

Licence Sciences Mathématique et Informatique

Mémoire de Projet de Fin d'Études



Application de Gestion des Stocks Sous Android
MyStock

Soutenu par: KHOUNA RIDA

KILANI HASSAN

Encadrant académique: Mr.Merzak Abdelaziz

Jury:

Mr .Nouh Said

Table des matières

<i>Table des matières.....</i>	2
<i>Table des tableaux.....</i>	3
<i>Table des figures.....</i>	4
<i>Remerciements.....</i>	6
<i>Introduction générale :</i>	7
<i>Chapitre 1: Cadre général</i>	8
1. Introduction :	8
2. Présentation du projet :.....	8
3. Méthodologie et formalismes adoptés :	8
3.1 Introduction :	8
3.2 Les processus unifiés :	9
4. Conclusion :	9
<i>Chapitre 2 : Etude du Projet</i>	10
1. Introduction :	10
2. Étude de l'existant :	10
3. Les critères de choix du système Android :.....	12
3.1 Flexibilité de la plateforme :	12
3.2 Facilité du développement :	12
3.3 Croissance dans le marché des Smartphones :	13
4. Web Service SOAP :	14
4.1 Introduction :	14
4.2 Le développement des protocoles applicatifs :	14
4.3 Choix de SOAP :	15
5. Conclusion :	15
<i>Chapitre 3 : Spécification et analyse des Besoins.....</i>	16
1. Introduction :	16
2. Spécification des besoins :	16
2.1 Besoins fonctionnels :	16
2.2 Besoins non fonctionnels :	16
3. Analyse des besoins :	17
3.1 Identification des acteurs et des cas d'utilisations :	17
3.2 Diagramme de cas d'utilisation :	17
4. Conclusion :	27
<i>Chapitre 4 : Conception.....</i>	28
1. Introduction :	28
2. Diagrammes de séquences	28

3.	Diagrammes d'activités :	35
4.	Diagramme de classes :	40
5.	Modèle relationnel :	41
6.	Conclusion :	42
	<i>Chapitre 5 : Réalisation</i>	43
1.	Introduction :	43
2.	Architecture du système :	43
3.	Environnements de travail :	44
3.1	Environnement matériel :	44
3.2	Environnement logiciel :	44
4.	Implémentation du web service SOAP :	45
5.	Implémentation coté Android :	46
5.1	Choix de la version Android :	46
5.2	Les bibliothèques utilisées :	47
6.	Interfaces de l'application :	48
6.1	Mode Online :	48
6.2	Mode Offline :	62
7.	Les Options Ajoutées :	68
7.1	L'option de recherche dans la Barre de Menu :	68
7.2	L'option de l'Actualisation	69
7.3	Le menu d'accès rapide.....	70
7.4	L'option de changement de couleur	70
	<i>Conclusions et perspectives</i>	71
	<i>Bibliographie et Webographie</i>	72

Table des tableaux

Tableau 1: Scénario d'authentification	19
Tableau 2: Scénario de la Consultation de l'Inventaire	23
Tableau 3: Scénario de Recherche de l'Article	24
Tableau 4: Scénario de l'Impression de l'Article	25
Tableau 5: Scénario de l'Initialisation de la Base	26
Tableau 6: Ensemble des outils logiciels utilisés en phase de réalisation	45
Tableau 7: Répartition des versions d'API Android	46

Table des figures

Figure 1 - Répartition par OS des livraisons mondiales de Smartphones (millions d'unité)	13
Figure 2:Diagramme de cas d'utilisation global relatif au Magasinier hôtel	18
Figure 3: Diagramme de cas d'utilisation de la Consultation de l'Inventaire	20
Figure 4: Diagramme de cas d'utilisation de la Recherche de l'Article	24
Figure 5: Diagramme de cas d'utilisation de l'Impression du Dépôt.....	25
Figure 6: Diagramme de cas d'utilisation de l'Initialisation de la Base	26
Figure 7: Diagramme de séquence d'authentification	29
Figure 8: Diagramme de séquence de la Consultation de l'inventaire.....	30
Figure 9: Diagramme de séquence de l'Ajout de l'Article	31
Figure 10: Diagramme de séquence de la Modification de l'Article.....	32
Figure 11: Diagramme de séquence de la Recherche de l'Article	33
Figure 12: Diagramme de séquence de l'impression du Dépôt.....	34
Figure 13: Diagramme de séquence de l'Initialisation de la Base de données	35
Figure 14: Diagramme d'activité d'authentification.....	36
Figure 15: Diagramme d'activité de la Consultation de l'Inventaire	37
Figure 16: Diagramme d'activité de la Recherche de l'Article.....	38
Figure 17: Diagramme d'activité de l'Impression de l'Inventaire	39
Figure 18: Diagramme de classes	40
Figure 19: MLD	41
Figure 20: Architecture du système	43
Figure 21: Répartition des versions d'API Android	46
Figure 22: Interface d'accueil d'authentification	48
Figure 23: Message d'erreur lorsqu'il y a échec d'authentification	49
Figure 24: Interface d'accueil du responsable si les paramètres sont valides	49
Figure 25: Interface d'authentification du responsable	49
Figure 26: Interface de choix de type de Recherche	50
Figure 27: Interface de consultation des Articles avec l'option de Filtrage NON	51
Figure 28: Interface pour sélectionner le dépôt Où se trouve l'Article	51
Figure 29: Interface liste des Catégories	51
Figure 30: : Interface liste des Familles	52
Figure 31: Interface liste des Sous Familles	52
Figure 32: : Interface liste des Articles	52
Figure 33: Interface de consultation des Articles avec l'option de Filtrage OUI	53
Figure 34: Interface liste des Articles	53
Figure 35: Interface les infos de l'article en mode modification	54
Figure 36: Interface les infos de l'article	54
Figure 37: Interface les infos de l'article en mode modification	55
Figure 38: Interface liste des articles.....	55
Figure 39: Interface liste des articles.....	56
Figure 40: Interface de Modification des Articles en Mode Online.....	56

Figure 41: Interface liste des articles.....	57
Figure 42: Interface ajout de l'article	57
Figure 43: Interface ajout de l'article	58
Figure 44: Interface d'Ajout d'Articles en Mode Online	58
Figure 45: Interface Scanner article.....	59
Figure 46: Interface de Recherche d'Article	59
Figure 47: Interface de Recherche d'Article	59
Figure 48: Interface Inventaire Menu.....	60
Figure 49: Interface liste des dépôts	60
Figure 50: Interface d'Impression de Dépôt.....	60
Figure 51: Interface Menu principal	61
Figure 52: Interface de Déconnexion	61
Figure 53: Interface liste des articles.....	62
Figure 54: Interface des infos de l'article	62
Figure 55: Interface des infos de l'article	63
Figure 56: Interface liste des articles.....	63
Figure 57: Alerte Envoie de la modification	63
Figure 58: Alerte Envoie de la modification	64
Figure 59: Interface de Modification des Articles en Mode Offline	64
Figure 60: Interface liste des articles.....	65
Figure 61: Interface Ajout de l'article	65
Figure 62: Interface liste des articles.....	66
Figure 63: Alerte Envoie de la modification	66
Figure 64: Alerte Envoie de la modification	67
Figure 65: Interface envoie de mise à jour au serveur	67
Figure 66: Interface de l'option de recherche optimisé.....	68
Figure 67: Interface liste des articles.....	69
Figure 68: Interface mise à jour des données	69
Figure 69: Interface de l'option d'Accès rapide	70
Figure 70: Interface Liste depots avec l'option de couleur.....	70

Remerciements

Ce rapport est le fruit de trois mois de travail sur un projet qu'on a eu le plaisir de réaliser.

Avant d'entamer le rapport de cette expérience professionnelle, il nous paraît important de commencer ce document en exprimant notre gratitude envers Notre encadrant académique Monsieur Marzak Abdelaziz, qui nous a accompagné tout au long de cette expérience avec beaucoup de patience et de pédagogie.

C'est grâce à son aide, sa disponibilité et ses recommandations précieuses que nous avons pu porter à terme ce travail. Nous espérons qu'il trouvera dans ce mémoire le témoignage de notre profonde reconnaissance.

Finalement, nous tenons à remercier les membres du jury d'avoir bien voulu évaluer notre travail.

Introduction générale :

Le mode d'accès continu à l'information de façon fiable et flexible, à travers divers services, devient un nouveau paradigme de programmation et d'organisation des opérations.

Les applications web et mobiles se popularisent et intéressent de plus en plus les entreprises soucieuses de soigner leur image technologique et de satisfaire des exigences de performance. Le marché des Smartphones connaît donc un véritable essor dans lequel les acteurs habituels essaient de s'engouffrer. Google, ayant réalisé le potentiel de ce marché, a décidé de s'y introduire en rachetant une startup travaillant sur un système d'exploitation ouvert pour terminal mobile : **Android**.

Durant le stage du projet de fin d'études, il nous a été demandé de concevoir, développer et faire l'intégration d'une application Android, permettant de suivre à chaque instant tous les rapports, en outre, elle permet aux magasiniers de consulter l'inventaire de l'hôtel.

Ce rapport peut ainsi être subdivisé en cinq chapitres :

- Le premier chapitre «**Cadre du projet**» consiste à la présentation du cadre du projet, puis la planification du projet.
- Le second chapitre intitulé «**Etude du projet**» dont on va y faire une étude de l'existant, les critères de choix de système Android ainsi que les web services SOAP.
- Le troisième chapitre intitulé «**Spécification et analyse des besoins**» est composé d'une description des différents modules de l'application (les besoins fonctionnels et non fonctionnels) liée à la nature de développement sur mobile ainsi qu'une description du comportement attendu du système.
- Le quatrième chapitre est consacré à **la conception**, dans lequel seront détaillées les spécificités conceptuelles et logiques du travail réalisé présentés par le diagramme de classe, ainsi que les diagrammes de séquence.
- Le cinquième chapitre «**Réalisation**» sera réservé à présenter l'environnement matériel et logiciel avec les étapes d'implémentation de la solution ainsi que les tests et la validation de l'application.

Enfin, nous clôturons ce rapport par une conclusion dans laquelle nous résumons notre solution et exposant quelques perspectives futures.

Chapitre 1: Cadre général

1. Introduction :

Ce projet s'inscrit dans le cadre du projet de fin d'études de Licence Sciences Mathématique et informatique au Faculté des Sciences Ben M'Sik pour l'obtention du diplôme de licence en SMI.

Dans ce chapitre nous introduirons le cadre général du projet. Puis, nous passerons à présenter la méthodologie et formalismes adoptés pour la planification du travail.

2. Présentation du projet :

De plus en plus les entreprises ont besoin de rendre leurs applications accessibles sur le web. Les motivations sont multiples : élargir l'audience des utilisateurs, faire communiquer des applications existantes.

Ainsi, assurer une gestion de stock via une application embarquée représente un atout considérable pour les grands responsables et pour les hôtels pour améliorer leur gestion de l'inventaire. C'est dans ce cadre que se situe ce projet de fin d'étude qui vise la conception et le développement d'une telle application sous Android. Plus particulièrement, cette application offre un accès direct aux Articles de l'inventaire pour le suivi instantané de tout type de modification appliquée à la quantité de ce dernier. Elle est aussi utile en méthode de recherche cela en assurant deux options de recherches, la première par le code de l'Article, et la deuxième par le code barre après l'avoir scanné sur l'Article.

L'application se caractérise par l'option STANDALONE qui permet au magasinier de consulter l'inventaire en deux modes :

1. Mode Online.

2. Mode Offline

3. Méthodologie et formalismes adoptés :

3.1 Introduction :

Un processus de développement définit une séquence d'étapes, en partie ordonnée, qui concoure à l'obtention d'un système logiciel ou à l'évolution d'un système existant, pour produire des logiciels de qualité, qui répondent aux besoins des utilisateurs dans des temps et des coûts prévisibles.

Bien qu'il existe une multitude de méthodologies de développement logiciel, le choix d'une méthode pour un projet donné est une décision cruciale. Du fait que, souvent, le produit final ne correspond pas au besoin réel du moment.

3.2 Les processus unifiés :

Un processus uniifié est un processus construit sur UML (Unified Modeling Language). Les processus unifiés sont le résultat de l'unification, non pas des processus, mais plus exactement les meilleures pratiques du développement objet.

Un processus uniifié se distingue par les caractéristiques suivantes :

- ✓ **Itératif** : Le logiciel nécessite une compréhension progressive du problème à travers des raffinements successifs et développer une solution effective de façon incrémentale par des itérations multiples.
- ✓ **Piloté par les risques** : les causes majeures d'échec d'un projet logiciel doivent être écartées en priorité.
- ✓ **Centré sur l'architecture** : le choix de l'architecture logicielle est effectué lors des premières phases de développement du logiciel. La conception des composants du produit est basée sur ce choix.
- ✓ **Conduit par les cas d'utilisation** : le processus est orienté par les besoins utilisateurs présentés par des cas d'utilisation.

4. Conclusion :

Le long de ce chapitre, nous avons abordé d'une façon générale le cadre de notre projet de fin d'études. Tout d'abord, nous avons introduit notre application ainsi que les besoins menant à sa réalisation. Nous avons présenté par la suite une brève étude sur les différentes méthodologies pour justifier notre choix méthodologique et conceptuel, qui nous permettra d'atteindre les objectifs de notre projet.

Nous passerons dans le prochain chapitre à la détermination des problématiques présentées par ce sujet ainsi que le besoin du passage d'une version logicielle à une version mobile et le critère de choix du système Android comme plateforme de développement.

Chapitre 2 : Etude du Projet

1. Introduction :

Pour réussir les phases de conception et de développement, une étude préalable doit être bien élaborée. Nous commençons, en premier lieu, par présenter ce qui est existant. Ensuite nous évoquons les nouvelles fonctionnalités apportées par notre solution logicielle. Enfin, nous justifions à travers une étude menée, notre choix des solutions techniques adoptées.

2. Étude de l'existant :

Avant d'entamer notre travail, il s'avère nécessaire de faire une présentation de INFOR VR/STOCKS (VISUAL RESTAURANT GESTION DES STOCKS) l'ERP qui a une relation avec notre application Android.

INFOR Visual Restaurant pour point de vente permet de :

- Éviter les erreurs entre les serveurs et le personnel de cuisine.
- Identifier les éléments de menu les plus appréciés.
- Gérer les exigences en matière de contrôle du stock quelle que soit la capacité d'accueil du restaurant.
- Booster le service, la productivité et les bénéfices.
- Rendre vos serveurs opérationnels rapidement et durablement.

Prévu pour être utilisé avec un écran tactile et doté d'une interface graphique conviviale pour l'utilisateur, INFOR Visual Restaurant est facile à maîtriser et à utiliser par les serveurs, les managers et les propriétaires, qu'ils travaillent en salle, en cuisine ou sur la gestion back office. Entièrement modulable et facile à faire évoluer, Visual Restaurant aide à exploiter chaque centre de restauration ou centre de profit des établissements en mode local ou via la technologie sans fil.

Il permet notamment de gérer :

- Service de chambres
- Bars
- Restaurants

Application Mobile Native

- Terrasses
- Banquets
- Boutiques

INFOR Visual Restaurant prend également en charge plusieurs types d'exploitation :

- ✓ Service à table
- ✓ Restauration rapide haut de gamme
- ✓ Vente à emporter
- ✓ Restaurant gastronomique
- ✓ Service au bar

INFOR Visual Restaurant peut être intégré à d'autres solutions INFOR, ce qui permet d'accéder à l'ensemble des informations indispensables. Les modules Fidélité client, Réservation et Liste d'attente qui permettent d'assurer le suivi des réservations, d'améliorer l'utilisation des tables et d'offrir un meilleur service aux clients.

Ce nouveau système informatique, conçu selon les normes et les technologies .NET les plus modernes avec notamment un regroupement de toutes les données au sein d'une base centralisée liée à des web service, s'appuie sur une infrastructure moderne et sécurisée.

Nous citons ci-dessous les faiblesses du système actuel :

- ✓ Une perte du temps et des coûts (au détriment du client ou du prestataire de service)
- ✓ Les responsables devaient puiser les informations de façon non-automatisée.
- ✓ Les décisions d'affaire étaient prises principalement sur la base de l'intuition du corps exécutif.
- ✓ Le Contrôle de l'organisation du travail et le fonctionnement de l'ensemble d'équipe est totalement manuelle (Perte de temps et possibilité de perte de l'information).

Pour ce fait, nous avons essayé de trouver une solution mobile et fiable.

Pour atteindre les objectifs et palier les exigences, l'idée majeure est de concevoir et développer une application mobile aux utilisateurs du terminal Android.

Le projet consiste à développer :

Application Mobile Native

- ✓ Une application Android fonctionnel pour tout hôtel, permettant d'offrir l'ensemble des fonctionnalités pour les magasins, responsables de la gestion du stock de l'hôtel.
- ✓ Un module web service SOAP sur la Framework .NET qui permet d'assurer la synchronisation et la communication entre l'application Android et le serveur d'application.

Nous allons présenter par la suite les critères qui nous ont amenés à choisir les systèmes Android ainsi que SOAP comme protocole pour le module web service.

3. Les critères de choix du système Android :

3.1 Flexibilité de la plateforme :

Contrairement à l'Apple Store, Google Play n'aura pas des restrictions pour le développement des tiers. Aux États-Unis, AT&T avait acquis les droits pour vendre les iPhones d'Apple pour les cinq prochaines années à partir de la date de sa libération. Et dans le cas du BlackBerry, ce n'est pas un appareil entièrement indépendant de l'opérateur, puisque la majeure partie de la vente passe à travers ses différents supports, dans le monde entier.

Cette approche avait laissé des gens frustrés, liés à un opérateur monopolistique mobile, indépendamment de leur volonté de choisir un autre opérateur. Mais, comme Android est un système d'exploitation open source, il pourrait tirer parti des avantages de l'indépendance de l'appareil et de l'indépendance du fournisseur de services.

3.2 Facilité du développement :

Bien que les consommateurs pourraient bénéficier d'un faible coût des Smartphones Android, les développeurs ont eu des droits de personnalisation sans restriction. D'un point de vue du développeur, Android a plusieurs avantages,

Dont on peut citer :

- ✓ La plateforme d'application entière peut être réutilisée et remplacée par des éléments sélectifs.
- ✓ La machine virtuelle DALVIK renforce le pouvoir des systèmes de gestion.
- ✓ Prise en charge des graphiques 2D et 3D (OpenGL ES 1.0), donc beaucoup de projets pour les développeurs d'animation.
- ✓ Stockage de données renforcé (en utilisant la plateforme SQLITE)
- ✓ Support des applications GSM, EDGE, GPRS, UMTS 3G, HSDPA, HSUPA, LTE, réseau Wi-Fi (Dépend du matériel)
- ✓ Environnement de développement Android comprend un émulateur, un débogueur et un plug-in pour l'IDE Android Studios.

Application Mobile Native

3.3 Croissance dans le marché des Smartphones :

Rien ne semble pouvoir entraver l'écrasante domination d'Android : avec une part de marché de 78,6% en 2013 (+9,6 points en un an), il écrase la concurrence. IOS limite malgré tout la Casse et monopolise le haut de gamme. Derrière, BlackBerry disparaît peu à peu (1,9%), dépassé par un Windows Phone (3,3%) utilisé principalement par Nokia (89,3%).

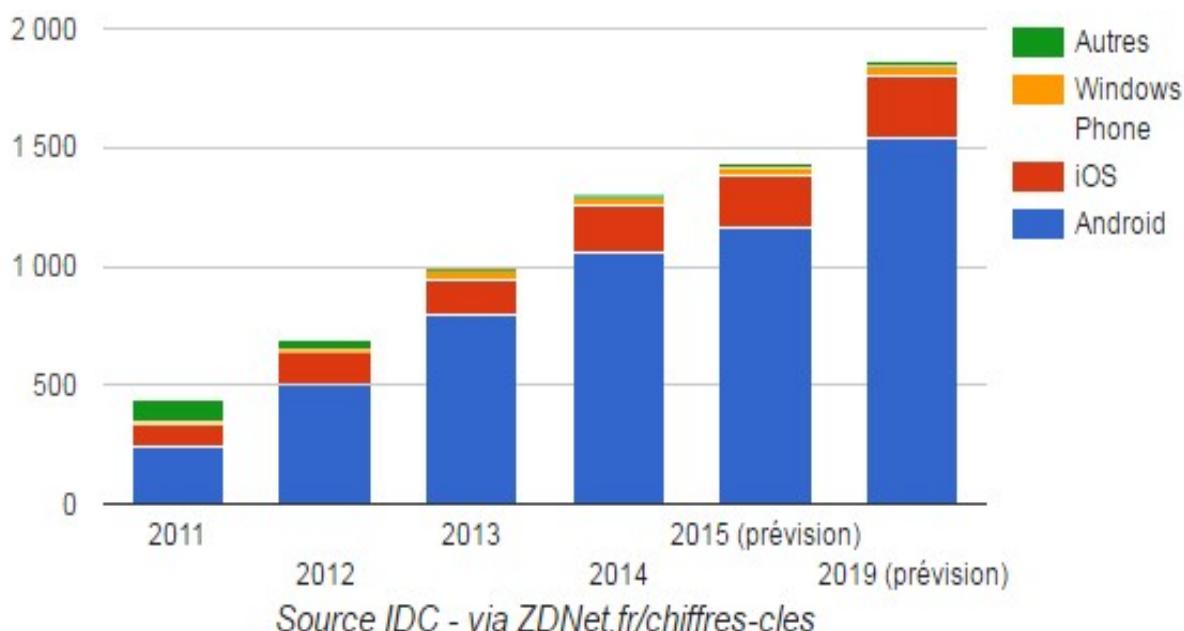


Figure 1 - Répartition par OS des livraisons mondiales de Smartphones (millions d'unité)

Plus d'un milliard d'Androphones en 2015 - Android a bien profité de la croissance du marché mondial des smartphones (+9,8%). Entre 2011 et 2015, le nombre d'androphones est ainsi passé de 243,5 à 1016 millions d'unités. La part de marché d'Android a sur la même période progressé de plus de 30 points, bondissant de 49,2% à plus de 81% en 2015.

Android bénéficie de l'entrée de gamme. Pour autant, et malgré une avance confortable, l'écosystème Android connaît quelques évolutions. En raison du fort ralentissement de Samsung en 2014, la croissance des livraisons de smartphones repose désormais plus sur des constructeurs de taille moindre. Le coréen reste malgré tout le numéro un mondial du marché.

4. Web Service SOAP :

4.1 Introduction :

Élargir l'audience des utilisateurs, vendre des services en ligne ou faire communiquer des applications existantes nous ramène à poser des questions : quelle architecture choisir où Concevoir ? Quel format utiliser pour échanger des données sur le web ? Et devrait-on utiliser un protocole applicatif existant ou développer un protocole adapté à nos besoins ?

4.2 Le développement des protocoles applicatifs :

Rendre ses applications accessibles sur le web consiste à définir un protocole applicatif et un format de données. Un protocole applicatif est un protocole de haut niveau, orienté utilisateur. Les aspects non-fonctionnels comme le transport de l'information sont délégués aux couches inférieures comme TCP/IP ou même HTTP et SMTP.

Nous avons vu que les motivations des entreprises pour déployer des web services aujourd'hui étaient nombreuses. Quelle que soit la motivation, la problématique de l'interopérabilité est présente. Même si aujourd'hui les applications qui doivent interagir sont connues, il est fort probable qu'une nouvelle application vienne un jour s'intégrer au système.

Le format des données échangées est un autre aspect de la problématique d'interopérabilité. Il existe tellement de domaines fonctionnels différents qu'il paraît difficile de spécifier un format de données qui permettrait d'exprimer efficacement tous les besoins. Le langage XML définit aujourd'hui un ensemble de règles de structuration de l'information qui nous permet d'entrevoir tous les bénéfices que l'on pourrait retirer de systèmes véritablement interopérables. Un exemple récent est le format de données RSS qui permet à différents clients de consommer un type d'information sans se soucier de la manière dont elle a été produite.

La première idée pour résoudre la problématique de l'interopérabilité des protocoles pourrait être de développer un protocole générique permettant à toutes les applications de communiquer sur le web. Ça a été l'approche adoptée par XML-RPC puis SOAP. Ces spécifications utilisent le paradigme d'appel de méthodes à distance. Les messages sont envoyés sur le web dans un format XML standardisé.

Cette approche a deux avantages :

- ✓ Elle facilite le travail du développeur dans la mesure où le modèle d'interaction par appel de méthodes métiers est le même que celui de l'application (le plus souvent développée dans le paradigme objet ou procédural)
- ✓ Et améliore un aspect de l'interopérabilité en spécifiant le format dans lequel les informations doivent être encodées.

Application Mobile Native

Cependant, en décrivant de plus en plus exhaustivement ces différents modèles d'interaction, SOAP est devenu un Framework de développement de protocoles applicatifs et non plus un protocole générique. Deux web services SOAP ne sont pas directement interopérables. Pour qu'ils le soient, ils doivent se mettre d'accord sur la façon dont ils utilisent SOAP.

4.3 Choix de SOAP :

Le choix de développement d'un module web service est principalement imposé puisqu'un ensemble des web services étaient au début déjà développé et utilisés dans le VR/STOCKS.

Nous avons vu que SOAP tient aujourd'hui plus du Framework de protocoles applicatifs que du protocole générique. Sa spécification décrit l'ensemble des types d'échange possibles entre applications. L'avantage de cette approche est qu'elle permet le développement d'outils de génération automatique de web services. Cependant, deux applications qui veulent discuter entre elles doivent se mettre d'accord sur la façon dont elles utilisent SOAP, freinant ainsi l'interopérabilité.

5. Conclusion :

Dans ce chapitre nous avons présenté une vue globale sur la solution logicielle existante et ce qui nous ramène à passer à une version mobile, ainsi que les critères du choix du système Android. Dans le prochain chapitre nous allons entamer la partie spécification et analyse des besoins.

Chapitre 3 : Spécification et analyse des Besoins

1. Introduction :

Suite à la phase de présentation du cadre du projet et l'étude du Projet, nous invoquerons dans cette partie la spécification de notre application. Cette étape est une étape nécessaire pour organiser le développement d'une application. Elle constitue la base de départ de notre travail. Elle permet de définir les différentes fonctionnalités qu'il est susceptible de réaliser pour répondre au mieux aux exigences de l'entreprise et aux besoins des clients. Ainsi, nous allons présenter les différents services et fonctionnalités que l'application doit fournir à l'utilisateur. Nous commençons par une description générale de l'application, avant de présenter une description détaillée des rubriques constituant l'application. Après, nous établirons une analyse fonctionnelle pouvant être modélisée ensuite par le diagramme des cas d'utilisation.

2. Spécification des besoins :

Cette partie prend en charge la partie descriptive des besoins exprimés par les clients. Ces besoins vont nous permettre d'identifier les fonctionnalités qui devront être implémentées dans notre application.

2.1 Besoins fonctionnels :

Une spécification fonctionnelle pour le système permet la description de ses différentes fonctionnalités en vue de sa réalisation.

A travers les diagrammes des cas d'utilisation du langage UML, nous allons présenter les fonctionnalités principales que le système doit assurer, ainsi que la liste d'acteurs qui interagissent avec. Un cas d'utilisation (use case) permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié.

Le diagramme de cas d'utilisation permet de représenter visuellement une séquence d'actions réalisées par un système, produisant un résultat sur un acteur, appelé acteur principal, et ceci indépendamment de son fonctionnement interne.

Un acteur, au sens UML, représente le rôle d'une entité externe (utilisateur humain ou non) interagissant avec le système.

2.2 Besoins non fonctionnels :

Outre les besoins fonctionnels, un ensemble de besoins non fonctionnels devra être vérifié. Ceci décidera de la qualité de réponse de notre application et le degré de satisfaction de l'utilisateur.

Application Mobile Native

- **Ergonomie et Souplesse** : Notre application doit offrir une interface conviviale et ergonomique exploitable par l'utilisateur en envisageant toutes les interactions possibles avec le Smartphone Android.
- **Fiabilité** : Nous sommes amenés à vérifier également le degré de fiabilité de notre application qui peut être exécutée tout au long des durées de travail sans erreurs.
- **Sécurité** : Notre application doit garantir la sécurité des données échangées.
- **Extensibilité** : Notre application est extensible. Elle permet l'ajout de nouveaux modules et fonctionnalités dans le futur.
- **Compatibilité** : L'un des points les plus importants lors du développement d'une application sur un environnement mobile, c'est d'assurer sa compatibilité avec n'importe quelle version du système sans oublier sa compatibilité avec les applications et éléments tierces (comme les web services).

3. Analyse des besoins :

3.1 Identification des acteurs et des cas d'utilisations :

Le magasinier : Cet acteur est l'un des responsables de l'hôtel. Il a accès aux modules suivants :

- **Consultation de l'Inventaire** : Retrouver toutes les informations pratiques (Code Article, Nom de l'Article, Son dépôt, Sa catégorie, Sa Famille et sa Sous Famille) sous forme d'un tableau.
- **Rechercher l'Article** : Elle contient deux options, soit avec le Code Article ou le Code Barre de l'Article.
 - **Scanner un Article** : Scanner le Code Barre de l'Article afin de modifier sa Quantité.
- **Initialiser la Base de données** : Permet de vider la base afin de la remettre à Zéro.
- **Déconnexion** : Permet de se déconnecter de l'établissement.

3.2 Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation général nous offre une vue globale des différents services offerts par notre application. Il expose une vision générale des fonctionnalités assurées par notre système vis-à-vis de l'utilisateur. Les figures suivantes illustrent les diagrammes de cas d'utilisations générales relatifs aux acteurs :

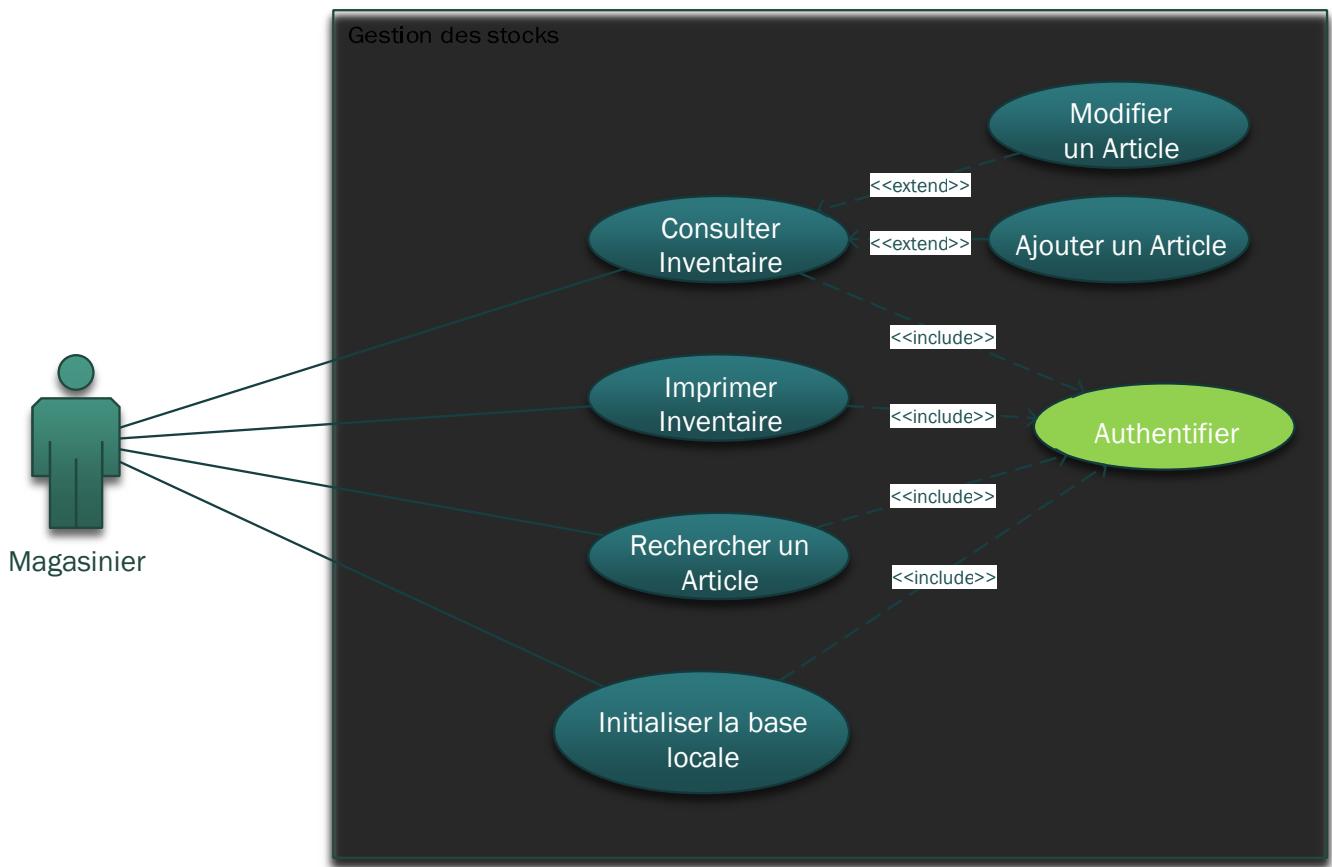


Figure 2:Diagramme de cas d'utilisation global relatif au Magasinier hôtel

- **Détails du cas d'utilisation « Authentification » :**

Numéro du cas	01
Nom du cas	Authentification
Acteur	Le magasinier
Description	Traiter l'accès de l'utilisateur à l'application
Précondition(s)	Installation de l'Application sur un mobile ou une tablette. La connexion internet est disponible
Post condition(s)	L'interface « Inventaire Menu » est affichée
Scenario	<ol style="list-style-type: none"> 1. L'utilisateur lance l'application 2. Le système affiche l'interface d'authentification 3. L'utilisateur saisit le numéro d'établissement et le mot de passe 4. Le système vérifie la validation du numéro d'établissement et le mot de passe 5. Le système récupère l'adresse IP du serveur 6. Le système vérifie la disponibilité d'accès Internet 7. Le système vérifie la connexion avec le serveur 8. Le système envoie le numéro d'établissement et le mot de passe au serveur 9. Le serveur vérifie les paramètres d'authentification 10. Le système sauvegarde les données d'utilisateur pour la prochaine entrée 11. Accéder à la page menu
Exceptions	4a- Le système affiche « champ vide » 4b- Le système affiche « login erroné » 5a- Le système affiche un message box pour saisir l'adresse IP 6a- Le système affiche « problème de connexion Internet » 6a- Le système affiche « problème de connexion avec le serveur » 11a- Le système affiche « Le mot de passe ou numéro d'établissement est erroné »

Tableau 1: Scénario d'authentification

Application Mobile Native

- Détails du cas d'utilisation «Consulter l'Inventaire» :

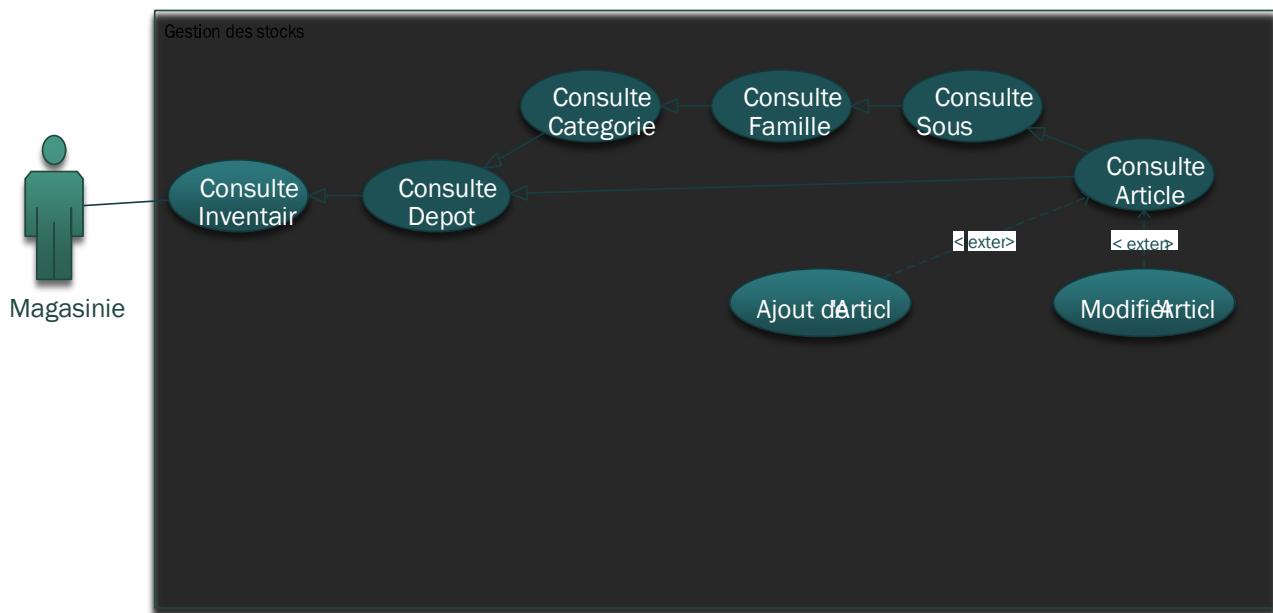


Figure 3: Diagramme de cas d'utilisation de la Consultation de l'Inventaire

Application Mobile Native

Numéro du cas	02
Nom du cas	Consulter l'Inventaire
Acteur	Le magasinier
Description	Permet au magasinier de consulter le groupe de dépôts, ses catégories, familles sous familles et articles
Précondition(s)	<ul style="list-style-type: none"> - Installation de l'Application sur un mobile ou une tablette. - Le magasinier se connecte à l'application.
Post condition(s)	<ul style="list-style-type: none"> - L'interface « Les Dépôts » est affiché - L'interface « Les Catégories » est affiché - L'interface « Les Familles » est affiché - L'interface « Les Sous Familles » est affiché - L'interface « Les Articles » est affiché - L'interface « Article » est affiché
Scenario	<ol style="list-style-type: none"> 1. L'utilisateur clique sur le bouton Saisir Inventaire 2. Le système affiche une liste des dépôts 3. L'utilisateur choisit un dépôt 4. Le système affiche une liste des catégories 5. L'utilisateur choisit une catégorie 6. Le système affiche une liste des familles 7. L'utilisateur choisit une famille 8. Le système affiche une liste des sous familles 9. L'utilisateur choisit une sous famille 10. Le système affiche une liste des Articles 11. L'utilisateur choisit un Article 12. Le système affiche une page qui contient les informations de l'Article
Exceptions	Afficher un message « liste vide »

Application Mobile Native

Numéro du cas	02.01
Nom du cas	Modifier Article
Acteur	Le magasinier
Description	Permet au magasinier de modifier la quantité d'un article
Précondition(s)	<ul style="list-style-type: none"> - Installation de l'Application sur un mobile ou une tablette. - Le magasinier se connecte à l'application.
Post condition(s)	Message « Envoie effectuée avec succès »
Scenario	<ol style="list-style-type: none"> 1. Le système affiche une page contient les informations de l'article 2. L'utilisateur clique sur le bouton Modifier 3. L'utilisateur saisit la nouvelle quantité 4. L'utilisateur clique sur le bouton Valider 5. Le système sauvegarde la nouvelle quantité 6. Retour à la liste des articles 7. L'utilisateur clique sur le bouton Envoie la modification 8. Le système envoie la modification au serveur
Exceptions	4a- L'utilisateur clique sur le bouton Annuler L'affichage du message « Envoie non effectuée »

Application Mobile Native

Numéro du cas	02.02
Nom du cas	Ajouter Article
Acteur	Le magasinier
Description	Permet au magasinier d'ajouter un nouvel article
Précondition(s)	<ul style="list-style-type: none"> - Installation de l'Application sur un mobile ou une tablette. - Le magasinier se connecte à l'application. - Le filtre de consultation est désactivé - L'accès à la page «Articles»
Post condition(s)	<p>Message « Ajout effectuée avec succès »</p> <p>Message « Envoie effectuée avec succès »</p>
Scenario	<ol style="list-style-type: none"> 1. L'utilisateur clique sur le bouton Ajouter 2. Le système affiche un formulaire 3. L'utilisateur saisit les informations de l'article et scanne le code barre si disponible 4. L'utilisateur clique sur le bouton Ajouter 5. Le système sauvegarde le nouvel article 6. Retour à la liste des articles 7. L'utilisateur clique sur le bouton Envoie la modification 8. Le système envoie la modification au serveur
Exceptions	<p>4a- L'utilisateur clique sur le bouton Annuler</p> <p>Afficher un message « Ajout effectuée non effectué»</p> <p>Afficher un message « Envoie effectuée non effectué»</p>

Tableau 2: Scénario de la Consultation de l'Inventaire

Application Mobile Native

- Détails du cas d'utilisation «Recherche de l'Article» :

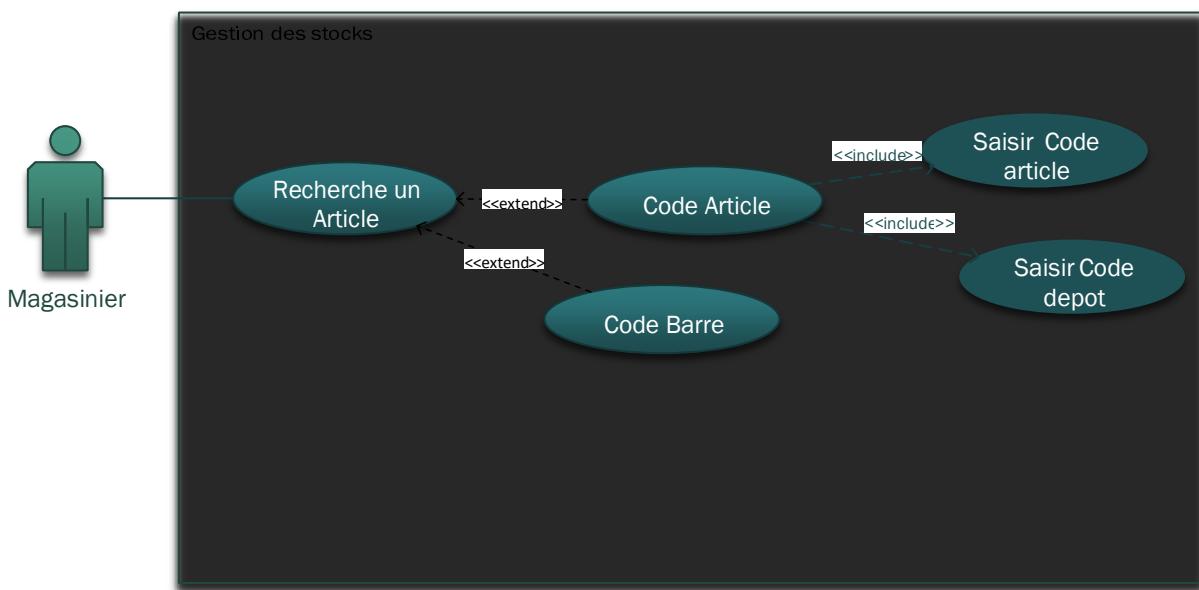


Figure 4: Diagramme de cas d'utilisation de la Recherche de l'Article

Numéro du cas	03
Nom du cas	Rechercher un article
Acteur	Le magasinier
Description	Permet au magasinier de chercher facilement un Article grâce à son code
Précondition(s)	<ul style="list-style-type: none"> - Installation de l'Application sur un mobile ou une tablette. - Le magasinier se connecte à l'application.
Post condition(s)	L'affichage des informations concernant l'Article
Scenario	<ol style="list-style-type: none"> 1. L'utilisateur clique sur le bouton Rechercher un Article 2. Le système affiche un formulaire 3. L'utilisateur saisit le code dépôt 4. L'utilisateur saisit le code de l'Article 5. L'utilisateur clique sur le bouton Rechercher 6. Le système affiche les informations concernant l'Article
Exceptions	5a- L'utilisateur clique sur le bouton Annuler 6a- Le système affiche «Article non trouvé»

Tableau 3: Scénario de Recherche de l'Article

Application Mobile Native

- Détails du cas d'utilisation « Imprimer l'Inventaire » :

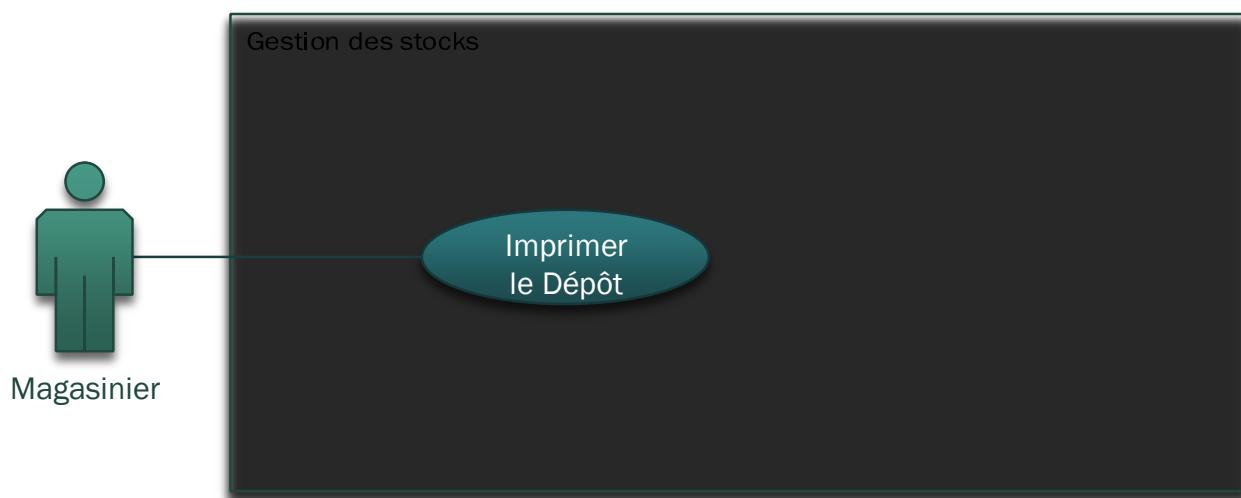


Figure 5: Diagramme de cas d'utilisation de l'Impression du Dépôt

Numéro du cas	04
Nom du cas	Imprimer l'inventaire
Acteur	Le magasinier
Description	Permet au magasinier d'imprimer une fiche des articles
Précondition(s)	- Installation de l'Application sur un mobile ou une tablette. - Le magasinier se connecte à l'application.
Post condition(s)	L'interface « Tableau » est affichée
Scénario	<ol style="list-style-type: none">1. L'utilisateur clique sur le bouton Imprimer l'inventaire2. Le système affiche une liste des dépôts3. L'utilisateur choisit un dépôt4. Le système affiche un tableau des articles et ses informations
Exceptions	4a- Le système affiche un tableau vide

Tableau 4: Scénario de l'Impression de l'Article

Application Mobile Native

- Détails du cas d'utilisation « Initialisation de la Base » :



Figure 6: Diagramme de cas d'utilisation de l'Initialisation de la Base

Numéro du cas	05
Nom du cas	Initialiser la base locale
Acteur	Le magasinier
Description	Permet au magasinier de vider la base de données locale.
Précondition(s)	<ul style="list-style-type: none"> - Installation de l'Application sur un mobile ou une tablette. - Le magasinier se connecte à l'application.
Post condition(s)	<ul style="list-style-type: none"> - Message « Base vidé » - Retour à L'interface « Inventaire Menu »
Scénario	<ol style="list-style-type: none"> 1. L'utilisateur clique sur le bouton Initialiser la base locale 2. Le système affiche un message box d'Initialisation de la base locale 3. L'utilisateur clique sur le bouton confirmer 4. Le système vide la base locale 5. Retour à la page home
Exceptions	4a- L'utilisateur clique sur le bouton annuler

Tableau 5: Scénario de l'Initialisation de la Base

4. Conclusion :

Au cours de la phase d'exploration, nous avons mis en évidence les exigences des acteurs du système. Ces exigences ont été traduites à travers la modélisation des cas d'utilisation UML. Ces modèles nous ont permis de bien cerner les besoins fonctionnels et de définir les scénarios utilisateurs. Les cas d'utilisation identifiés nous guideront, par la suite dans la conception et l'implémentation de notre projet.

Chapitre 4 : Conception

1. Introduction :

Suite à la phase de spécification et analyse des besoins, nous invoquerons dans cette partie la conception de notre application. Cette phase nous aide à comprendre notre système, le clarifier et éliminer les ambiguïtés en spécifiant sa structure et son comportement.

Pour ce faire, nous débuterons par la conception de notre système qui a été effectuée à l'aide du langage UML (Unified Modeling Language) qui est une méthode de modélisation orientée objet utilisée pour concevoir les logiciels. Nous proposerons une conception détaillée composée des différents diagrammes de classes, de séquences et d'activité.

2. Diagrammes de séquences :

Les diagrammes de séquences peuvent servir pour décrire les cas d'utilisation. Ils permettent de modéliser l'aspect dynamique du système et présentent une dimension temporelle qui met l'accent sur la chronologie des envois des messages entre les objets.

Pour notre modélisation, on présentera les différents diagrammes de séquence par cas d'utilisation.

Nous avons choisi de modéliser les diagrammes de séquences des cas d'utilisation «Authentification», «Consultation de l'Inventaire», « Recherche Article», «Imprime Article» et «Initialiser la base de données ».

Application Mobile Native

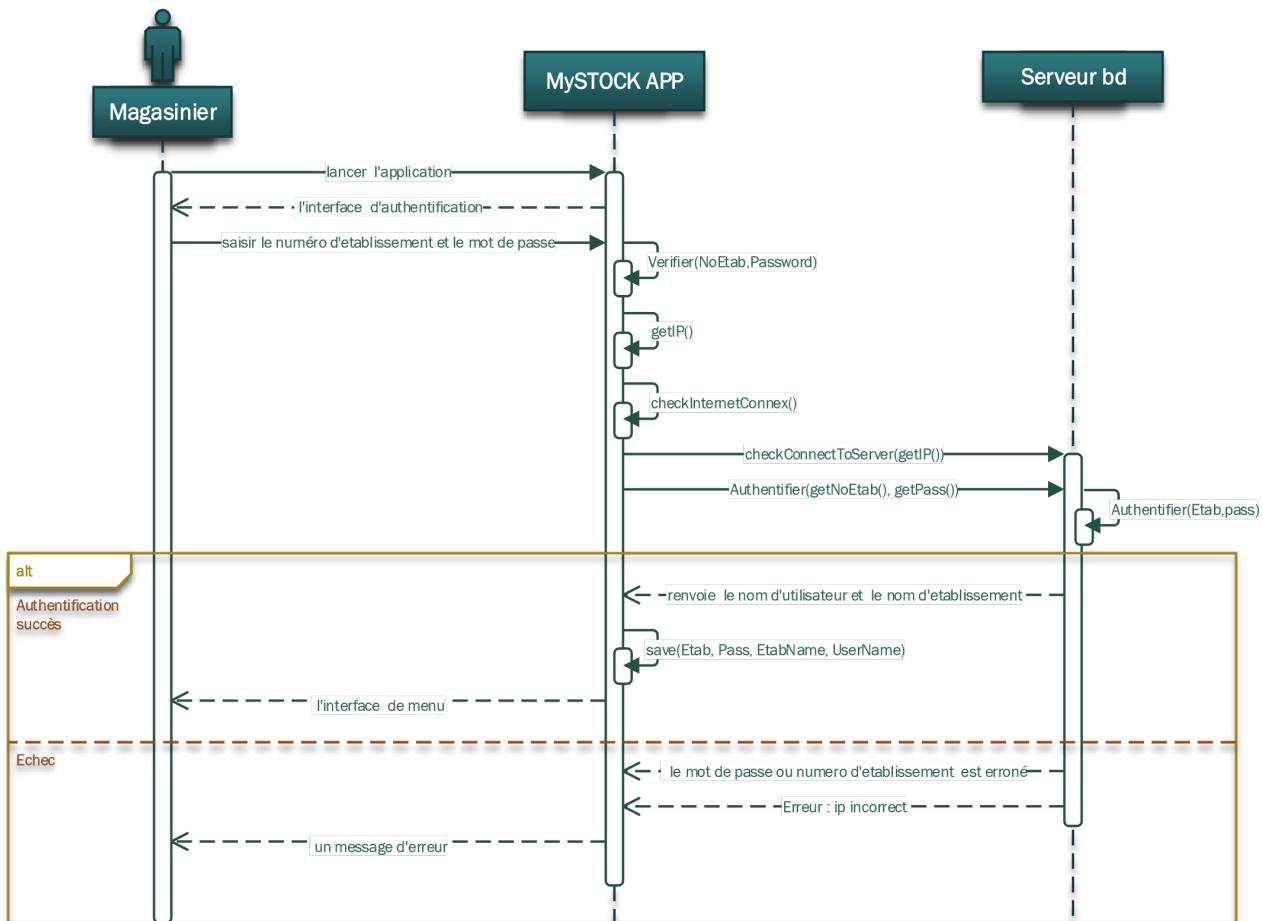


Figure 7: Diagramme de séquence d'authentification

Lorsque l'utilisateur demande l'accès à l'application, il doit tout d'abord s'identifier par son login et le mot de passe via le serveur d'application qui prend en charge de vérifier et consulter la base de données. S'il est accepté, donc il aura l'accès au système et aux applications du menu correspondant.

Sinon, le serveur d'application lui affiche un message d'erreur afin de rectifier ses données.

Application Mobile Native

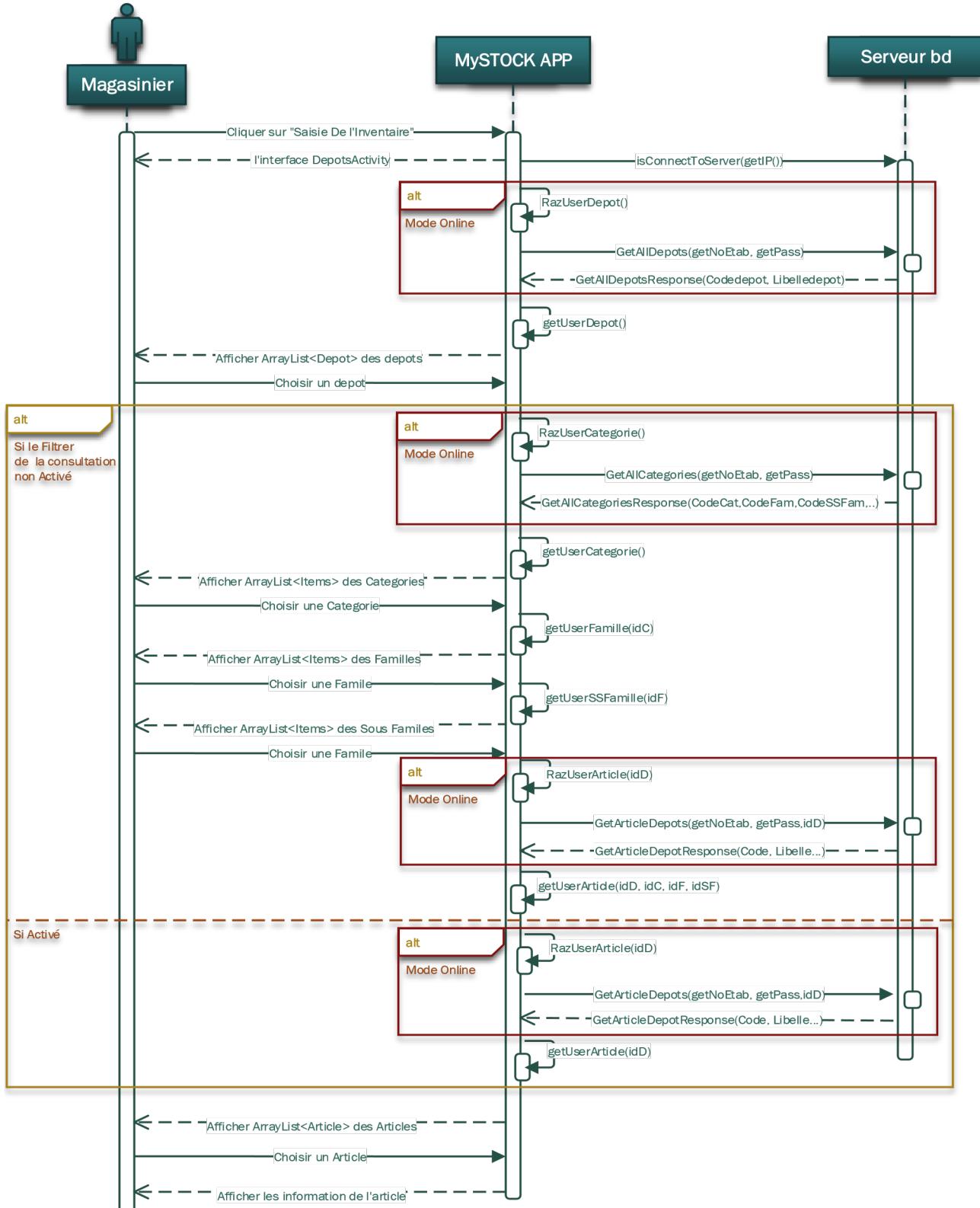


Figure 8: Diagramme de séquence de la Consultation de l'inventaire

Application Mobile Native

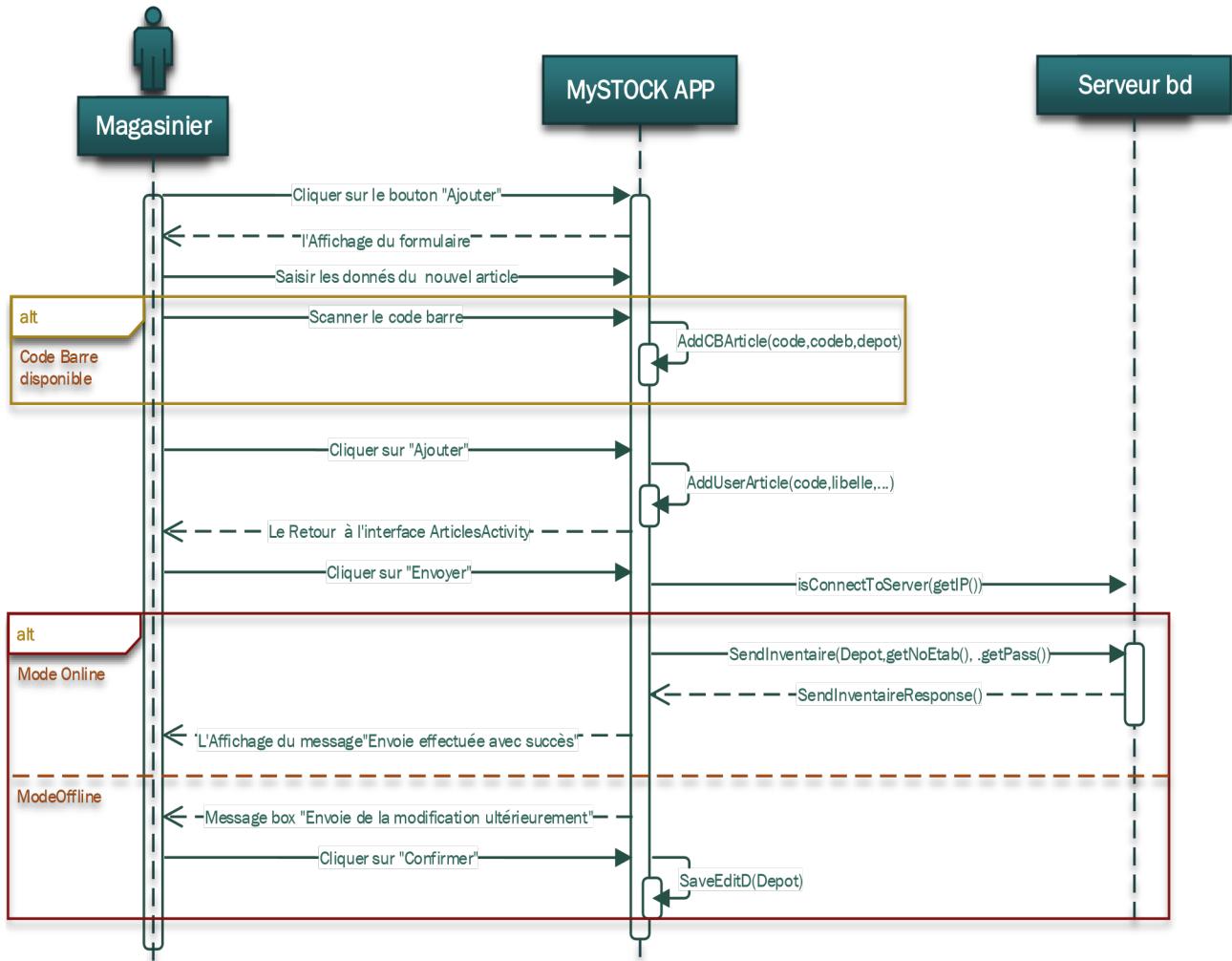


Figure 9: Diagramme de séquence de l'Addition de l'Article

Application Mobile Native

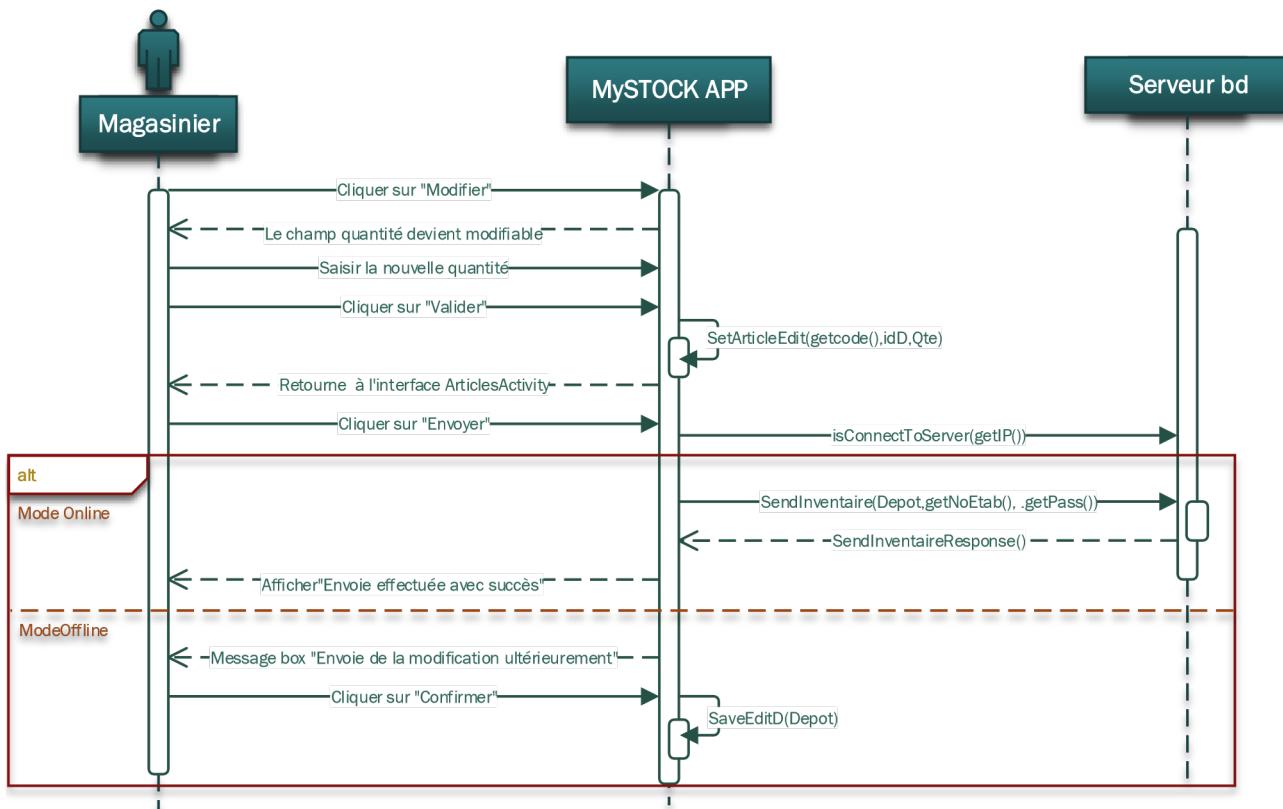


Figure 10: Diagramme de séquence de la Modification de l'Article

Pour afficher les données concernant les Articles de l'hôtel, le Magasinier doit tout d'abord ouvrir l'interface de la Consultation de l'Inventaire, puis choisir l'option de Filtrage (sélectionner « OUI » pour accéder aux Articles directement depuis le dépôt, sélectionner « NON » pour accéder aux articles depuis les catégories)

Si l'option de Filtrage a la valeur OUI alors pour afficher les Articles, il doit sélectionner le dépôt où se trouve l'Article. Ainsi l'application interroge le web service « ServerItération » qui se charge de récupérer les données de la base de données qui seront ensuite affichés à l'utilisateur sous forme d'une liste des Articles, ensuite il doit sélectionner l'Article qu'il désire consulter.

Sinon Pour afficher les Articles, il doit sélectionner le dépôt où se trouve l'Article. Ainsi l'application interroge le web service « ServerItération » qui se charge de récupérer les données de la base de données qui seront ensuite affichés à l'utilisateur sous forme d'une liste, ce dernier récupère les Catégories, les Familles et les Sous Familles. Après le Magasinier choisit la Catégorie puis la Famille et la Sous Famille de l'Articles, et encore l'application interroge le web service

« ServerItération » pour récupérer les Articles qui seront ensuite affichés à l'utilisateur sous forme d'une liste, ensuite il doit sélectionner l'Article qu'il désire consulter.

Application Mobile Native

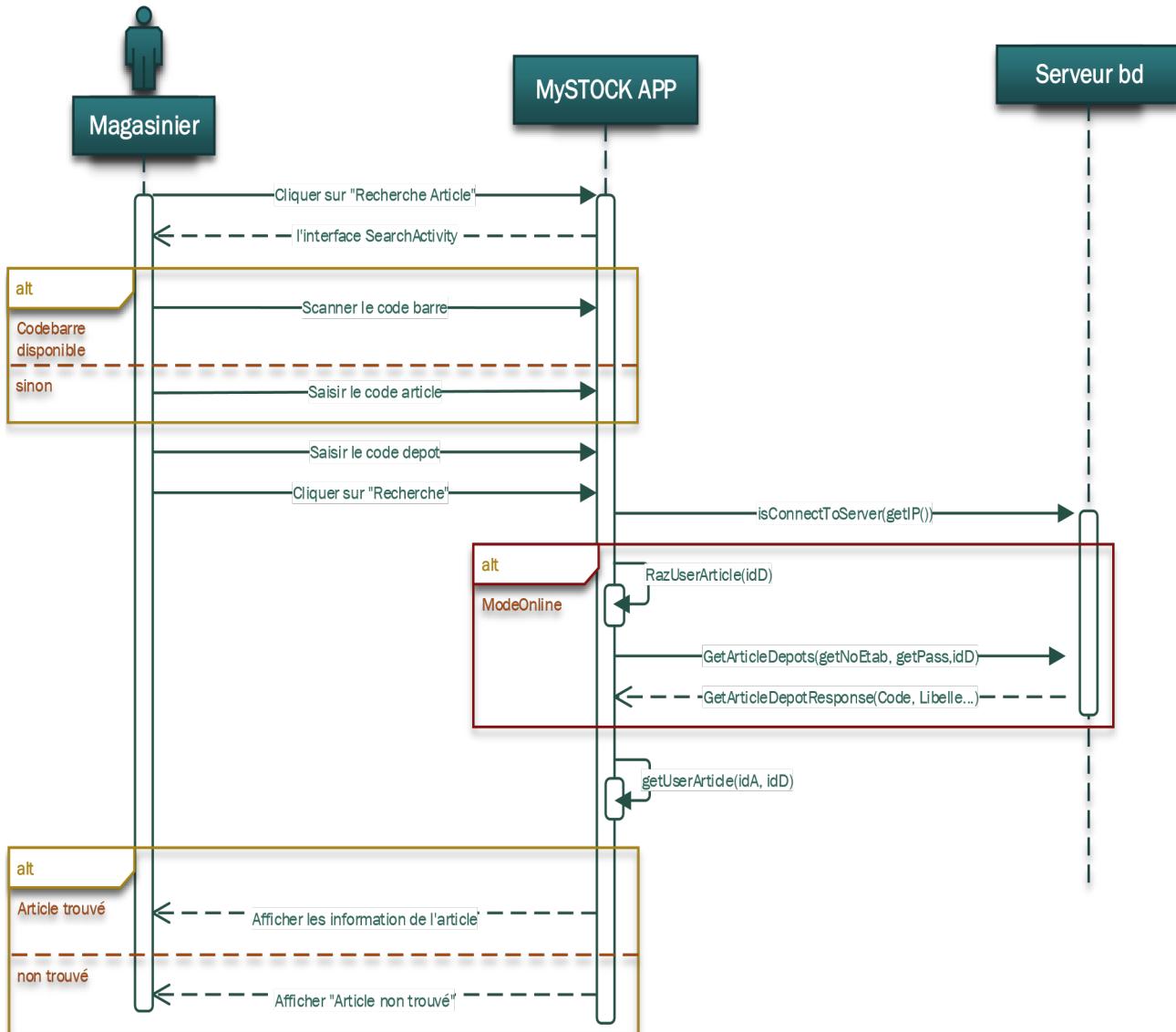


Figure 11: Diagramme de séquence de la Recherche de l'Article

Pour afficher les données concernant les Articles de l'hôtel, le Magasinier doit tout d'abord ouvrir l'interface de la Recherche de l'Article, en suite choisir l'option de Recherche.

Si le Magasinier choisit l'option de Recherche avec le Code de l'Article, alors il doit saisir le Code de l'Article et le Code du dépôt.

Sinon s'il choisit l'option de Recherche avec le code Barre de l'Article, alors il doit scanner l'Article afin de récupérer le code Barre et saisir le code dépôt.

Application Mobile Native

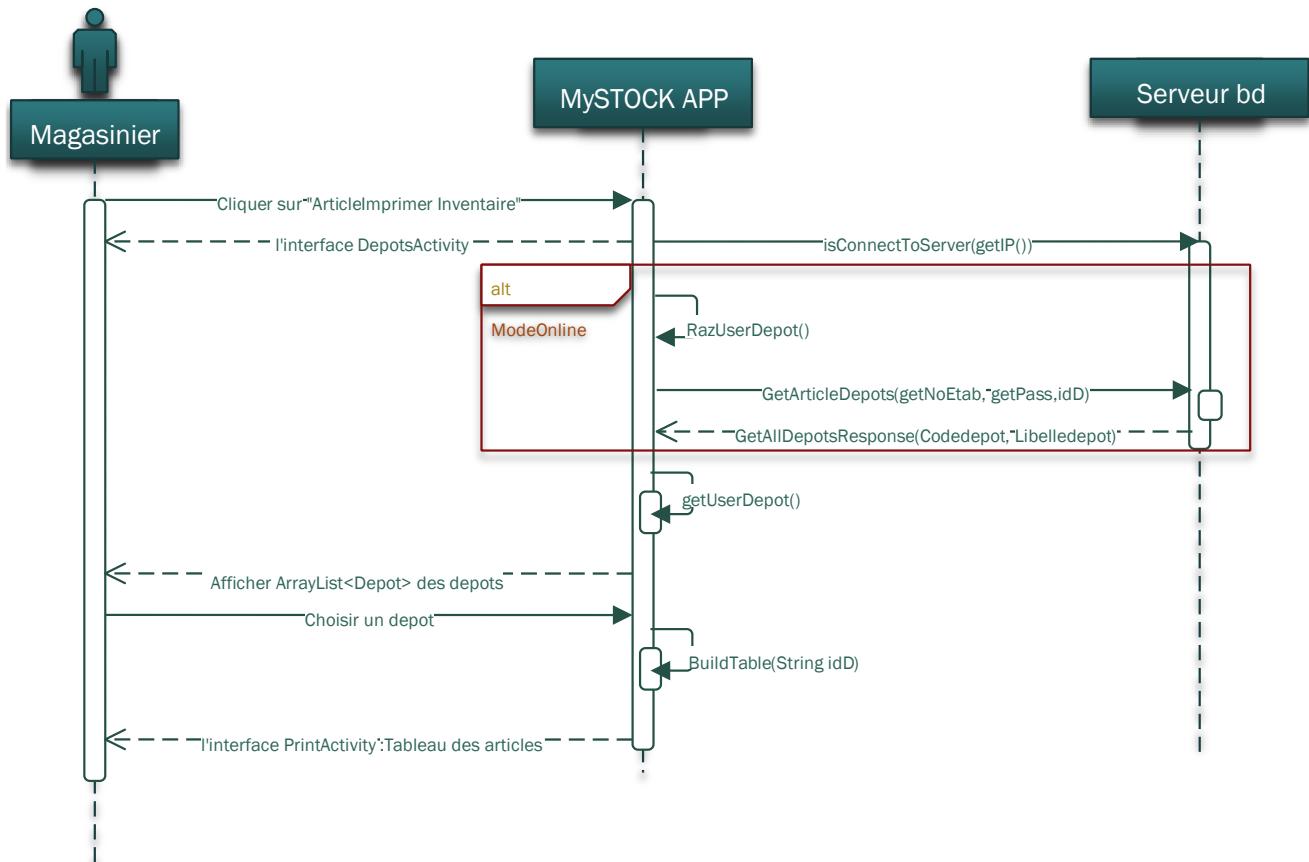


Figure 12: Diagramme de séquence de l'impression du Dépôt

Pour Imprimer les Articles, l'utilisateur doit cliquer sur le Bouton Imprimer Inventaire, puis il faut qu'il sélectionne un dépôt.

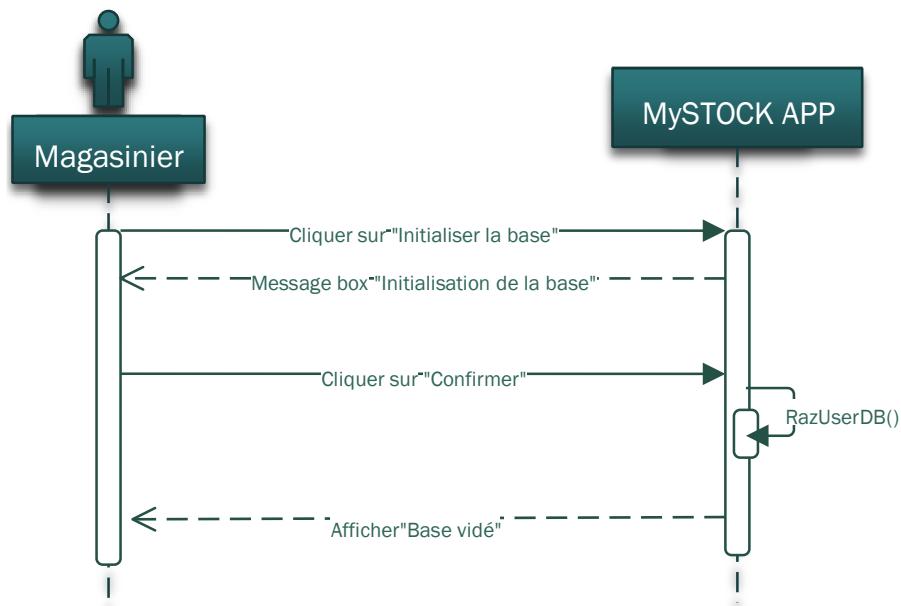


Figure 13: Diagramme de séquence de l'Initialisation de la Base de données

Pour vider la Base de données local, le Magasinier doit cliquer sur le Bouton de L'Initialisation de la Base de données.

3. Diagrammes d'activités :

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Dans ce qui suit, on présente les diagrammes d'activités pour quelques cas d'utilisation dans notre système.

Pour accéder à notre application, l'utilisateur doit s'authentifier en entrant son login et son mot de passe. Le processus d'authentification peut être résumé dans le diagramme d'activités suivant :

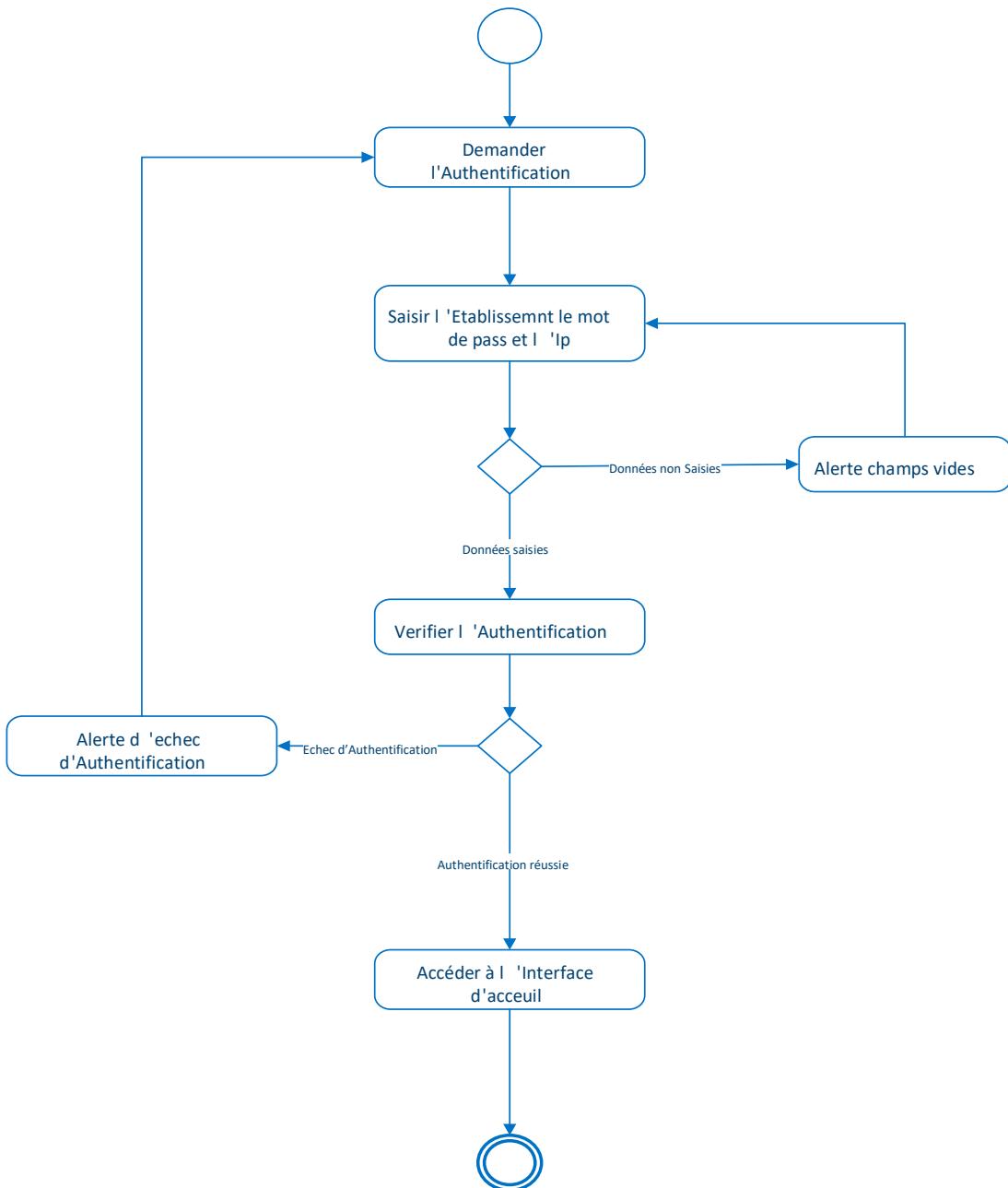


Figure 14: Diagramme d'activité d'authentification

Après s'être authentifié, l'utilisateur peut accéder à l'interface d'accueil qui lui correspond et qui lui permet de choisir l'une des fonctionnalités offertes par l'application.

- Nous commençons par le diagramme d'activité du cas d'utilisation «Consulter l'Inventaire». Ce cas d'utilisation commence par le choix de l'option du Filtrage.

Application Mobile Native

Si la valeur de Filtrage est OUI alors il faut sélectionner le dépôt qui va nous mener à une liste d'Articles afin de choisir l'Article Désiré ce qui aboutit à l'affichage des données sous forme de tableau.

Sinon si la valeur de Filtrage est NON alors il faut sélectionner le dépôt, la Catégorie, la Famille, et la Sous Famille ou ce dernier va nous mener à une liste d'Articles afin de choisir l'Article Désirés ce qui aboutit à l'affichage des données sous forme de tableau.

La figure développe le diagramme d'activité du cas d'utilisation «Consulter de l'Inventaire».

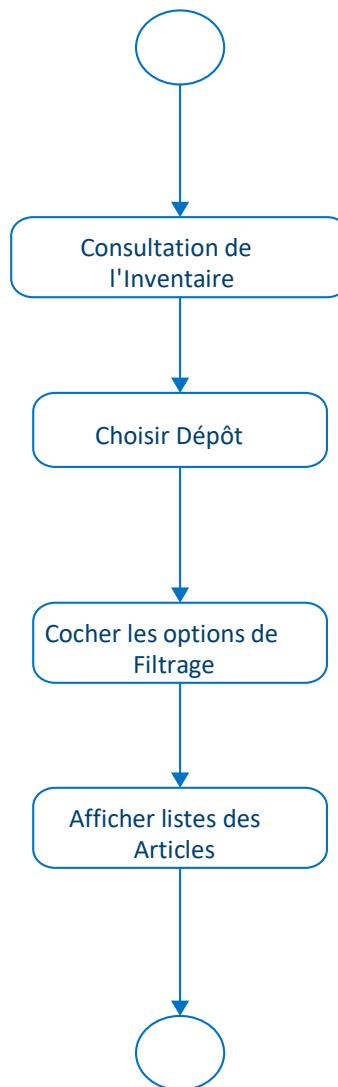


Figure 15: Diagramme d'activité de la Consultation de l'Inventaire

Application Mobile Native

La figure suivante montre le diagramme d'activité du cas d'utilisation «Recherche de l'Article». Le Magasinier doit saisir le Code de l'Article, le Code dépôt. Le processus se termine par l'affichage de l'Article sous forme d'un tableau de donnée.

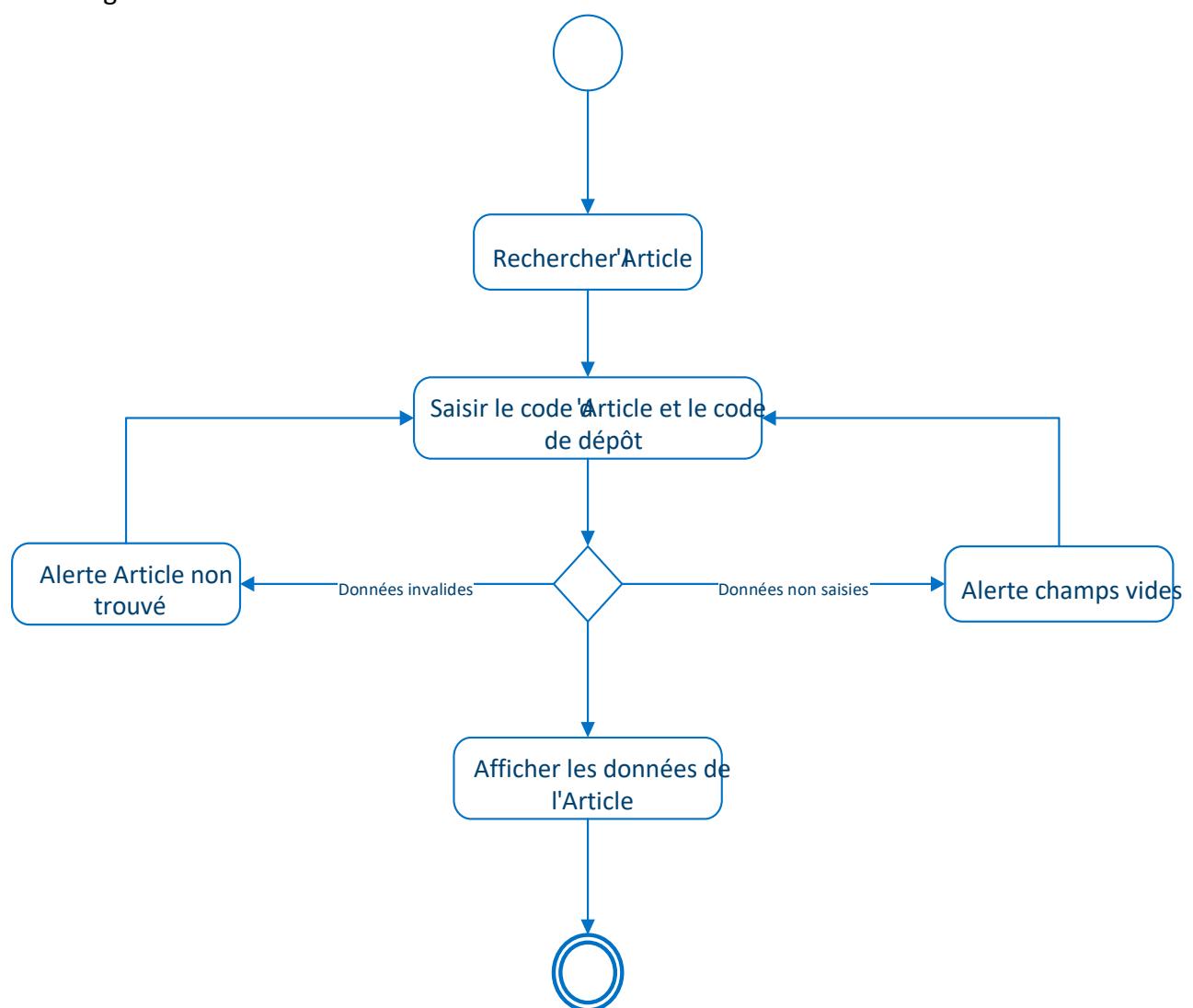


Figure 16: Diagramme d'activité de la Recherche de l'Article

Application Mobile Native

La figure suivante montre le diagramme d'activité du cas d'utilisation «Imprimer Inventaire». Le Magasinier doit choisir un Dépôt, ce qui va mener à une page qui affiche les Articles du Dépôt.

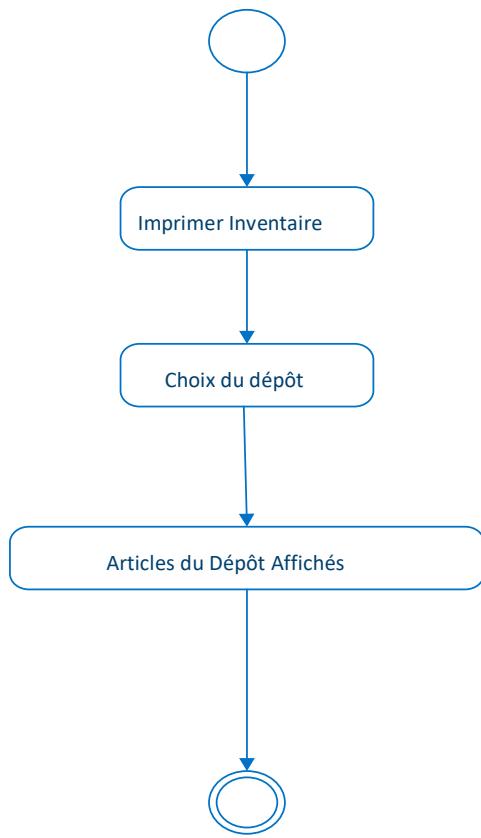


Figure 17: Diagramme d'activité de l'Impression de l'Inventaire

4. Diagramme de classes :

Un diagramme de classes donne une vue statique du système. Il est considéré comme le plus important de la modélisation orientée objet. Il contient principalement des classes. Une classe contient des attributs et des opérations. Il permet de définir ainsi quelles seront les composantes du système final et de structurer le travail de développement de manière très efficace.

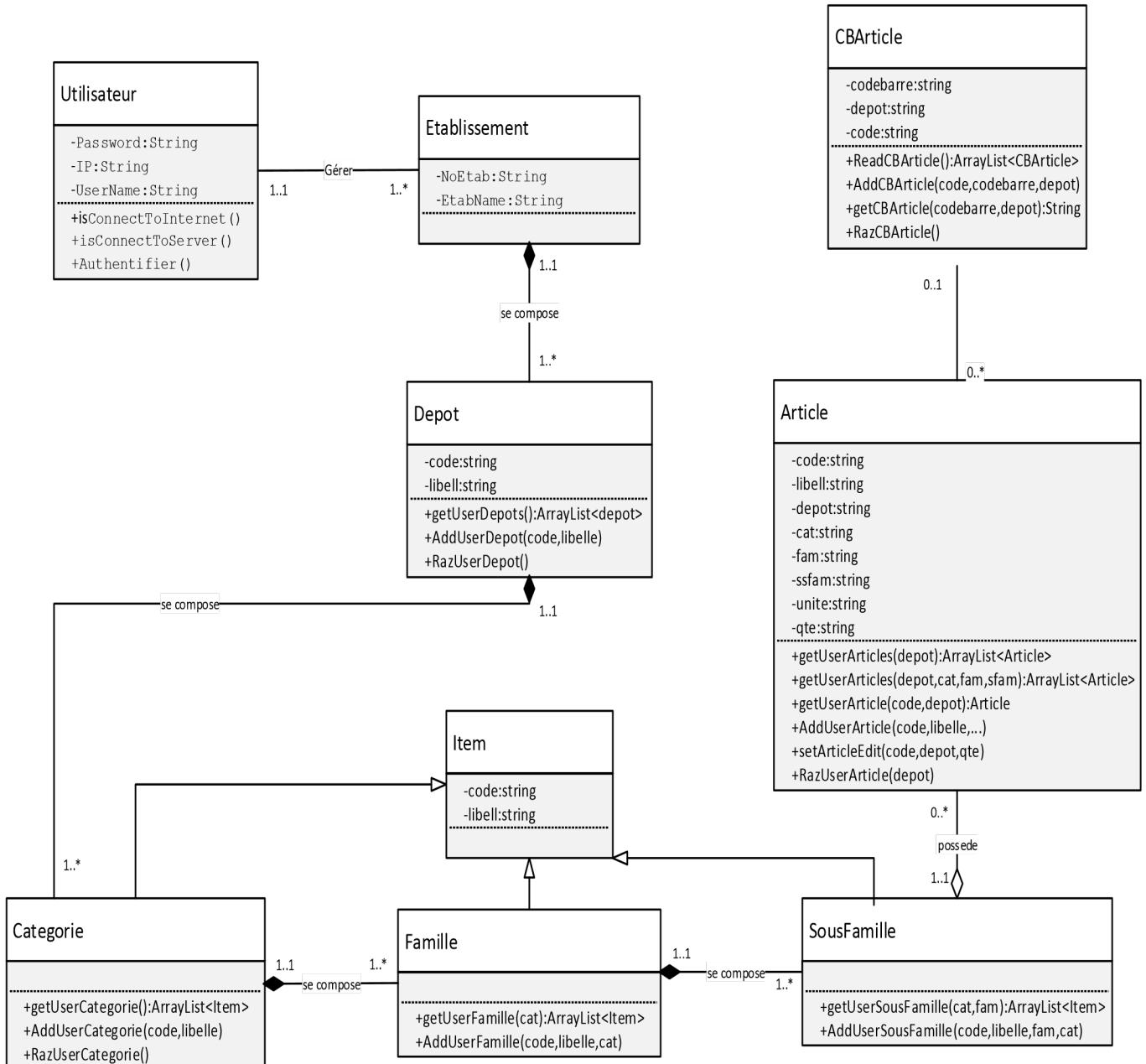


Figure 18: Diagramme de classes

Application Mobile Native

5. Modèle relationnel :

Le modèle relationnel Le MLD ou modèle relationnel consiste à décrire la structure de données utilisées. Il prend en compte les contraintes informatiques de structuration des données. Il s'agit donc de préciser le type de données utilisées lors des traitements (texte, entier, réel...). Ainsi, le modèle logique est dépendant du SGBD utilisé, qui est dans notre cas MS Access. Ce modèle intègre aussi les contraintes des données qui sont directement définies dans les paramètres des tables de la base de données : type de données attributaires (texte, numérique ou booléen), valeurs limites des données et unités de mesure pour les données quantitatives, format.

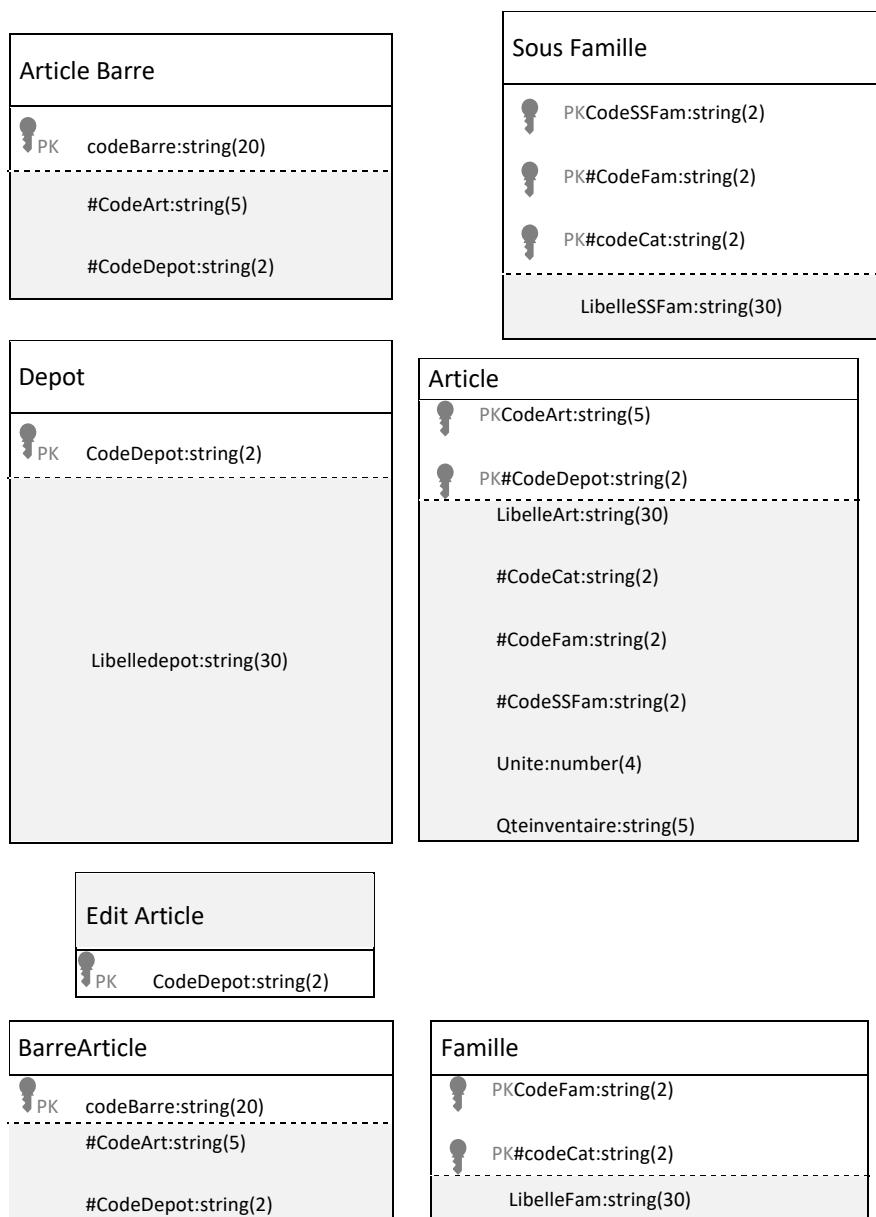


Figure 19: MLD

6. Conclusion :

Nous avons mis en valeur, dans ce chapitre, une conception étendue des principaux cas d'utilisation de l'application **MyStock**, au cours de laquelle nous avons développé les diagrammes de classes, les diagrammes de séquences et les diagrammes d'activités de notre système ainsi que la conception de la base de données utilisée.

Dans le chapitre suivant, nous détaillerons l'environnement de développement ainsi que les outils qui nous ont conduits à la réalisation de notre projet.

Chapitre 5 : Réalisation

1. Introduction :

Ce chapitre permet de réaliser les objectifs et les solutions demandées. Nous commençons par une présentation de l'environnement du travail sur le plan matériel et le plan logiciel. Ensuite, nous évoquons les pratiques adoptés durant la programmation. Enfin nous détaillons ce que nous avons développé.

2. Architecture du système :

L'architecture de notre système est illustrée par la figure... elle est partagée entre :

- **Le client ANDROID** : Conteneur d'application et demandeur de ressources.
- **Le module service web** : Vu que les données seront communiquées entre deux environnements hétérogènes, le rôle principal de ce module est de gérer la communication entre le client ANDROID et le serveur d'application d'INFOR.

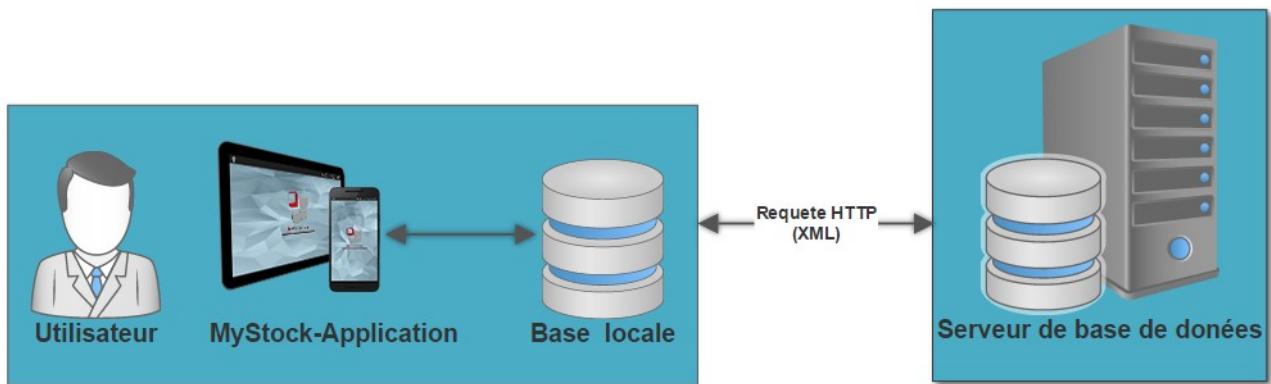


Figure 20: Architecture du système

3. Environnements de travail :

L'environnement de travail est essentiel pour la réalisation des solutions demandées.

3.1 Environnement matériel :

Durant le projet, nous avons utilisé :

- Ordinateur portable COMPAQ: processeur Intel Core i5 / 2,53 GHz
- Système d'exploitation : Microsoft Windows 8.1

3.2 Environnement logiciel :

Pour concrétiser notre application, nous avons eu recours à un ensemble d'outils techniques.

Le tableau ci-dessus inclue ces outils :

Outils	Logo	Description
Android Studios		Android studios est un environnement de développement intégré libre, extensible, universel, polyvalent et multiplateforme.
Adobe Photoshop CC		Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur édité par Adobe. Il est principalement utilisé pour le traitement de photographies numériques.
Microsoft SQL Server		Microsoft SQL Server est un système de gestion de base des données puissant et fiable et complet et fiable pour les applications bureautiques et les sites Web.
SQLite		SQLite est une bibliothèque écrite en C qui propose un moteur de base de données relationnelle accessible par le langage SQL.

Application Mobile Native

Microsoft Visio 2016 Professionnel		Simplifiez les informations complexes avec des diagrammes professionnels que vous pouvez créer en quelques clics seulement. Visio simplifie la création de diagrammes, qu'il s'agisse de capturer rapidement un diagramme de flux que vous venez d'élaborer sur un tableau blanc, de schématiser un réseau informatique, de construire un organigramme, de documenter un processus métier ou de dessiner un plan d'étage.
---	---	---

Tableau 6: Ensemble des outils logiciels utilisés en phase de réalisation

4. Implémentation du web service SOAP :

Les web services sont hébergés sur le serveur locale d'application d'INFOR.

Cette liste représente l'ensemble des web services qui ont été déjà développé et utilisés dans le PMS:

- ✓ Authentifier
- ✓ PingRQ

La figure suivante présente la liste des web services qui ont été ajoutés pour assurer la communication entre le client Android et Serveur d'application.

- ✓ GetAllWarehouses
- ✓ GetItemWarehouse
- ✓ GetAllCategories
- ✓ SendInventory
- ✓ SendBCInventory

5. Implémentation coté Android :

5.1 Choix de la version Android :

On a choisi de développer l'application avec la version Android 4.4.2 connue sous le nom KitKat vue que les terminaux Android qui supportent cette version représentent plus que 32.5% dans tout le monde.

La figure suivante représente une statistique depuis le site officiel de la plate-forme Android sur ce que nous avons mentionné précédemment.

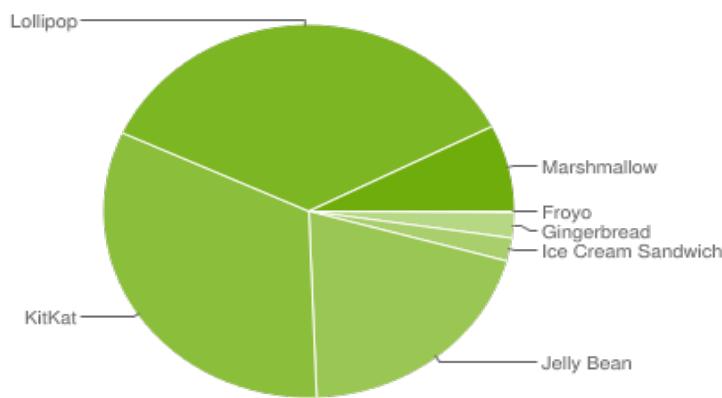


Figure 21: Répartition des versions d'API Android

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.0%
4.1.x	Jelly Bean	16	7.2%
4.2.x		17	10.0%
4.3		18	2.9%
4.4	KitKat	19	32.5%
5.0	Lollipop	21	16.2%
5.1		22	19.4%
6.0	Marshmallow	23	7.5%

Tableau 7: Répartition des versions d'API Android

5.2 Les bibliothèques utilisées :

- **kSOAP2** : est une bibliothèque légère qui est utilisée pour communiquer l'application Android avec les web services SOAP. Elle est conçue pour interagir avec la plupart des moteurs SOAP populaires.
- **ZXING** : est un projet open-source multi-format de code-barres 1D/2D de traitement d'images mis en œuvre en Java. Ce projet met l'accent sur l'utilisation de la caméra intégrée sur les téléphones mobiles et de décoder les codes-barres sur l'appareil, sans communiquer avec un serveur.

6. Interfaces de l'application :

L'application a deux modes d'accès :

- ✓ Le mode online.
- ✓ Le mode offline.

Commençant d'abord par le mode Online.

6.1 Mode Online :

Ce Mode est utilisé dans le but de charger ou d'envoyer des données au serveur.

La première ouverture de l'application se fait en Mode Online

- Interfaces d'authentification :

La figure présente l'interface qui nous permet de choisir le paramétrage l'application.

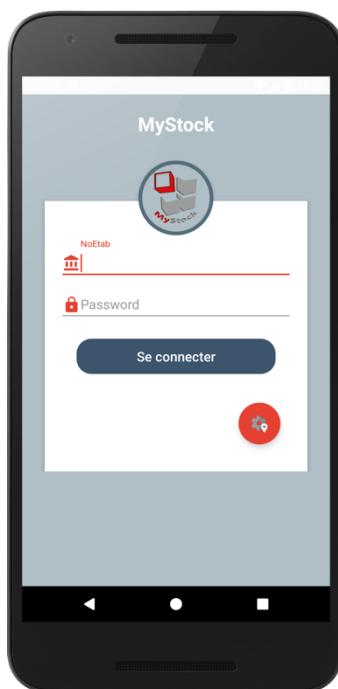


Figure 22: Interface d'accueil d'authentification

Application Mobile Native

La figure suivante (Figure 24) est l'interface d'authentification avec laquelle le magasinier peut se connecter au système. Après l'appui sur le bouton « Connexion », si les paramètres sont valides, le système affiche l'interface d'accueil (Figure 23) qui montre les différents modules que nous détaillerons par la suite. Sinon, le système affiche un message d'erreur d'échec d'authentification ou un message d'erreur d'un champ vide (Figure 25).

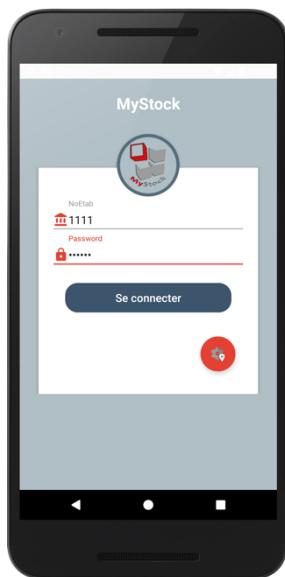


Figure 25: Interface d'authentification du responsable



Figure 24: Interface d'accueil du responsable si les paramètres sont valides



Figure 23: Message d'erreur lorsqu'il y a échec d'authentification

Application Mobile Native

- Interfaces des modules liés au magasinier :

- Consultation de l'Inventaire :

La figure suivante présente l'interface qui permet d'accéder aux différents Articles.

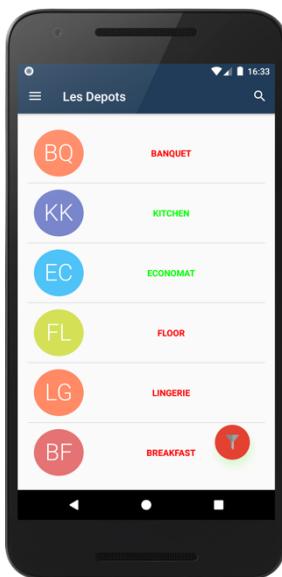


Figure 26: Interface de choix de type de Recherche

Le processus de consultation d'Articles se fait avec deux méthodes qui se basent sur l'option de

Filtrage :

- ✓ Si l'utilisateur sélectionne **NON** (Figure 27) (par défaut non) dans l'option de Filtrage alors le processus sera le suivant :
 1. Premièrement on sélectionne le dépôt Où se trouve l'Article (Figure 28).
 2. Le dépôt sélectionné va nous afficher une liste de Catégories (Figure 29).
 3. Maintenant l'utilisateur doit sélectionner la Catégorie, ce qui va nous afficher les familles suivantes (Figure 30).
 4. Après ça sera le tour de la sélectionner la Famille, afin de nous donner le résultat suivant (Figure 31).
 5. A la fin la sélection de la Sous Famille va nous afficher les Articles (Figure 32).

Application Mobile Native

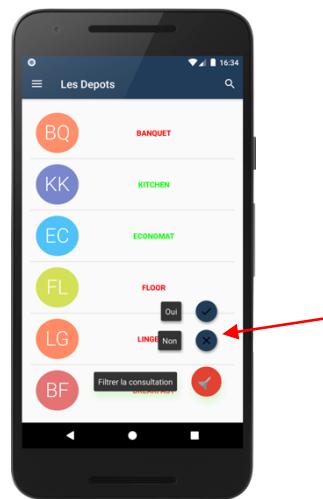


Figure 27: Interface de consultation des Articles avec l'option de Filtrage NON

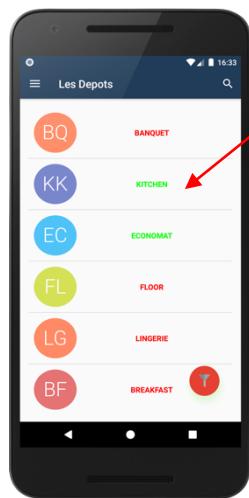


Figure 28: Interface pour sélectionner le dépôt Où se trouve l'Article

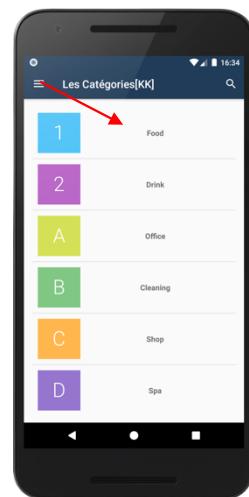


Figure 29: Interface liste des Catégories

Application Mobile Native

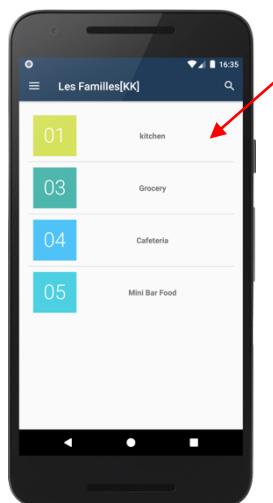


Figure 30: : Interface liste des Familles

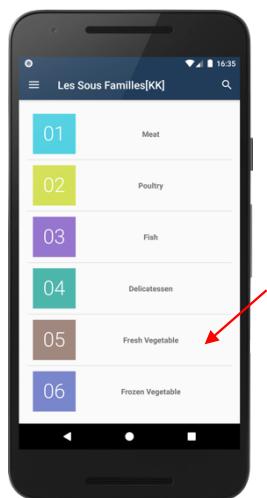


Figure 31: Interface liste des Sous Familles

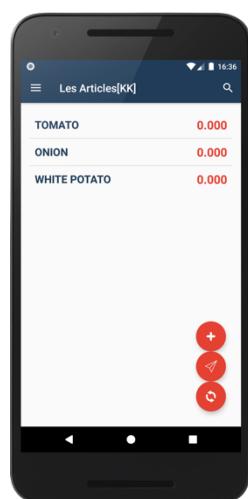


Figure 32: : Interface liste des Articles

Application Mobile Native

- ✓ Si l'utilisateur sélectionne **OUI** (Figure 33) dans l'option de Filtrage alors le processus sera le suivant :
 1. Le dépôt sélectionné va nous afficher une liste des Articles (Figure 34).

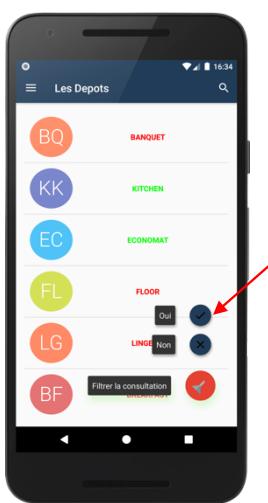


Figure 33: Interface de consultation des Articles avec l'option de Filtrage OUI

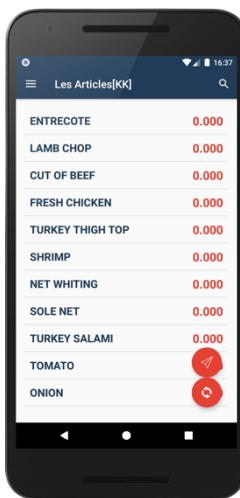


Figure 34: Interface liste des Articles

Application Mobile Native

➤ Modification d'Articles :

- ✓ Le processus de Modification d'Articles vient après la consultation des Articles :
 1. La consultation peut se faire avec les deux options de Filtrage, ici on a utilisé l'option OUI du Filtrage.
 2. Quand L'utilisateur sélectionne un Article de la liste précédente, l'interface suivante va afficher les infos de l'Article (Figure 35), Pour modifier la quantité, l'utilisateur doit d'abord cliquer sur le Bouton Modifier.
 3. Après il doit saisir la nouvelle quantité (Figure 36).
 4. Pour valider la modification cliquant sur le boutant Valider (Figure 37).
 5. Après avoir Validé la modification, l'application va retourner vers l'interface précédente, où on va distinguer que la quantité a changé (Figure 38).

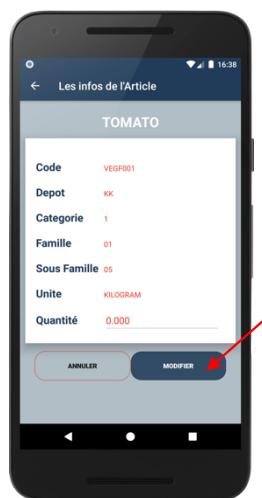


Figure 36: Interface les infos de l'article

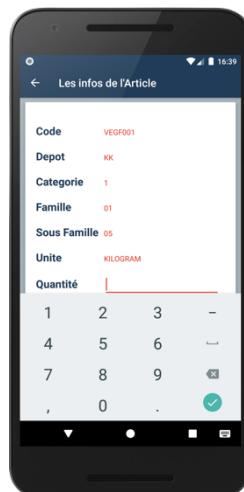


Figure 35: Interface les infos de l'article en mode modification

Application Mobile Native

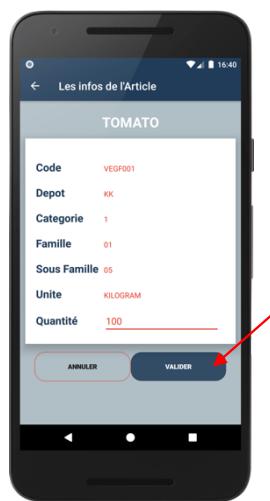


Figure 37: Interface les infos de l'article en mode modification

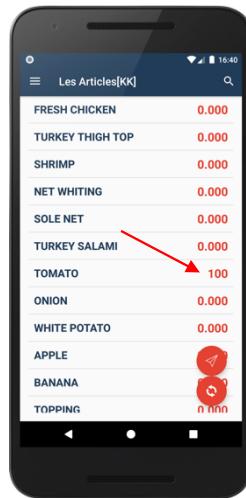


Figure 38: Interface liste des articles

Application Mobile Native

- ✓ Après la modification l'utilisateur peut envoyer la modification au serveur :
 1. Pour envoyer la modification au serveur, l'utilisateur doit cliquer sur le bouton envoyer (Figure 39).
 2. Qui va changer de forme après l'envoie (Figure 40).

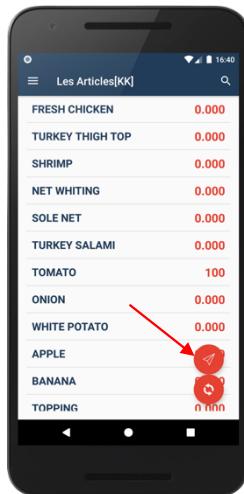


Figure 39: Interface liste des articles

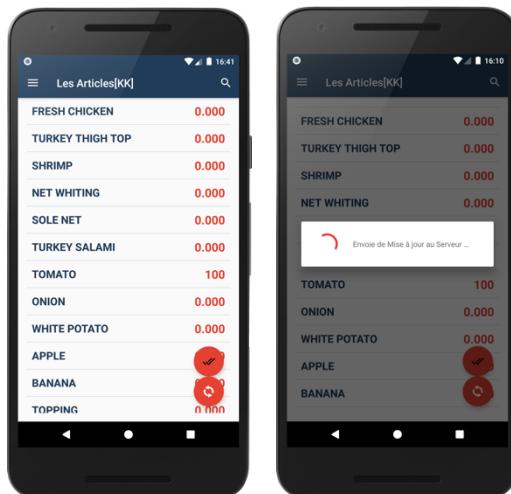


Figure 40: Interface de Modification des Articles en Mode Online

Application Mobile Native

➤ L'Addition d'un Article :

- ✓ Le processus d'Addition d'Articles vient après la consultation des Articles :
 1. Il faut que la consultation soit avec l'option de Filtrage NON.
 2. Pour Ajouter un Article, l'utilisateur doit cliquer sur le Bouton Ajouter (Figure 41).
 3. Quand l'utilisateur appuie sur le bouton d'Addition, l'interface suivante va afficher les infos de l'Article (Figure 42), Ici l'Addition peut se faire avec deux options : Ajouter un Article en donnant son code Ou Ajouter un Article en scannant son code Barre.
 4. L'utilisateur doit saisir le Code, Libelle, Unité, et la Quantité. A la fin il faut appuyer sur le bouton Ajouter (Figure 44).
 5. Après avoir Validé l'ajout, l'application va retourner vers l'interface précédente, où on va distinguer que l'article a été ajouté (Figure 45).

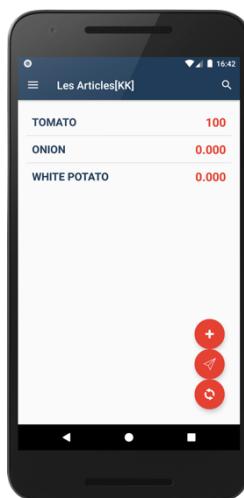


Figure 41: Interface liste des articles



Figure 42: Interface ajout de l'article



Figure 43: Interface ajout de l'article

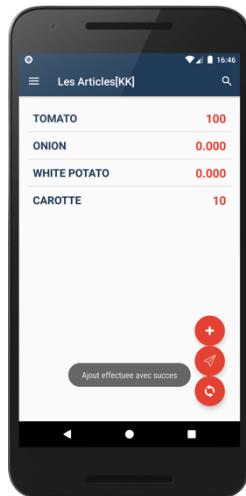


Figure 44: Interface d'Ajout d'Articles en Mode Online

Remarque : L'envoie de la requête d'Ajout se fait de la même manière que la Modification.

- Recherche de l'Article :
- ✓ Le processus de recherche d'Article se fait lui aussi avec deux options comme l'Ajout :
 1. Pour Rechercher l'Article, l'utilisateur doit d'abord scanner le code Barre de l'Article (Figure 45) ou saisir le code article, et saisir le code du Dépôt, puis cliquer sur le bouton rechercher (Figure 46).
 2. D'où l'affichage des infos de l'Article de la façon suivante (Figure 47).

Application Mobile Native



Figure 45: Interface Scanner article

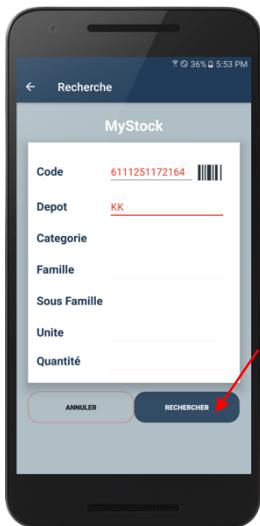


Figure 46: Interface de Recherche d'Article



Figure 47: Interface de Recherche d'Article

Application Mobile Native

- Imprimer un Dépôt :
- ✓ Le processus de l'impression du dépôt se fait de la manière suivante :
 1. Pour Imprimer un Dépôt, l'utilisateur doit cliquer sur le Bouton IMPRIMER L'INVENTAIRE sur le menu principal (Figure 48).
 2. Puis il faut choisir un Dépôt (Figure 49).
 3. Après l'interface suivante va s'afficher (Figure 50).



Figure 48: Interface Inventaire Menu

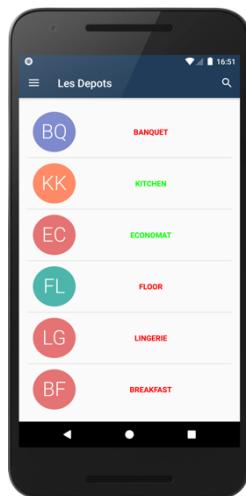


Figure 49: Interface liste des dépôts

Figure 50: Interface d'Impression de Dépôt

Application Mobile Native

- Déconnexion :

Pour se Déconnecter il faut cliquer sur le Bouton de Déconnexion (Figure 51).



Figure 51: Interface Menu principal

Ce qui va mener à l'interface d'Authentification (Figure 52).

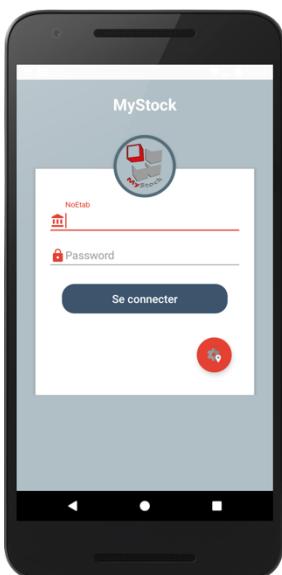


Figure 52: Interface de Déconnexion

Application Mobile Native

6.2 Mode Offline :

Ce mode a de gros avantages par rapport au Mode Online, il permet une grande lisibilité de l'application, car les opérations sur les données se fait sur la Base de données du téléphone (SQLITE).

- Les Opérations effectués dans ce Mode sont les mêmes que celles du Mode Online sauf deux opérations qui sont :

➤ La Modifications de la Quantités de l'Article :

1. On utilise la même démarche que dans le Mode Online pour l'accès à la liste des articles.
2. On consulte alors l'article désiré (Figure 53).
3. Pour modifier la Quantité, l'utilisateur doit cliquer sur le bouton Modifier (Figure 54).
4. Après il doit saisir la nouvelle quantité, puis la valider en cliquant sur le bouton valider (Figure 55).
5. Après avoir validé la modification au serveur, l'utilisateur doit cliquer sur le Bouton envoyer (Figure 56).
6. Une alerte sera affichée, l'utilisateur doit confirmer l'envoi lorsque le téléphone sera connecté à l'internet (Figure 57).

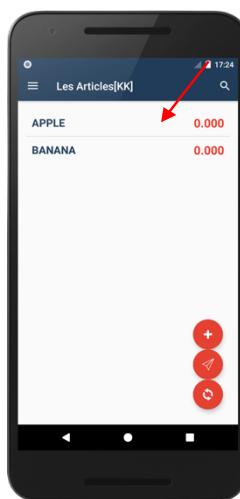


Figure 53: Interface liste des articles

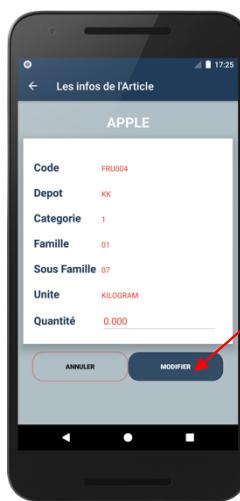


Figure 54: Interface des infos de l'article

Application Mobile Native

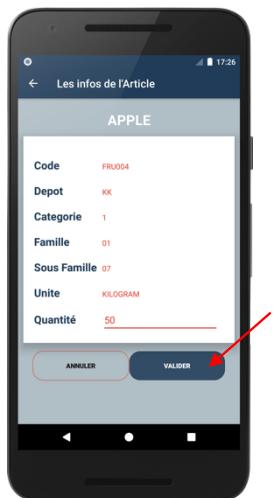


Figure 55: Interface des infos de l'article

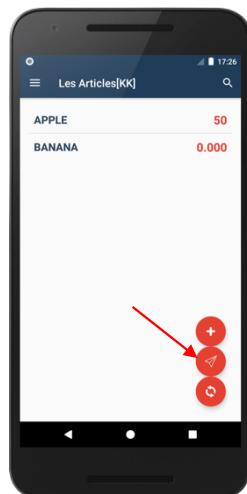


Figure 56: Interface liste des articles



Figure 57: Alert de l'envoi de la modification

Application Mobile Native

- ✓ Alors Quand le téléphone détecte la présence de la connexion Internet.
 1. L'Application affiche le message de confirmation (Figure 58).
 2. L'utilisateur doit cliquer sur confirmer pour envoyer la modification au serveur (Figure 59).



Figure 58: Alerta Envoie de la modification

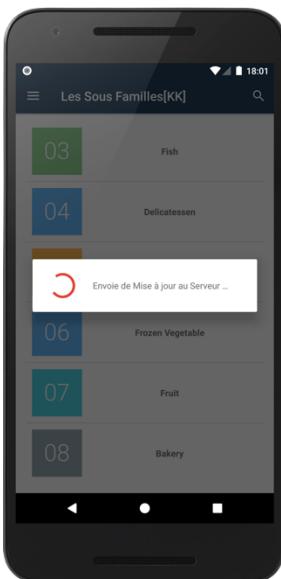


Figure 59: Interface de Modification des Articles en Mode Offline

Application Mobile Native

➤ l'Ajout de l'Article :

Le processus d'Ajout de l'Article se fait de la manière suivante :

1. Il faut d'abord consulter les Articles pour choisir l'Article correspondant (Figure 60).
2. Puis l'utilisateur doit cliquer sur Ajouter après la saisie des infos de l'Article (Figure 61).
3. Pour envoyer l'Ajout au serveur, l'utilisateur doit d'abord cliquer sur le Bouton envoyer (Figure 62), Qui va afficher un dialogue, ou celui-là va demander de confirmer l'envoi lorsque le téléphone sera connecté à internet (Figure 63).
4. Alors Quand l'application détecte la présence de la connexion Internet, elle affiche une alerte de confirmation (Figure 64), L'utilisateur doit cliquer sur confirmer pour envoyer la modification au serveur (Figure 65).

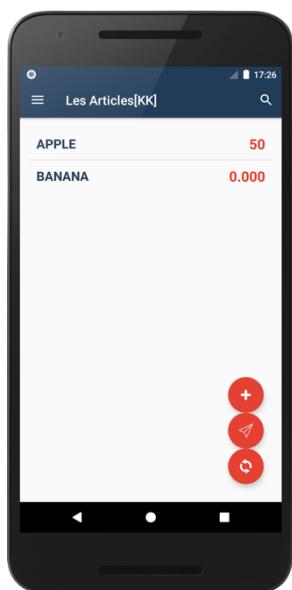


Figure 60: Interface liste des articles



Figure 61: Interface Ajout de l'article

Application Mobile Native

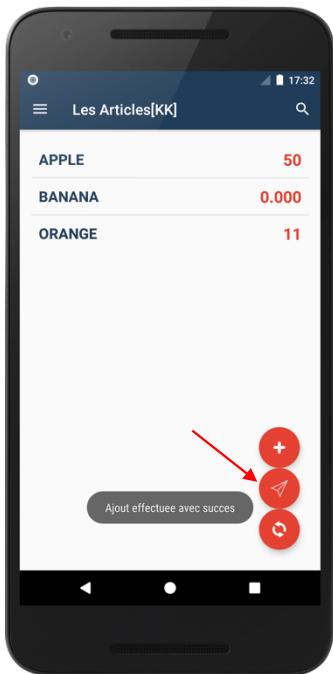


Figure 62: Interface liste des articles

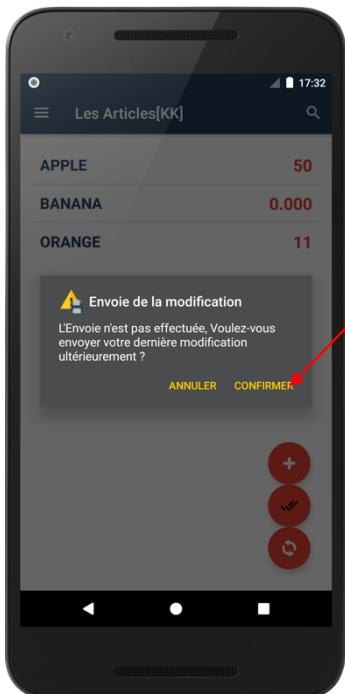


Figure 63: Alerta Envoie de la modification

Application Mobile Native

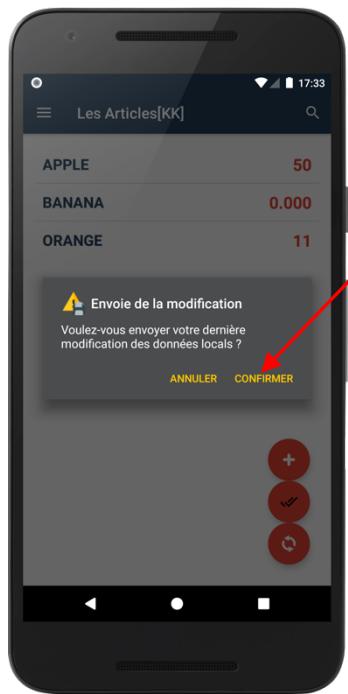


Figure 64: Alerta Envoie de la modification

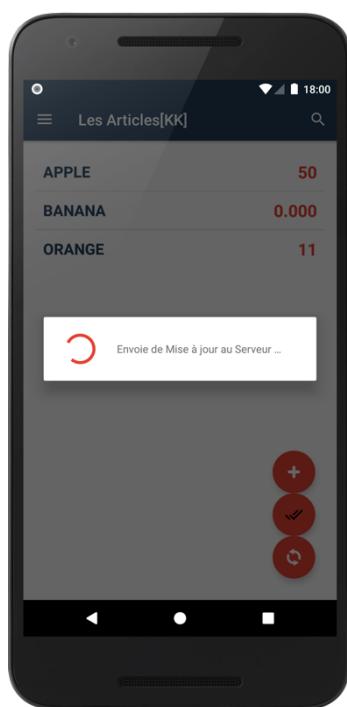


Figure 65: Interface envoie de mise à jour au serveur

7. Les Options Ajoutées :

7.1 L'option de recherche dans la Barre de Menu :

- ✓ Elle permet de faciliter la recherche en écrivant le nom dans la Barre.

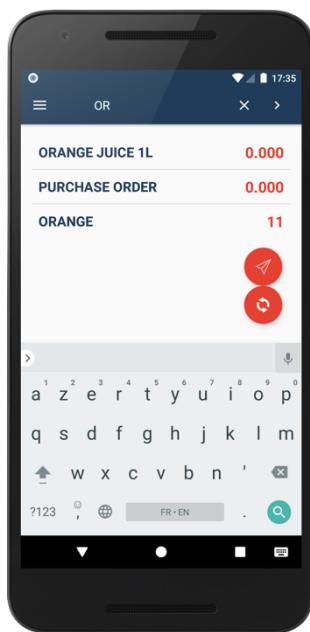


Figure 66: Interface de l'option de recherche optimisé

7.2 L'option de l'Actualisation

- ✓ Cette option actualise la base de données :
 1. Pour Actualiser la base de données, l'utilisateur doit cliquer sur le bouton actualiser (Figure 67), Qui va mettre à jour les données (Figure 68).

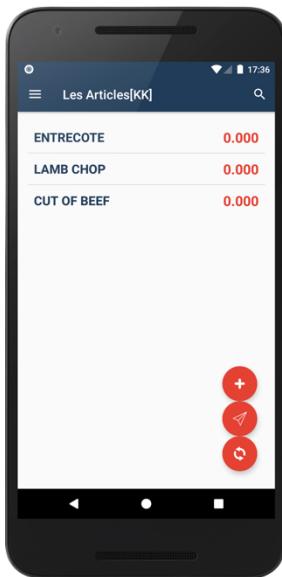


Figure 67: Interface liste des articles

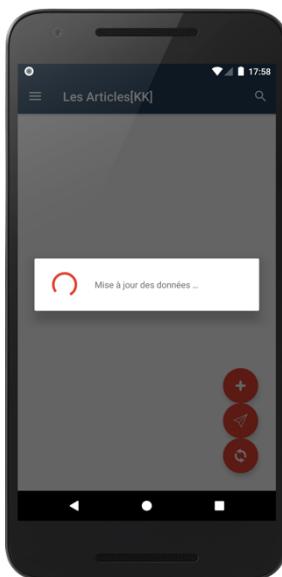


Figure 68: Interface mise à jour des données

Application Mobile Native

7.3 Le menu d'accès rapide

- ✓ Ce menu permet d'accéder à des tâches d'une façon plus rapide.

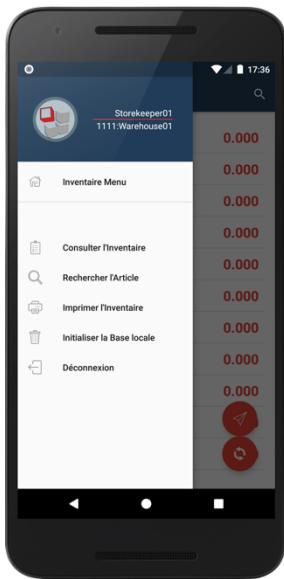


Figure 69: Interface de l'option d'Accès rapide

7.4 L'option de changement de couleur

- ✓ Cette option est incluse seulement dans l'interface d'affichage des dépôts, elle indique le dépôt plein et les dépôts vides.
 1. La couleur verte indique que le dépôt est plein.
 2. Le Rouge indique que le dépôt est vide

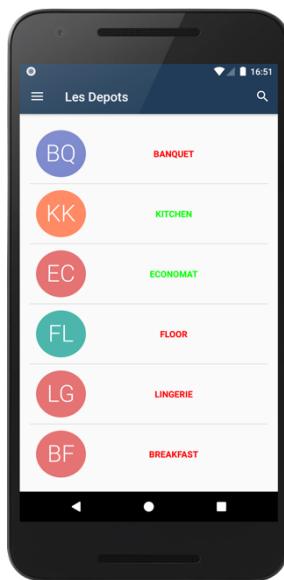


Figure 70: Interface Liste depots avec l'option de couleur

Conclusions et perspectives

Tout au long de ce rapport, nous avons essayé de présenter les différentes étapes que nous avons effectuées pour aboutir à la réalisation de notre application Android.

Nous avons commencé ce rapport par une étude du contexte du projet et une analyse des besoins fonctionnels et non fonctionnels tout en élaborant des diagrammes de cas d'utilisation pour le système, ainsi que ses différentes parties. Une fois la conception est élaborée, nous avons passé à la partie réalisation tout en présentant les différentes étapes du développement de l'application mobile.

Certes, la solution, que nous sommes parvenus à réaliser, répond à la majorité des objectifs que nous avons fixé dès le début. Néanmoins il s'avère important de signaler que certaines tâches, au cours de ce travail, nous ont posé de véritables défis.

Pour conclure, ce projet nous a été un apport indéniable sur tous les plans : personnel, fonctionnel, méthodologique et particulièrement sur le plan technique. En termes de perspectives, notre application pourra par la suite être adaptée par des améliorations. Il est possible de l'enrichir avec d'autres fonctionnalités et nous pourrons aussi généraliser le développement de notre application sur d'autres systèmes d'exploitation mobiles.

Bibliographie et Webographie

[1] Infor: <https://www.infor.com/>

[2] Android : <https://developer.android.com>

[3] SOAP: <http://www.java2s.com/Code/Jar/k/Downloadsoap2android252jar.htm>

[4] BarreCode scanner : <https://github.com/dm77/barcodescanner>

[5] Icon Letters: <https://github.com/amulyakhare/TextDrawable>

[6] Message box: <http://developer.android.com/reference/android/app/ProgressDialog.html>

[7] Les icônes : <https://www.iconfinder.com/>