

République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid– Tlemcen

Faculté des Sciences

Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option : Génie Logiciel (GL)

Thème

**Développement d'une application mobile / Tablette pour la
gestion de production intégrée à Odoo.**

Réalisé par :

- BELFADEL Wafa.
- LARBI Fatima Zohra.

Présenté le 06 Juillet 2019 devant le jury composé de :

- *Mr Maatallah Hocine* (Président).
- *Mme. Halfaoui Amel* (Examinatrice).
- *Mr MESSABIHI Mohamed* (Encadrant).
- *Mr BRAHMI Noredine* (Maitre de stage).

Année universitaire : 2018-2019

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

صَدَقَ اللَّهُ الْعَظِيمَ

Remerciement

Tout d'abord nous remercions notre créateur le dieu de nous avoir donné la volonté et la patience d'achever notre objectif et réaliser ce travail.

Nous présentons nos profonds respects et reconnaissance à notre Encadreur Mr MESSABIHI Mohammed pour son soutien et son aide tout au long de nos parcours.

Nos vifs remerciements vont aussi à Monsieur Brahmi Noureddine, le directeur de la société SOGESI qui a été plus qu'un maitre de stage, il nous a guidé, encouragé de faire mieux et offert une expérience professionnelle et enrichissant. Nous tiendrons aussi a remercié l'équipe SOGESI pour le support et les conseils au long de déroulement de ce stage.

Aussi nous adressons nos sincères remerciements à tous nos enseignants de Abou Bakr Belkaid et tout le staff administratif pour ses suivis et ses orientations durant toutes les années de nos études.

Nos profonds remerciements pour les membres de jury qui ont accepté d'évaluer ce travail.

Que tous ceux qui ont contribué à mener à bien ce mémoire trouvent ici l'expression de notre parfaite considération

Dédicace

Je dédie ce travail :

*A ma chérie, mon espoir, ma réussite, ma source de
tendresse, à ma mère ...*

A mon cher, mon pilier dans cette vie, mon destin, à mon père

...

*Ils sont tous les deux mon soutien tout au long de mes années
d'études. Que dieu les garde et les protège.*

A mes adorables, mes sœurs Manel, Imene, et Ilheme.

A toute ma famille.

A mon binôme Fatima, pour sa gentillesse, et son honnêteté.

A mes ami(e)s.

A tous ceux qui m'aiment, à tous ceux que j'aime.

Wafa

Dédicace

Je dédie ce travail :

À celle qui m'a donné la vie, qui s'est sacrifié pour mon bonheur et ma réussite, ma tendre mère.

À celui qui a toujours été à mes côtés pour me soutenir et qui a été mon ombre durant toutes les années des études et ma vie ... mon cher père, que dieu les gardes et les protèges.

À Mon adorable sœur et mes frères qui m'ont chaleureusement supporté et encouragé tout au long de mon parcours.

A Mme Labraoui Nabila ... sans son conseil je ne serais pas ici aujourd'hui.

A mon Binôme Wafa pour son sérieux et pour prouver sa Sincérité et professionnalisme.

À ma famille et toutes mes amies pour l'amour, le respect et le support qu'ils m'ont accordé

A tous ceux que j'aime et ceux qui m'aiment

Merci

Fatima Zohra

Table des matières

Liste de figures.....	3
Liste des tableaux	4
Liste des abréviations	5
Introduction générale	6
Chapitre I : La gestion de production	10
I.1 Introduction.....	11
I.2 La gestion de production.....	11
I.2.1 Les modes de production.....	11
I.2.2 Les objectifs de production	12
I.3 La gestion de production assistée par ordinateur (GPAO).....	13
I.3.1 définition de GPAO.....	13
I.3.2 Les fonctionnalités de GPAO	13
I.5 Conclusion	14
Chapitre II : Entreprise Resource Planning (ERP)	15
II.1 Introduction	16
II.2 Les ERP	16
II.2.1 Les caractéristiques d'un ERP	17
II.2.2 Les différents environnements d 'ERP	17
II.2.3 Les types d 'ERP :.....	18
II.3 ODOO.....	19
II.3.1 Architectures fonctionnelles et techniques	19
II.3.2 Points forts d'ODOO	24
II.3.3 Les Services	24
II.3.4 Les avantages d'avoir Odoo comme base d'application.....	25
II.4 Conclusion	25
Chapitre III : Conception et réalisation.....	26
III.1 Introduction	27
III.2 Techniques utilisées	27
III.2.1 Web services	27

III.2.2 Android	30
III.3 Méthode de réalisation de notre application	31
III.3.1 Présentation de la méthode Scrum.....	31
III.3.2 Les sprints	32
III.3.2.1 Sprint 1 : Installer et configurer les outils & Gaps Analyses	34
III.3.2.2 Sprint 2 : Améliorer le module de la gestion de production	37
III.3.2.3 Sprint 3: Importer les données de Odoo vers l'application.....	42
III.3.2.4 Sprint 4: Enregistrement des données de l'application vers Odoo	46
III 3.2.5 Sprint 5 : Ajuster le design et réaliser la version tablette.....	55
III.4 Conclusion.....	58
<i>Conclusion Générale</i>	59
<i>Références</i>	61

Liste de figures

figures	pages
Figure I.1 : Les aspects de production organisés par GPAO	13
Figure II.1 : Caractéristiques des ERP	18
Figure II.2 : ERP type Open Source	19
Figure II.3 : ERP type propriétaire	20
Figure II.4 : Architecture technique d'Odoo	21
Figure II.5 : Architecture Modulaire d'Odoo	22
Figure II.6 : Les composants d'un module sous Odoo	23
Figure II.7 : Architecture MVC d'Odoo	24
Figure III.1 : Architecture des Fonctionnements des Web-services	30
Figure III.2 : Schéma SOAP	31
Figure III.3 : Schéma XML-RPC	32
Figure III.4 : Interface Trello	35
Figure III.5 : Diagramme cas d'utilisation pour responsable de production	36
Figure III.6 : Diagramme de cas d'utilisation pour les opérateurs	40
Figure III.7 : Classes participantes pour système de production	42
Figure III.8 : Création d'Ordre de fabrication	43
Figure III.9 : Création d'Ordre de travail	43
Figure III.10 : Les droits d'accès ajoutés	44
Figure III.11 : Diagramme de séquence « s'authentifier »	45
Figure III.12 : Fonction pour configurer XML-RPC	46
Figure III.13 : Appelle de la fonction de configuration	46
Figure III.14 : Connexion au serveur Odoo	46
Figure III.15 : Scan le code QR de l'opérateur	46
Figure III.16 : Page des listes OT d'un opérateur	47
Figure III.17 : Interface principale d'un OT	47
Figure III.18 : Menu	48
Figure III.19 : Consigne d'un OT	48
Figure III.20 : Plan d'un OT	48
Figure III.21 : Diagramme de séquence « gérer les quantités »	49
Figure III.22 : Diagramme de séquence « gérer temps réels »	50
Figure III.23 : Diagramme de séquence « gérer les demandes d'interventions »	50
Figure III.24 : Créer une note et la sauvegarder	51
Figure III.25 : Créer une note par vocal et la sauvegarder	51
Figure III.26 : Les quants déclaré dans un emplacement du stock	52
Figure III.27 : Enregistrer les quantités de production	53

Figure III.28 : Créer les temps d'une opération	54
Figure III.29 : Bouton floating	54
Figure III.30 : Sauvegarder les temps et sortir	54
Figure III.31 : Créer demande d'intervention	55
Figure III.32 : Création d'une demande d'intervention	56
Figure III.33 : Annuler la production	56
Figure III.35 : Des captures d'écran pour la version tablette d'application	58

Liste des tableaux

Tableaux	pages
Tableau III.1 : Les sprints du projet	34
Tableau III.2 : Les règles de gestion	38

Liste des abréviations

Abréviation	Désignation
ERP	Enterprise Resource Planning
OF	Ordres de Fabrication
OT	Ordres de Travail
DI	Demande d'Intervention
XML	L'EXtensible Markup Language
RPC	Remote Procedure Call
GPAO	Gestion de Production Assistée par Ordinateur
CBN	Le Calcul des Besoins Nets
MVC	Model Controller View
SOAP	Simple object Access Protocol
WSDL	Web Services Description Language
IDE	Integrated Development Environment

Introduction générale

Contexte :

Notre mémoire représente une description du travail que nous avons réalisé lors de la préparation de notre projet de fin d'étude de master en informatique option génie logiciel (GL) au sein de l'université de Tlemcen Abou Bakr Belkaid . Le stage s'est déroulé dans l'entreprise SOGESI située à Tlemcen depuis le **03 Février** jusqu'au **23 Juin 2019**.

Nous avons choisi ce sujet afin de découvrir le monde professionnel, prendre la conscience de l'intérêt du travail en équipe, et acquérir des connaissances sur le domaine de la gestion de production industrielle. Notre but est de développer une solution mobile/tablette destiné aux opérateurs de la production. Cette solution est proposée afin de répondre aux besoins d'un établissement industriel client de l'entreprise SOGESI dans laquelle s'est déroulé notre stage.

Notre projet s'inscrit dans le cadre de la réalisation d'une solution mobile/tablette qui utilise l'ERP open source Odoo comme base pour permettre aux opérateurs de la production d'accomplir leurs tâches liées à la production très facilement, dans un temps minimal, et avec une grande efficacité, ainsi de minimiser le travail du gestionnaire de production et de lui faciliter la communication à distance avec les opérateurs qui sont dans des ateliers parfois très dispersés géographiquement.

Présentation de l'organisme d'accueil :

SOGESI* est une société de développement destinée pour intégrer des solutions informatiques pour ses clients.

Elle permet de :

- Mettre en place et gérer des projets, conseiller et orienter les clients, communiquer et propulser de nouvelles entreprises.
- proposer une suite complète de Solution ERP Odoo pour les particuliers et les Entreprises nationales et internationales, ainsi les nouvelles technologies (application mobiles et site E-commerce).

* : <http://www.sogesi-dz.com>

Cette société fournit à ses clients les services suivants :

- **Conception** : inclus la rédaction des cahiers de charges aux clients.
- **Développement** : soit des applications mobiles ou des applications modulaires.
- **Formation Fonctionnelle** : Après l'installation des systèmes au sein des bureaux du client, SOGESI garantit que l'utilisateur a bien acquis les connaissances nécessaires concernant les fonctionnalités du système.
- **Formation Technique** : SOGESI offre une formation autour ERP Odoo, les langages de programmation (Python, Java...), les technologies web et autre pour l'installation et utilisation Git.
- **Maintenance** : SOGESI propose un contrat de maintenance pour rassurer ses clients.
- **Support.**

Problématique :

Notre travail traite un cas réel d'un client de l'entreprise SOGESI. Le cœur de métier de ce client est la production de tuyaux en PVC et en Polyéthylène. Mais la gestion de cette production connaît plusieurs problèmes.

Parmi ces problèmes, nous indiquons :

- Gestion manuelle des ordres de fabrication
- Difficulté de suivi d'état des ordres de travaux,
- Retard dans la déclaration par les opérateurs des quantités des produits fabriqués et des rebuts,
- Manque de fiabilité par rapport aux temps de travail passés par chaque opérateur,
- Gestion manuelle des demandes d'intervention en cas de panne d'un équipement ce qui provoque des problèmes d'intégrité de l'information,
- Manque de fiabilité dans l'information remontée par les opérateurs,
- Problème dans la fiabilité du calcul du prix de revient.

La gestion actuelle engendre une perte de temps considérable et des erreurs à répétition ce qui altère la fiabilité dans la gestion de l'information.

Notre travail a permis de réduire considérablement des difficultés et nous allons donc vous le détailler dans les chapitres à suivants.

Objectif du Projet :

Les objectifs de notre PFE sont :

- Faciliter la gestion de production par les opérateurs.
- Proposer une interface mobile/tablette ergonomique.
- Créer le lien entre l'application mobile/tablette et la base de données Odoo.

Plan de mémoire :

Notre mémoire est constitué de trois chapitres. Dans le premier chapitre, nous allons décrire la gestion de la production. Dans le deuxième chapitre, nous allons présenter les ERP en général et Odoo en particulier comme une base de notre application. Dans le dernier chapitre, nous allons expliquer les étapes de la réalisation de notre projet en utilisant la méthode Scrum.

Ce mémoire est clôturé par une conclusion générale et quelques perspectives.

Chapitre I :
**La gestion de
production**

I.1 Introduction

La gestion de la production est le processus de planification, organisation, direction et contrôle d'activité de production qui s'appliquent sur un ensemble de matière première pour les transférer en un produit fini, elle se résume par la conception, la combinaison entre les ressources matérielles, humaines et financières dans un plan pour assurer la fabrication dans les meilleures conditions.

La gestion de production regroupe les différentes activités liées à la préparation de la production (budget hommes et machines, mise à disposition d'équipements prêts à produire), l'ordonnancement (la mise des ordres de fabrication sur machines), la production elle-même avec son lot de tâches annexes : contrôle des produits en ligne, l'enregistrement ou déclaration des produits qui entrent en stock, des produits consommés (sorties de stock), des heures,....

Pour mieux comprendre ce qu'est la production, sa gestion, quels sont ses modes etc, nous avons réalisé ce chapitre pour présenter un peu plus de détail dans ce domaine.

I.2 La gestion de production

La production est l'opération qui permet de transformer un ensemble des composants (matières premières) en un produit fini qui satisfait tous les besoins du client et respecte certaines normes (qualité, délai, cout).

I.2.1 Les modes de production

Les modes de la production sont :

- **la production par unité** : Permet de fabriquer un produit unique ou d'une faible quantité, l'organisation ne produit l'article si seulement si ce dernier est commandé, les produits concernés par ce type de production sont les complexes de haute couture (avion, bateau ...).

- **la production en série** : permet de fabriquer un grand nombre de produit. La fabrication se fera en série peut être unique ou répétitive selon le besoin du client.
- **la production continue** : Cette méthode consiste à fabriquer sans arrêt 24h/24h la production industrielle comme électriques, nucléaires concerné par ce genre de production qui nécessite la continuation pour conserver leur intégrité.

I.2.2 Les objectifs de production

Les objectifs de productions les plus importants se résument dans la phrase suivante « produire des bons services avec bonne qualité et quantité au bon moment avec un meilleur cout ».

- **Bonne Qualité** : La qualité du produit est un élément important pour chaque organisation pour satisfaire les besoins clientèle. Sans qualité, le produit ne peut pas s'établir dans les compétitions des marques.

La bonne qualité ne représente pas forcément la meilleure qualité, il se détermine par le cout et les caractéristiques techniques des besoins.

- **Bonne Quantité** : Le processus de production assure une fabrication d'un convenable nombre de produit de la quantité demandé. Si le nombre est plus grand que celui de la demande ça posera un problème dans le stock ..., par contre si la quantité est plus petite ça conduit vers une pénurie de production.

- **Bon délai** : Afin d'évaluer l'effectivité de production, le temps joue un rôle très important dans ces jugements .devrant être réalisé dans des délais raisonnables pour être conforme au niveau de la livraison.

- **Meilleur coût** : Le coût de production est défini avant que le produit soit fabriqué. L'entreprise doit proposer des coûts qui garantissent la compétition avec les autres entreprises, pour cela le coût est considéré comme une majeure caractéristique de ce produit.

I.3 La gestion de production assistée par ordinateur (GPAO)

I.3.1 définition de GPAO

GPAO (Gestion de Production Assistée par Ordinateur) est une solution permet d'organiser tous les aspects de la production : la conception, la planification des différentes ressources (matérielles, humaines, financières), l'ordonnancement, la traçabilité, l'enregistrement et le contrôle de la production.



Figure I.1 : Les aspects de production organisés par **GPAO**

L'utilisation de cette solution assure le suivi opérationnel des produits et respecte les délais et les coûts et coordonne les services pour arriver à l'objectif demandé.

La GPAO, destinée particulièrement aux entreprises industrielles et de production[1].

I.3.2 Les fonctionnalités de GPAO

GPAO regroupe un ensemble de fonctionnalités qui gère le cycle de production. Quelles sont alors les **fonctionnalités essentielles** ?

- **La gestion des données techniques** : Un logiciel GPAO prend en compte l'élaboration des **Gammes** opératoires à la réalisation de produit (temps prévu de fabrication et de réglage, temps d'attente, ressources nécessaires)

et des **Nomenclatures** (quantité de composant nécessaire, quantité de rebuts prévisionnels, ...), définir les **articles** (la matière première ...).

- **Le Calcul des Besoins Nets (CBN)** : Permet de définir un planning de lancement des articles nécessaires à la production (les quantités et le moment auquel il faut lancer la fabrication).

- **La gestion des Stocks** : GPAO prend en compte la gestion des commandes. Cela intègre notamment les quantités, délais,...

Concernant le stock GPAO prend en considération le calcul des besoins net et le stock disponible dans l'entrepôt, le numéro de lots (série) et mesurer la rotation des produits : FIFO(**F**irst **I**n **F**irst **O**ut), LIFO(**L**ast **I**n **F**irst **O**ut).

- **La gestion des coûts** : GPAO intègre le calcul des coûts réels à partir des informations enregistrées en fabrication.

Cette **gestion des coûts** peut analyser des frais par ordre de fabrication, article, client...

- **La traçabilité** : permet de poursuivre toutes les étapes de fabrication (mesurer les consommations des composants, contrôler l'avancement du produit, assurer la traçabilité). [2]

I.5 Conclusion

La gestion de production est un domaine très vaste. Dans ce chapitre, nous avons donné une idée sur le monde de production qui se relie de plus en plus au mode numérique. Ensuite, nous avons parlé de leur gestion en informatique qui permet de garder les traces de ses opérations à l'aide des logiciels. Et à la fin nous avons montré que les ERP sont considérés comme une solution dynamique pour faciliter la gestion de production.

Dans le chapitre qui s'en suit, nous allons présenter plus d'informations sur ces ERP.

Chapitre II :
**Entreprise
Resource
Planning
(ERP)**

II.1 Introduction

Actuellement, toutes les entreprises veulent gérer ses processus internes comme la gestion de production, la gestion des ressources humaines, la gestion comptable, la gestion des ventes, l'approvisionnement ou encore du e-commerce de manière assez simple.

Cette gestion sera difficile car il existe plusieurs processus fonctionnels qui doivent être prise en charge, et chaque processus a son propre système d'information, mais si ces entreprises peuvent gérer ces processus fonctionnels dans un seul système d'information, cela peut éliminer toutes les difficultés possibles.

Pour cela l'utilisation d'ERP est nécessaire car il permet d'optimiser la gestion interne de l'entreprise en s'assurant que :

- Tous ces processus fonctionnels sont centralisés dans un seul système d'information.
- Les données enregistrées dans le système d'information seront envoyés vers tous les modules qui auront besoins de cette donnée.

II.2 Les ERP

L'acronyme ERP nommé le **PGI (Progiciel de Gestion Intégré)** en français est un regroupement de Logiciels qu'intègre les fonctions fondamentales pour la gestion des procédures de l'entreprise (production, comptabilité, vente ... etc.).

Son particularité est d'embarquer sur un même logiciel et une même base de données ces fonctions. Les ERP en général ont une réponse rapide aux besoins de l'entreprise

Systématiquement les ERP utilisent un moteur de WORKFLOW qui permet de propager les données entrées dans tous les modules des systèmes. A l'aide de ce système les utilisateurs peuvent travailler dans un même environnement, il assure l'intégrité et l'absence de redondance des données. [3]



II.2.1 Les caractéristiques d'un ERP

Un ERP répond aux caractéristiques suivantes :

- Assurer l'unicité d'information (une seule BDD).
- Garantir la mise à jour en temps réel en cas d'impact d'un module.
- Connecter les fonctions en utilisant des données communes.
- Avoir des outils de développement ou de personnalisation de compléments applicatifs.
- Automatiser les processus de l'entreprise.
- Diffuser l'information a tous les modules concernés lorsqu'il y'a des nouvelles données.
- Avoir une uniformisation des IHM(Interface **H**omme **M**achine) pour tous les modules (même boutons, même menu, même ergonomie des écrans).
- Il permet la traçabilité de toutes les informations transmises. [4]

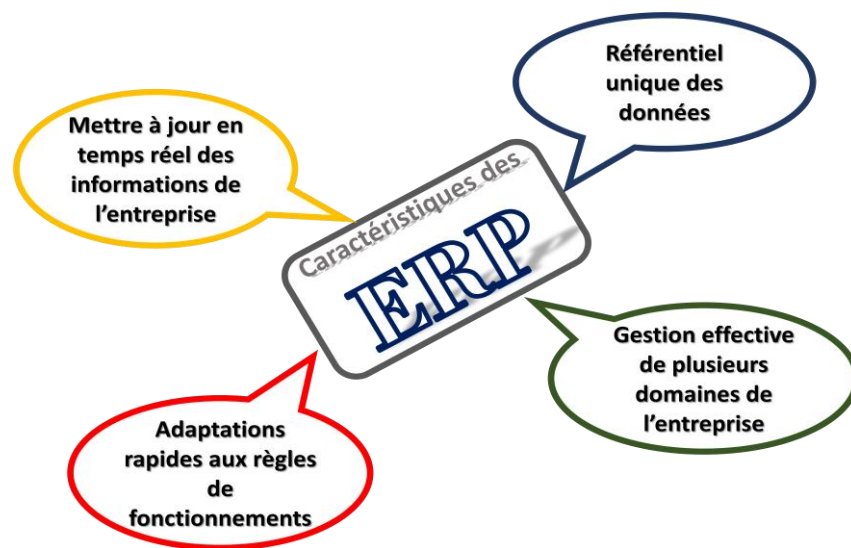


Figure II.1 : Caractéristiques des ERP

II.2.2 Les différents environnements d'ERP

Un ERP comprend généralement 3 environnements de travail : [5]

- **Environnement de Développement** : permet d'ajuster l'ERP standard à des besoins spécifiques de l'entreprise.

Par exemple, les opérateurs d'une entreprise développent des modules sous l'environnement Odoo de cette entreprise, dans ce cas Odoo représente un environnement de développement.

- **Environnement de Test (recette) :** permet de tester de nouveaux paramétrages et vérifier le fonctionnement correct du progiciel.

Par exemples, une nouvelle fonctionnalité développée sera testé dans la BDD du client, cet environnement est l'environnement de test.

- **Environnement de Production :** qui correspond au progiciel utilisé par les gestionnaires de l'entreprise au quotidien.

Par exemple, opérateurs d'une entreprise développent un module, et ses tests sont validés, il sera installé sous Odoo de l'entreprise cliente qui est totalement différent de Odoo de l'environnement de développement.

Dans ce cas Odoo de l'entreprise client représente l'environnement de Production de ce client.

II.2.3 Les types d 'ERP :

- **ERP libre :** dont le code source est rendu public. Le modèle open source permet aux développeurs d'accéder au code du système ERP et de le personnaliser à l'aide de leur propre service informatique. il existe souvent une version gratuite « communautaire » et une version professionnelle qui n'impose pas l'acquisition d'une licence. Il est nettement moins cher, mais il inclut les frais de maintenance et d'assistance technique et les services associés.

La figure suivante représente des exemples d'ERP libres sur le marché



Figure II.2 : ERP type Open Source

- **ERP Propriétaire :** édité par des sociétés et nécessite l'achat d'une licence. La figure suivante représente des exemples d'ERP Propriétaires sur le marché :



Figure II.3 : ERP type propriétaire

Les modules et les fonctionnalités des ERP propriétaires sont identiques aux ERP open sources :

Cependant, les ERP propriétaires incluent une prise en charge complète et un service personnalisé.

Ce qui différencie un ERP propriétaire d'un ERP open source, c'est la licence. Une licence ERP est une forme de contrat par laquelle une société :

- donne l'autorisation à un client d'utiliser le PGI commercialisé,
- détermine les limites de l'utilisation du logiciel,
- liste les obligations et responsabilités du client et de la société[6].

II.3 ODOO



Anciennement connu sous le nom TinyERP (2005) puis OpenERP (2009) puis Odoo en mai 2014 est un PGI open source qui permet de faciliter la gestion d'entreprise grâce à ses nombreux modules et intégrer les données dans une base de données unique accessible par un navigateur.

Il est fondé par Fabien Pinckaers en Belgique 2005, et sa dernière version stable 12.0 en octobre 2018.

Odoo est disponible en deux versions :

- **Odoo Community** : la version open source, téléchargeable et entièrement gratuite.
- **Odoo Enterprise** : la solution déployée sur un site (installée localement) qui est recommandée pour les entreprises sous licence propriétaire.[7]

II.3.1 Architectures fonctionnelles et techniques

- **Architecture technique** : composé de 3 niveaux (3-tiers) : niveau de base de données, niveau de traitement et niveau de présentation.

Cette architecture est montrée dans la figure suivante :

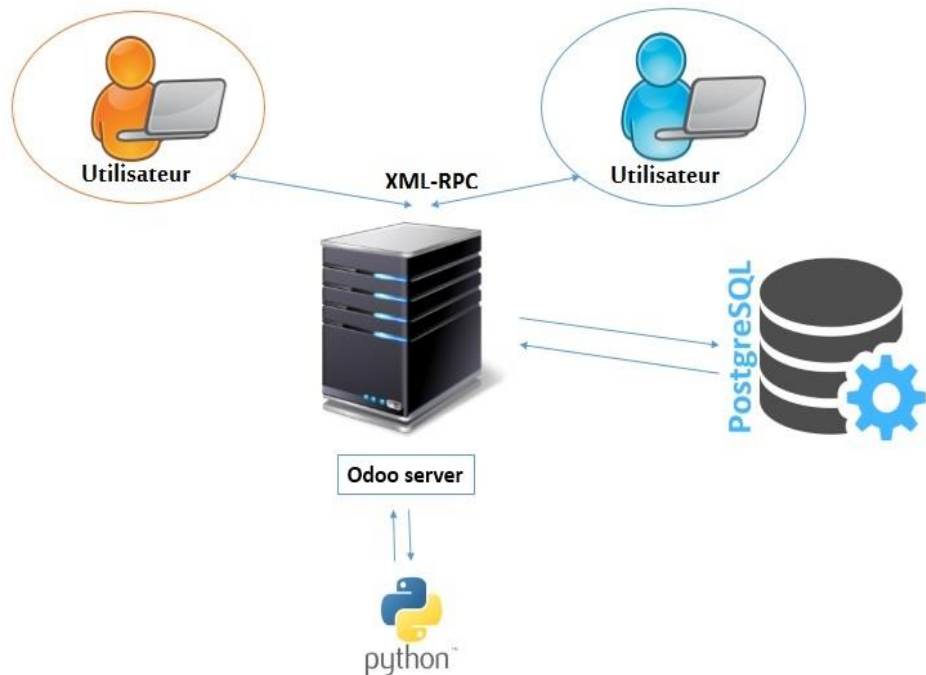


Figure II.4 : Architecture technique d'Odoo

Ces niveaux sont des couches distinctes tel que :

- ✓ **Base de données PostgreSQL : la couche mapping** est un système de gestion de base de données relationnelle objet. la plupart des accès à la base de données sont effectués via L'ORM (Object-Relational Mapping).
- ✓ **Serveur OpenERP** : est un cadre de développement complet, offrant un ensemble de techniques pour l'écriture de ces applications. Le serveur contient une couche spécifique pour communiquer avec le client basé sur le navigateur Web. Cette couche connecte les utilisateurs à l'aide de navigateurs standard vers le serveur.
- ✓ **Clients** : peuvent communiquer avec le serveur grâce au web services en utilisant le protocole XML-RPC. Ils envoient une demande au serveur et affichent le résultat de différentes manières (sous forme de formulaires, listes, calendriers, ...).

- **Architecture Modulaire :** Cette architecture est trouvée sous le serveur OpenERP qui fait parti de l'architecture précédente (architecture technique). Elle consiste à construire plusieurs modules indépendants mais partage la même base de données comme c'est montré dans la figure ci-dessous.

Ce qui obvie les saisies multiples (répétition) et évite l'ambiguïté des données de même nature (clarté).

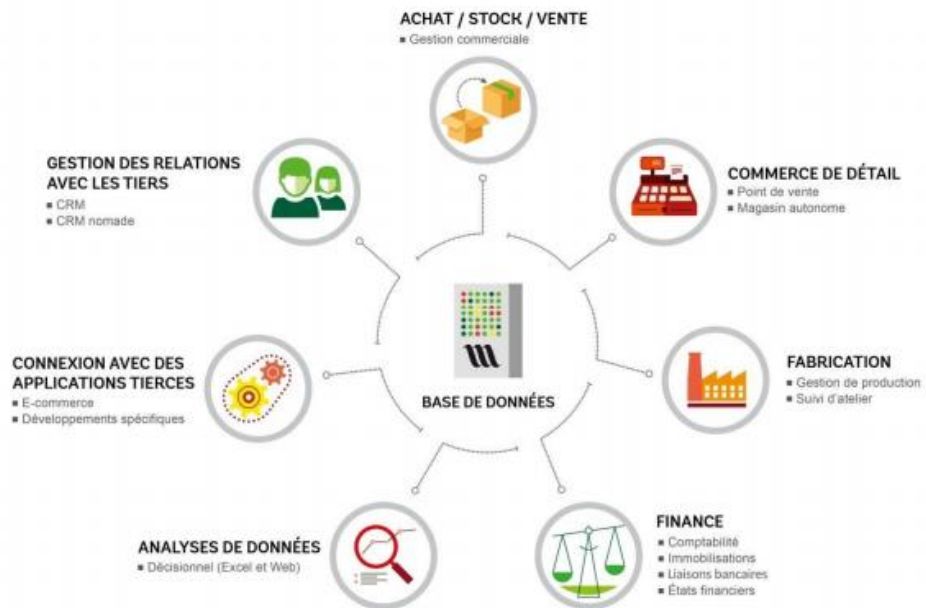


Figure II.5 : Architecture Modulaire d'Odoo

Un module est structuré sous forme d'un ensemble de composants comme c'est montré dans la figure suivante.

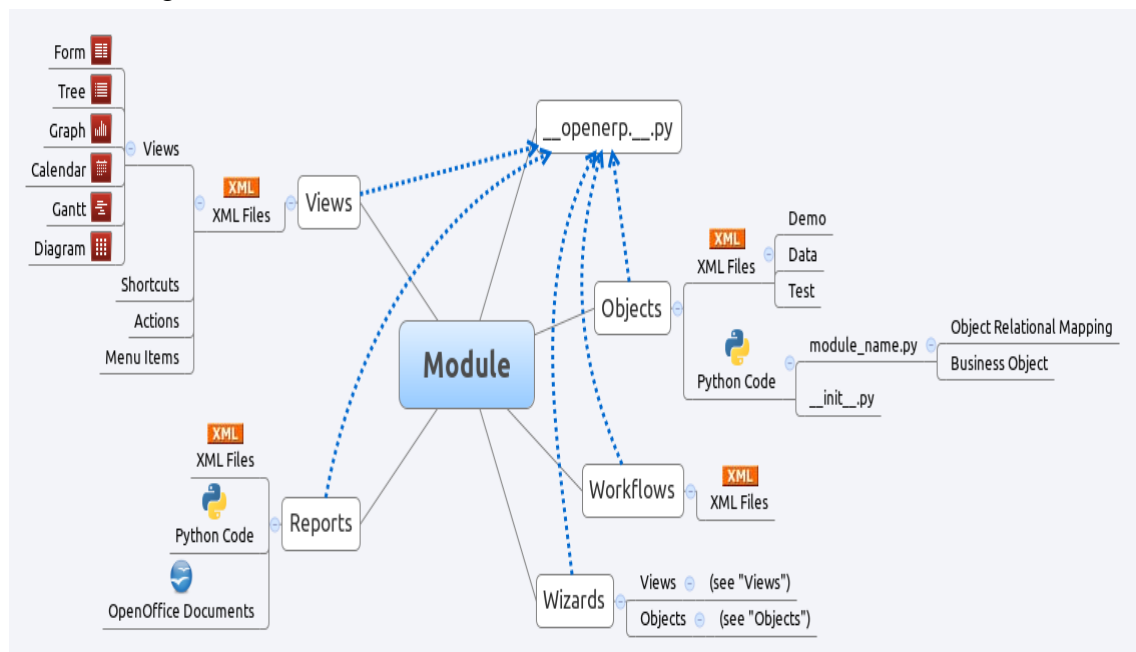


Figure II.6 : Les composants d'un module sous Odoo

Chaque composant possède son rôle tel que :

- ✓ **Objet** : c'est une classe python contient les traitements des modules (champs, fonctions ...).
- ✓ **Vue** : c'est l'affichage des objets dans le navigateur sous forme d'un formulaire, arbre, calendrier.... .Coté programmation les vues sont des fichiers XML.
- ✓ **Wizard** : est une fenêtre de dialogue contient des vues ou des objets. C'est des fichiers XML
- ✓ **Rapport** : document à imprimer s'écrit à l'aide de Qweb (html) et XML.
- ✓ **Sécurité** : gérer les droits d'accès (créer, modifier, supprimer, lire) sur les vues ou partie de la vue. [8]

- **Architecture MVC** : Chaque module fait partie de l'architecture modulaire. Odoo adopte à ce module la modélisation MVC comme c'est montré dans la figure ci-dessous.

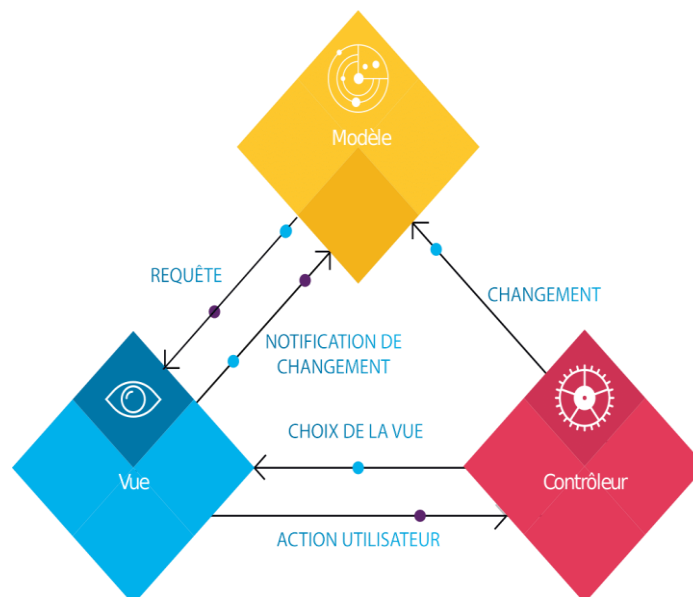


Figure II.7 : Architecture MVC d'Odoo

- ✓ **Modèle** : comme nous avons cité précédemment, nous avons utilisé le PGI Odoo comme une base de notre application. Ce PGI a des objets formulés, ces objets sont appelés des **Modèles**, et transformés à une table dans la base de données **PostgreSQL**.



Ce PostgreSQL c'est le système de gestion de bases de données relationnelles, il est basé sur POSTGRES [8]. Il est open-source, et il offre de nombreuses fonctionnalités modernes :

- requêtes complexes.
- clés étrangères.
- déclencheurs (triggers).
- vues.
- contrôle des accès

De plus, PostgreSQL peut être étendu de plusieurs façons par l'utilisateur, par exemple en ajoutant de nouveaux :

- types de données.
- Fonctions.
- Opérateurs.
- fonctions d'agrégat.
- méthodes d'indexage.
- langages de procédure.

Ce système peut être utilisé, modifié et distribué par tout le monde gratuitement quelque soit le but visé, qu'il soit privé, commercial ou académique, grâce à sa licence libérale

✓ **Vue** : sont les vues des modules dans Odoo, ils sont des fichiers écrits en **XML** tel que ce dernier est l'acronyme de e**X**tensible **M**arkup **L**anguage (qui signifie: langage de balisage extensible).

Nous avons utilisé ce langage quand nous avons adapté le module de production pour être compatible avec notre application, tel que ce langage nous a facilité la tâche d'afficher dans la vue tout ce que nous avons ajouté.



C'est un langage informatique qui sert à enregistrer des données textuelles. Il s'agit d'un langage orienté texte il est formé de balises qui permettent d'organiser les données de manière structurée. Aussi bien utilisé pour le stockage des documents, l'enregistrement des données textuelles et aussi

la transmission des données entre applications. Ainsi, XML permet de séparer le contenu de la présentation.[10]

- ✓ **Contrôleur** : les fichiers qui contrôlent Odoo.

Ces fichiers sont écrits en **Python**, tel que ce dernier est un langage de programmation interprété. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet [11].

Ce langage nous a facilité la tâche de déterminer les fonctionnalités ajoutées au module de production.



II.3.2 Points forts d'ODOO

Odoo dispose les points forts suivants :

- **Open Source** : on peut le télécharger gratuitement depuis internet sans des frais de licence ni autre coût .et le plus important c'est que Odoo est modifiable, et par conséquence il permet de satisfaire les besoins d'entreprise.
- **Hébergement flexible** : Odoo peut être hébergé sur le site ou sur une plate-forme.
- **Conception adaptée au fonctionnement d'entreprise** : nous installons seulement les modules qui nous intéressent.
- **Facile à apprendre** : grâce à son interface ergonomie très soignée.
- **Efficacité** : simplifier et éliminer la redondance dans le processus des affaires, réduire le besoin aux manuels outils, rapidité de réponses aux requêtes.

II.3.3 Les Services

Les services offerts par Odoo sont :

- **L'installation** : depuis site officiel(**), avec des étapes prédéfinis.
- **Configuration** : configurer le système pour convenir aux besoins.
- **Développement** : ajouter vos propres modules ou modification des modules existés.
- **Documentation** : Odoo offre un guide pour nous aider au développement

(**) https://www.odoo.com/fr_FR/

II.3.4 Les avantages d'avoir Odoo comme base d'application

Le serveur utilisé par sogesi est l'open source Odoo. Ce dernier nous a aidés énormément afin de réaliser notre application pour les raisons suivants :

- **Pour ses fonctionnalités :** Notre application consiste à utiliser le module de la gestion de production, d'entrepôt, et de maintenance à la fois. Odoo nous a permis d'utiliser ces trois modules facilement parce qu'il regroupe de nombreux modules.

- **Pour son intégration :** Les fonctionnalités de notre application sont reliées entre elles, et chacune utilise des données enregistrées dans des modules différents au même temps. Ce qui provoque des difficultés pour manipuler ces données à la fois, mais en utilisant Odoo ces difficultés seront éliminées parce qu'il permet d'intégrer les données dans une base de données unique et rendre le travail plus simple, à l'aide d'une interface agréable et unifiée.

- **Pour son modularité :** Odoo permet d'ajouter, modifier, supprimer des modules selon le besoin. C'est pour cette raison que nous avons pu ajouter des attributs aux objets d'un module selon nos besoins sans soucis.

- **Pour son modèle économique avantageux :** L'utilisation d'Odoo n'implique aucun frais de licence.[12]

II.4 Conclusion

Dans ce chapitre nous avons présenté les ERP en général et l'open source Odoo en particulier pour comprendre leurs fonctionnalités et leurs architectures afin d'avoir une idée claire sur l'usage et l'importance de l'utilisation des ERP.

Dans le chapitre suivant, nous allons montrer que nous avons utilisé cet open source "Odoo" comme une base pour notre application.

Chapitre III :
**Conception et
réalisation**

III.1 Introduction

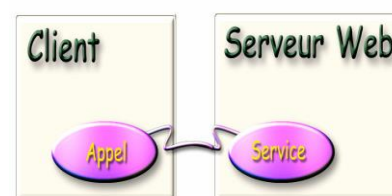
Notre mission au sein de l'entreprise Sogesi consiste à passer par plusieurs étapes : faire une étude de l'existant (le module existant) pour pouvoir le modifier et l'adapter, faire une analyse des besoins pour extraire les exigences fonctionnelles, et enfin réaliser l'application adaptable aux besoins initiaux.

Dans ce chapitre nous allons montrer plus d'informations sur ces étapes, et à la fin nous allons présenter notre application avec des captures d'écran.

III.2 Techniques utilisées

III.2.1 Web services

Dans notre projet, nous avons utilisé les web services pour avoir une communication entre l'application et la base de données Odoo. Donc il est intéressant de décrire ce qu'est un web service.



Un **web service** est un protocole d'interface informatique de la famille des technologies web [13] permettant d'interconnecter deux applications distantes via internet indépendamment des plates-formes et des langages sur lesquels elles se reposent. Cette connexion est assurée par les protocoles (HTTP, XML), et elle est basée sur le principe de demandes et réponses, effectuées avec des messages transportés sur **http** en utilisant un langage tel que **SOAP** (*Simple object Access Protocol*) ou **XML-RPC** généralement en utilisant XML. Ces services web sont décrits par des documents **WSDL** (*Web Services Description Language*) [14].

Ce **web service** nous a facilité l'appel des fonctions sur un serveur web distant en utilisant **XML-RPC**.

III.2.1.1 caractéristiques d'un web service :

Un web service répond aux caractéristiques suivantes :

1. Accessibilité via le réseau.
2. Interface publique (écrite en XML).

3. Ses fonctionnalités et comment les appeler sont stockées dans un annuaire.
4. Les messages de communication avec le web services sont écrits en XML.
5. le demandeur du service ne connaît pas forcément le fournisseur.

III.2.1.2 Architecture des Fonctionnement des Web-services :

La figure ci-dessous montre une architecture des fonctionnements des web services.

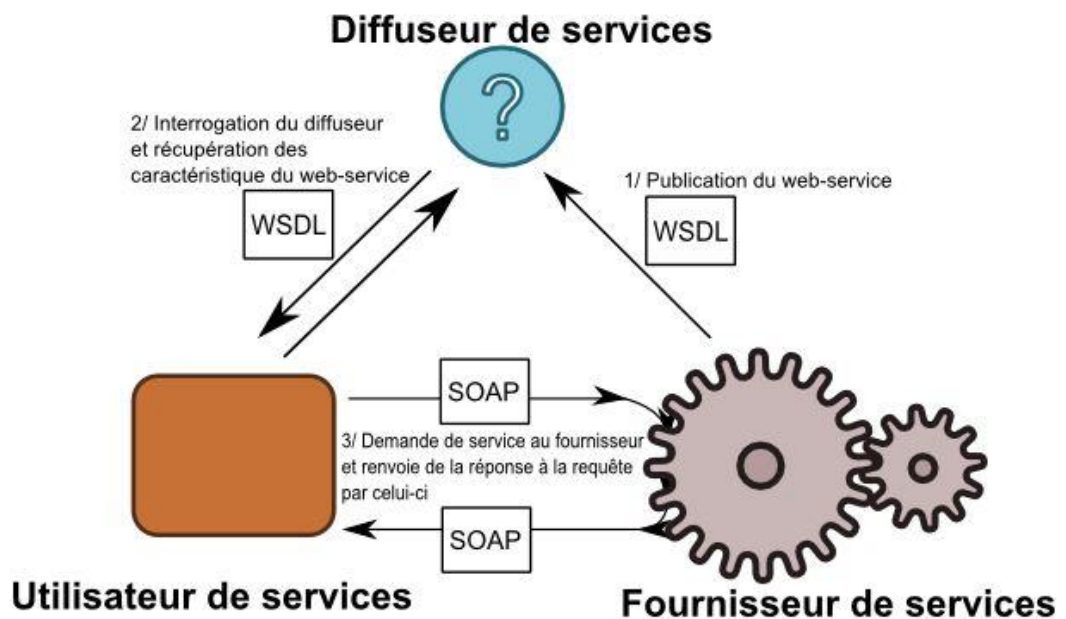


Figure III.1 : Architecture des Fonctionnements des Web-services

1. Le fournisseur fournit un document au format WSDL permettant de décrire le service proposé (adresse, arguments acceptés en entrée, type de retour etc...).
2. Le client interroge le diffuseur afin de récupérer la description du service.
3. Après avoir construit sa requête selon le schéma demandé le client l'envoie au fournisseur du service qui traite la demande et réexpédie la réponse.[15]

III.2.1.3 Les technologies utilisées par les services Web :

Parmi la technologie utilisées par les services web on trouve :

- **WSDL:** (*Web Services Description Language*) est un langage de description standard. C'est l'interface présentée aux utilisateurs. Il indique comment utiliser

le service Web et comment interagir avec lui. WSDL est basé sur XML et permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyées.[16]

- **SOAP** : (*Simple object Access Protocol*) est un protocole standard de communication. C'est l'épine dorsale du système d'interopérabilité. SOAP est un protocole décrit en XML et standardisé par le W3C. Il se présente comme une enveloppe pouvant être signée et pouvant contenir des données ou des pièces jointes. Il circule sur le protocole HTTP et permet d'effectuer des appels de méthodes à distance[15]

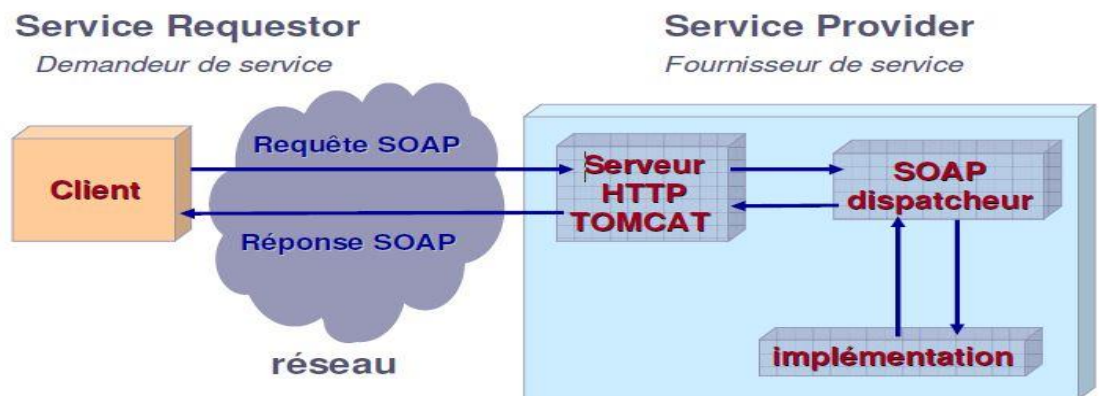


Figure III.2: Schéma SOAP.

- **XML-RPC**: XML-RPC est un protocole simple utilisant XML pour effectuer des messages RPC. Les requêtes sont écrites en XML et envoyées via HTTP POST. Les requêtes sont intégrées dans le corps de la réponse HTTP. XML-RPC est indépendant de la plate-forme, ce qui lui permet de communiquer avec diverses applications. Par exemple, un client Java peut parler de XML-RPC à un PerlServer[16]

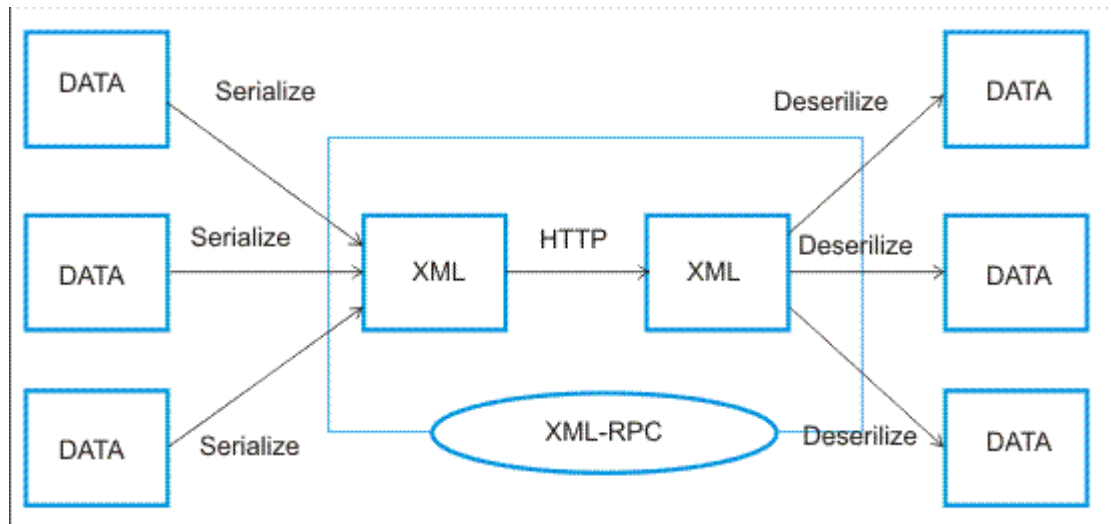


Figure III.3: Schéma XML-RPC

III.2.2 Android

• Pour notre projet nous avons développé une application mobile/tablette sous **Android**. Ce dernier est basé sur Linux. C'est un système d'exploitation ouvert dont le code source est librement accessible (contrairement aux systèmes de Apple ou Microsoft) ce qui permet à n'importe quel fabricant de l'intégrer dans son système gratuitement. Ce modèle est opposé au modèle de Apple, et explique en grande partie la forte croissance que connaît Android.[17]



Android appartient à Google qui l'a racheté en 2005 à une start up du même nom, et son propre marché d'applications nommé Google Play. [17]

Android connaît la plus forte croissance actuelle sur le marché des OS mobiles et est en train de dépasser Apple en parts de marchés mondiales.

• Ce système d'exploitation c'est le noyau de l'environnement de développement **Android Studio (IDE - Integrated Development Environment)** que nous avons utilisé pour développer notre application mobile/tablette.

Android Studio est annoncé le 15 Mai 2013 lors du Google I/O.

Il permet d'éditer des fichiers JAVA et KOTLIN et des fichiers de configuration d'une application android. Comme il a des outils qui permettent de développer des applications multilingues et de visualiser rapidement la mise en page des écrans sur des écrans de résolution différentes.[18]



- Avec cet environnement de développement nous avons utilisé les deux langages **JAVA et XML**

- **JAVA** : un langage de programmation orienté objet que nous avons utilisé pour la gestion du fonctionnement d'application[19]



- **XML** : nous avons l'utilisé pour la gestion des affichages[20]



III.3 Méthode de réalisation de notre application

III.3.1 Présentation de la méthode Scrum

Durant notre stage, nous avons géré notre projet en utilisant la méthode **Agile (Scrum)**, elle nous a facilité la tâche de spécifier, planifier et réaliser les tâches nécessaires dans les délais.

Cette méthode est considérée comme un processus empirique qui se base sur l'expérience du terrain. Elle s'appuie sur trois piliers :

- **Transparence** : Scrum met l'accent sur le fait d'avoir un langage commun entre les membres de notre équipe tel que chacun d'entre nous obtient rapidement une bonne compréhension du projet.
- **Inspection** : Scrum nous a permis de mettre le point sur les différents artefacts produits, afin de détecter toute variation indésirable.
- **Adaptation** : cette méthode (Scrum) nous a permis d'être adapté en cas où une dérive est constatée pendant l'inspection parce que Scrum fournit des événements, durant lesquels cette adaptation est possible. Il s'agit de la réunion de planification de **sprint(= itération)**, de la mêlée quotidienne, de la revue de Sprint ainsi que de la rétrospective de sprint.[21]

La méthode Scrum définit 3 rôles :

i. Le Product Owner :

- Qui porte la vision du produit à réaliser (joue le rôle du client).
- Pour notre projet il représente le client du Sogesi.

ii. Le Scrum Master :

- Garant de l'application de la méthodologie Scrum.
- Pour notre projet il représente le directeur de Sogesi.

iii. L'équipe de développement :

- Une équipe qui réalise le produit.
- Pour notre projet elle représente les membres de notre équipe.

III.3.2 Les sprints

Les sprints que nous avons réalisés pour notre projet sont représentés dans le tableau ci-dessous :

N° du Sprints	Durée	Sprints	Tâches
01	01/02/2019 - 24/02/2019	Installer et configurer les outils & <i>Gaps Analyses</i>	Installation et configuration Odoo
			Installation et configuration Android Studio 8
			Installation des émulateurs virtuels (tablette et mobile)
			Apprentissage sur développement mobile
			<i>Gaps Analyses</i>
02	25/02/2019 - 06/03/2019	Améliorer le module de la gestion de production	Conception sur les fonctionnalités à ajouté au module
			Ajouter des champs dans le module hérité
			Ajouter les droits d'accès pour les opérateurs
			Test
			Déploiement sur le serveur
03	07/03/2019 - 07/04/2019	Importer les données de Odoo vers l'application	Conception
			Formation sur connexion et importation des données de Odoo vers l'application avec XML-RPC
			Configuration du XML-RPC dans le projet.
			Connexion au serveur local Odoo via l'application
			Authentification des opérateurs
			Récupérer la liste des OT d'un opérateur
			Récupération des informations d'un OT
			Test
Déploiement sur le serveur			
04	08/04/2019 - 08/05/2019	Enregistrement des données de l'application vers Odoo	Conception
			Formation sur l'enregistrement des données de l'application vers Odoo avec XML-RPC
			Enregistrer les lignes de temps d'un opérateur
			Enregistrer les quantités de production dans stock
			Enregistrer les données d'un OT
			Sauvegarder des notes par écrit et vocal

			Créer une demande d'intervention
			Annuler la production
			Test
			Déploiement sur le serveur
05	09/05/2019 - 24/05/2019	Design et version tablette	Améliorer le design de l'application
			Réaliser la version tablette
			Test
			Déploiement sur le serveur

Tableau III.1 : Les sprints du projet

Chaque sprint regroupe un certain nombre des tâches à réaliser. Ces derniers sont spécifiées, suivies, et partagées entre les membres de notre équipe en utilisant **Trello** qui est un outil de gestion de projet en ligne.

Le tableau de bord de **Trello** est composé d'une série de liste et chaque liste est composée d'un ensemble de cartes que nous déplaçons d'une colonne à l'autre. Par défaut il existe trois colonnes « to do », « doing » et « done » et nous pouvons ajouter autant de colonnes selon notre besoin [22]. La figure ci-dessous montre une capture de l'interface Trello que nous avons utilisé durant notre projet.



Figure III.4 : Interface Trello.

Dans la partie qui suit, Les sections représentent les sprints et les sous sections décrivent les taches les plus importants du sprint correspondant.

III.3.2.1 Sprint 1 : Installer et configurer les outils & Gaps Analyses

Dans ce sprint nous avons installé tous les outils que nous avons cités dans les deux chapitres précédents et nous avons fait une formation sur ces outils et sur le développement des applications mobile. Ensuite, nous avons procédé au *Gaps Analyses*.

- **Gaps Analyses** : consiste à faire une étude de l'existant et une analyse des besoins du client et puis mesurer l'écart entre les deux.

a. Etude de l'existant « module de gestion de production actuel »:

Le système actuel de la gestion production est un module développé sous Odoo par Sogesi. Les fonctionnalités offertes par ce module sont représentées dans la figure ci-dessous :

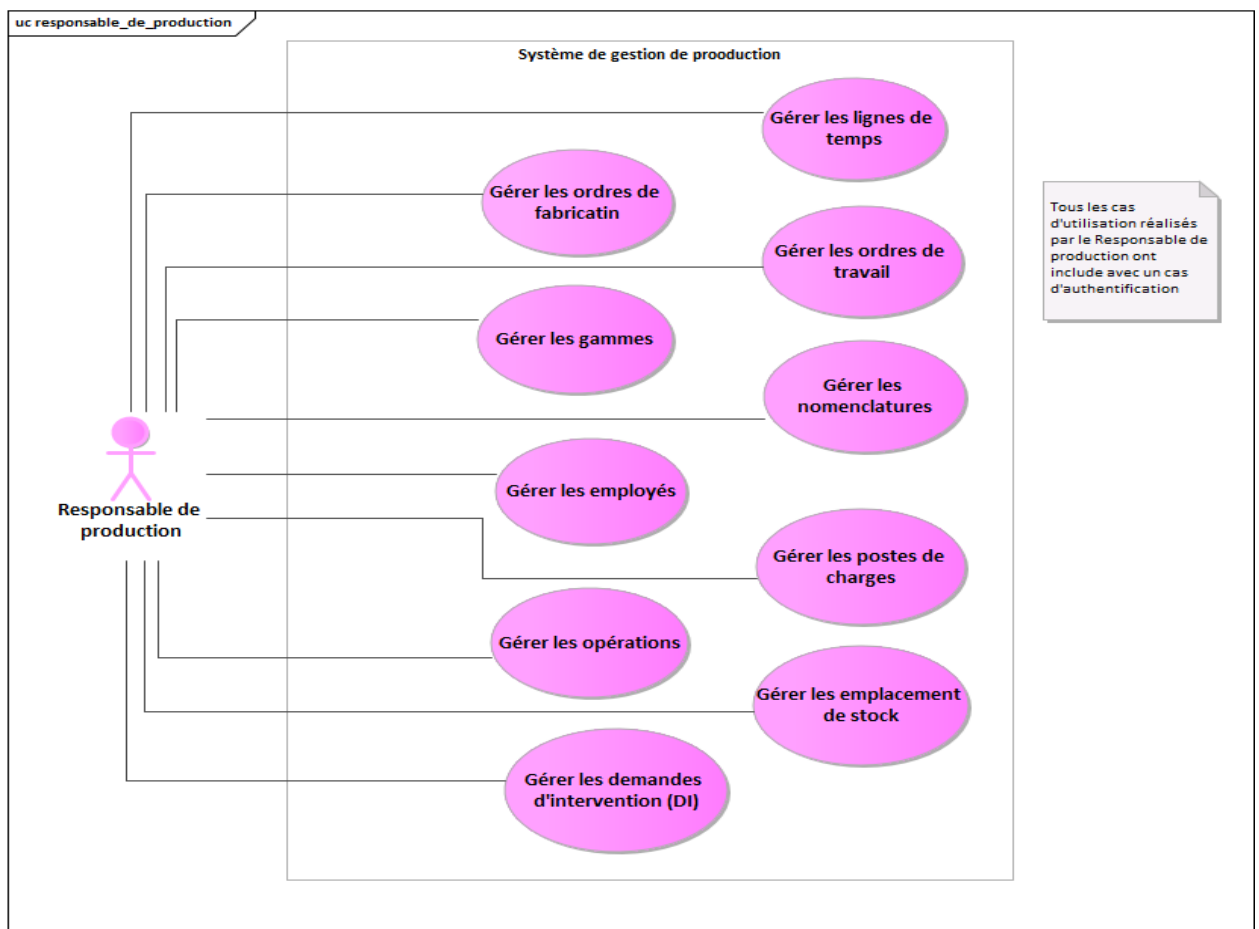


Figure III.5 : Cas d'utilisation pour le module de la gestion de production

Description des cas d'utilisations :

- ✓ **Gérer les Ordres de fabrication :** Suite à une demande de fabrication d'un produit, Le responsable de production va créer un ordre de fabrication pour ce produit à fabriquer, suivre son état et le mettre à jour quand il est dans l'état 'terminé'. Un **Ordres de fabrication** représente un **Ordre** autorisant un atelier de **fabriquer** des produits.
- ✓ **Gérer les nomenclatures :** Pour chaque produit à fabriquer, le responsable va créer ou modifier les nomenclatures. Une nomenclature définit les matières premières à consommer pour produire ce produit à fabriquer.
- ✓ **Gérer les postes de charges :** responsable de production définit les postes de charges nécessaires pour la fabrication d'un produit fini. Un poste de charge représente l'ensemble des ressources machines/humains avec ses informations (cout de chacun et les temps théoriques de production, de préparation, de nettoyage qui correspond à chacun d'entre eux).
- ✓ **Gérer les opérations :** Le responsable de production crée ou modifie les opérations nécessaires pour fabriquer un produit fini, tel que ces opérations représentent un ensemble de poste de charges.
- ✓ **Gérer les gammes :** Le responsable de production va gérer les gammes pour définir l'ensemble des opérations réalisées pour fabriquer un produit fini.
- ✓ **Gérer l'emplacement de stock :** Le responsable de production peut définir des emplacements dans le stock pour être considérés comme des emplacements de stockage des produits, les annuler, ou définir les quantités des produits dans chaque emplacement.
- ✓ **Gérer les lignes de temps :** Le responsable de production va déclarer ou modifier les lignes de temps associées à un ordre de travail. Une ligne de temps représente les temps de travail (de production, de préparation, de nettoyage) d'un opérateur pour un certain ordre de travail.
- ✓ **Gérer les Ordres de travail :** Le responsable de production peut créer, modifier, supprimer des ordres de travail associés à un ordre de fabrication et suivre ses états. Chaque ordre de travail doit avoir un poste de charge et les lignes de temps nécessaires.

- ✓ **Gérer les opérateurs :** Le responsable de production peut ajouter ou annuler des opérateurs au sein de l'établissement ou modifier leurs données.
- ✓ **Gérer les demandes d'interventions :** Le responsable de production peut créer des demandes d'interventions en cas où il y'a une panne d'une machines utilisée et mettre à jour cette demande une fois cette machine est rétablie.

Notons bien que dans ce système actuel :

- Il n'y'a pas une partie pour les opérateurs pour avoir un accès.
- Le responsable de production fait des tâches qui sont normalement réalisées automatiquement par les opérateurs eux même.

Notre rôle c'est d'inclure la partie d'opérateur dans le système en lui offrant une application mobile afin de lui donner l'accès à ce module et d'automatiser leurs tâches liées à la gestion de production pour gagner du temps et éviter les erreurs à répétitions de la saisie des informations.

Dans la tâche qui suit, nous avons analysé les besoins définis par le Product Owner pour spécifier exactement les tâches des opérateurs qui doivent être automatisées.

b. Analyse des besoins :

Nous avons analysé les besoins définis par Product Owner pour extraire les règles de gestion qui ont été validées conjointement avec le Scrum Master .

Ces règles de gestion sont représentées dans le tableau ci-dessous :

N°	Les règles de gestions
1.	L'opérateur accède à l'application en scannant leur QR.
2.	L'opérateur peut visualiser la liste des ordres de travail (OT) associés à lui, et avoir une interface ergonomique quand il accède à tel OT pour faciliter la navigation entre les fonctionnalités de l'application.
3.	L'opérateur peut suivre et déclarer les temps (avant production (préparation) / de production /après production (nettoyage)) des OT associés à lui et recevoir les consignes et le plan définis par le responsable de production pour cet OT
4.	l'opérateur peut faire une déclaration de production dans un temps réel

	(mentionner les quantités produites et les quantités rebutes) et peut savoir quelles sont les matières premières définies par le gestionnaire de production pour être utilisées pour fabriquer un certain nombre d'un produit fini.
5.	l'opérateur peut définir et suivre l'état des OT, comme il peut définir l'état de l'OF après la fin des OT associés à cet OF.
6.	En cas de défaillance d'un matériel, l'opérateur peut facilement créer une demande d'Intervention (DI) sans passer par le gérant, comme il peut imprimer un rapport qui contient un résumé de OT.
7.	Toutes ces actions faites par l'opérateur sont enregistrées dans l'ERP Odoo via les web services et comme ça le responsable de production s'assure de l'état actuel de la production sans consulter les opérateurs.

Tableau III.2 : Les règles de gestion

Dans les sprints qui suivent nous allons montrer la réalisation de ses règles, et à la fin de chaque sprint nous aurons un livrable qui sera testé pour être déployé sur le serveur.

III.3.2.2 Sprint 2 : Améliorer le module de la gestion de production

En premier lieu, nous avons fait une conception à l'aide du diagramme de cas d'utilisation montré dans la figure ci-dessous pour déterminer les fonctionnalités qui seront allouées aux opérateurs tout en se basant sur les règles de gestion validés par le Scrum Master.

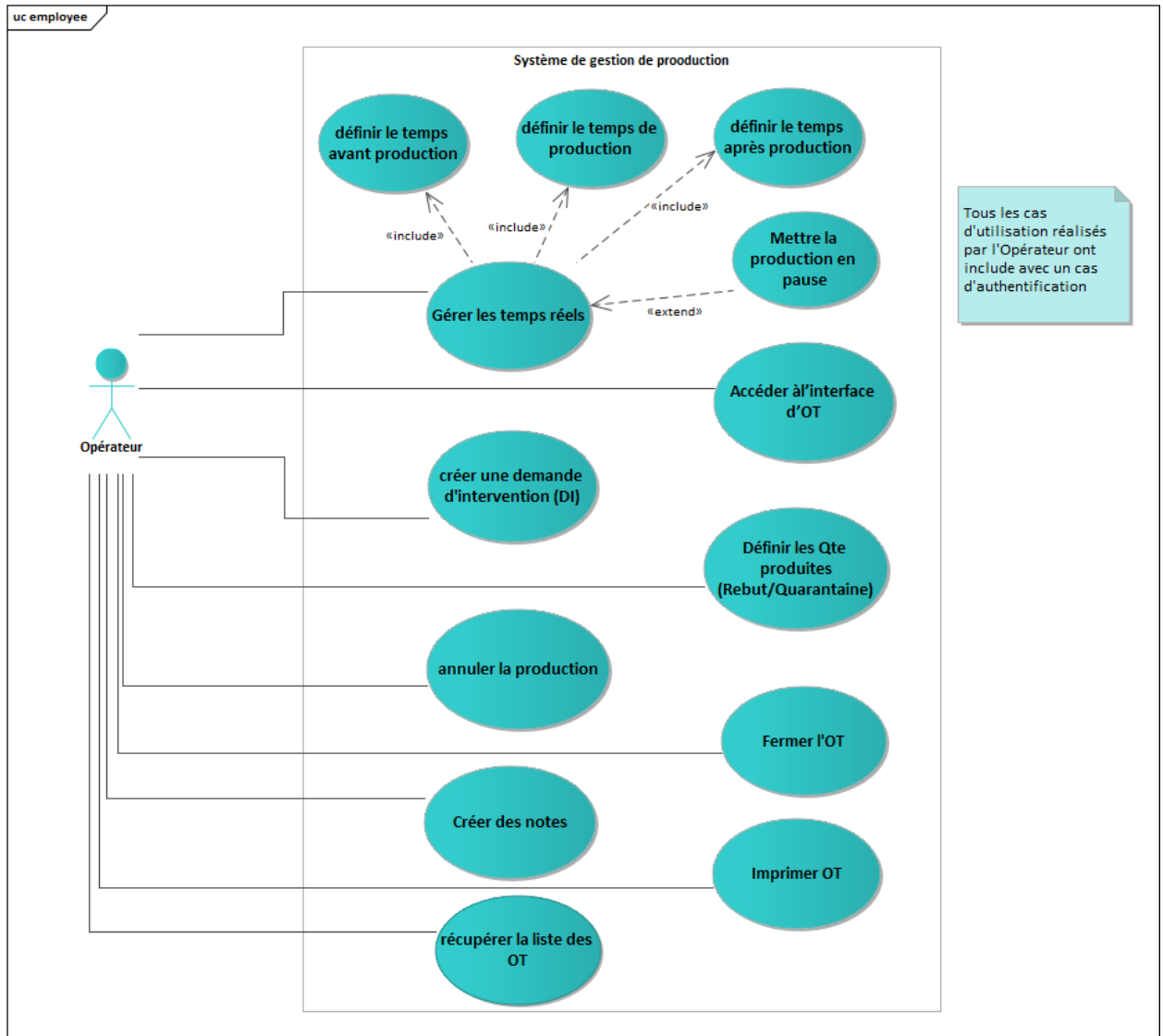


Figure III.6 : Diagramme de cas d'utilisation pour les Opérateurs

1/ Description des cas d'utilisations :

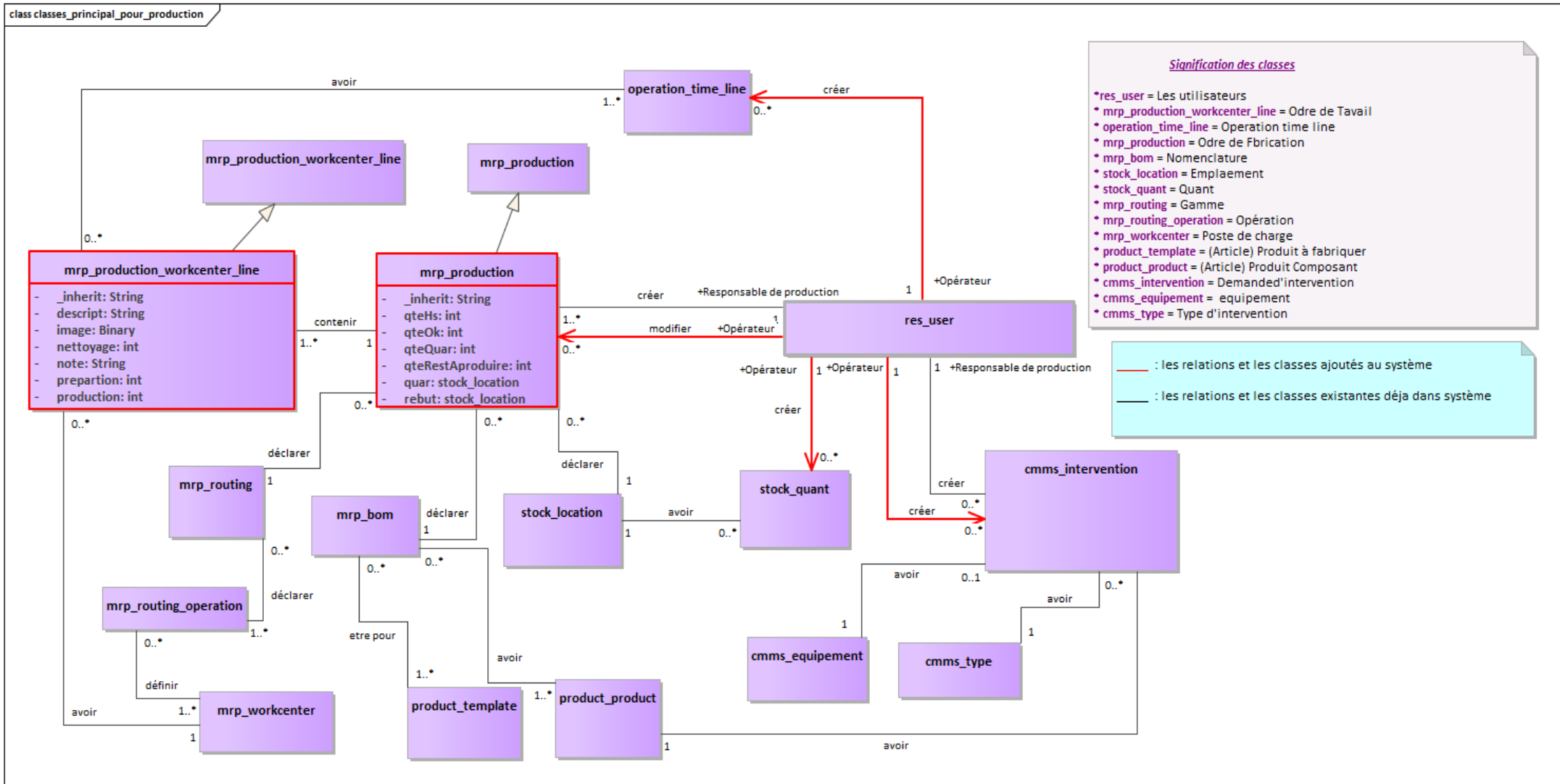
- **Accéder à l'interface d'OT** : Chaque opérateur a la possibilité d'accéder à l'interface d'un OT (Ordre de travail) associé à lui pour visionner les informations et faire les traitements nécessaires.
- **Fermer l'OT** : après que l'opérateur accède à l'interface d'OT, fait les traitements nécessaires, et termine son travail, il faut qu'il ferme OT pour valider son travail avant la sortie.
- **Annuler la production** : l'opérateur peut annuler la production à cause d'une certaine raison.

- **Gérer les temps réels** : l'opérateur peut définir et modifier les anomalies de leurs temps de travail au sein de l'établissement.
- **Définir le temps avant production** : l'opérateur peut définir et modifier les anomalies du temps qu'il a fait pour la préparation (avant production).
- **Définir le temps de production** : l'opérateur peut définir et modifier les anomalies du temps qu'il a fait pour la production.
- **Définir le temps après production** : l'opérateur peut définir et modifier les anomalies du temps qu'il a fait pour le nettoyage (après production).
- **Définir les quantités produites** : l'opérateur peut définir et modifier la quantité des produits fabriqués/rebuts pour être gardés dans le stock.
- **Créer une demande d'intervention(DI)** : l'opérateur peut créer une demande d'intervention en cas où il y'a une panne d'un équipement (une machine).
- **Créer des notes** : l'opérateur peut mentionner des notes par écrit ou vocale.
- **Imprimer OT** : l'opérateur peut imprimer un rapport qui contient tout les informations nécessaire pour un OT.
- **Récupérer la liste des OT** : l'opérateur peut récupérer la liste des OT(en état brouillon, en cours, ou en attente). associés à lui

Ensuite, à partir de diagramme de cas d'utilisation précédent nous avons fait une conception pour savoir quel sont les droits d'accès et les champs à ajouter au module de la gestion de production existant pour qu'il soit adaptable avec l'application.

Les champs ajoutés et les relations des opérateurs établies avec le nouveau système de la gestion de production sont représentés dans la figure ci-dessous.

Chapitre III : Conception et réalisation



III.7 : Classes participantes pour système de production

a. Ajouter des champs dans le module hérité

Comme nous l'avons dit auparavant, nous avons ajouté quelques champs dans les objets créés dans le module (objet d'Ordre de fabrication et d'Ordre de travail). Ces deux objets peuvent être créés par le responsable de production.

La figure ci-dessous montre la création de l'Ordre de Fabrication (qui contient les nouveaux champs ajoutés entourés en rouge) par le responsable de production :

Qte OK	Qte HS	Qte Quarantaine
0.00	0.00	0.00

Figure III.8 : Création d'Ordre de fabrication

La figure ci-dessous montre la création de l'Ordre de Travail (qui contient les nouveaux champs ajoutés entourés en rouge) par le responsable de production :

Temps de machine	Écart de temps	Information
------------------	----------------	-------------

Figure III.9 : Création d'Ordre de travail

b. Ajouter les droits d'accès pour les opérateurs :

Dans cette partie, nous avons ajouté d'autres droits d'accès au module pour donner aux opérateurs la possibilité d'enregistrer des quantités des produits dans un certain emplacement du stock, créer une demande d'intervention, déclarer leur temps de travail...

La figure suivante représente les différents droits d'accès ajoutés au module à l'aide du fichier CSV.

```
access_stock_warehouse_user,stock.warehouse.user,model_stock_warehouse,base.group_user,1,1,1,1
access_stock_location_user,stock.location.user,model_stock_location,base.group_user,1,1,1,1
access_stock_quant_all,stock.quant all users,model_stock_quant,base.group_user,1,1,1,1

access_cmms_equipment,cmms.equipment.user,model_cmms_equipment,base.group_user,1,1,1,1
access_cmms_intervention,intervention,model_cmms_intervention,base.group_user,1,1,1,1
access_cmms_type,typeintervention,model_cmms_type,base.group_user,1,1,1,1
```

Figure III.10 : Les droits d'accès ajoutés

Chaque ligne de la liste de control d'accès respecte la forme suivante :

Id,name,model_id :id,group_id :id,perm_read, perm_write, perm_create, perm_unlink

Prenons l'exemple de la ligne entourée en rouge dans la figure ci-dessus :

Id : access_cmms_type.	Perm_read:1
Model_id : typeintervention.	perm_write:1
name : Model_cmms_type.	perm_create:1
Group_id : base.group_user	perm_unlin:1

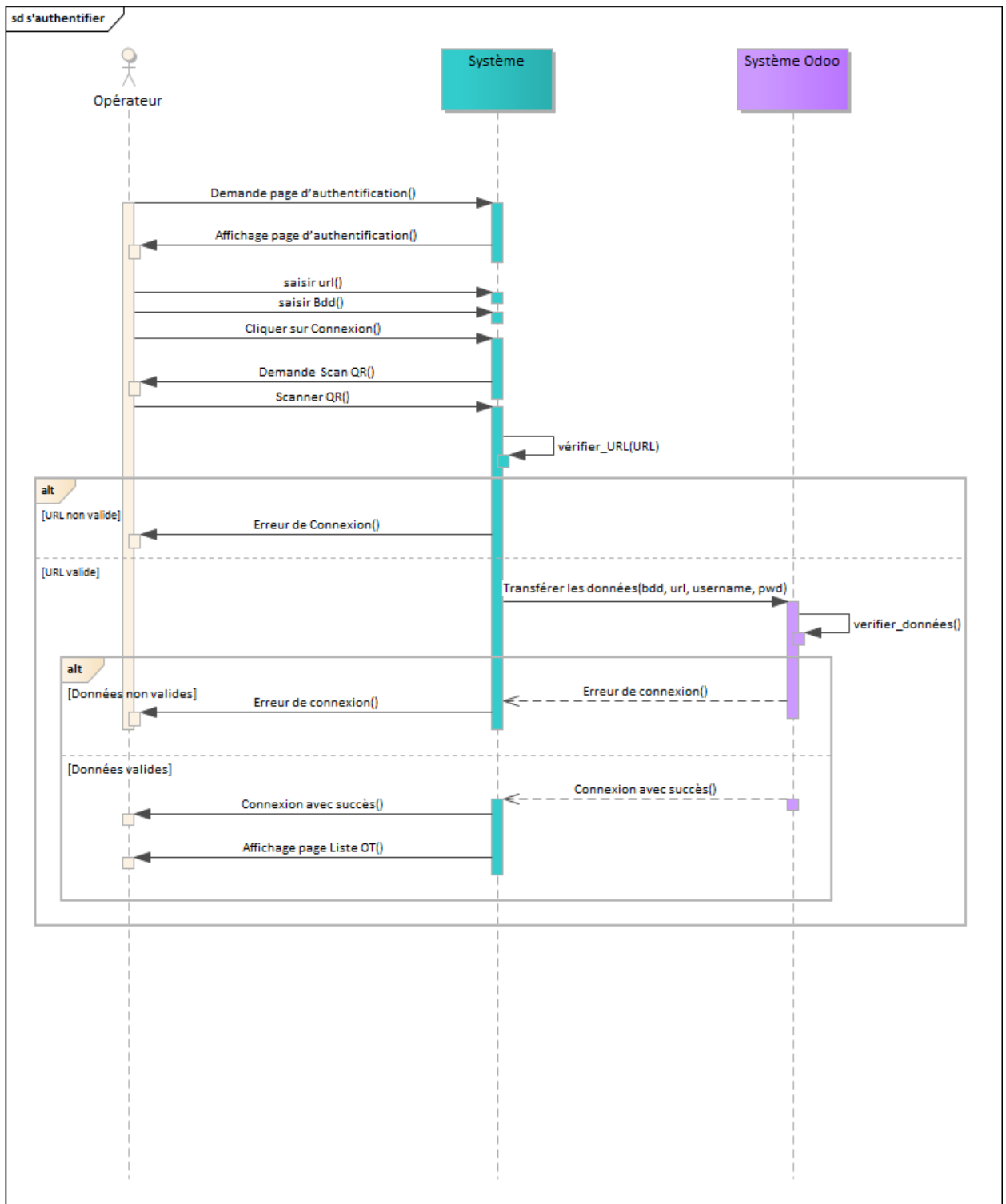
Cet exemple montre comment donner des droits d'accès aux opérateurs par rapport à l'objet cmms_type (type d'intervention).

Le livrable de ce sprint est le module amélioré.

III.3.2.3 Sprint 3: Importer les données de Odoo vers l'application

Avant d'entamer la partie de développement nous avons proposé une solution de connexion avec le serveur Odoo et une conception de l'importation des données depuis ce serveur.

La figure ci-dessous représente le processus pour que l'opérateur ait une connexion à l'application avec succès.



- Figure III.11 : Diagramme de séquence « s'authentifier »

Après cette tâche, nous avons fait une formation sur le fonctionnement des web services pour pouvoir intégrer et configurer le protocole XML-RPC dans notre projet, et pour pouvoir connecter et enregistrer des données de l'application vers Odoo.

a. configurer XML-RPC dans le projet :

Après l'amélioration du module, nous avons configuré XML-RPC avec le serveur d'application Odoo (soit locale ou en ligne) dans notre application mobile. Voici

une copie du code :

```
OdooUtility(String serverAddress, String path) {  
    try{  
        url = new URL(serverAddress + "/xmlrpc/2/" + path);  
        client = new XMLRPCClient(url);  
    } catch(Exception ex) {  
        Log.e("ODOO UTILITY: ", ex.getMessage());  
    }  
}
```

Figure III.12 : Fonction pour configurer XML-RPC

```
odoo = new OdooUtility(serverAddress, path: "common");
```

Figure III.13 : Appelle de la fonction de configuration

b. Connexion au serveur local via l'application et Authentification des opérateurs

Quand l'opérateur accède à l'application la première page affichée lui demande de saisir l'URL du serveur (locale ou en ligne) et le nom de la BDD trouvé dans ce serveur puis clique sur CONNEXION.

Avant d'établir la connexion il faut que l'opérateur scanne son code QR, tel que chaque opérateur a son propre code QR généré en ligne via (<https://qrcode.tec-it.com/fr>).



Figure III.14 : Connexion au serveur Odoo

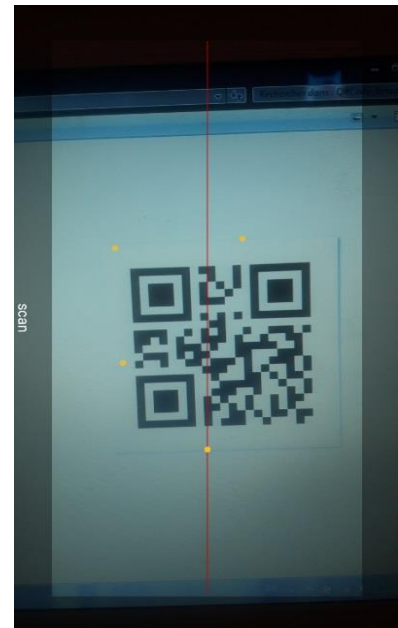


Figure III.15 : Scan le code QR de l'opérateur.

c. Récupérer la liste des OT d'un opérateurs :

Si cet opérateur est déclaré au sein de l'établissement et son code QR scanné est valide, il sera redirigé vers la page qui affiche la liste des OT non terminés qui lui sont associés.

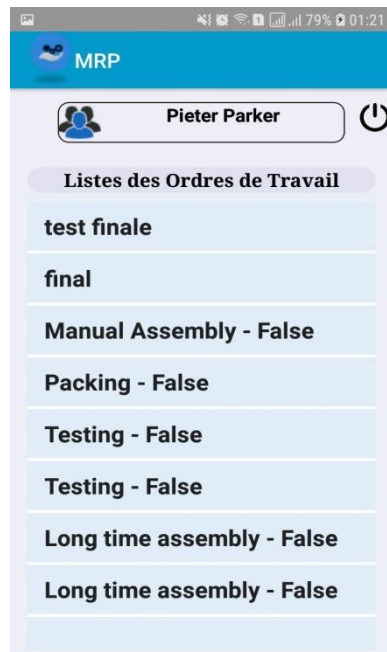


Figure III.16 : Page des listes OT d'un opérateur

Si l'opérateur sélectionne un OT il sera redirigé vers la page qui affiche Les temps théoriques désignés, les matières premières à consommer pour fabriquer le produit fini (les matières première qui demandent une traçabilité sont affichées avec une couleur rouge) et les temps réels qu'ils soient mis à jour au moment du travail de l'opérateur au sein de l'établissement.



Figure III.17 : Interface principale d'un OT

d. Récupération des informations d'un OT :

Dans l'interface d'OT nous avons importé tous ces informations nécessaires.

La figure à gauche est un menu qui contient les informations d'OT et le nom d'opérateur et donner la possibilité à l'opérateur de consulter les consignes ou le plan désigné pour cet OT en cliquant sur Consigne et Plan.

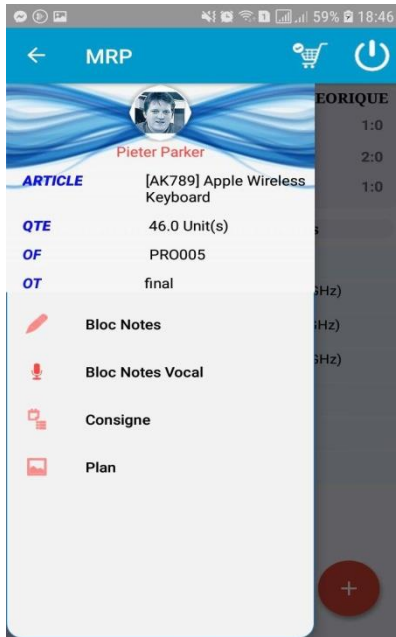


Figure III.18 : Menu



Figure III.19 : Consigne d'un OT

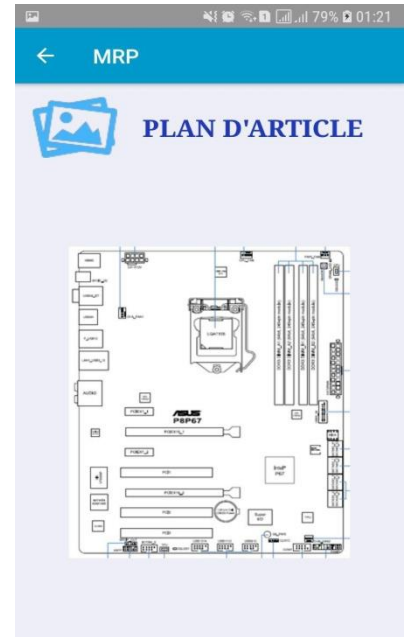


Figure III.20 : Plan d'un OT

Le livrable de ce sprint est la 1^{ère} partie fonctionnelle de notre application qui permet de connecter au serveur Odoo et importer les données de Odoo vers l'application.

III.3.2.4 Sprint 4: Enregistrement des données de l'application vers Odoo

Avant d'entamer la partie de développement nous avons fait une conception sur l'enregistrement des données de l'application vers Odoo avec XML-RPC.

i. gérer les quantités :

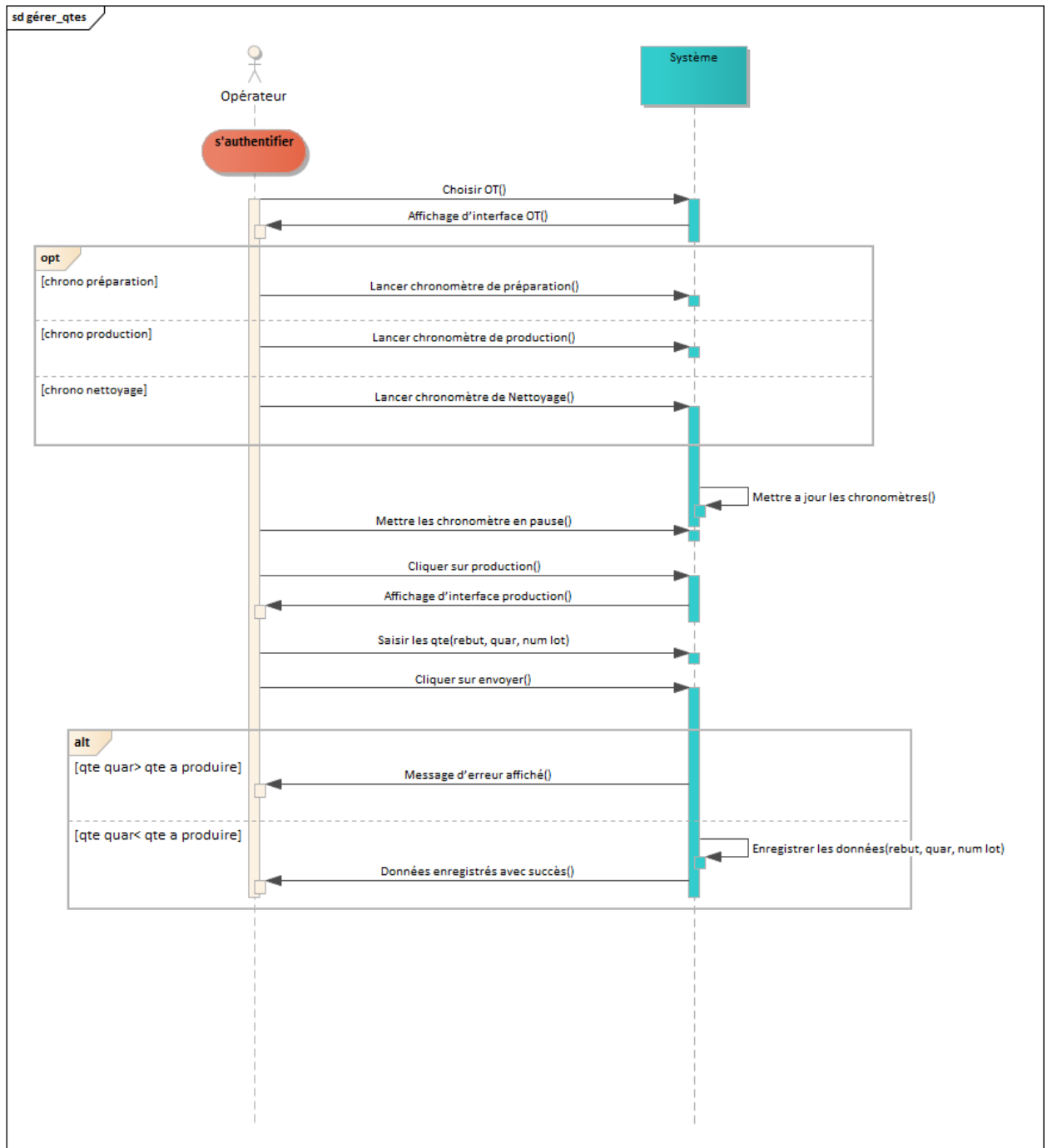


Figure III.21 : Diagramme de séquence «gérer les quantités »

ii. gérer les temps réels :

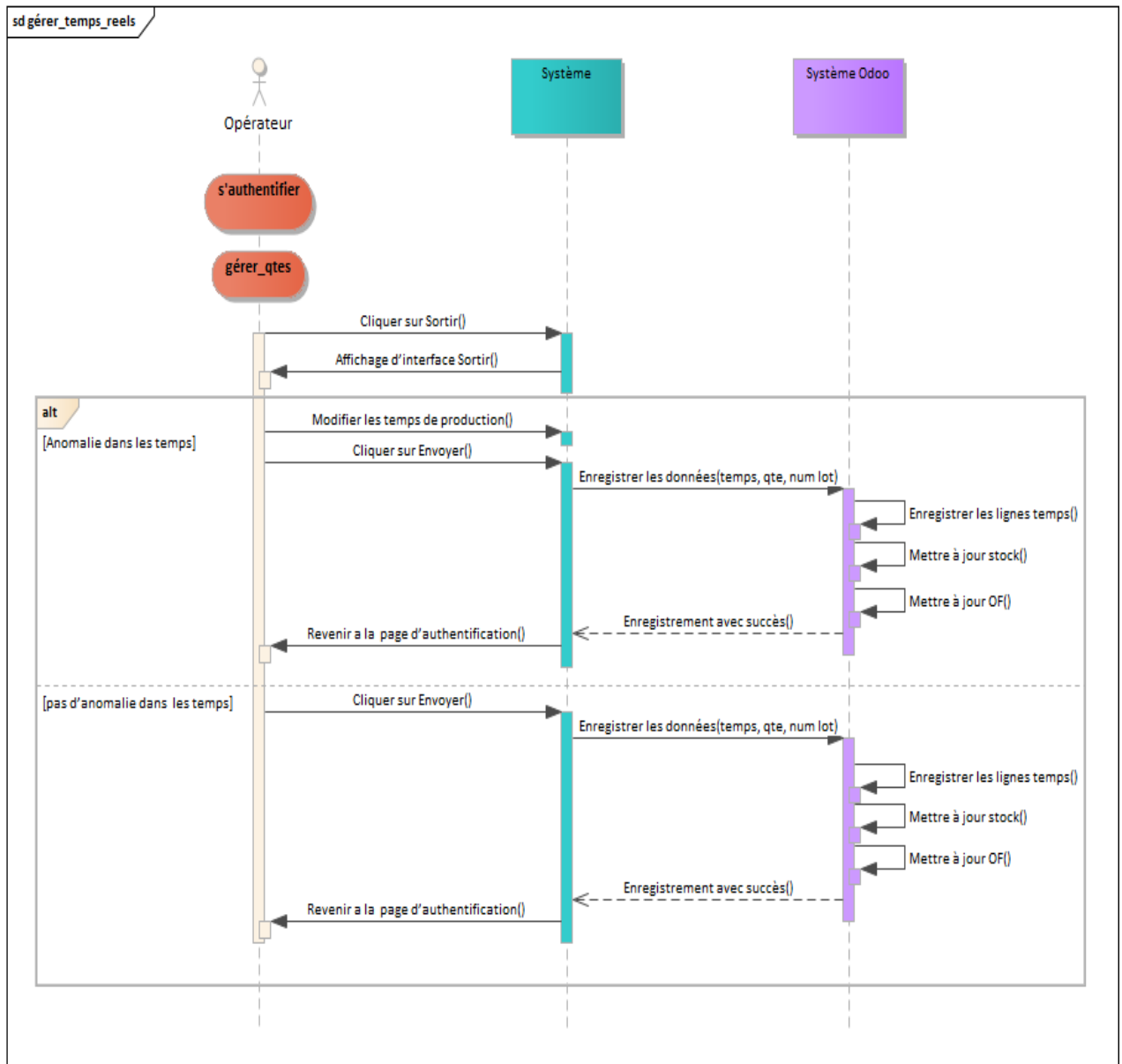


Figure III.22 : Diagramme de séquence «gérer temps réels »

iii. gérer les demandes d'interventions :

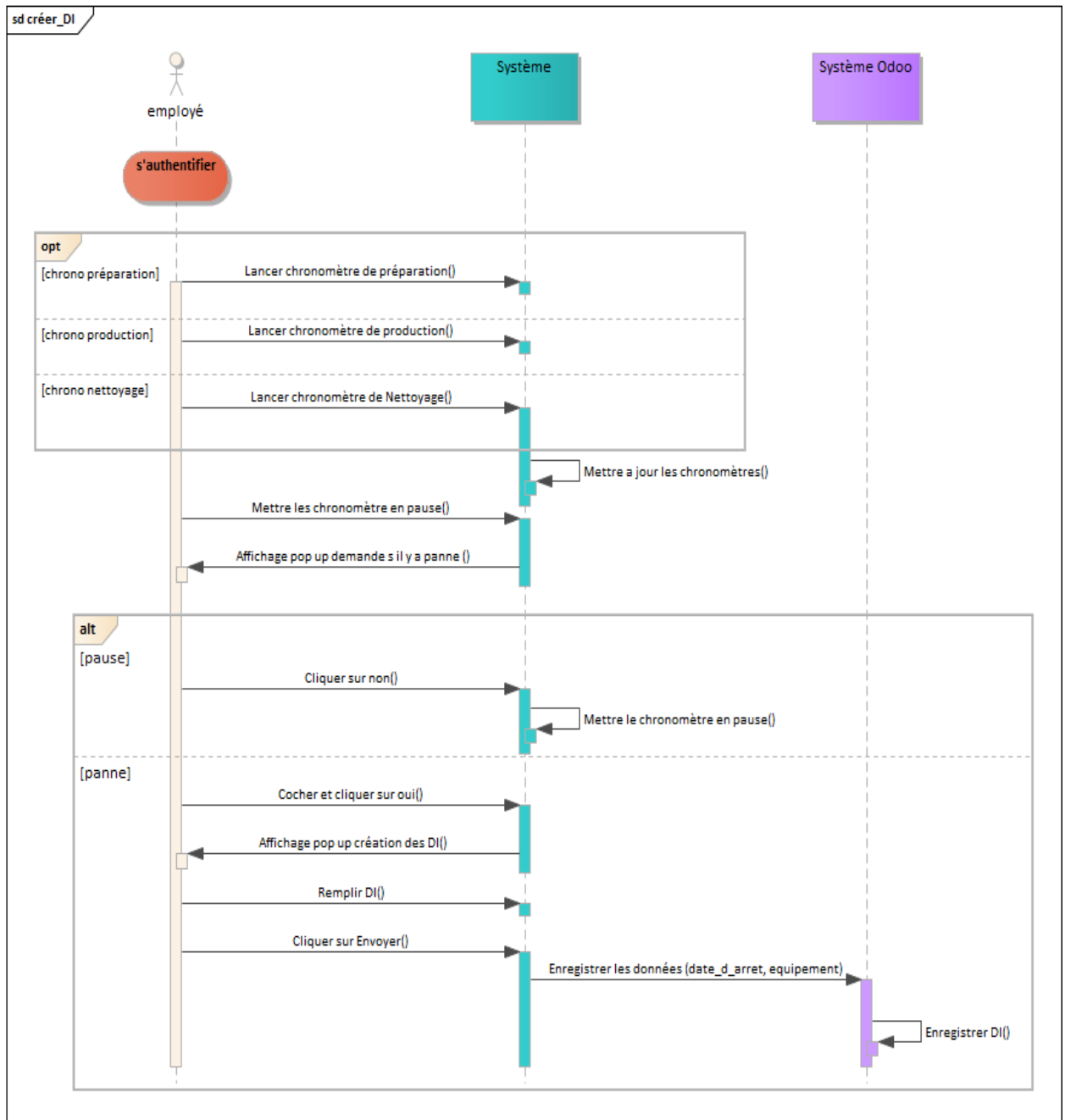


Figure III.23 : Diagramme de séquence « gérer les demandes d'interventions»

Après la tâche de la conception, nous avons fait une formation sur le fonctionnement des web services pour pouvoir intégrer et configurer le protocole XML-RPC dans notre projet, et pour pouvoir connecter et enregistrer des données de l'application vers Odoo.

a. Sauvegarder des notes par écrit ou vocale :

A partir du menu, l'opérateur peut aussi écrire une note et la sauvegarder sous Odoo.

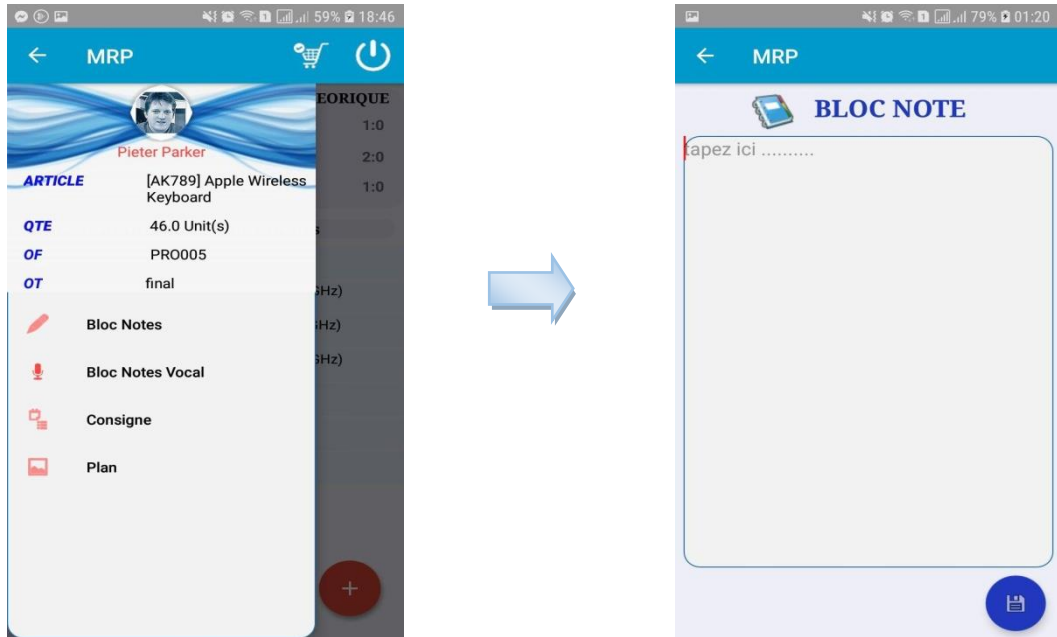


Figure III.24 : Créer une note et la sauvegarder

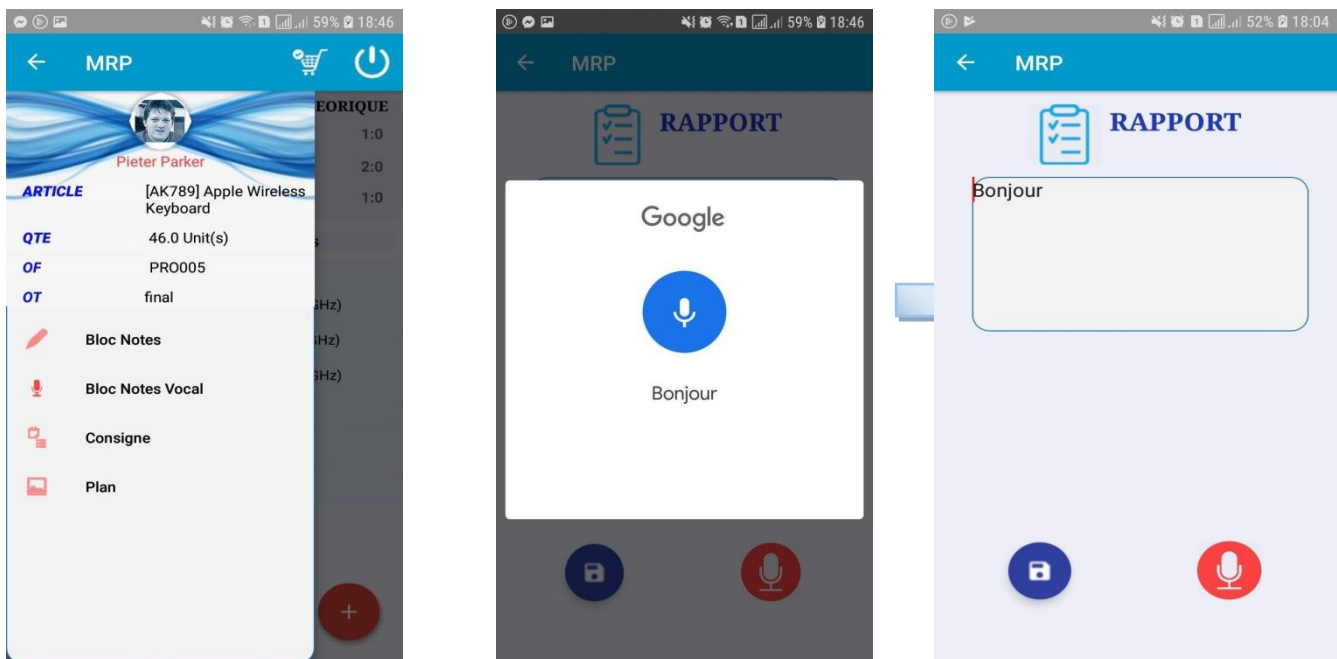


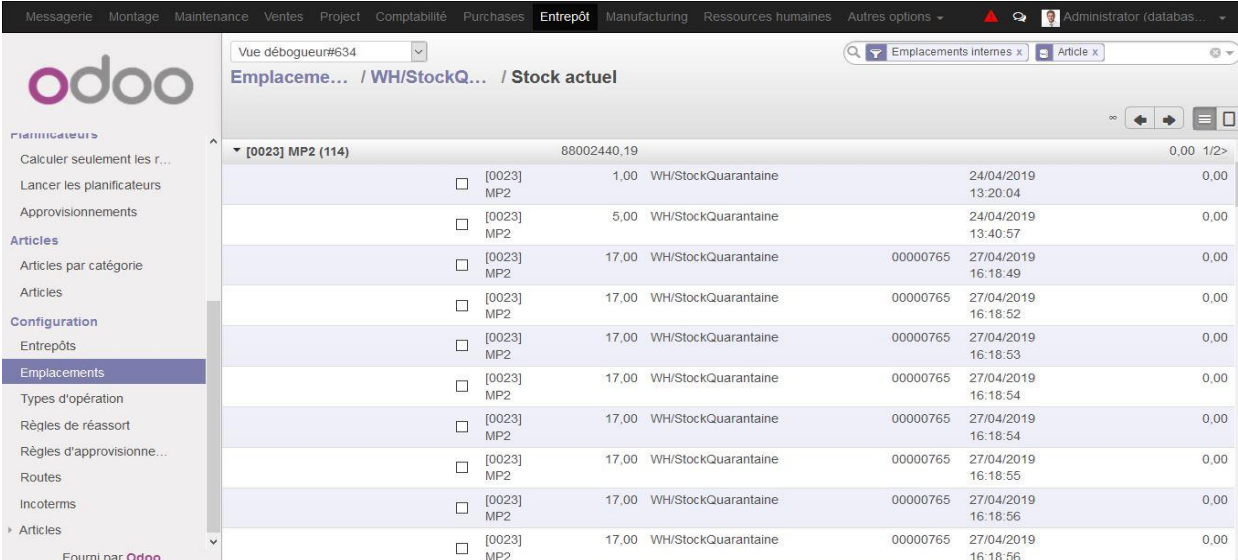
Figure III.25 : Créer une note par vocal et la sauvegarder

Dans les figures ci-dessus, nous avons montré qu'à partir du menu, l'opérateur peut aussi écrire une note manuellement comme c'est montré dans la figure 24 ou par enregistrement vocal comme illustré dans la figure 25, et quand il clique sur save un message sera affiché pour lui indiquer que la note est sauvegardé sous Odoo.

b. Enregistrer les quantités de production dans le stock :

Avant l'intégration de notre application à Odoo, le responsable de production consulte un certain nombre des opérateurs pour savoir les quantités de production fabriquées afin de les enregistrer sous Odoo manuellement.

La figure ci-dessous montre que la quantité de production déclarée par le responsable de production sera remontée au stock.



[0023] MP2 (114)		88002440.19		0.00 1/2>	
<input type="checkbox"/>	[0023] MP2	1,00	WH/StockQuarantaine	24/04/2019 13:20:04	0,00
<input type="checkbox"/>	[0023] MP2	5,00	WH/StockQuarantaine	24/04/2019 13:40:57	0,00
<input type="checkbox"/>	[0023] MP2	17,00	WH/StockQuarantaine	00000765 27/04/2019 16:18:49	0,00
<input type="checkbox"/>	[0023] MP2	17,00	WH/StockQuarantaine	00000765 27/04/2019 16:18:52	0,00
<input type="checkbox"/>	[0023] MP2	17,00	WH/StockQuarantaine	00000765 27/04/2019 16:18:53	0,00
<input type="checkbox"/>	[0023] MP2	17,00	WH/StockQuarantaine	00000765 27/04/2019 16:18:54	0,00
<input type="checkbox"/>	[0023] MP2	17,00	WH/StockQuarantaine	00000765 27/04/2019 16:18:54	0,00
<input type="checkbox"/>	[0023] MP2	17,00	WH/StockQuarantaine	00000765 27/04/2019 16:18:55	0,00
<input type="checkbox"/>	[0023] MP2	17,00	WH/StockQuarantaine	00000765 27/04/2019 16:18:56	0,00
<input type="checkbox"/>	[0023] MP2	17,00	WH/StockQuarantaine	00000765 27/04/2019 16:18:56	0,00

Figure III.26 : Les quantités déclarées dans un emplacement du stock

Mais en utilisant notre application, la déclaration des quantités de production sera faite directement par l'opérateur lui-même tel que après un certain temps de travail (préparation, production, ou nettoyage), si l'opérateur veut quitter l'interface de l'OT, il met les chronomètres en pause et il va y avoir le bouton de production cliquable. quand l'opérateur clique sur production une pop up sera affichée pour lui permettre de mentionner le numéro du lot du produit fini et déclarer la quantité des produits fabriqués qui seront mis dans un emplacement pour quarantaine et les quantités des rebuts pour être stockés dans l'emplacement des rebuts.

Si cet OT est terminé il va cocher même la case à cocher 'fermer OT' pour qu'il ne soit pas importé encore une fois pour être traité.

Si l'opérateur veut enregistrer les informations de cet OT sous forme d'un pdf il suffit de cocher même la case à cocher 'imprimer'.

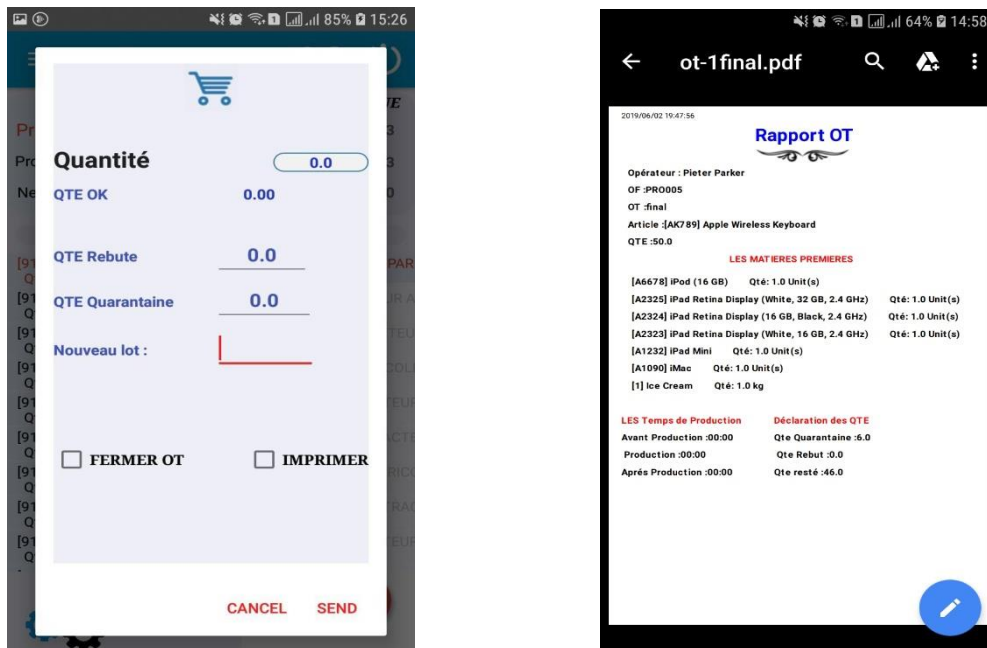


Figure III.27 : Enregistrer les quantités de production

c. Enregistrer les lignes de temps d'un opérateur :

Avant d'intégrer notre application à Odoo, l'opérateur travaille sur son poste de travail et quand il termine, le responsable de production va le consulter pour savoir les temps pris d'une opération faite concernant la production afin de les enregistrés dans la base odoo manuellement.

La figure ci-dessous montre que le responsable de production a la possibilité de déclarer les temps d'une opération faite concernant la production.

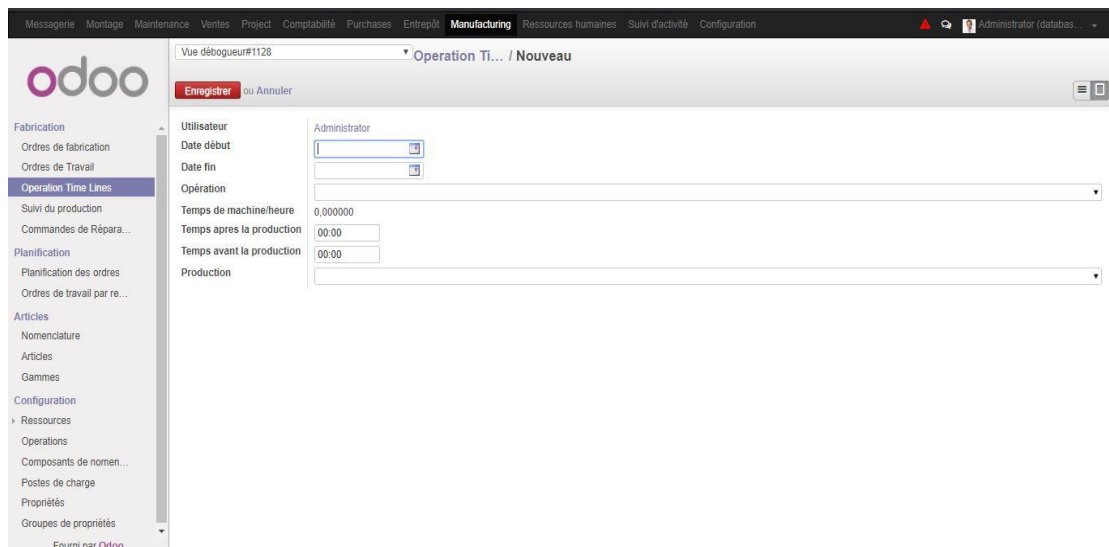


Figure III.28 : Créer les temps d'une opération

Mais en utilisant notre application, lorsque l'opérateur commence son travail au sein de l'établissement il clique sur bouton floating associé au type de travail qu'il est entrain de faire (préparation, production, ou nettoyage), et par la suite le chronomètre associé sera lancé (voir la figure 29 montrée ci-dessous). Quand il termine son travail et il veut sortir (Déconnecter) il n'aura pas la possibilité de sortir car ce bouton est non cliquable. Donc il est obligé de passer par le bouton production pour déclarer les quantités de production qu'il a fait, et lorsqu'il valide ça, il aura le bouton de déconnexion cliquable. A ce moment là, lorsqu'il clique sur déconnexion, il va y avoir une pop up qui affiche le résumé des temps qu'il a passé pour chaque action (préparation, production, et nettoyage), et s'il y'a des anomalies il peut les modifier. Quand il clique sur envoyer, les temps du pop-up, les quantités de production qui ont été déclarées, et l'état de OT seront remonté vers Odoo via les web services, et l'opérateur sera redirigé vers la page de connexion, et un message s'affichera pour assurer à l'opérateur que les données sont bien enregistrés (voir la figure 30 montrée ci-dessous).

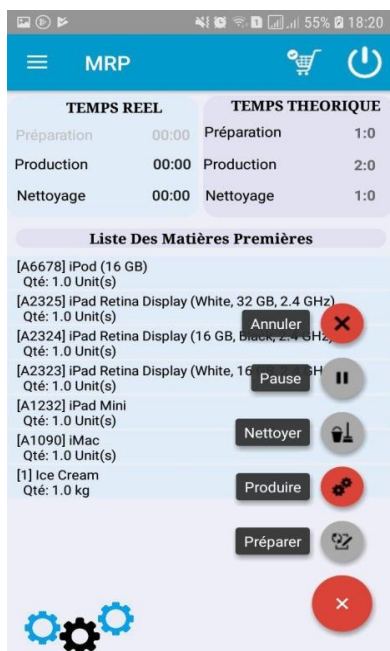


Figure III.29 : Menu floating



Figure III.30 : Sauvegarder les temps et sortir

d. Créer une demande d'intervention :

Avant d'intégrer notre application à Odoo, l'opérateur n'avait plus le droit de créer la demande d'intervention s'il y'a une panne d'un matériel, mais il va déclarer cette panne pour être créer par le responsable de production manuellement sous Odoo.

La figure ci-dessous montre que le responsable de production a la possibilité de créer une demande d'intervention :

The screenshot shows the Odoo web interface for creating a maintenance request. The form is titled 'Créer demande d'intervention' and is located in the 'Maintenance' module. The form includes several sections: 'Référence intervention' with fields for 'Type maintenance', 'Émetteur', 'Domaine d'intervention', and 'Date souhaitée'; 'Type d'intervention' with fields for 'Equipement interne', 'Equipement', 'Intervenant', and 'Intervenant externe'; 'Arrêt de la machine?' with a checkbox and fields for 'Date d'arrêt' and 'Date de redémarrage'; 'Temps d'arrêt' with a field for '0,00 Heure'; 'Information générale' with fields for 'Destinataire', 'Priorité', 'Coût', 'Date d'intervention', 'Date de fin d'intervention', and 'Durée'; and an 'Observations' text area. There are also buttons for 'Enregistrer', 'Annuler', 'Accepter', 'Refuser', 'Brouillon', 'Acceptée', 'Convertie en bon de travail', 'Clôturée', and 'Refusée'.

Figure III.31 : Créer demande d'intervention

Mais en utilisant notre application, le fait que l'opérateur fait une pause une pop up sera affiché pour lui demander d'indiquer si cette pause c'est à cause d'une panne ou pas. Quand cette pause est suite à une panne, il coche la case à cocher et clique sur oui, il sera redirigé vers la création d'une demande d'intervention dans laquelle, nous introduisons les détails suivants : la date d'arrêt de l'équipement (machine), l'équipement est interne ou loué, le nom de l'équipement concerné, le nom de l'opérateur comme un émetteur, et enfin la date de création de l'intervention. Une fois, la demande est créé, un message sera affiché pour montrer que la demande d'intervention est créée.

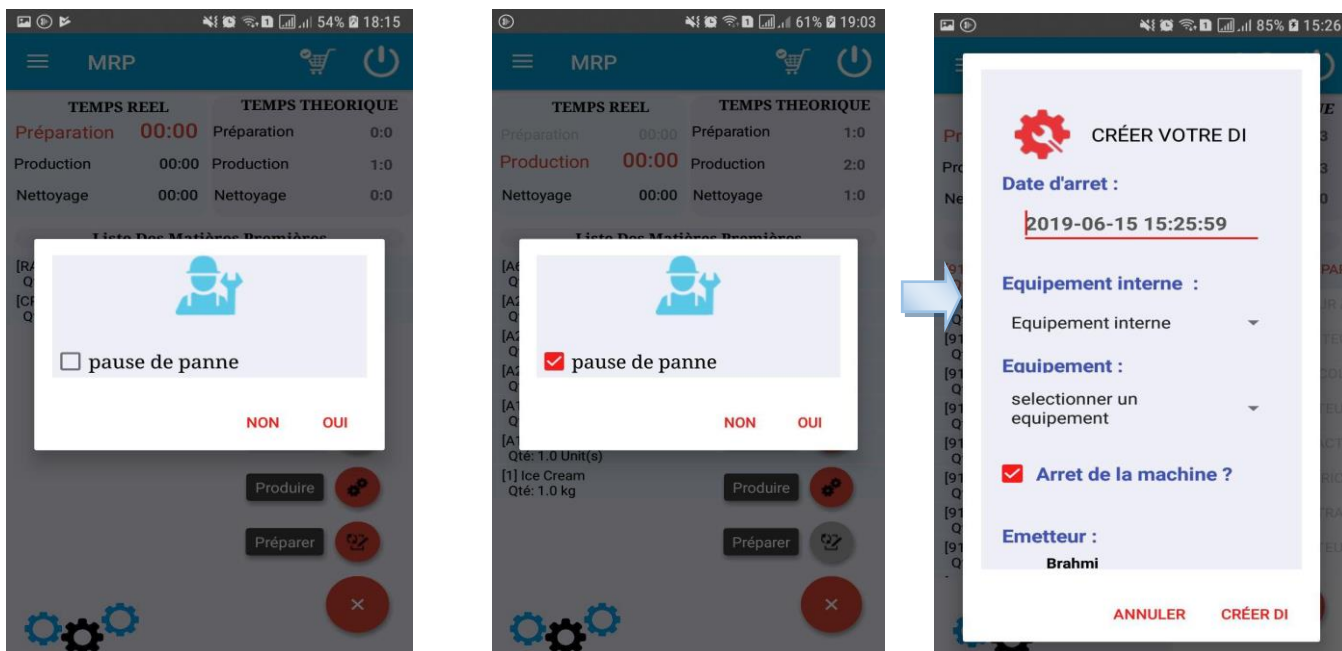


Figure III.32 : Création d'une demande d'intervention

e. Annuler la production :

Comme c'est montré dans la figure ci-dessous, si l'opérateur veut annuler la production à cause d'une certaine raison, il va cliquer sur 'Annuler' de menu floating. Par la suite une pop up sera affiché pour lui demander s'il veut vraiment annuler cette production. S'il clique sur 'oui' cette production sera annulée et l'information sera remontée vers le serveur Odoo. La figure ci-dessus montre.

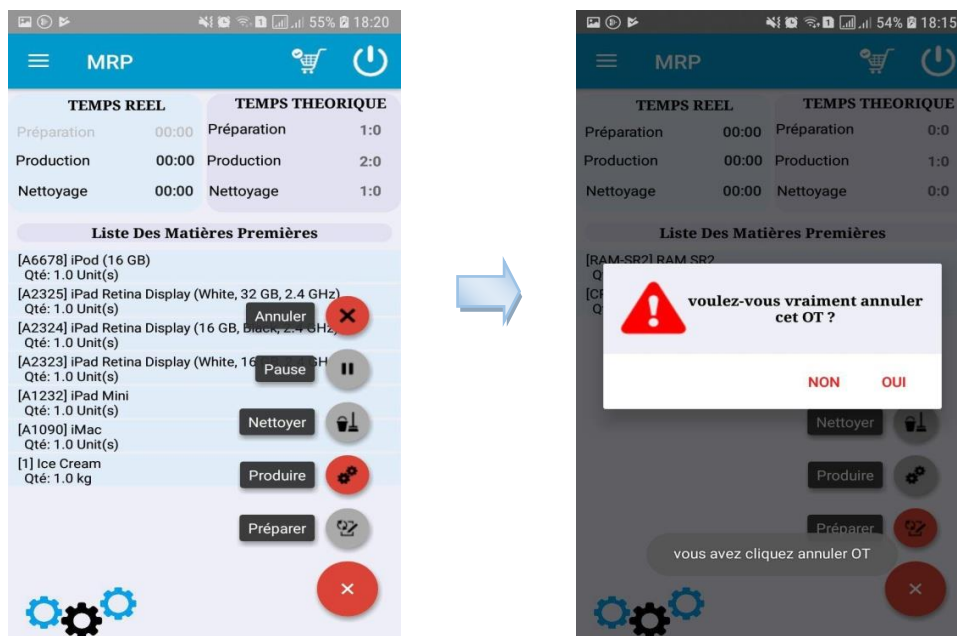


Figure III.33 : Annuler la production

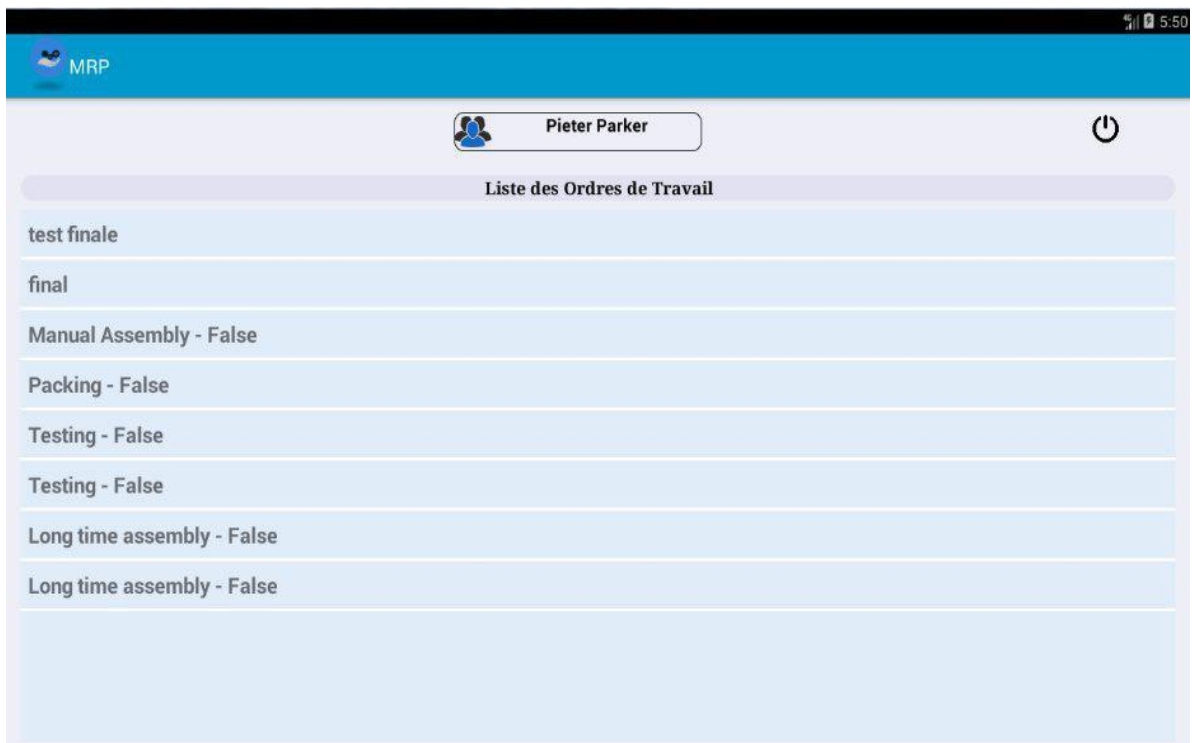
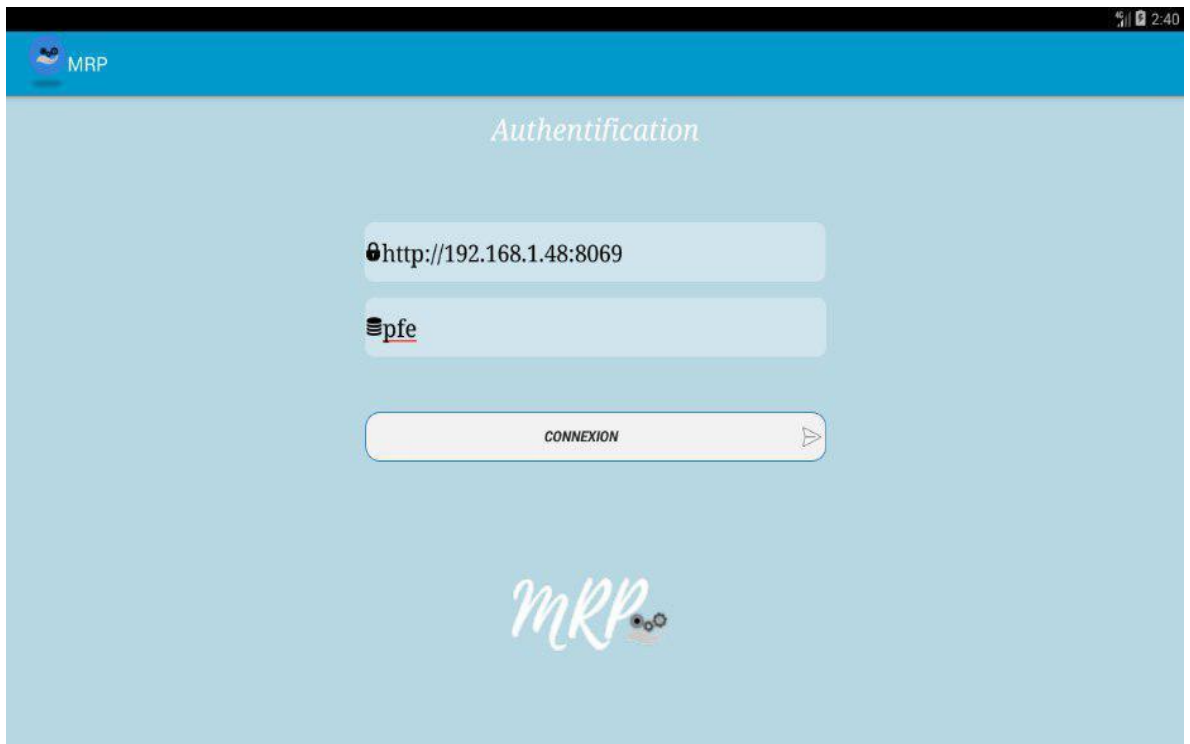
Le livrable de ce sprint est la nouvelle version de notre application qui permet de faire en plus l'enregistrement des données de l'application vers Odoo.

III 3.2.5 Sprint 5 : Ajuster le design et réaliser la version tablette

Dans cette partie, nous avons amélioré le design de l'application. Et par la suite nous avons réalisé la version tablette de notre application pour qu'elle soit utilisable dans l'atelier de l'établissement client.

a. Réalisation version tablette :

Dans les figures ci-dessous nous allons montrer la version tablette de notre application:



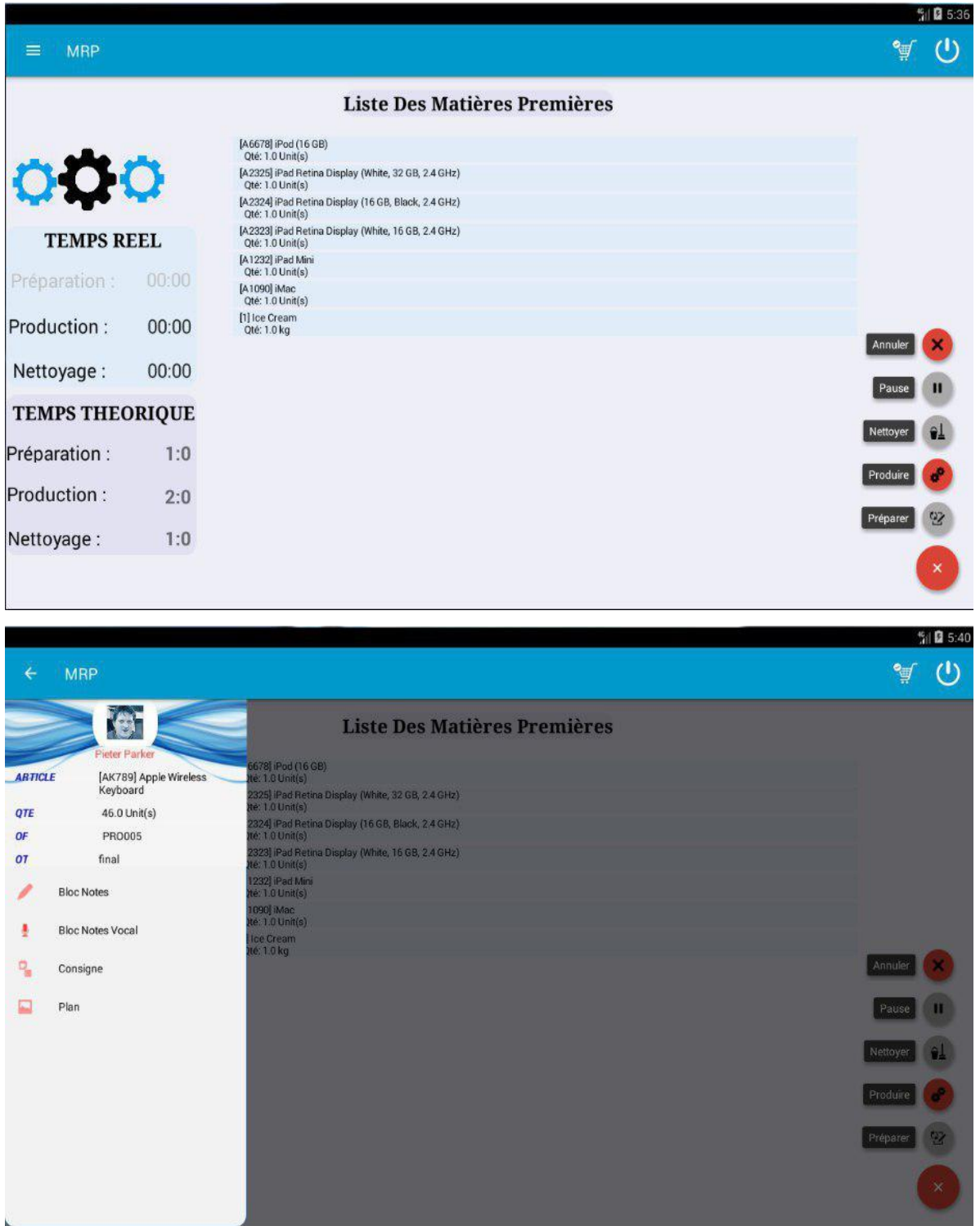


Figure III.34: des captures d'écran pour la version tablette d'application

Le livrable de ce sprint est la nouvelle version de notre application qui a un design amélioré, avec la possibilité d'être utilisé sur une tablette.

III.4 Conclusion

Dans ce chapitre nous avons présenté les étapes réalisées pour répondre aux besoins de la société, en commençant par une définition de la méthode Scrum qui nous a aidé pour gérer notre projet. Ensuite, nous avons expliqué les tâches réalisés pour le déploiement de notre solution. Enfin, nous avons donné une présentation de toutes les fonctionnalités développées.

Conclusion Générale

a. Conclusion

Notre projet de fin d'étude est réalisé au sein de la société SOGESI.

Durant tout notre stage au sein de cette société, nous avons utilisé la méthode Scrum. Nous avons commencé notre travail par l'analyse des besoins du client et l'extraction des exigences fonctionnelles.

Ensuite, nous avons fait une recherche sur les technologies et les solutions existantes possibles, Ensuite, nous avons procédé à une analyse du module existant de production et les autres modules qui ont une relation avec ce dernier afin de pouvoir le modifier ce le rendre plus approprié.

Après l'analyse, la conception, et l'adaptation de module de la gestion de production, nous avons pu réaliser notre application (basée sur Odoo) qui permet aux opérateurs de gérer leur production de manière simple et efficace directement depuis leur mobile/tablette..

Enfin, nous pensons que ce projet nous a permis d'avoir une expérience dans le domaine professionnel, de découvrir le monde des ERP et des applications mobiles/tablettes, comme il nous a permis également de mettre en pratique les connaissances que nous avons acquises durant notre cursus universitaire.

b. Perspectives

- A court terme, nous voudrions ajouter la notification dans l'application. Par exemple, quand le responsable de production crée un nouvel ordre de travail associé à un opérateur. Ce dernier reçoit une notification. Et réciproquement, lorsque l'opérateur termine un ordre de travail le responsable de production reçoit une notification le module de la gestion de production.

- Intégrer le chat dans l'application pour faciliter la communication.

Conclusion générale

- Interfacier notre application avec l'application de la gestion de stock et l'application de la gestion de maintenance.
- A moyen terme, nous souhaitons également ajouter à l'application une partie pour le responsable de production pour qu'il puisse réaliser toutes ses tâches à distance en utilisant cette application intégrée à Odoo.
- A long terme, nous souhaitons ajouter le BI(Business Intelligence) à l'application pour que les employés peuvent voir toutes les statistiques concernant la production.

Références

- [1] <https://www.celge.fr/actualite/quels-avantages-utilisation-logiciel-gpao>, consulté le 13/04/2019.
- [2] <https://www.celge.fr/article-conseil/logiciel-gpao-fonctionnalites>, consulté le 13/04/2019.
- [3] Gérard Baglin, et Samir Lamouri, Maîtriser les progiciels ER, 19 mars 2015.
- [4] Bouchaib Rivami, PGI,ER : Master CCA, 10/17/2016.
- [5] <http://khamarerp.eclablog.com/1-definition-d-un-erp-p1151374>, consulté le 20/04/2019.
- [6] <https://erp.ooreka.fr/comprendre/erp-propretaire>, consulté le 11/06/2019.
- [7] Gerardus Blokdyk, Odoo a Complete Guide - 2019 Edition, 20 décembre 2018.
- [8] HORIYASOFT EDITION, Ramzi msiyah, et al.Comment Réaliser un module Odoo de A à Z, 1 septembre 2018
- [9] Sébastien LARDIÈRE ,PostgreSQL - Administration et exploitation de vos bases de données (4e édition), 13 décembre 2017.
- [10] Thierry BOULANGER, XML par la pratique - Bases indispensables, Concepts et cas pratiques (3ième édition), 8 avril 2015
- [11] Gérard Swinnen , Apprendre à programmer avec Python 3 Broché, 2 février 2012.
- [12] BENMANSOUR Hadjer Selsabil, et BABA AHMED Djazila, développement d'une solution ERP pour la gestion de maintenance assistée par ordinateur, le 21 Juin 2016
- [13] https://fr.wikipedia.org/wiki/Service_web, consulté le 04/05/2019
- [14] <http://www-igm.univ-mlv.fr/~dr/XPOSE2004/woollams/definition.html>, consulté le 04/05/2019

Références

- [15] http://biodev.extra.cea.fr/docs/epims/doku.php?id=wiki:epims4_0:developer:ep-webservices, consulté le 04/05/2019.
- [16] Lambert M Surhone,), Miriam T Timpledon, et Susan F Marseken , XML - RPC (Anglais) Broché, 25 juillet 2010
- [17] Sylvain HEBUTERNE, Développez une application Android - Programmation en Java sous Android Studio (2e édition), 11 octobre 2017
- [18] Neil Smyth , Android Studio 3.0 Development Essentials - Android 8 Edition (Anglais), 25 septembre 2017
- [19] Delannoy, Programmer en Java : Couvre les nouveautés de Java 8, streams, Claude expressions, 12 juin 2014
- [20] Thierry BOULANGER, XML par la pratique - Bases indispensables, Concepts et cas pratiques (3ième édition), 8 avril 2015
- [21] Bassem El Haddad, et Julien Oger, Scrum, de la théorie à la pratique: Initiation. Perfectionnement. Agilité.avec un mémento de 14 pages à emporter partout !
- [22] Marc Maxmeister ,Trello for Project Management (English Edition), 28 juillet 2014.
- [23] Marc Maxmeister ,Trello for Project Management (English Edition), 28 juillet 2014.

Résumé

Ce travail s'inscrit dans le cadre de notre projet de fin d'étude de Master en informatique option Génie Logiciel (GL), qui s'est déroulé au sein de l'entreprise SOGESI.

Cette entreprise a développé un module sous l'ERP Odoo pour la gestion de production qui devait être utilisé par le responsable de production. Il nous a été demandé de développer une version mobile/tablette pour les opérateurs. Cette application est développée avec Android. Elle est intégrée avec le module de production pour faciliter l'échange des informations entre les opérateurs de l'atelier et le responsable de production.

Mots clés : ERP, Odoo, MRP, Android, Production

Abstract

This work is part of our Master's degree final project in Computer Science field in Software Engineering, which took place in SOGESI company.

This company has developed a module under the Odoo ERP for the production management that will be used by production manager. It wants us to developed a mobile / tablet version for the operators. This application is developed with Android. It is integrated with the module 'Manufacturing' to facilitate the exchange of information between the operators of the workshop and the production manager.

Key words : ERP, Odoo, MRP, Android, Production.

ملخص

يُدرج هذا العمل في إطار مشروع نهاية الدراسة درجة ماستر في الاعلام الالي تخصص هندسة البرامج ، الذي تم إنجازه ضمن شركة Sogesi.

طورت هذه الشركة وحدة برمجية في Odoo لإدارة الإنتاج الذي سيتم استعماله من قبل أحد عمالها. طلب منا برمجة نسخة لأجهزة الهاتف و اللوحات الإلكترونية للعمال. تم إشراك أندرويد في تطوير هذا التطبيق ودمج وحدة الإنتاج لتسهيل تبادل المعلومات بين عمال الورشة و مسؤول الإنتاج

الكلمات المفتاحية : ERP, Odoo, MRP, أندرويد, انتاج