

Société : SOPHYSA
Etudiant : Florian BACLE

Rapport de stage

- Développement d'applications sur appareils mobiles lecteurs de codes barres
- Installation d'un réseau sans fil



REMERCIEMENTS

Je tiens à remercier l'ensemble du personnel du site qui m'a accueilli avec la plus grande sympathie, et notamment l'équipe recherche et développement avec qui j'ai passé de bon moments pendant ce stage. Par ailleurs, un grand merci à mon tuteur Xavier Valentin qui m'a fait confiance sur ce projet et qui m'a accordé beaucoup de temps et d'attention pour me soutenir dans la réalisation de ce travail.

SOMMAIRE

Introduction	1
I. Presentation de Sophysa	2
1.1 Historique	2
1.2 Organisation de la Société.....	2
1.3 La chaine de production.....	3
1.4 Le service informatique	3
1.4.a Le parc informatique	3
1.4.b Ses missions.....	3
1.4.c Ses activités.....	4
II. Le cahier des charges	5
1. Le système actuel	5
1.1 La préparation des commandes	5
1.2 La gestion de la Stérilisation	8
1.3 La consultation des dépôts.....	10
2. Les besoins	11
2.1 La préparation des commandes	11
2.2 La gestion de la Stérilisation	12
2.3 La consultation des dépôts.....	12
2.4 La logistique.....	12
3. Les outils à disposition	12
3.1 Le Datalogic memor.....	12
3.2 Les bornes Netgear WG302.....	13
3.3 SwingMobility	13
4. Les objectifs	14
4.1 Le réseau	14
4.2 Les appareils mobiles	14
4.3 Les applications	14
5. Les contraintes	15
5.1 Le wifi.....	15
5.2 Les applications	16

6. Le calendrier	16
III. Solutions.....	18
1. Mise en place du wifi.....	18
1.1 Mise en place des bornes	18
1.2 Paramétrage des bornes	19
2. Configuration des appareils mobiles.....	20
2.1 Les appareils sur le réseau.....	20
2.2 La lecture des codes barre ean128.....	20
2.3 Leur administration	20
3. L'environnement de développement	21
3.1 Les différentes solutions étudiées.....	21
3.2 Le compact Framework 3.5	22
4. Les applications	23
4.1 Gestion des commandes	23
4.2 Gestion retour de stérilisation	25
4.3 La gestion des codes barres.....	28
4.4 La récupération des données	31
4.5 Le fichier de configuration.....	33
3.7 Ce qu'il reste à faire	34
Conclusion	35
Netographie	36
Développement Java sur appareils mobiles.....	36
Codes barres GS1 128.....	36
C sharp.....	37
WebService.....	37
Serveur FTP.....	37
Glossaire.....	38
Annexe	41

INTRODUCTION

Dans le cadre de la validation de ma troisième année de licence informatique à l'UFR-st de Besançon, il m'a été donné d'entreprendre la recherche d'un stage d'au moins 14 semaines. D'abord optimiste quant à la signature d'une convention, je me suis rapidement inquiété face au nombre de réponses négatives. Bien que souhaitant effectuer un master sécurité et sûreté du logiciel, il a été difficile d'orienter mes recherches dans ce sens. J'ai finalement été reçu dans les locaux de Sophysa, usine de composants neurochirurgicaux situé à TEMIS où j'ai travaillé en tant que magasinier l'été passé. Ayant de bonnes relations avec le personnel, respectant finalement mon projet professionnel et possédant déjà quelques connaissances vis-à-vis du projet qui m'a été confié, j'ai pu obtenir un stage jusqu'au premier juillet.

Poussé par la multiplication de la quantité de produits à gérer au sein de l'usine et par le désir d'améliorer la productivité, le projet est né à la demande de la direction. Il vise à mettre à la disposition des magasiniers un système leur permettant la gestion des stocks via l'utilisation d'appareils mobiles lecteurs de codes barres fonctionnant sous Windows Mobile 6.1. Il se divise en deux parties, la première étant à la charge de l'entreprise SwingMobility* qui doit fournir des applications sur les appareils permettant d'effectuer des inventaires et des transferts de dépôts.

C'est sur la seconde partie que nous interviendrons. Elle vise à mettre en place trois autres applications ayant chacune une visée précise. La vérification et le suivi de la préparation des commandes, la vérification et le transfert des produits en retour de stérilisation et la consultation de l'état des stocks. La partie comprend, l'établissement du cahier des charges, la mise en place du réseau sans fil, la configuration des appareils, la construction des applications et bien sur la formation des utilisateurs.

Dans ce sens, ce rapport fera état de la situation de la PME afin de mettre en place le contexte du stage. Vous y trouverez l'historique de l'entreprise Sophysa, son organisation, sa chaîne de production et la description du service informatique. Le cahier des charges sera le second point abordé. Primordial quant à la réalisation du projet, il constituera ici une explication du fonctionnement des protocoles ciblés pour en mettre en valeur les besoins. A partir de ces besoins, les différentes contraintes seront détaillées en fonction des objectifs qui auront été exprimés. Il sera nécessaire de revenir sur les outils à disposition et le calendrier prévisionnel. Finalement, la dernière partie se concentrera sur les solutions qui ont été apportées jusqu'à présent en revenant sur l'installation du réseau sans fil, la configuration des appareils mobiles et les applications développées. Il conviendra de terminer sur le travail restant pour ce dernier mois d'activité au sein de la société.

I. PRESENTATION DE SOPHYSA

Sophysa (SOciété de PHYsiologie Appliquée) est une entreprise spécialisée dans la conception, la fabrication et la commercialisation de produits destinés à la neurochirurgie, en particulier au traitement de l'hydrocéphalie*.

1.1 Historique

Fondée en Mai 1976 par Bernard Marion, ancien Directeur Général de Cordis en France, la société Sophysa s'est développée à partir d'une innovation majeure pour le traitement de l'hydrocéphalie : la valve programmable SOPHY®, première valve au monde à permettre le réglage de la pression à travers la peau, sans ré-intervention chirurgicale.

Rachetée en 1989 par le groupe japonais TOKIBO, Sophysa s'est rapidement développée en France et à l'export et a acquis au fil des ans une reconnaissance internationale dans le domaine très spécialisé de la neurochirurgie.

L'Extension de l'unité de Production de Besançon s'effectue en 1995. En 2008, le dépôt situé à Orsay est transféré vers TEMIS et commencent les expéditions clients de Besançon.

Peu à peu Sophysa s'étend dans les autres pays et prend une importance grandissante. La création de Sophysa-Bénélux (site de Nivelles) et de Sophysa-USA a lieu en 2002. Trois ans plus tard, la société acquiert de nouveaux distributeurs au Brésil et en Allemagne.

En 2010, Sophysa rachète la Société Vygon Neuro et transfère son activité sur le site de TEMIS. Ainsi, elle récupère plus de 50 ans de savoir faire dans le silicone et complète sa gamme de produit pour la neurochirurgie, notamment avec des valves destinées aux pays en voie de développement.

Aujourd'hui, Sophysa exporte ses produits dans plus de 70 pays (Europe, Japon, Brésil, USA...) et détient en France, la première place sur le marché des valves neurologiques. Elle emploie 67 personnes en France et à l'étranger.

1.2 Organisation de la Société

La jeune entreprise s'organise désormais sur plusieurs sites. Le siège social, situé à Orsay, centralise les activités administratives, marketing, commerciales et réglementaires.

Le site industriel de Besançon regroupe les activités de Production, de Contrôle Qualité ainsi que le bureau d'étude.

Les sites de Brain l'Alleud (Belgique) et de Crown point (USA) possèdent une activité purement commerciale sur leurs territoires respectifs.

1.3 La chaîne de production

Centralisée à Besançon, la chaîne de production s'étale sur de nombreux services. Elle démarre au service « Recherche et Développement » qui organise la réalisation de nouveaux produits, mais aussi l'amélioration et le suivi des projets antérieurs. La mise en production de ces projets est effectuée au niveau du service « méthode » qui lui va mettre en place un système et un protocole de fabrication. On arrive ensuite sur la production en elle-même, se faisant en grande partie en salle blanche (400 m², deux équipes de 12 personnes), étroitement lié au magasin qui organise le stockage, la réception et l'emballage des produits. Autour de cela, de nombreux contrôles sur les articles sont effectués en amont et en aval de cette chaîne via le service « qualité ».

1.4 Le service informatique

Basé sur le site de production à TEMIS, il n'est composé que d'une seule personne, Xavier Valentin. Son rôle au sein de l'entreprise est d'organiser la maintenance, le suivi et la mise à jour de l'installation informatique dans sa globalité.

1.4.a Le parc informatique

Le parc informatique est réparti sur trois sites et se compose de :

- 15 clients légers*.
- 25 portables.
- 55 pc de bureau.
- 10 serveurs.
- 25 imprimantes.
- 3 routeurs.

Le réseau comprend plusieurs environnements. Tous les postes étant sous Windows, l'authentification est gérée par Active Directory. Le partage des applications et des fichiers se fait sous environnement Citrix*. Un server Lotus Domino* permet d'apporter une messagerie, un annuaire intégré, un agenda collectif et un gestionnaire de documents organisé en base (base documentaire).

La production, la comptabilité, les ventes et les stocks sont gérés via l'ERP* SwingAgility qui utilise une base de données hébergée sous Microsoft® SQL Server 2005.

1.4.b Ses missions

L'administrateur du réseau a pour mission d'assurer une qualité de service constante et d'améliorer la productivité du réseau en organisant la maintenance préventive et en

contrôlant la sécurité de ce dernier. Il est le garant, au quotidien, de l'utilisation optimale et la plus large possible des moyens de communications et de productions.

D'un point de vue utilisateur il doit fournir un support leur garantissant de pouvoir travailler dans les meilleures conditions. Il est le seul intermédiaire entre le personnel et le système, il doit donc assumer la formation et le conseil sur l'ensemble des fonctionnalités du réseau.

1.4.c Ses activités

Les activités de l'administrateur sont extrêmement diverses puisqu'elles s'étendent du simple support utilisateur à la mise en place de nouveaux serveurs.

- Il étudie, préconise et supervise l'implantation de nouveaux matériels et logiciels.
- Il suit et analyse les performances du système, s'assure du traitement des incidents ou anomalies (diagnostic et résolution) en pilotant les interventions.
- En relation avec les fournisseurs, il assure une veille technologique permanente sur les produits matériels et logiciels.
- Il assure un support utilisateur constant et garanti la résolution de problème mais aussi une analyse des données via la récupération d'informations sur la base.

II. LE CAHIER DES CHARGES

Afin de mener à bien ce projet, il était avant tout nécessaire d'établir un cahier des charges aussi clair que possible. Sa mise en place a constituée la première étape de la réalisation du sujet. Il se présentera ici avec tout d'abord, une étude du système actuel afin d'en dégager les besoins. Pour continuer, une revue des éléments à disposition en revenant sur le matériel disponible et le système de gestion des données. Face aux différents constats effectués, les objectifs et les contraintes qui en découlent seront mis en places. Pour finir, il conviendra d'établir les différentes étapes à suivre dans la réalisation de ce travail.

1. Le système actuel

Le sujet se situe sur deux secteurs de l'entreprise, le magasin expédition et le magasin réception. Sur ces deux secteurs, les protocoles à étudier sont respectivement « la préparation des commandes » et « la gestion des produits en stérilisation ». A côté de cela, on distingue aussi un protocole commun aux deux secteurs, la « consultation des dépôts ».

1.1 La préparation des commandes

C'est au magasin expédition qu'est organisée la préparation des commandes, en collaboration évidemment avec le service commercial qui gère ces commandes via l'ERP* en place. Il existe en réalité 3 types de documents avec lesquels le magasinier doit organiser son travail. Ces trois types de documents correspondent à trois types de commandes différents.

1.1.a Les commandes France

Lorsqu'une commande France est passée par un client, les secrétaires la saisissent dans l'ERP*. Pour répondre à cette commande, un bon de livraison (BL) est créé, elles ajoutent à ce bon de livraison la liste des articles à envoyer, identifiés par références, lots et quantités (voir annexe [Image 1 : Bon de livraison](#)). Au moment où la commande doit partir, elles envoient un mail au magasinier contenant le numéro de ce BL. Le magasinier peut alors imprimer ce document et dispose de la liste des articles à emballer. Il va alors chercher les articles dans le stock suivant cette liste pour les mettre en carton et envoyer le ou les colis accompagnés du BL.

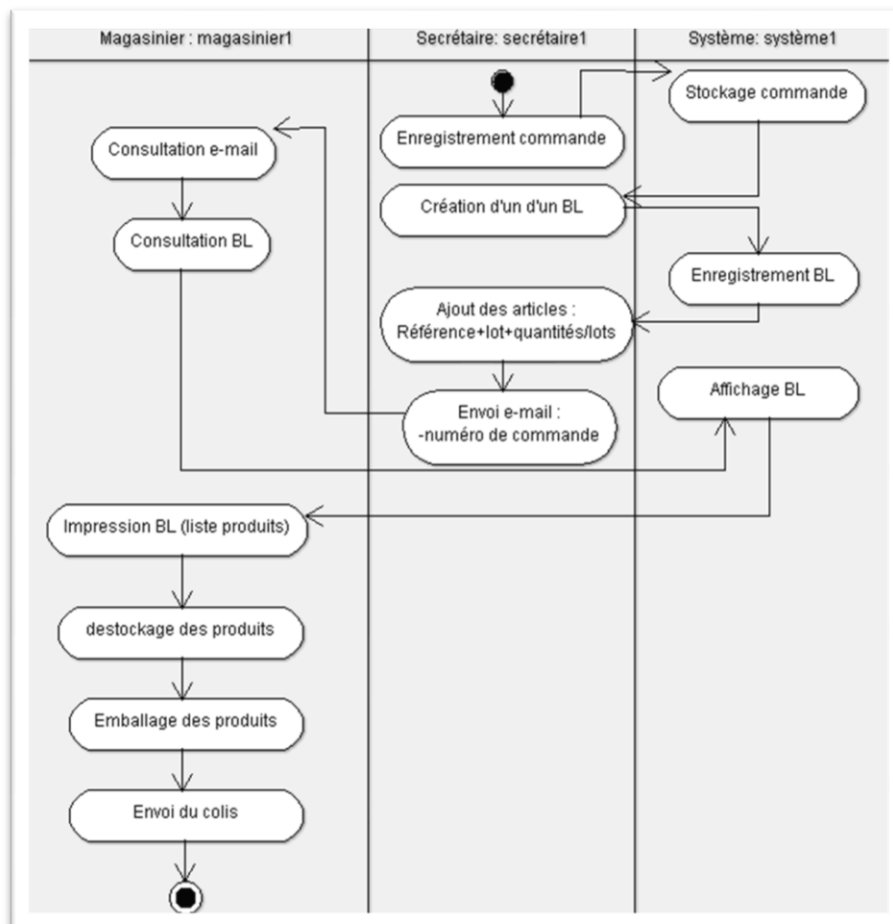


Figure 1 : Diagramme d'activité du protocole d'emballage d'une commande France.

1.1.b Les commandes France avec dépôt externe

Les commandes à destination de la France avec dépôt externe fonctionnent sur un document différent du BL. On attribut au client un dépôt, les produits sont donc entreposés chez ce client. A tout moment, il peut faire une demande d'approvisionnement de son dépôt. Les secrétaires organisent un transfert des produits demandés vers ce dépôt et le magasinier reçoit alors un « bordereau de mise en dépôt » (voir annexe [Image 2 : Bordereau de mise en dépôt](#)) où les produits à envoyer sont identifiés de la même manière que pour le BL. Le fonctionnement est donc le même, mais cette fois à partir du bordereau.

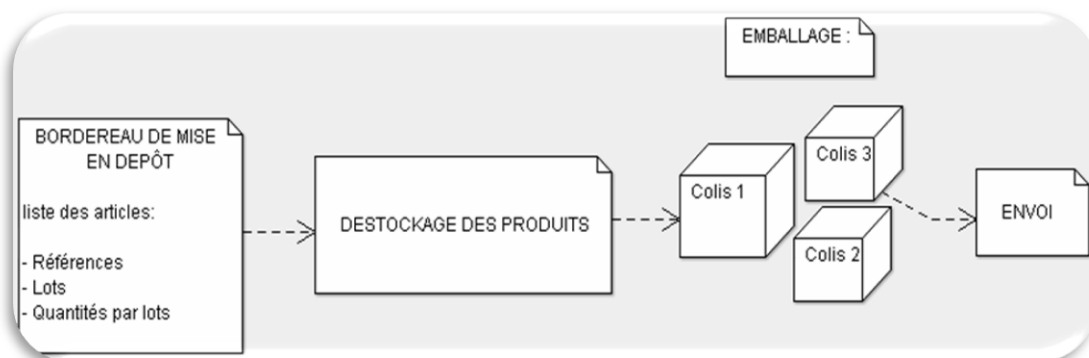


Figure 2 : Schéma du protocole d'emballage avec un bordereau de mise en dépôt

1.1.c Les commandes hors France

Pour ce type d'envoi, un dépôt est attribué à chacun des clients étrangers, ce dépôt n'est pas externe, il constitue juste une manière de « réserver des produits » pour une destination. Les opératrices vont y ajouter les articles des commandes passées par le client, en fonction des disponibilités des stocks. Ce dépôt contient donc les articles de plusieurs commandes, mais pour ce même client. Lorsqu'elles estiment qu'il faut envoyer les produits, elles fournissent au magasinier le numéro du dépôt à envoyer. Il disposera donc d'un numéro de dépôt, dépôt dont il récupère le détail (voir annexe [Image 3 : Détail d'un dépôt](#)) pour avoir la liste des articles à emballer. De manière générale, il va préparer les produits à partir de cette liste, sans savoir à quelle commande ils se réfèrent, il mélange donc les commandes du client dans ses colis.

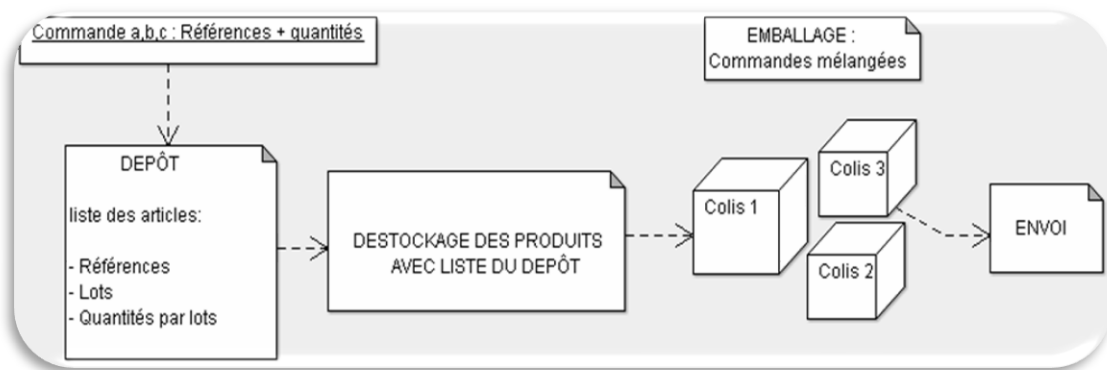


Figure 3 : Schéma du protocole d'emballage d'une commande hors France

Mais, pour certaine destination, il lui est demandé d'organiser les colis selon les commandes, dans ce genre de cas, les opératrices vont lui fournir, en plus du numéro de dépôt, le numéro des commandes associées au client, à partir desquelles elles auront rempli le dépôt. Ces commandes fournissent au magasinier une référence et une quantité globale des articles sans détail de lot. Le magasinier utilise les deux documents, commande et le détail du dépôt, pour trier ses colis par commande.

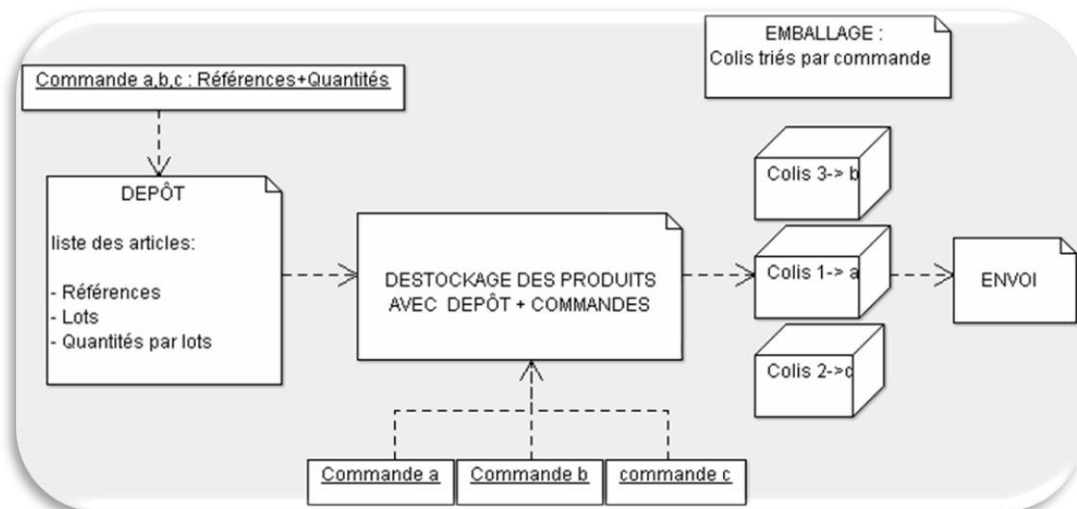


Figure 4 : Schéma du protocole d'emballage d'une commande triée par commande.

A partir de ces informations, on peut dégager plusieurs problèmes quant à ce mode de fonctionnement.

Premièrement, le déstockage des produits par le magasinier se fait sans aucune sécurité, le magasinier ne fait qu'un contrôle visuel lorsqu'il choisit ses articles. Si il fait une erreur, ils seront emballés et envoyés au mauvais endroit, et ce, sans aucune trace.

Le deuxième problème réside justement dans la traçabilité des produits. Avec ce système, impossible de remonter une erreur sur l'emballage ou de suivre un article dans le processus d'envoi. Par ailleurs, il n'existe aucun moyen de connaître l'avancement de la préparation d'une commande.

Pour finir, concernant les commandes hors France, les lots qui ont été choisis par le magasinier (dans le dépôt du client) pour chacune des commandes ne sont pas répertoriés. On ne sait pas précisément quels couples (références, lots) le magasinier a attribués aux commandes.

1.2 La gestion de la Stérilisation

Travaillant dans le domaine médical, l'entreprise doit gérer une grande quantité de matériel stérile. Cette stérilisation ne se fait pas sur place, elle est sous-traitée. Toutes les semaines, environ 8 caisses de produits sont envoyées au sous-traitant après préparation.

1.2.a La préparation à la stérilisation

Ce processus mobilise le service conditionnement qui emballe, enregistre et étiquète les produits sortis de production et nécessitant un envoi en stérilisation. Chaque envoi en stérilisation est identifié par un numéro du type « X0123 ». Pour pouvoir suivre les produits, leurs étiquettes doivent indiquer ce numéro, ainsi qu'une date de péremption.

Afin d'organiser l'enregistrement de la liste des produits d'une stérilisation, les opératrices vont saisir les articles sur un fichier Excel (voir annexe [Image 5](#) : Fichier EXCEL des produits en stérilisation).

Une fois les actions d'emballage, d'étiquetage et d'enregistrement effectuées, le suivi informatique se fait par le transfert de cette liste d'article au dépôt 002 – Désorption, via l'utilisation d'un module d'intégration sur l'ERP* capable d'effectuer des transferts de dépôts à partir d'un fichier Excel. Parallèlement, Les produits sont envoyés au sous-traitant.

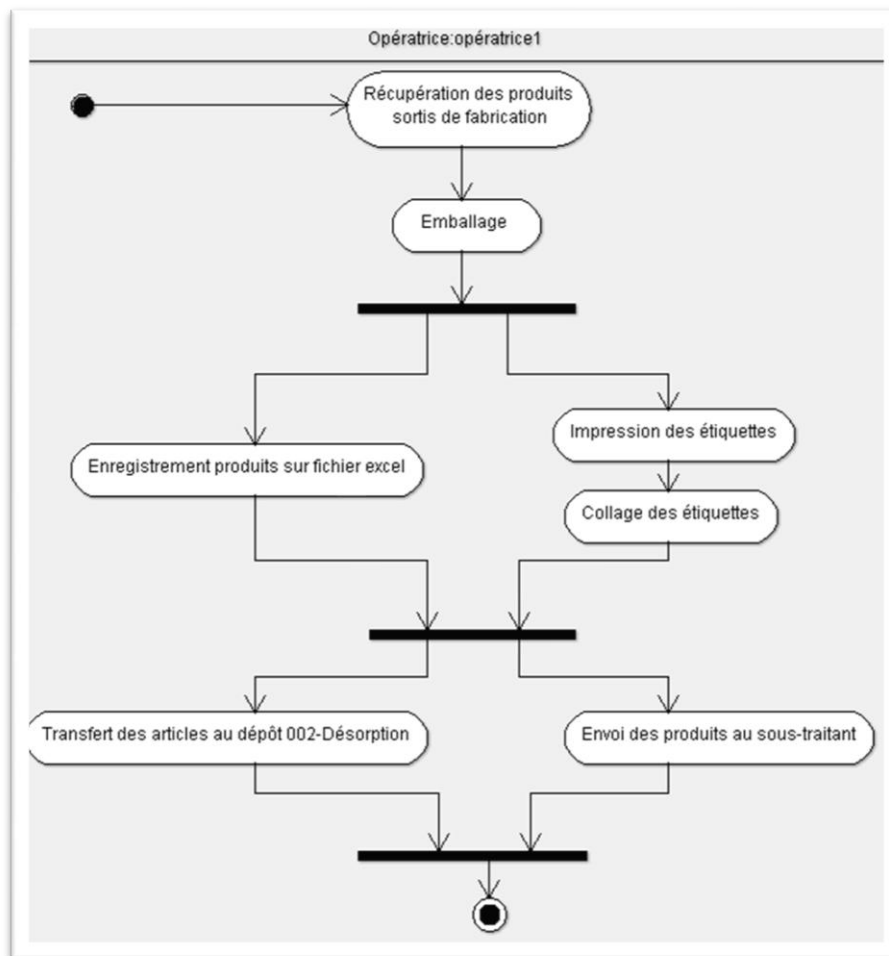


Figure 5 : Diagramme d'activité du protocole d'envoi des produits en stérilisation.

A la fin de la préparation on a :

- La liste des produits enregistrés dans un fichier Excel.
- Une indication de numéro de stérilisation et date de péremption sur les produits.
- Un dépôt 002 auquel on a ajouté les produits envoyés.

1.2.b Le retour de stérilisation

C'est le magasin réception qui se charge de la mise en place du retour de stérilisation et c'est en particulier sur cette partie que notre travail se concentrera. Le magasinier s'organise en différentes étapes.

D'abord, les colis arrivent, il les ouvre, puis sors les produits des cartons pour déterminer à quel numéro de stérilisation il à affaire. Il peut alors récupérer le fichier Excel préalablement rempli à la préparation. Il lui faudra vérifier qu'il a bien reçu la totalité des articles envoyés, il doit donc cocher un à un les produits sur sa liste après les avoir identifiés. Il peut tout à fait recevoir une stérilisation en plusieurs fois, il conserve la liste qu'il a commencée pour un numéro de stérilisation tant qu'il ne l'a pas complètement reçue.

Lorsqu'il a effectivement reçu l'intégralité des articles d'une stérilisation, il va alors les sortir du dépôt 002 pour les envoyer vers les dépôts nécessaires (les destinations étant déterminées par de nombreux facteurs) en utilisant le fichier Excel et l'intégrateur sur l'ERP*.

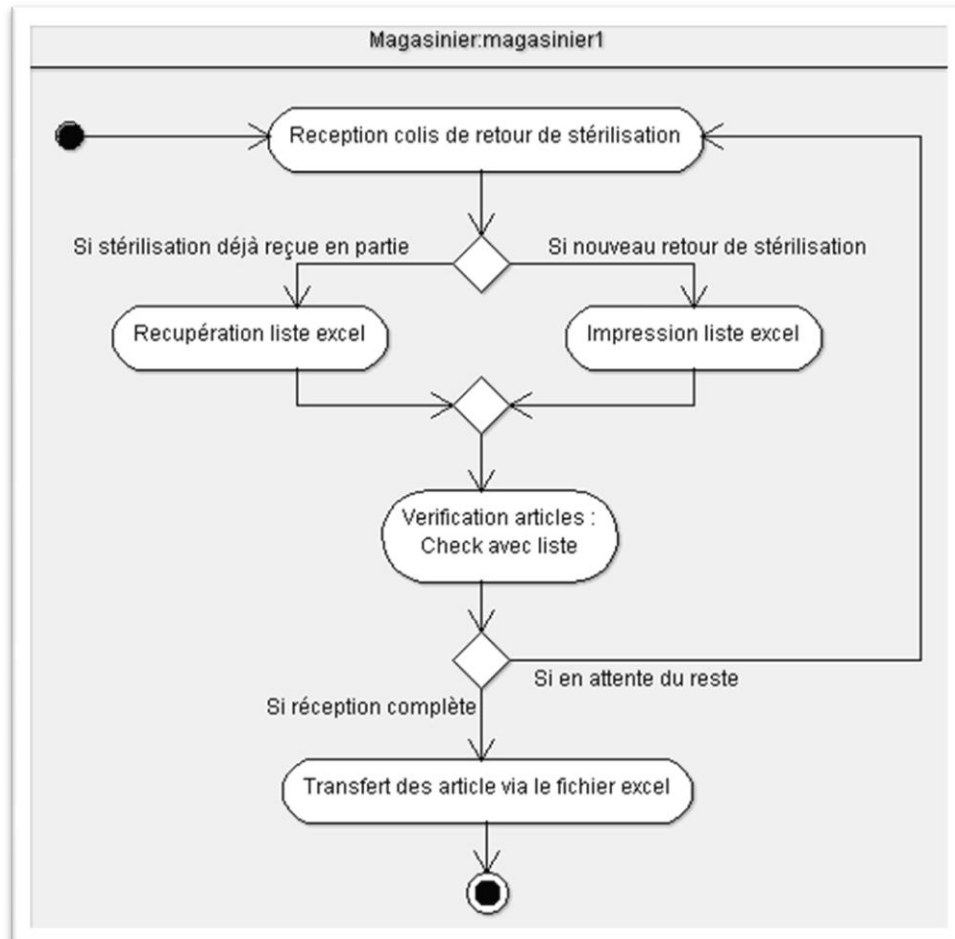


Figure 6 : Diagramme d'activité du protocole de gestion des produits en retour de stérilisation.

Le problème sur ce processus est qu'il est très long. En effet, le magasinier doit prendre sa liste et y cocher un à un les articles qu'il a reçu pour une stérilisation. Sachant qu'une réception contient en moyenne une centaine d'article, il perd un temps significatif.

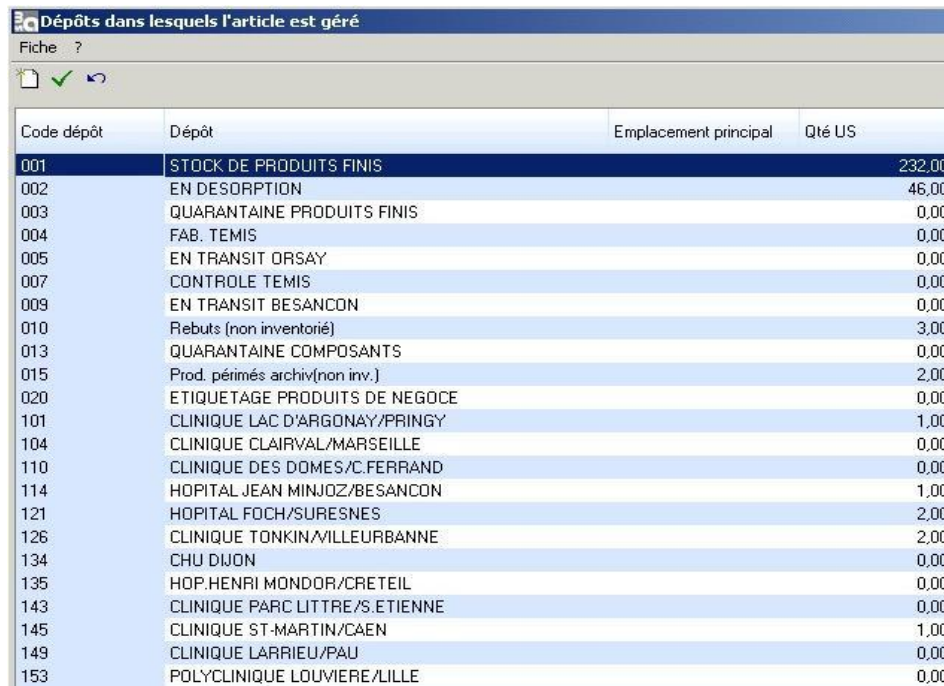
Par ailleurs, sa vérification est, comme pour la gestion des commandes, uniquement visuelle, il n'y a aucune sécurité.

Pour finir, le magasinier est obligé d'organiser ses transferts un à un via le fichier Excel, un travail extrêmement peu pratique.

1.3 La consultation des dépôts

Pendant leurs heures de travail, il arrive très fréquemment aux magasiniers d'avoir à consulter l'état des stocks. A l'heure actuelle, ils utilisent l'ERP* qui comprend un module de

consultation, mais ils doivent passer par leurs postes, ce qui leurs fait faire de nombreux aller et retour et la perte de temps est non négligeable. Concrètement, la manipulation est très simple. Il suffit de se connecter sur l'ERP*, d'entrer dans le module et d'y saisir une référence pour laquelle on souhaite le détail des stocks, on obtient la quantité de l'article pour tous les dépôts :



Code dépôt	Dépôt	Emplacement principal	Qté US
001	STOCK DE PRODUITS FINIS		232,00
002	EN DESORPTION		46,00
003	QUARANTAINE PRODUITS FINIS		0,00
004	FAB. TEMIS		0,00
005	EN TRANSIT ORSAY		0,00
007	CONTROLE TEMIS		0,00
009	EN TRANSIT BESANCON		0,00
010	Rebuts (non inventorié)		3,00
013	QUARANTAINE COMPOSANTS		0,00
015	Prod. périmés archiv(non inv.)		2,00
020	ETIQUETAGE PRODUITS DE NEGOCE		0,00
101	CLINIQUE LAC D'ARGONAY/PRINGY		1,00
104	CLINIQUE CLAIRVAL/MARSEILLE		0,00
110	CLINIQUE DES DOMES/C.FERRAND		0,00
114	HOPITAL JEAN MINJOZ/BESANCON		1,00
121	HOPITAL FOCH/SURESNES		2,00
126	CLINIQUE TONKIN/VILLEURBANNE		2,00
134	CHU DIJON		0,00
135	HOP.HENRI MONDOR/CRETEIL		0,00
143	CLINIQUE PARC LITRE/S.ETIENNE		0,00
145	CLINIQUE ST-MARTIN/CAEN		1,00
149	CLINIQUE LARRIEU/PAU		0,00
153	POLYCLINIQUE LOUVIERE/LILLE		0,00

Figure 7 : Ecran de résultat d'une interrogation de stock pour un article.

2. Les besoins

On a pu observer sur les procédés précédents de nombreux problèmes qu'une informatisation pourrait corriger. Les besoins qui découlent de ce constat sont les suivants.

2.1 La préparation des commandes

La préparation des commandes se fait de manière non contrôlée et non rigoureuse, sa gestion nécessiterait :

- Un contrôle du contenu des colis, un moyen de s'assurer que le ou les colis contiennent effectivement les articles du BL, de la commande ou du dépôt, avant l'envoi.
- Un système qui permettrait de tracer les produits pris par le magasinier pour préparer sa commande.
- Un système qui permettrait aux secrétaires de savoir l'état d'avancement du travail qu'elles lui fournissent, de manière à pouvoir maîtriser le flux des commandes d'un service à l'autre.

2.2 La gestion de la Stérilisation

La gestion du retour de stérilisation souffre d'un temps d'exécution trop lourd, par ailleurs, on ne distingue aucun contrôle autre que celui du magasinier, les besoins se définissent par :

- Un gain de temps sur la vérification des produits reçu.
- Un système qui ferait office d'historique des entrées, afin de remplacer la vérification manuelle sur papier.
- Un système différent de l'utilisation du fichier Excel pour faire les transferts de dépôts.

2.3 La consultation des dépôts

La consultation des dépôts étant fréquente dans la journée et impliquant l'utilisation d'un poste de travail fixe, il est nécessaire d'installer un système qui permettrait cette consultation n'importe où dans le magasin.

2.4 La logistique

Les systèmes d'assistance à mettre en place doivent pouvoir suivre les magasiniers. Mais l'équipement informatique actuel ne prévoit pas de transmissions sans fil. L'utilisation d'un appareil mobile pour héberger ces systèmes sera nécessaire. D'un point de vu matériel on aura donc besoin :

- De bornes wifi.
- Appareil mobile équipé wifi et permettant la lecture des codes barres utilisés.

3. Les outils à disposition

En réalité, le matériel nécessaire à l'établissement du wifi à déjà été acheté. L'achat des appareils s'est fait au préalable et s'est avec lui qu'il faut composer. Ce matériel comprend, trois appareils mobiles Datalogic Memor et deux bornes wifi Netgear WG302. D'autre part, il faut noter que tout le système de production est centralisé dans l'ERP*, afin d'être au clair sur ces éléments, en voici une brève description.

3.1 Le Datalogic memor

L'achat de ces appareils s'est fait sur la demande de SwingMobility*, les développeurs de l'ERP*, qui ont eux aussi des applications à mettre en place. En voici les caractéristiques :



Figure 8 : Appareil mobile Datalogic Memor.

Système :

- Fonctionne sous Windows mobile 6.1.
- Microprocesseur Marvell PXA 310 624MHz.
- Ram : 128MB.
- Flash : 256MB.

Ecran :

- LCD.
- 320 X 240 pixel.
- Tactile.

Communications :

- WLAN : IEEE 802.11b/g.
- WPAN : Bluetooth IEEE 802.15, classe 2, version 2.0.

3.2 Les bornes Netgear WG302

Des bornes wifi ont également été achetées, en voici la description.



Figure 9: Borne WIFI Netgear WG302.

- Support WPA* et WPA – PSK.
- Compatible 54 Mbit/s (norme 802.11g) et 11Mbit/s (norme 802.11b).
- ACL : authentification par adresse MAC des cartes autorisées à la connexion.
- Support de la fonction Wireless isolation : pas d'échange réseau entre les clients wifi.
- Serveur DHCP*.

3.3 SwingMobility



Figure 10 : Logo de l'ERP SwingMobility.

L'ERP utilisé à Sophysa se nomme SwingAgility et a été développé par SwingMobility*, c'est lui qui concentre toutes les activités de l'entreprise depuis 1997. C'est une solution de gestion intégrée permettant la mise en œuvre de tout ou partie du système d'information des filiales et départements des entreprises exerçant une activité industrielle, de distribution, de négoce, ou de service.

SwingAgility permet une gestion multi-organisationnelle : multi-sociétés, multi-départements, multi-dépôts, multi-emplacements, multi-conditionnements et multi-lots. Les modules utilisés à Sophysa sont les suivants :

- Gestion des ventes.
- Gestion des achats.
- Gestion des stocks.
- Gestion par affaire.
- Gestion comptabilité.
- Gestion de production.
- Gestion de la planification/ ordonnancement.
- Contrôle de la qualité et suivi des non-conformités.

D'un point de vu informatique, le système fonctionne sur une base de données hébergé par un server Microsoft SQL Server 2005. L'intégralité des informations est répartie sur 974 tables auxquelles on peut avoir accès via un système de bureau à distance.

4. Les objectifs

Les objectifs de se projet comprennent avant tout la mise en place du réseau wifi, l'idée étant de fournir ensuite une interface logicielle répondant aux besoins des magasiniers, entre des terminaux mobiles de saisie et la base de données de l'ERP*. Il faudra mettre en place, pour chacun des procédés énoncés précédemment, une application permettant de répondre aux problèmes qui en découlent.

4.1 Le réseau

Mettre en place le réseau wifi, positionner et paramétrer les deux bornes wifi. Il faudra avant tout étudier leur fonctionnement.

4.2 Les appareils mobiles

Configurer les appareils mobiles et organiser leur gestion à distance.

4.3 Les applications

Il faudra construire 3 applications, une pour chacun des processus étudiés précédemment. Afin de les réaliser, il sera nécessaire de déterminer un environnement de développement adapté, en étudiant les différentes possibilités.

4.3.a Application 1 : « Gestion des commandes »

Cette application, destinée au magasin expédition devra apporter :

- Un contrôle du contenu des cartons avant envoi.

- Une indication d'avancement de la préparation des commandes aux opératrices qui travaillent avec le magasinier.
- Un historique des sorties de stock des produits.

4.3.b Application 2 : « Gestion retour de stérilisation »

Destinée au magasin réception, elle doit apporter :

- Une substitution du système de vérification visuelle des produits par un contrôle informatique.
- Un gain de temps quant au protocole global.
- Une substitution au transfert de dépôt via le fichier Excel.

4.3.c Application 3 : « Consultation dépôt »

Cette dernière application devra permettre la consultation des détails de stock de la même manière que sur l'ERP*.

5. Les contraintes

Autour de ces objectifs, de nombreuses contraintes ont été établies en collaboration avec les acteurs du système, autant sur le réseau que sur les applications. Les solutions devront donc respecter certaines conditions.

5.1 Le wifi

La couverture du wifi doit comprendre l'ensemble des salles de travail des magasiniers, notamment le stock et la salle blanche, et ce, avec les deux bornes à disposition, voir image ci-dessous.

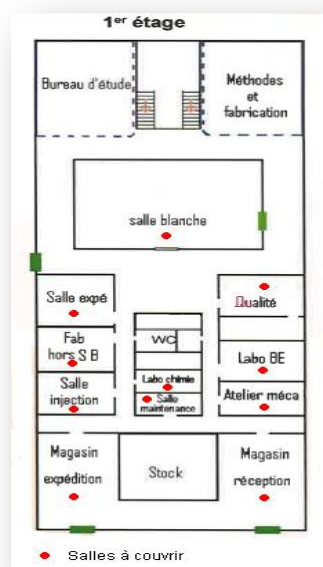


Figure 11 : Schéma des différentes salles à couvrir.

- Afin d'éviter une surexposition du personnel, il faudra limiter la puissance des bornes au strict nécessaire.
- Dans les soucis de garder le réseau sécurisé, la couverture devra se limiter autant que possible à l'intérieur de l'établissement et les communications devront être cryptées.
- L'émission devra être optimale au niveau du magasin, afin d'offrir les meilleures performances autour du stock, où les appareils seront utilisés le plus souvent.

5.2 Les applications

Le choix de l'environnement devra prendre en compte plusieurs contraintes :

- Permettre un accès à une base de données SQL server 2005.
- Avoir des outils permettant la mise en place d'une interface graphique.
- Être le moins cher possible.
- Être assez haut niveau pour permettre à l'administrateur, sans connaissance en programmation, de pouvoir comprendre un minimum le code.
- Permettre une portabilité aussi grande que possible.

De manière générale les applications devront être :

- Sécurisées.
- Simples à utiliser.
- Utilisables au doigt (sans le stylet).
- Rapides.

6. Le calendrier

La durée complète du projet est limitée à 16 semaines. Le calendrier prévisionnel s'étale sur toute la durée du stage. Chronologiquement, il se présente ainsi :

- 1) Compréhension de l'environnement réseau existant et installation du wifi : 2 semaines.
- 2) Définition des besoins avec les utilisateurs finaux : 2 semaines.
- 3) Etude du fonctionnement des modules concernés de l'ERP (commandes clients, gestion des stocks, état des stocks) : 1 semaine.
- 4) Etude de l'architecture de la base de données à interfacier : 1 semaine.
- 5) Analyse des possibilités des terminaux de saisie : 1 semaine.
- 6) Revue des différents environnements utilisables pour la mise en place des applications : 3 semaines.
- 7) Mise en œuvre des solutions retenues : 4 semaines.
- 8) Tests et débogage : 2 semaines.

- Formation des utilisateurs : 1 jour.

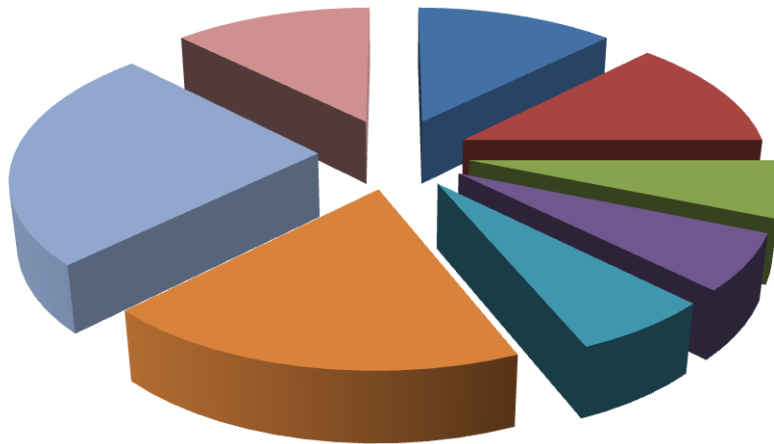


Figure 12 : Diagramme du temps en semaines prévu pour chacune des étapes du projet.

III. SOLUTIONS

Il sera présenté ici l'ensemble des solutions qui ont été trouvées et implémentées pour répondre au cahier des charges. Avant tout, l'installation du réseau sans fil et les décisions qui ont été prises pour le paramétrage des bornes seront détaillées, en prenant soin de revenir sur la configuration des appareils. Ensuite, il conviendra d'expliquer les choix de programmation qui ont été effectués pour le développement des applications, avant d'expliquer leur fonctionnement et les différentes fonctionnalités qu'elles offrent. Finalement, il sera établi un bilan sur ce qu'il reste à faire pour le dernier mois de stage.

1. Mise en place du wifi

La mise en place du Wifi s'est organisée en 2 parties. Le premier travail a consisté à déterminer un emplacement optimal des deux bornes en prenant en compte les possibilités du réseau actuel et les contraintes exprimées quant à la couverture. Le paramétrage des bornes sur le réseau est intervenu à posteriori pour garantir la sécurité et la stabilité de l'installation.

1.1 Mise en place des bornes

Le choix de l'emplacement des bornes s'est arrêté sur le positionnement d'une borne au stock même, et une borne au dessus de la salle blanche (dans les combles). De cette manière, la zone de couverture est respectée, et le débordement extérieur est évité. La puissance minimale des bornes suffit pour entretenir cette configuration, on empêche donc toute surexposition. Cependant, aucun accès réseau n'est disponible à ces deux endroits, de nouveaux câbles réseau depuis la baie de brassage* ont été tirés jusqu'aux deux bornes, voir schéma ci-dessous.

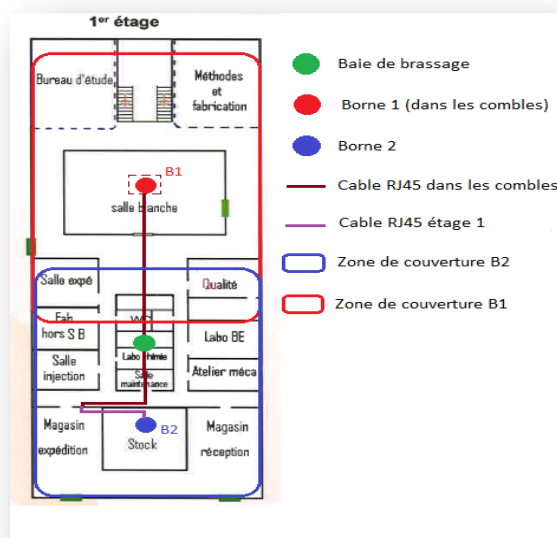


Figure 13 : Schéma de l'installation et la couverture du réseau sans fil.

1.2 Paramétrage des bornes

Le paramétrage des bornes s'est organisé en 3 parties. Des choix ont été effectués quant à la représentation des bornes sur le réseau, de manière à pouvoir les repérer et les identifier. Pour permettre la cohabitation des deux signaux, quelques réglages sur les paramètres d'émission ont été effectués. Par ailleurs, dans le souci de garder une sécurisation correcte, un contrôle de l'accès a été mis en œuvre.

1.2.a Identification sur le réseau

Pour commencer, les deux bornes sont identifiées par un nom différent à savoir « netgear1 » (dans les combles) et « netgear2 » (au stock). Le serveur peut fournir des adresses IP à la volée via le DHCP*, mais, pour garder une plus grande accessibilité et une organisation plus propre, nous avons choisi un adressage fixe, les deux bornes seront donc accessibles sur :

- netgear1 : 130.0.0.231
- netgear2 : 130.0.0.233
- Le masque de sous-réseau est : 255.255.255.0
- La passerelle par défaut : 130.0.0.254
- DNS 1 : 130.0.0.16
- DNS 2 : 130.0.0.13

1.2.b Paramètres d'émissions

Les paramètres d'émission sont réglés de manière à éviter les interférences dans les zones de couvertures communes. Les deux bornes ont ainsi le même SSID* « netgear », mais émettent sur des fréquences différentes.

- netgear1 : channel11/Fréquence : 2.462GHz
- netgear2 : channel1/Fréquence : 2.412GHz

Comme prévu dans le cahier des charges, la puissance du signal est au minimum possible.

1.2.c Contrôle de l'accès et sécurisation

Plusieurs précautions ont été prises pour sécuriser le réseau sans fil. Avant tout, l'accès au hot spot de configuration des bornes est limité à un compte administrateur. Au niveau du cryptage des informations, nous utilisons une authentification par clé WPA2-PSK* 64 bits avec un cryptage TKIP*+AES*, ce qui garantit une sécurisation convenable.

Afin de limiter l'accès au réseau, un contrôle d'accès par adresse mac a été ajouté, ainsi, seuls les appareils enregistrés pourront obtenir une connexion.

2. Configuration des appareils mobiles

La configuration des mobiles a été plutôt rapide à mettre en œuvre. Elle a impliqué une configuration de la connexion, en fonction des paramètres des bornes. Le réglage du lecteur pour garantir la lecture des codes barre EAN128*. Pour finir, la sauvegarde des paramètres et l'installation d'un server FTP, primordial pour une gestion à distance.

2.1 Les appareils sur le réseau

De la même manière que pour les bornes, il est important de pouvoir retrouver rapidement les appareils sur le réseau, ils leur sont donc attribuée une adresse IP fixe :

- Appareil magasinier1 : 130.0.0.36
- Appareil magasinier2 : 130.0.0.37
- Appareil salle Blanche : 130.0.0.38
- Masque de sous réseau : 255.255.255.0

En ce qui concerne la connexion aux bornes, un profil utilisant le SSID* « netgear » est attribué aux deux points d'accès. Il respecte le cryptage WPA2-PSK* défini.

2.2 La lecture des codes barre ean128

Le paramétrage du lecteur de code barre ne prévoit pas, avec les réglages d'origine, la lecture des codes barre EAN128*. Pour assurer le bon fonctionnement des applications, les scanners ont été configurés avec les valeurs suivantes :

- EAN128* : Enable.
- ScanAlwaysOn : Enable.
- KeyboardEmulation : Yes.
- NotPrintableChar : Leave.

2.3 Leur administration

Pendant leur durée de vie, il est très probable que des mises à jour voient le jour, aussi bien sur le firmware* que sur les applications. D'autre part, il ne faut pas exclure d'éventuelles remises aux paramètres d'usines.

2.3.a Les sauvegardes

Via les utilitaires fournis avec les mobiles, une exportation de l'ensemble des configurations de l'appareil a été effectuée, ce qui permettra à l'administrateur de ne pas perdre trop de temps sur la réinitialisation si besoin est. Cette exportation comprend :

- Une sauvegarde des paramètres WIFI de l'appareil.

- Une sauvegarde des paramètres relatifs à l'appareil, notamment pour le scanner.

2.3.b Le server FTP

Afin de garantir un transfert de fichiers à distance, un serveur FTP* est installé sur les appareils. C'est un serveur open source fiable et paramétrable nommé « Mocha Ftp Server » (voir netographie, Serveur FTP). Pour garder un accès sécurisé, un compte administrateur est utilisé. Le contenu des mobiles est donc gérable à partir d'un client FTP quelconque et à l'adresse IP du mobile, ici : 130.0.0.36.

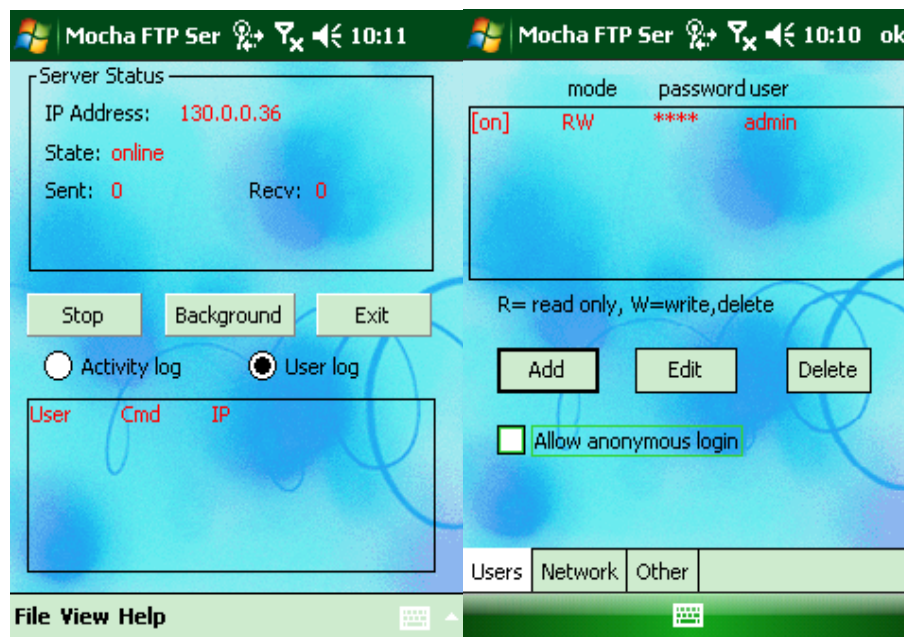


Figure 14 : aperçu du server FTP Mocha FTP Server.

3. L'environnement de développement

Parmi les objectifs de ce projet, l'un des points les plus importants concerne le choix de l'environnement de développement. Dans la multitude de possibilités envisageables, quelques une ont été étudiées vis-à-vis des contraintes exprimées.

3.1 Les différentes solutions étudiées

3.1.a Le web service

La mise en place d'un web service s'est avérée être une solution plutôt intéressante, elle permet en effet de fournir une grande portabilité si le choix de l'implémentation côté client est judicieux. A priori, aucun problème concernant La connexion à une base de données, compte tenu du fait que la technologie utilisée côté serveur est aussi au choix. Une telle solution sous-entend cependant la mise en place d'un serveur web. Ce service est

disponible à Sophysa, en réalité, un service web est déjà implanté, utilisé par Swingmobility* pour mettre en place leurs applications. Mais ce service n'est pas documenté, et en aucun cas il est prévu une utilisation externe, la mise en place d'autres applications ne pourrait pas être maintenue en cas de mise à jour de leur service. Le problème sur cette solution est qu'elle est plutôt compliquée à mettre en place, notamment sur les échanges client/serveur sur lesquels l'administrateur n'aurait probablement pas été à l'aise.

3.1.b Le Java

L'utilisation du Java est une solution que nous avons très vite envisagée, bien qu'assez complexe à mettre en œuvre compte tenu des nombreux profils et configurations existant, et sur laquelle nous avons finalement perdu beaucoup de temps. Le Java sur appareil mobile est plutôt mal documenté, mais techniquement, il existe de quoi le mettre en place. Au niveau des machines virtuelles, il y en a quelques-unes fiables, on trouve CrEme™ (voir netographie, Développement Java sur appareils mobiles) et J9™ (Ibm) autour d'une trentaine de dollars la licence, avec à disposition des API graphiques tel que SWING ou AWT. Jusqu'ici, le Java aurait dû apporter, une excellente portabilité, un coût relativement faible, une API graphique suffisante, une maintenance plutôt aisée. Le gros problème réside en fait sur l'accès à une base de données, il n'existe aujourd'hui aucun pilote JDBC* pour SQL Server compatible avec les machines virtuelles disponibles, il est donc impossible d'établir une quelconque communication avec un serveur de base de données.

3.1.c Le c#

C'est finalement sur le C# que notre choix s'est arrêté. Ce langage, créé par Microsoft®, est un moyen d'exploitation de l'architecture .NET* de Microsoft® (sur lequel nous reviendrons). Tout comme le Java, il est orienté objet, il s'exécute sur le Runtime Compact Framework de la même manière que sur la machine virtuelle pour le Java, mais est supporté par Windows CE 4.2, 5.0, 6.0, Microsoft Pocket PC, Microsoft Pocket PC 2002 et Smartphone 2003, Windows Mobile 5.0 et Windows Mobile 6. Le c# permet de palier à toutes les contraintes énoncées sauf évidemment la portabilité puisque non utilisable sur des plateformes autres que Windows.

3.2 Le compact Framework 3.5

Le .Net Compact Framework de Microsoft (.Net CF) est une version du .NET Framework* spécialement conçue pour être exécutée sur des terminaux mobiles de type Windows CE ou Windows Mobile tels que des PDA, des téléphones portables, des panels PC industriels. Le Framework* gère tous les aspects de l'exécution d'une application dans un environnement d'exécution dit « managé ».

- Il alloue la mémoire pour le stockage des données et des instructions du programme.
- Il autorise ou refuse des droits à l'application.
- Il démarre et gère l'exécution.
- Il gère la réallocation de la mémoire pour les ressources qui ne sont plus utilisées.

Il est possible de développer des applications qui utilisent le Compact Framework avec Visual Studio 2003, Visual Studio 2005 et Visual Studio 2008 en langage C#, mais aussi avec des IDE* open source tel que SharpDevelop. Les applications développées ici ont été codées sur Visual studio 2008.

4. Les applications

Les deux applications réalisées jusqu'à présent respectent un concept commun, flasher, vérifier, stocker. Pour être plus clair, dans les deux cas, une table a été ajoutée dans la base de données de façon à pouvoir mémoriser les articles saisis. Table sur laquelle on va pouvoir faire des consultations, faire des modifications. Les deux applications ont beaucoup d'autres points communs, notamment sur la lecture, la gestion et le traitement des codes barres, mais aussi au niveau du fichier de configuration et la sécurisation des requêtes.

4.1 Gestion des commandes

Pour rappel, cette application doit encadrer la vérification du contenu des colis préparés par le magasinier expédition. Le principe de fonctionnement et les différentes possibilités seront les deux points à aborder.

4.1.a Le principe

Avec son mobile, le magasinier va scanner un à un les produits qu'il prend. Chacun des produits est enregistré dans la table « SOPHYSCANCDE » hébergée sur le server. Dans cette table on enregistre, le numéro du document de référence (commande, BL ou dépôt saisi auparavant), le numéro système de l'article, son code lot, la quantité et la date de la saisie.

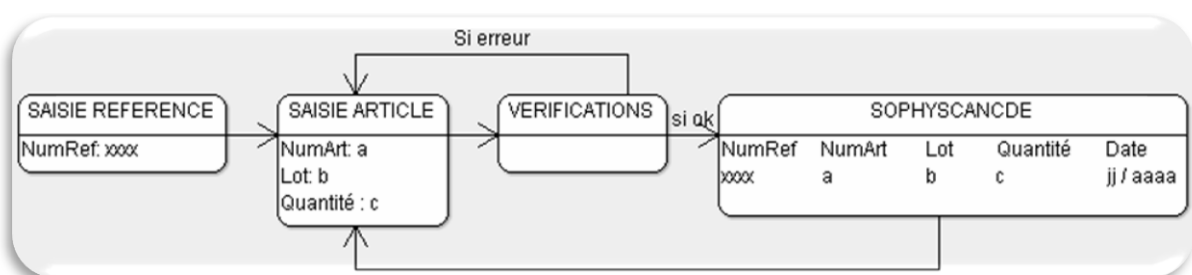


Figure 15 : Schéma du fonctionnement général de l'application « Gestion des Commandes ».

Avant tout ajout dans la base, une première sécurité est mise en place et différentes erreurs peuvent être levées. On vérifie que :

- L'article existe bien dans la base de données.
- Le document de référence existe bien dans la base de données.
- Le couple article/lot appartient bien à la liste des articles de la référence.
- La quantité ne dépasse pas celle de la référence.

Si une erreur est levée, le magasinier est informé visuellement sur une « console » et auditivement par un bip erreur. On empêche ainsi toute erreur à la saisie.

Cette configuration permet de garder une trace des produits pris par le magasinier pour chacun des documents qu'il a utilisés.

4.1.b Les fonctionnalités

Evidemment, cette application ne se limite pas à une saisie à la volée. En effet, il est important pour l'utilisateur d'avoir des fonctionnalités quant à cette table pour savoir où il en est dans la préparation de sa commande.

Dans ce sens, plusieurs options sont offertes. Depuis le menu, trois fenêtres sont accessibles, « Vérification », « Saisie » et « Suppression », voir figure 16 ci-dessous.

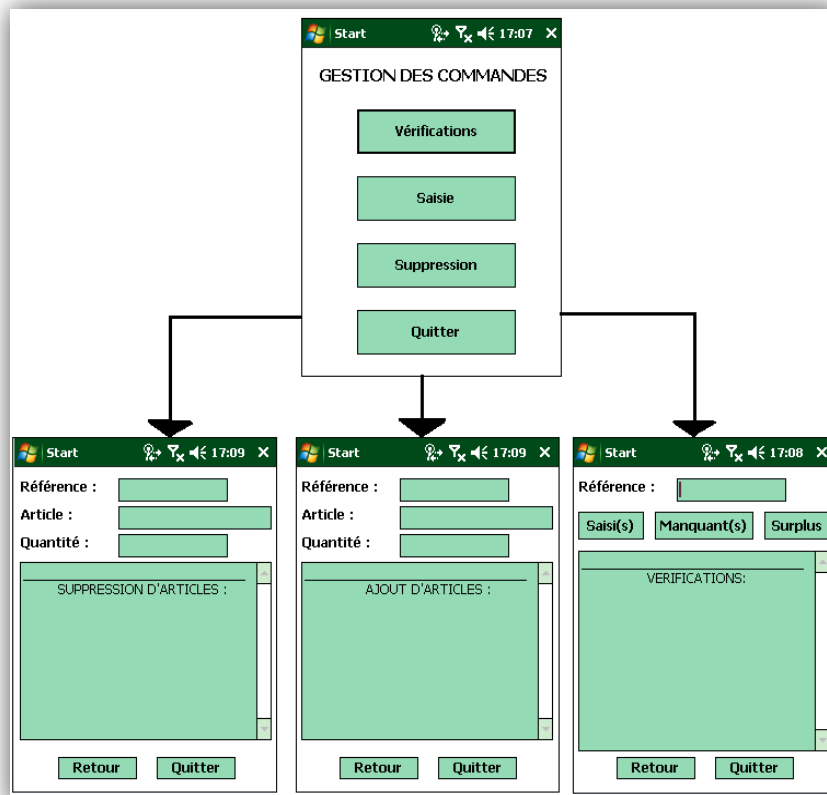


Figure 16 : Capture d'écran des différentes fenêtres de l'application « Gestion des commandes ».

La fenêtre « Saisie »

Depuis cette fenêtre, il est possible d'ajouter des articles à la table « SOPHYSCANCDE ». Compte tenu du fait qu'un couple numéro article/lot est unique, l'ajout d'un article pourra, selon les cas, faire l'objet d'une nouvelle ligne ou de l'incrémentation de la quantité de la ligne existante. Lorsque les vérifications ne dégagent aucune erreur, un message est affiché, récapitulant l'ajout qui vient d'être effectué. L'appareil émet alors bip caractéristique. Pour son confort, l'utilisateur n'a pas à ressaisir la référence à chaque fois qu'il entre un code barre, il peut flasher les articles à la chaîne. Il peut aussi choisir de définir une quantité pour un produit en la saisissant avant de scanner ce produit, la quantité du code barre sera ignorée.

La fenêtre « Suppression »

D'un point de vue programmation cette fenêtre est en fait la même que la fenêtre saisie, elle envoie simplement sur un traitement inverse. De la même manière que l'on incrémente la quantité ou ajoute une ligne, on va ici décrémenter ou retirer la ligne. Un message récapitulatif et un bip indique la suppression effectuée.

La fenêtre « Vérifications »

Cette fenêtre est en fait une fenêtre de contrôle. Le magasinier peut afficher, pour un document donné, la liste des articles qu'il a déjà saisis, la liste des articles qu'il n'a pas saisi ou la liste des articles qu'il a saisi mais qui n'appartiennent plus à la liste des articles du document (bouton « Surplus »). Il peut ainsi gérer son avancement et sa liste d'article.

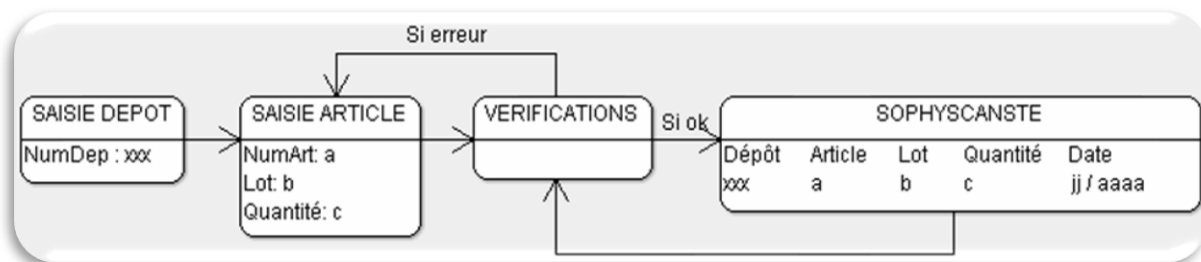
4.2 Gestion retour de stérilisation

4.2.a Le principe

Lorsque le magasinier reçoit ses produits en retour de stérilisation, il va, plutôt que de les cocher sur sa liste, les flasher un par un en indiquant leurs dépôt de destination. La référence, le lot, la quantité et la destination sont enregistrés dans la table « SOPHYSCANSTE » sur le serveur. Avant chaque enregistrement des vérifications sont mises en œuvre pour s'assurer que :

- Le dépôt de destination existe.
- L'article existe.
- L'article est bien au dépôt 002 – Désorption et en quantité suffisante.

Tout comme précédemment, si une erreur est levée, elle est indiquée à l'utilisateur via un message et un bip, voir schéma du protocole ci-dessous.



Cette configuration garantie donc une vérification automatique des articles saisie par le magasinier mais aussi un gain de temps considérable.

Lorsque le magasinier a effectivement flashé tous ses articles, il va pouvoir alimenter l'intégrateur de l'ERP* via la table « SOPHYSCANSTE » en transformant les lignes de la table en lignes exploitables par l'intégrateur. Il pourra alors lancer l'intégrateur depuis son poste de travail qui va organiser les mouvements des articles depuis le dépôt 002-Désorption jusqu'au dépôt de destination indiqué sur chacune des lignes. Cette fonction lui évite d'avoir à utiliser le fichier Excel pour organiser un à un les transferts de dépôt.

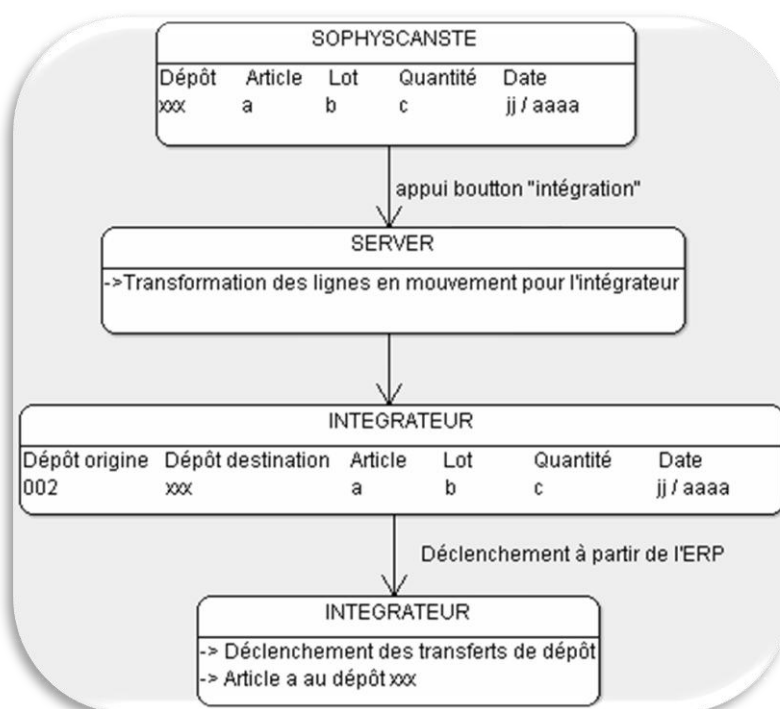


Figure 17 : Schéma du fonctionnement de la partie serveur de l'application « Gestion des retour de stérilisation ».

4.2.b Les fonctionnalités

De manière à pouvoir gérer la table SOPHYSCANSTE, plusieurs fonctionnalités sont implémentées, le tout sur 3 fenêtres.

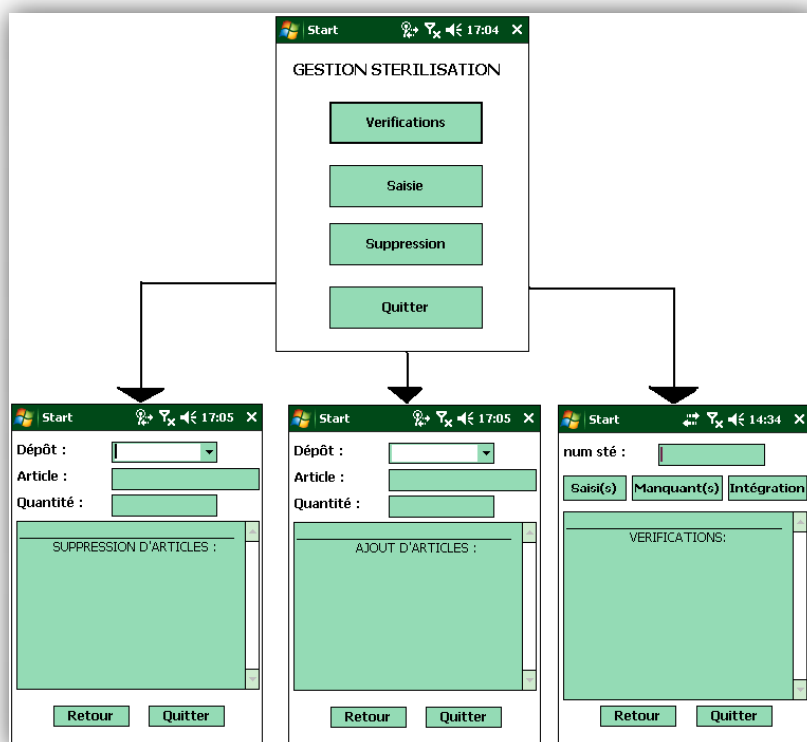


Figure 16 : Capture d'écran des différentes fenêtres de l'application « Gestion des retour de stérilisation ».

La fenêtre « Saisie »

Via cette fenêtre, le magasinier va pouvoir ajouter des articles dans la table. Il indique avant tout un dépôt vers lequel il souhaite envoyer les produits qu'il va saisir. Il peut alors les scanner un par un sans avoir à redéfinir son dépôt à chaque fois. Si les vérifications sont bonnes, un récapitulatif de son ajout lui est donné, l'appareil émet un bip favorable. Il peut bien sur redéfinir son dépôt de destination à tout moment.

La fenêtre « Suppression »

Comme pour l'application de gestion des commandes, l'utilisateur a la possibilité de supprimer un article de la table. Il indique le dépôt de destination de l'article, puis le scanne. Si l'article appartient bien à SOPHYSCANSTE en quantité suffisante, la ligne correspondante est supprimée dans le cas où la quantité tombe à 0, la quantité est décrémentée si elle reste supérieure à 0. Un message récapitulatif est affiché, un bip favorable est émis.

La fenêtre « Vérifications »

C'est à partir de cette fenêtre que l'on va pouvoir obtenir des informations quant aux articles enregistrés. En précisant un numéro de stérilisation (présent dans le code lot), il est possible d'obtenir, la liste des articles saisis pour cette stérilisation ou la liste des articles manquants. Le bouton intégration permet le remplissage de l'intégrateur via le traitement des lignes de la table sur le serveur (via une procédure stockée), qui est ensuite vidée.

4.3 La gestion des codes barres

Les deux applications nécessitent la possibilité de lire, décomposer et interpréter deux types de codes barres. Ces codes barres peuvent être saisis avec le scanner, ou à la main. Dans les deux cas, l'envoi d'un code barre se fait via une détection du retour chariot, caractère ascii <CR>, qui est rajouter en fin de lecture via le paramétrage du scanner, ou envoyer via l'appui de la touche « ENTER » sur le mobile.

4.3.a Les codes barres EAN128

Description

Il s'agit d'un code barre répondant à la norme GS1-128. Il est composé d'une série de champs (identifiant/ Valeur), la valeur étant de longueur prédéfinie ou variable. A Sophysa, seuls les champs dont les identifiants ci-dessous sont utilisés :

- 01 : Code EAN14, (nombre qui identifie de manière unique un produit ou un ensemble de produits référencés par un fabricant).
- 17 : Date de péremption.
- 30 : Quantité.
- 10 : Code lot.



Figure 17 : Code barre de type EAN128* .

Parmis ces champs, la quantité et le code lot sont à taille variable. A la lecture, on obtient la série de caractère suivante :

0113760124133407**17**150600**30**20**10**10195B/X0249

Gestion des champs à taille variable

La première difficulté quant au traitement de ce type de code est la détection d'une fin de champ à taille variable. Lorsque le scanner lis une fin de champ, il envoi un caractère GS (code ascii 29). Ce caractère n'est pas paramétrable et il ne possède aucune représentation symbolique, il n'est donc exploitable qu'au moment de sa lecture.

Le mobile étant paramétré pour émuler la saisie clavier lors d'une lecture, il est malgré tout possible de repérer, plutôt que le caractère GS lui-même, la suite de touche qui permet son ajout, à savoir « Menu » (correspond à « ALT »), « Numpad2 », « Numpad9 ».

Pour parvenir à sa détection, un mini Keylogger* est utilisé. Ce Keylogger* va intervenir sur les événements « appui touche » du champ de saisie du code barre. Aucun traitement ne doit être lancé avant que les touches « Menu », « Numpad2 » et « Numpad9 » soit récupéré dans l'ordre et consécutivement, afin qu'il n'y ait aucune interférence avec d'autres symboles spéciaux.

Pour se faire, un compteur est utilisé, il joue le rôle d'historique. Lorsque la touche « Menu » est détectée, le compteur passe à un. Si la touche suivante n'est pas « Numpad2 » il repasse à zéro, dans le cas contraire, il passe à 2. Si la touche d'après est bien « Numpad9 », le caractère est détecté, on ajoute alors un « # » au code barre, sinon, le compteur repasse à zéro. Voir algorithme ci-dessous.

```
//Contrôle des champs à taille variable
private void tb_CodeArt_KeyDown(object sender, KeyEventArgs e)
{
    //Détection touche "ALT"
    if (e.KeyCode == System.Windows.Forms.Keys.Menu)
    {
        this.cptChamp = 1;
    }
    //Détection touche "2"
    else if (this.cptChamp == 1 && e.KeyCode == System.Windows.Forms.Keys.NumPad2)
    {
        this.cptChamp = 2;
    }
    //Détection touche "9"
    else if (this.cptChamp == 2 && e.KeyCode == System.Windows.Forms.Keys.NumPad9)
    {
        //Ajout du caractère '#'
        this.tb_CodeArt.Text += "#";
        tb_CodeArt.SelectionStart = tb_CodeArt.TextLength;
        this.cptChamp = 0;
    }
    else
    {
        this.cptChamp = 0;
    }
}
```

Figure 18 : Vue de l'algorithme de détection du caractère GS.

Via cet algorithme on obtient en fin de lecture un code barre dont la fin des champs à taille variable est représentée par un « # ». Dans le cas du code ci-dessus le résultat est :

➤ 0113760124133407171506003020#1010195B/X0249#

Gestion du découpage

Il y a deux manières de saisir un code barre. La première consiste à taper son code barre à la main en veillant à ajouter un dièse à la fin des champs à taille variable. La seconde se fait via le scanner, paramétré pour ajouter un retour chariot à la fin du code barre. Dans les deux cas, le découpage du code intervient après détection du caractère de retour chariot « CR », via l'événement « appui de la touche « ENTER ».

L'algorithme de découpage est récursif. Partant du code barre formaté, il récupère un à un les identifiants pour en extraire la valeur. La longueur de la sous chaîne exprimant la valeur est déterminée par les identifiants eux même pour les champs dont la taille est fixe, par le « # » suivant dans le cas contraire. Lorsqu'un identifiant et sa valeur ont été récupérés, on procède au même traitement avec la chaîne restante. Si un identifiant est non valide, une erreur est levée, l'algorithme s'arrête. Si la sous chaîne restante est nulle, le découpage s'est bien déroulé, l'algorithme s'arrête. Voir [Image 6](#) : Algorithme de lecture des codes barre EAN128*.

4.3.b Les codes barres Sophysa

Description

Les codes barre Sophysa n'utilisent aucun identifiant, ils se construisent de manière fixe. Ils fournissent 3 informations visuelles, la référence de l'article, le code lot et la quantité. Parmi ces trois champs, seul deux sont codés, la référence et le lot. A la lecture, Les deux champs sont séparé par ':'.



Figure 18 : Code barre Sophysa.

A la lecture du code ci-dessus, le scanner renvoie :

➤ C855 :111970C/Y0248

Traitement

Afin de différencier les deux codes et pouvoir lancer l'algorithme de traitement qui leur est propre, on recherche avant tout le caractère ':'. Ce caractère n'est pas utilisé dans les code de type EAN128*, il permet donc d'identifier sur quel type de code barre le

traitement doit se faire. Dans le cas où ce caractère séparateur viendrait à changer, il est modifiable dans le fichier de configuration des applications (voir plus bas).

Pour récupérer les valeurs des deux champs, il suffit de découper la chaîne en deux sous-chaînes pour les stocker. La quantité n'étant pas codée, elle sera par défaut à 1, excepté si le magasinier la redéfinit avant de flasher son article.

4.4 La récupération des données

La communication avec le serveur occupe une place primordiale sur le fonctionnement des deux applications. Via du code exemple, le protocole connexion avec le serveur et une requête de sélection des articles seront détaillés ici. La mise en place d'un fichier de configuration, indispensable à la pérennité de ce travail sera le second point abordé.

4.4.a La communication avec le serveur

La communication se fait sur un compte utilisateur avec des droits restreints nommé « Consult ». L'API « Data.SqlClient » garantit la possibilité d'établir une connexion avec une chaîne de connexion dans laquelle on va fournir, l'adresse IP du serveur de base de données, la base de données à utiliser et le compte de connexion. L'adresse IP est identifiée par la « Data Source », la base de données par le « Catalog » et le compte par « User ID », voir figure 19 ci-dessous.

```
string connectionString = "Data Source=130.0.0.13;Initial Catalog=PYRA;User ID=Consult";
SqlConnection cn = new SqlConnection(connectionString);
try
{
    cn.Open();
}
catch (SqlException ex)
{
    for (int i = 0; i < ex.Errors.Count; i++)
    {
        this.tb_Console.Text += "\r\n Error #" + i + ex.Errors[i].Message;
    }
    Beep(0);
    this.VideChamps();
}
```

Figure 19 : Exemple de connexion à la base de données PYRA.

Les deux applications ont été programmées pour effectuer une connexion au démarrage des fenêtres d'ajout, de suppression et de vérification. De cette manière on s'assure, avant tout traitement, d'avoir une connexion opérationnelle.

Le compte « Consult » utilisé est paramétré pour n'autoriser que des actions de sélection, d'update ou d'insert.

De plus, On s'assure à chaque connexion que l'action s'est bien déroulée, dans le cas contraire l'exception capturée est affichée et bip d'erreur est envoyé.

4.4.b Les requêtes

A chaque fois qu'un article est scanné, plusieurs vérifications sont effectuées. Chacune de ces vérifications fait l'objet d'une requête. Mais ces requêtes diffèrent lorsque l'on doit agir sur un BL, une commande ou un bordereau de mise en dépôt. Il faut, dans tous les cas, toujours veiller à sécuriser les requêtes effectuées et se prévenir de l'injection SQL*.

Exemple du BL

La base de données PYRA comprend environ 1000 tables. Parmi ces tables seules une vingtaine sont utilisées. En prenant l'exemple du bon de livraison numéro XXXXXXXX, la requête de sélection des articles avec référence, lot et quantité est :

```
SELECT ARTIC.C_ARTIC, LOT.C_LOT, BLVLG.QTE_US
FROM BLVLG, ARTIC, BLVET, DEPMV, DEPMVD, LOT
WHERE BLVLG.T_ARTIC=ARTIC.T_ARTIC --1
AND BLVLG.T_DOCET=BLVET.T_DOCET --2
AND BLVLG.T_DOCLG=DEPMV.T_DOCLG --3
AND DEPMV.T_DEPMV=DEPMVD.T_DEPMV --4
AND DEPMVD.T_LOT=LOT.T_LOT --5
AND BLVLG.C_TYPAR='DIV' --6
AND BLVET.NO_DOCET=XXXXXXX --7
```

Cette requête se fait sur 6 tables qu'il faut lier une à une par jointure. Concrètement, le traitement effectué par cette requête est :

- 1 : Récupération des informations sur l'article en liant la ligne du BL avec la table ARTIC.
- 2 : La liaison entre le document BLVET et ses lignes BLVLG en utilisant le numéro système du document, T_DOCET.
- 3 : La liaison de chaque ligne stockée dans BLVLG avec la table des mouvements de stock DEPMV.
- 4 : La liaison des mouvements DEPMV avec leur détail DEPMVD
- 5 : La liaison des détails de mouvement avec la table LOT, qui permet une récupération du code lot de l'article.
- 6 : Restriction des lignes à sélectionner aux lignes de type article, de manière à ne pas prendre les commentaires et les informations complémentaires.
- 7 : Restriction aux informations ne concernant que le document 'XXXXXXX'.

La sécurisation

L'ensemble des requêtes utilisées sont de type paramétrées. Ce paramétrage permet d'attribuer aux variables un type et une taille. De cette manière les saisies utilisateur ne sont pas envoyées directement au serveur, elle sont encadrées et contrôlées. Si le type ou la taille ne correspond pas, une exception est capturée, un message est envoyé.

Dans le cas de la requête vu ci-dessus, la mise en place du paramétrage sur la variable numéro de BL se fait via le code suivant :

```
SqlParameter ParamNumBl = new SqlParameter("@ParamNumBl", SqlDbType.Int, 8 );  
ParamNumBl.Value = ParamNumBl.Value = Int32.Parse(this.tb_NumRef.Text);
```

Le paramètre @ParamNumBl est ajouté à la requête et il est utilisé à la place de la saisie utilisateur. Sa taille est restreinte à 8 octet et, le type à entier.

4.5 Le fichier de configuration

Afin d'assurer la pérennité des applications, il est indispensable de pouvoir permettre à l'administrateur de gérer des paramètres de configuration. Pour chacune des applications, il existe donc un fichier XML* lui permettant de gérer certaines valeurs.

Les paramètres

On trouve 6 paramètres potentiellement modifiables :

- L'adresse IP du server de base de données.
- La base de données à utiliser.
- Le compte à utiliser pour la connexion.
- Le champ séparateur des codes barre Sophysa (actuellement ':', voir plus haut).
- Les chemins d'accès aux sons utilisés, le bip 'ok' et le bip 'erreur'.

L'implémentation

L'ajout de cette fonctionnalité se fait via deux éléments, un fichier XML*, config.xml, et une classe ConfigurationManager, permettant d'aller agir sur ce fichier.

Le fichier config est formaté pour fournir un ensemble de couple (clé, valeur), il se présente ainsi :

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>

    <add key="Ip"      value="130.0.0.13"/>
    <add key="Bd"      value="PYRA"/>
    <add key="User"     value="consult"/>
    <add key="SeparatorCodeSophysa" value=":"/>
    <add key="PathBeepOk" value="SmartDeviceProject3/beep-ok.wav"/>
    <add key="PathBeepError" value="SmartDeviceProject3/beep-error.wav"/>
    <!-- ETC... -->

  </appSettings>
</configuration>

```

Figure 20 : Fichier de configuration config.xml.

La classe ConfigurationManager permet d'aller récupérer une valeur sur ce fichier, en fournissant une clé (voir [Image 7 : Classe ConfigurationManager](#)). Deux fonctions sont utilisées pour mettre en place ce système.

La première consiste à aller chercher le fichier .config pour l'ouvrir en lecture via la classe FileStream de la bibliothèque system.IO. Sur ce fichier, un lecteur XML* est ouvert, il va permettre de naviguer sur les nœuds du fichier, en les parcourant un à un. Si le nœud visité est un nœud de type « add » et que la clé correspond au paramètre d'appel, la valeur est retournée. Cette valeur est récupérée et convertie selon le type qui est indiqué à l'appel du ConfigurationManager, voir ci-dessous.

```
string ip = ConfigurationManager.GetKey<string>("Ip");
```

3.7 Ce qu'il reste à faire

Etant donné que ce projet est programmé sur 16 semaines, rien n'indique un retard sur l'avancement du travail. Il reste malgré tout quelques étapes à mettre en œuvre.

Tout d'abord, l'application de gestion du retour de stérilisation nécessite l'ajout d'une procédure stockée sur le server permettant la transformation des lignes de la table SOPHYSCANSTE en lignes exploitables par l'intégrateur. Ces lignes seront exportées au format csv et formatées selon des contraintes à définir. Il faudra ensuite mettre en place une formation et un test utilisateur.

Pour continuer, le développement de l'application de gestion de la préparation des commandes étant arrivé à son terme, une formation du magasinier et une phase de test sur une semaine est prévu. Des corrections ou des ajouts pourrons intervenir si nécessaires.

Finalement, la troisième application doit être implémentée afin d'offrir une consultation des dépôts. L'analyse des besoins et des objectifs étant dors et déjà effectuée, une semaine de développement est planifiée. Une formation et une période d'essais devront suivre.

CONCLUSION

Ces 12 premières semaines de stage au sein du service informatique de Sophysa ont permis un avancement correct vis-à-vis du calendrier prévisionnel. Peu précis au départ, la mise en place d'un cahier des charges détaillé a pu révéler l'ensemble des besoins vis-à-vis des trois processus étudiés. Les objectifs et contraintes qui en ont été retirés garantissent la mise en place de solutions fiables et durables.

Le réseau Wifi est désormais fonctionnel, l'usine dispose d'une couverture contenue à l'enceinte de ses murs et sécurisée par un cryptage WPA. Les appareils mobiles sont configurés et permettent une connexion stable dans l'ensemble des salles à couvrir. D'un point de vue administration, un accès aux bornes sur adresse IP fixe est opérationnel, les mobiles sont accessibles à distance via un serveur FTP.

En ce qui concerne les applications à fournir sur les appareils, leur état d'avancement est variable. Développées en c sharp, faute de pilote JDBC*, elles permettent la lecture et le traitement des codes barres spécifiques à Sophysa et EAN128*. La communication avec le serveur de base de données est assurée et un fichier de configuration est à la disposition de l'administrateur afin de garantir leur pérennité.

L'application de gestion des commandes permet d'ores et déjà le contrôle des produits emballés et leur référencement. Elle garantit une sécurité et une traçabilité des articles expédiés sur tous les types de commandes via l'utilisation d'une table d'enregistrement. Elle nécessite une phase de test utilisateur prévue dans les plus brefs délais. L'application de gestion de la réception des produits en stérilisation implémente elle aussi un système de contrôle et de référencement des produits. Pour la compléter, un traitement serveur devra permettre le transfert des articles de façon automatique en utilisant la table de référencement préalablement remplie par le magasinier. La dernière application n'a pas été développée bien que ces objectifs soient définis. Une période d'une semaine est estimée pour sa construction.

D'un point de vue personnel ce stage aura été l'occasion de mettre à l'épreuve l'ensemble des connaissances que j'ai pu obtenir durant mes 6 semestres de licence. Bien qu'ayant rencontré de nombreuses difficultés, j'ai pu constater que ma formation très généraliste me permettait une grande adaptabilité vis-à-vis des technologies informatiques. Par ailleurs, le sujet qui m'a été confié s'inscrit parfaitement dans la tendance actuelle, le développement sur appareil mobile. Ne faisant pas parti de ma formation initiale je pense que ce travail constitue un plus non négligeable dans la réalisation de mon projet professionnel.

Développement Java sur appareils mobiles

[1] http://www.blueboard.com/javame/pdf/guide_sip_gsg_javame_cdc_r1a.pdf

Getting started with Java™ CDC development P990, consulté le 21/03/2010.

[2] <http://java.developpez.com/faq/javame/?page=Definitions>,

JavaMe, consulté le 22/03/2010.

[3] <http://www.slideshare.net/andreasjakl/java-me-01-overview>

Java ME Overview, consulté le 23/03/2010.

[4] <http://www.slideshare.net/ter216/coursJ2ME>

cours J2ME, consulté le 25/03/2010.

[5] <http://www.vikdavid.com/mobile/>

Présentation java mobile, consulté le 28/03/2010.

[6] <http://blog.helyx.org/wp-content/BiendbuteravecJavapourPocketPC.pdf>

Bien débuter avec Java pour PocketPC, consulté le 30/03/2010.

[7] <http://www.cs.umanitoba.ca/~eclipse/1-Install.pdf>

Eclipse Install, consulté le 04/04/2010.

[8] <http://www.eclipse.org/swt/faq.php#pocketpclibrary>

Pocket pc library, consulté le 04/04/2010.

[9] <http://www.nsicom.com/Default.aspx?tabid=138>

CrEme™ - The Java™ Enabler for Windows® CE, consulté le 06/04/2010.

[10] <http://www.commentcamarche.net/contents/jdbc/jdbcacces.php3>

Pilote JDBC, consulté le 03/04/2010.

[11] <http://www-01.ibm.com/software/wireless/wece/features.html>

WebSphere Everyplace Custom Environment, consulté le 06/04/2010.

Codes barres GS1 128

[12] http://www.gs1.fr/gs1_fr/assistance_technique/les_codes_a_barres_gs1/le_code_a_barres_gs1_128, Code barres GS1 128, consulté le 06/04/2010.

C sharp

[13] [http://msdn.microsoft.com/fr-fr/vcsharp/msdn.coachcsharp.laboratoire-coach c sharp](http://msdn.microsoft.com/fr-fr/vcsharp/msdn.coachcsharp.laboratoire-coach-c-sharp), consulté le 07/04/2010.

[14] <http://nicopyright.developpez.com/tutoriel/vc2005/configurationsectioncsharp/#LbEginWithvisualcsharp>, [Tutoriel : Travailler avec les fichiers de configuration en C#](#), consulté le 10/05/2010.

[15] <http://www.asp-php.net/tutorial/pocket/windows-mobile-premier-pas.php> [Windows Mobile 5.0 & Visual Studio 2005](#), consulté le 08/04/2010.

[16] <http://www.packtpub.com/article/pocket-pc-application-visual-studio-2005> [A Simple Pocket PC Application using Visual Studio 2005](#), consulté le 08/04/2010.

[17] [http://msdn.microsoft.com/fr-fr/library/67ef8sbd\(v=vs.80\).aspx](http://msdn.microsoft.com/fr-fr/library/67ef8sbd(v=vs.80).aspx) [Guide de programmation C#](#), consulté le 06/04/2010.

[18] <http://www.dot-netfrance.com/Documents/Mobile/Approche%20sur%20les%20applications%20Windows%20Mobile.pdf>, [Applications sur Windows mobile](#), consulté le 05/04/2010.

[19] <http://blogs.codes-sources.com/pi-r/archive/2010/09/27/d-eacute-veloppement-windows-mobile-6-x-comment-interagir-avec-un-fichier-de-configuration-app-config.aspx>, [Développement Windows Mobile 6.x : Comment interagir avec un fichier de configuration app.config ?](#), consulté le 11/05/2010.

[20] http://www.codeproject.com/KB/database/sql_in_csharp.aspx [Beginners guide to accessing SQL Server through C#](#), consulté le 10/04/2010.

WebService

[21] <http://www.commentcamarche.net/contents/web-services/soa-architecture-orientee-services.php3>, [SOA - Architecture Orientée Service](#), consulté le 21/03/2010.

[22] <http://www.journaldunet.com/developpeur/tutoriel/php/040513-php-nexen-nusoap3.shtml>, [Créer des services Web avec SOAP et PHP](#), consulté le 22/03/2010.

Serveur FTP

[23] <http://www.mochasoft.dk>, [Mocha Ftp Server](#), consulté le 07/05/2010.

GLOSSAIRE

[1] AES: L'AES (Advanced Encryption Standard) est, comme son nom l'indique, un standard de cryptage symétrique destiné à remplacer le DES (Data Encryptions Standard) qui est devenu trop faible au regard des attaques actuelles.

[2] Baie de brassage: Barrette métallique ou ensemble de barrettes métalliques à laquelle sont accrochées les arrivées des câbles d'un réseau local (ou téléphonique).

[3] Client léger: Un client léger est un ordinateur qui, dans une architecture client-serveur, n'a presque pas de logique d'application. Il dépend donc surtout du serveur central pour le traitement.

[4] DHCP: Dynamic Host Configuration Protocol (DHCP) est un terme anglais désignant un protocole réseau dont le rôle est d'assurer la configuration automatique des paramètres IP d'une station, notamment en lui affectant automatiquement une adresse IP et un masque de sous-réseau.

[5] EAN128: Egalement connu sous le nom de "UCC-128" est un standard international de codification d'unités logistiques pour faciliter les échanges de données dans la relation "fournisseur-distributeur".

[6] Environnement Citrix: Citrix XenDesktop est une solution de virtualisation de postes qui transforme les postes de travail Windows en service à la demande. Que l'utilisateur soit un utilisateur standard, un cadre aux besoins plus complexes ou un travailleur mobile (utilisant une tablette tactile, un Smartphone, un ordinateur portable ou un client léger), XenDesktop lui délivre rapidement et en toute sécurité des applications individuelles ou des postes de travail complet.

[7] ERP: Entreprise ressource Planning, produit logiciel qui permet de gérer l'ensemble des processus d'une entreprise, en intégrant l'ensemble des fonctions de cette dernière comme la gestion des ressources humaines, la gestion comptable et financière, l'aide à la décision, mais aussi la vente, la distribution, l'approvisionnement, le commerce électronique. Les différents modules et fonctions sont traités autour d'une base de données unique.

[8] Firmware: Sur un appareil électronique (baladeur MP3 ou lecteur DivX, par exemple), le firmware est le logiciel interne qui permet à l'appareil de fonctionner. Il gère différentes fonctions : affichage, navigation dans les menus, mais aussi décodage ou encodage.

[9] Framework: En programmation informatique, un framework est un kit de composants logiciels structurels, qui servent à créer les fondations ainsi que les grandes lignes de tout ou partie d'un logiciel (architecture). En programmation orientée objet un framework est typiquement composé de classes mères qui seront dérivées et étendues par héritage en fonction des besoins spécifiques à chaque logiciel qui utilise le framework.

[10] FTP: Le File Transfer Protocol (protocole de transfert de fichiers), ou FTP, est un protocole de communication destiné à l'échange informatique de fichiers sur un réseau TCP/IP. Il permet, depuis un ordinateur, de copier des fichiers vers un autre ordinateur du réseau, d'alimenter un site web, ou encore de supprimer ou de modifier des fichiers sur cet ordinateur.

[11] Hydrocéphalie: Augmentation du volume de liquide céphalo-rachidien, qui provoque une dilatation des ventricules cérébraux, une augmentation du volume du crâne et une insuffisance du développement intellectuel.

[12] IDE: L'IDE (Integrated Development Environment) est une interface qui permet de développer, compiler et exécuter un programme dans un langage donné.

[13] Injection SQL: Une injection SQL est un type d'exploitation d'une faille de sécurité d'une application interagissant avec une base de données, en injectant une requête SQL non prévue par le système et pouvant compromettre sa sécurité.

[14] JDBC: JDBC est une API Java (ensemble de classes et d'interfaces défini par SUN et les acteurs du domaine des BD) permettant d'accéder aux bases de données à l'aide du langage Java via des requêtes SQL.

[15] Keylogger: En informatique, un enregistreur de frappe (en anglais, keylogger) est un équipement ou un logiciel espion qui enregistre les touches frappées sur le clavier d'un ordinateur sous certaines conditions et les transmet via un réseau ou via des ondes électromagnétiques.

[16] Lotus: Lotus Notes est un logiciel de travail collaboratif, utilisé dans des entreprises ou des administrations pour gérer les projets, les courriels et les échanges d'informations autour d'une base commune.

[17] Microsoft .NET: Microsoft .NET est le nom d'un ensemble de produits et de technologies de l'entreprise Microsoft dont la plupart dépendent du framework .NET, un composant du système d'exploitation Windows constituant un équivalent de machine virtuelle.

[18] SSID: SSID est l'acronyme de Service Set Identifier. C'est un nom identifiant un réseau sans fil selon la norme IEEE 802.11. Ce nom peut être composé jusqu'à 32 caractères.

[19] SwingMobility: SSII, l'entreprise développe et édite des solutions logicielles de mobilité pour les populations commerciales et techniques itinérantes.

[20] TKIP: (Anglais : Temporal Key Integrity Protocol), protocole de cryptage utilisé pour le WPA (Wi-Fi Protected Access). Le WPA est utilisé dans le cadre de réseaux WiFi pour corriger la déficience du standard WEP. TKIP a été développé pour remplacer le WEP sans que des modifications matérielles ne soient nécessaires.

[21] WPA-PSK: Wi-Fi Protected Access (WPA et WPA2) est un mécanisme pour sécuriser les réseaux sans-fil de type Wi-Fi. Il a été créé en réponse aux nombreuses et sévères faiblesses que des chercheurs ont trouvées dans le mécanisme précédent, le WEP. WPA respecte la majorité de la norme IEEE 802.11i et a été prévu comme une solution intermédiaire pour remplacer le WEP en attendant que la norme 802.11i soit terminée.

[22] XML: Extended Markup Language. Langage de description des documents qui utilise des balises, permet l'utilisation de balises personnalisées et permet l'échange des données.

ANNEXE

Image 1 : Bon de livraison

SOPHYSA TEL : (33) 01 69 35 35 00
FAX : (33) 01 69 35 36 90

NUMERO DU BL

BON DE LIVRAISON N° **11 111 179** du **16/05/2011**


Adresse de livraison
 HOPITAUX DE LA TIMONE
 SERVICE PHARMACIE
 1ER ETAGE
 264 RUE SAINT-PIERRE
 F - 13385 MARSEILLE CEDEX 5

Cde client : 38113	du 16/05/2011	Code client : 11301
Suivie par : Sophie ALAZARD		N° intracommun. client : FR 90 261 300 081
Nos réf. : 11121081	du 16/05/2011	Conditions de livraison : Franco de port
		Mode d'expédition : GLS

Référence	Désignation	Qté en Cde	Qté livrée	Serie/Lot	Qté S/L	Reste à livrer
533115	CELLUSTRIP RADIO-OPAQUE + FIL 15 x 15 mm	200	200	101209	200	0

NOUS VOUS DEMANDONS DE BIEN VOULOIR NOUS ENVOYER CE BON DE LIVRAISON DATE ET SIGNE DANS L'ENVELOPPE CI-JOINTE. MERCI

Image 2 : Bordereau de mise en dépôt



SOPHYSA

Tel : (33) 01 69 35 35 00
Fax : (33) 01 69 35 36 90

DATE DU BORDEREAU

BORDEREAU DE MISE EN DEPOT DES PRODUITS au 12/05/2011

PHOENIX - THAILANDE

Adresse du dépôt

Nous vous prions de bien vouloir nous retourner le document original daté et signé dans les plus brefs délais.

Référence	Désignation	Qté	Sens	Serie/Lot	Date exp.
BO19-10	CATH.VENTRI.DROIT STANDARD	55	entrée	110610/Y0086	01/02/2016
BO19-10	CATH.VENTRI.DROIT STANDARD	1	entrée	110887A/Y0132	01/03/2016
BO19-10	CATH.VENTRI.DROIT STANDARD	4	entrée	110887B/Y0132	01/03/2016

REFERENCE
ARTICLE

QUANTITE DU LOT

NUMERO DE LOT

ATTENTION

Conditions particulières de stockage :
A l'abri de la poussière et de l'humidité, à température ambiante.

Maintien en stock :
Les produits en dépôt doivent avoir une péremption supérieure à 12 mois. En deçà de ce délai, les produits en dépôt doivent être retournés à Sophysa.

Sortie de stock (en vue de l'utilisation immédiate du produit) :
Les produits en dépôt doivent être impérativement utilisés par ordre croissant de date de péremption.

Lu et approuvé

Nom :

Service :


Date :

Signature :

Image 3 : Détail d'un dépôt

Etat détaillé des produits en stock dans le dépôt 608 (SIAD Healthcare ITALIE)				
Edition du 16/05/2011				
Article		Qté totale	Code Lot	Qté lot
RE-1021	RES. LAT. FD PLAT PT MD / CA1	2,00	103070B/X0402	2,00
SM8B-2010	KIT SOPHY SM8B PRECONN/B0	2,00	131863/X0211	1,00
			131879/X0211	1,00
TUG-65	GAINE UU P.TUNNELLEUR 65	50,00	104590/X0596	25,00
			110425A/Y0098	25,00

Image 4 : Commande



SOPHYSA

PHONE : (33) 01 69 35 35 00
FAX : (33) 01 69 35 36 90

CONFIRMATION OF ORDER
dated 24/03/2011

NUMERO DE
LA COMMANDE

N° 11 120 652

Ship to
SANIFICO SA
AV 8 DE OCTUBRE 2573

UY - 11600 MONTEVIDEO

Purchase order : SPO17189 dated 23/03/11

Your contact : Sophie ALAZARD

Requested delivery date : 24/03/2011

Customer N° : 9UY02

Terms of delivery : EXW

Customer VAT number :

We thank you for your order. In the absence of particular instruction from you, the order will be executed as specified hereafter.

Customer department

Ref N°	Description	Qty	Week of delivery	Unit price	Extended price
SM1-2010M	SOPHY SM1-M KIT PREATTACHED/BO19-10	3	12	133,00	399,00

REFERENCE

QUANTITE

%VAT	Base	VAT Amount	Sale Amount	VAT amount	Freight	Amount to pay
0,00%	399,00	0,00	399,00	0,00		EUR 399,00

Terms of payment : Comptant

Payment by : Bank transfer

SophysaENR 915-007

1 / 1

HEAD OFFICE: 22,RUE JEAN ROSTAND - PARC CLUB ORSAY UNIVERSITE - 91893 ORSAY CEDEX FRANCE
S.A. AU CAPITAL DE 500.000 € - R.C.S. EVRY B 306 979 584 - V.A.T. N° : FR 05 306979584

Image 5 : Fichier EXCEL des produits en stérilisation

SYNT	REF	REFER	REFSOPH	LOTPYRA	SERIE	LOTSTE	QTE	PER	(17)	PYRA	ANBOX	valid.etiq
BO50	BO50	BO50	C506	100083/X0033	100083	X0033	5	2015/01	150100	B3/X0033		
B905S	B905S	B905S	C478	100081A/X0033	100081A	X0033	1	2015/01	150100	A/X0033		
B905S	B905S	B905S	C478	100081B/X0033	100081B	X0033	40	2015/01	150100	B/X0033		
B905S	B905S-10	B905S-10	C442	100082B/X0033	100082B	X0033	10	2015/01	150100	B/X0033		
PLO6	PLO6	PLO6	C482	100084A/X0033	100084A	X0033	6	2015/01	150100	A/X0033		
PLO6	PLO6	PLO6	C482	100084B/X0033	100084B	X0033	9	2015/01	150100	B/X0033		
PRO6	PRO6	PRO6	C464	100085A/X0033	100085A	X0033	3	2015/01	150100	A/X0033		
PRO6	PRO6	PRO6	C464	100085B/X0033	100085B	X0033	12	2015/01	150100	B/X0033		
Pastilles CV290 sur boites de regroupement seulement												
Total Produits:							86					
IFA777 ind. 54												
place utilisée												
place restante							960					
Remplir uniquement les cellules non colorées.												
Co-X0X0 ou TU-9321: Quantité = Nombre d'étuis de 5 ou10							(1 étui = 5 COXX ou 10 TU-9321)					
CO-3010 : étui individuel												
Fonctions/utilisation des colonnes :												
- SYNTHESE : pour aperçu des informations réf, série et lotsté dans Codsof6. CETTE COLONNE EST A COPIER												
- LOTPYRA : pour macro (n° série/lotsté) et édition étiquettes (n° série/lotsté)												
- @SERIALQTY pour information sur Codsof6 du nb d'étiquettes à imprimer (cette colonne est masquée)												
- REF : pour édition des étiquettes												
- REFER : pour macro d'entrée des produits au dépôt "02" (désorption)												
- REFSOPH pour macro de sortie des produits du dépôt "014" (prdts à expédier)												
- SERIE : pour saisie du n° de série du lot												
- LOTSTE : pour saisie du n° de stérilisation												
- QTE : qté par référence et par lot												
- PER : après saisie de la date de per ds l'onglet "Indice", date de péremption mise à jour automatiquement. Utilisée dans étiquette												
Enregistrement du contrôle à réception :							Nb de caisses:					
Pastilles indic de sté (C/NC):							Date:					
Qté par réf (C/NC):							Opérateur :					
Etat des caisses/étuis/boites (C/NC):							Emargement:					

Image 6 : Algorithme de lecture des codes barre EAN128

```

private int LectureCodeBarreEan128(String code)
{
    if (code.Length > 1)
    {
        String Aim = code.Substring(0, 2);
        switch (Aim)
        {
            case "01": this.tb_CodeArt.Text = code.Substring(2, 14);
                       LectureCodeBarreEan128(code.Substring(16, code.Length - 16));
                       break;
            case "17": this.s_DatePer = code.Substring(2, 6);
                       LectureCodeBarreEan128(code.Substring(8, code.Length - 8));
                       break;
            case "30": if (this.tb_Quantite.Text == "")
                       {
                           this.tb_Quantite.Text = code.Substring(2, code.IndexOf("#") - 2);
                       }
                       LectureCodeBarreEan128(code.Substring(code.IndexOf("#") + 1, code.Length - (code.IndexOf("#") + 1)));
                       break;
            case "10": this.s_Lot = code.Substring(2, code.IndexOf("#") - 2);
                       LectureCodeBarreEan128(code.Substring(code.IndexOf("#") + 1, code.Length - (code.IndexOf("#") + 1)));
                       break;
            default: this.tb_Console.Text += "\r\n" + "identifiant inconnu : " + code; LectureCodeBarreEan128("");
                     this.VideChamps(); Beep(0); return 0;
        }
    }
    else if (code.Length == 1)
    {
        Beep(0);
        this.tb_Console.Text += "\r\n" + "symbole inconnu : " + code; this.VideChamps(); return 0;
    }
    return 1;
}

```

Image 7 : Classe ConfigurationManager

```
class ConfigurationManager
{
    private static XmlTextReader GetReaderConfigFile()
    {
        // Récupération du chemin de l'exécutable:
        string appPath = System.IO.Path.GetFullPath(System.Reflection.
            Assembly.GetExecutingAssembly().GetName().CodeBase) + ".config";
        // Ouverture du fichier de config en lecture:
        FileStream file = new FileStream(appPath, FileMode.Open,
            FileAccess.Read, FileShare.Read);

        return new XmlTextReader(file);
    }

    public static T GetKey<T>(string key)
    {
        XmlTextReader xmlRead = null;
        try
        {
            // Récupère le flux XML du fichier de config:
            xmlRead = GetReaderConfigFile();
            // Recherche la clef dans le fichier de config:
            while (xmlRead.Read())
            {
                XmlNodeType nodeType = xmlRead.NodeType;

                if (nodeType == XmlNodeType.Element)
                {
                    if (xmlRead.Name.Equals("add") &&
                        xmlRead.GetAttribute("key").Equals(key))
                    {
                        // Retourne la valeur de la clef sous le bon type:
                        return (T) Convert.ChangeType(xmlRead.GetAttribute("value"),
                            typeof(T), null);
                    }
                }
            }
        }
        catch (XmlException ex)
        {
            throw ex;
        }
        finally
        {
            if (xmlRead != null)
                xmlRead.Close();
        }

        return default(T);
    }
}
```

RESUME

Ce rapport de stage s'inscrit dans le cadre de l'obtention de ma troisième année de licence informatique à l'université de Franche-Comté. Il fait état de l'avancement du projet qui m'a été confié durant mes 12 premières semaines de travail au sein site de production de l'entreprise Sophysa situé à Besançon. Cette PME active dans le domaine des matériaux neurochirurgicaux voit actuellement un accroissement conséquent de la quantité de produits à gérer. Poussé par un ralentissement et un manque de sécurisation de la chaîne de production, notamment au niveau de la gestion de l'expédition et la réception des produits, le projet vise à informatiser trois processus à l'aide d'appareils mobiles lecteurs de codes barre fonctionnant sous Windows mobile 6.1. A disposition des deux magasiniers, les applications à développer doivent encadrer la préparation des commandes, la réception des produits en retour de stérilisation et la consultation des dépôts. Il est relaté ici les différentes étapes qui ont été nécessaires à la réalisation de la première partie de ce projet, depuis l'établissement du cahier des charges et la mise en place du réseau sans fil, jusqu'au développement des applications nécessaire à la correction de deux des trois processus étudiés.

Mots-clés : Développement mobile, réseau wifi, codes barre, Windows mobile 6.1, gestion de stocks.

SUMMARY

This report takes place on the third year of my bachelor program at the University of Franche-Comté and it aim to explain the state of progress of my project during the twelve first weeks of my internship at the company Sophysa. This SME, which organize the production of neurosurgical materials has seen a huge increase of the quantity of product to manage. With the desire of keeping efficiency and security on its production process, particularly with the exporting and importing service, the project aims to computerize three processes with wireless and bar-code reader devices working on Windows Mobil 6.1. The three applications to develop must help the two warehousemen on the preparation of the orders, the control of the sterilized products return and the stock's consultation. You will find here the explanation of the project realization since the specifications document's construction, the wireless network deployment and until the development of the applications required to correct two of the three processes.

Keywords: Mobil development, wireless network, bar-code, Windows Mobil 6.1, stock control.