIE D'IDENTIFICATION AUTOMATIQUE 22
TABLEAU 3 : LES DIFFERENTS TYPES DE FREQUENCE DE LA RFID 24

Résumé

Le progiciel Open source Odoo permet aux développeurs de réaliser des solutions sur mesure aux besoins des clients en leur procurant une réduction dans le coût et un gain en matière de productivité et d'efficience par rapports à d'autres logiciels.

Dans ce travail, nous nous sommes intéressés à l'enrichissement de l'ERP Odoo en l'adaptant à une nouvelle technologie qui est la RFID. Une technologie largement déployée actuellement et qui ne cesse de croître dans de multiples domaines.

À travers l'intégration de la RFID dans Odoo, nous avons pu assurer le pointage des employées d'une manière automatique et simplifié.

Mots clés : Odoo, ERP, RFID.

Abstract

Odoo is an open source software that enables developers to realize solutions to customer needs by reducing costs and improving productivity and efficiency compared to other software.

In this work, we were interested in enriching the Odoo ERP by adapting it with a new technology that is RFID, a technology widely deployed today and which is constantly growing in multiple fields.

Through the integration of RFID in Odoo, we were able to ensure that employees were automatically and easily scored.

Keywords: Odoo, ERP, RFID.

ملخص

Odoo هو برنامج مفتوح المصدر يتيح للمطورين لتحقيق حلول لاحتياجات العملاء من خلال خفض التكاليف وتحسين الإنتاجية والكفاءة مقارنة مع البرامج الأخرى.

في هذا العمل، نحن مهتمون في إثراء التخطيط لموارد المؤسسات Odoo من خلال تكييفه مع التكنولوجيا الجديدة التي هي التعريف المفعل بتردد الراديو، وهي تكنولوجيا منتشرة على نطاق واسع اليوم والتي تتزايد باستمرار في مجالات متعددة.

من خلال دمج التعريف المفعل بتردد الراديو في Odoo ، كنا قادرين على ضمان تسجيل حضور الموظفين تلقائيا وبسهولة تامة.

الكلمات المفتاحية: Odoo ، التعريف المفعل بتردد الراديو، التخطيط لموارد المؤسسات.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES