

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire
Abd elhafid Boussouf Mila

Institut des sciences et de la technologie

Département de Mathématiques et Informatique

Mémoire préparé en vue de l'obtention du diplôme de Master

en : Informatique

Spécialité : Sciences et Technologies de l'information et de la communication (STIC)

Thème

**Développement d'un système de gestion
d'inventaire basé sur les codes barres**

Préparé par : Zebouchi Abdelhadi.

Maiche Hadjer.

Soutenu devant les jurys :

Encadré par : Mr.Boukhechem Nadhir.....M.A.A

Président : Mme.Yassaadi Sabrina.....M.A.A

Examineur : Mr.Bouridah Adel.....M.A.A

Année universitaire : 2015/2016

Remerciements

*Nous tenons à remercier toujours et par cette occasion, en
premier et avant tout,*

*<<ALLAH>>, qui nous a donné la force et la patience
d'accomplir ce Modeste travail.*

*En second lieu, nous tenons à remercier notre encadreur Mr :
Boukhechem Nadhir pour*

*Son précieux conseil et son aide durant toute la période du
travail.*

*Nos vifs remerciements vont également aux membres du jury pour
l'intérêt qu'ils ont*

*Porté à notre recherche en acceptant d'examiner notre travail et
de l'enrichir par leurs
Propositions.*

*Enfin, nous adressons nos plus sincères remerciements à tous nos
proches et amis, qui*

*Nous ont toujours soutenus et encouragés au cours de la
réalisation de ce mémoire.*

Merci à tous et à toutes.

Abdelhadi&Hadjer

Dédicace

Je remercie dieu qui a toujours été à mes côtés

Je dédie ce travail à

Mon père et ma mère qui ont toujours cru en moi et ont mis à ma

Disposition tous les moyens nécessaires pour que je réussisse

dans mes études.

A mes frères et mes sœurs.

A tout ma famille.

*En reconnaissance de tous les sacrifices consentis par tous et
chacun pour me permettre d'atteindre cette étape de ma vie.*

*Spécial dédicace à mon binôme, et tous les membres de ma
promotion.*

A tous mes amis.

Hadjer

Dédicace

Je remercie dieu qui a toujours été à mes côté

A mes très chers parents qui m'ont beaucoup aidé, soutenu,

encouragé et qui ont

fait de moi ce que je suis aujourd'hui

A mes chères sœurs

A mon frère

A toute ma famille.

A tous ceux que j'aime tant et que je n'ai pas cités ;

Je dédie ce mémoire ...

Abdelhadi

Résumé

Dans ce mémoire nous avons réalisé un système de gestion d'inventaire basé sur les codes-barres QR.

Notre système se compose de deux parties : une partie serveur (fixe) et une partie Smartphone (mobile).

Nous allons mener une étude conceptuelle en suivant la démarche 2TUP, utilisé UML comme langage de modélisation, NetBeans et android studio comme IDE, MySQL et SQLite comme système de gestion de bases de données.

Les mots clés : UML, JAVA, 2TUP, IDE.

Summary

In our project, we release an inventory management system based on bar codes QR. Our system content two parts: part sever, and part smartphone.

We will resort to the creation, following the approach 2TUP also by using UML as modeling language, NetBeans and android studio as IDE, MySQL and SQLite as management system.

Keywords: UML, JAVA, 2TUP, IDE.

Sommaire

| | |
|---|----|
| Introduction Générale | |
| Chapitre 1 : Généralités sur les codes-barres. | |
| Introduction | 1 |
| 1. Historique | 1 |
| 2. Définition du code-barres | 2 |
| 3. La structure d'un code-barres | 2 |
| 4. Les différents types de codes-barres | 3 |
| 4.1. Les codes-barres unidimensionnels ou linéaires | 3 |
| 4.1.1. Le code EAN | 3 |
| 4.1.2. Le code 39 | 4 |
| 4.2. Les codes-barres linéaires empilés | 5 |
| 4.2.1. Le code PDF 417 | 5 |
| 4.2.2. Le code 16K | 6 |
| 4.3. Les codes-barres à deux dimensions | 6 |
| 4.3.1. Le Code One | 6 |
| 4.3.2. Le code DataMatrix | 7 |
| 4.3.3. Le QR code | 7 |
| 5. Une comparaison entre les différents types de codes-barres | 8 |
| 6. Le type de code barre choisit pour notre application | 9 |
| 7. Les lecteurs de codes-barres | 9 |
| 7.1. Le crayon lecteur | 9 |
| 7.2. Le lecteur CCD | 9 |
| 7.3. Le lecteur laser | 10 |
| 8. L'utilité des codes à barres | 10 |
| Conclusion | 11 |
| Chapitre 2 : Technologies Utilisées. | |
| Introduction | 12 |
| 1. La plate-forme J2EE | 12 |
| 2. Le système Android | 12 |
| 2.1. L'architecture Android | 13 |

| | |
|---|----|
| 2.1.1. La couche Applications | 13 |
| 2.1.2. La couche "Application Framework" | 13 |
| 2.1.3. La couche "Libraries" (bibliothèques) | 14 |
| 2.1.4. Le noyau Android | 14 |
| 2.2. Les versions d'Android | 14 |
| 2.3. Développement d'applications pour Android | 15 |
| 2.3.1. Le Sdk Android | 15 |
| 2.3.2. Comment ça marche ? | 15 |
| 2.3.3. Composants du Sdk Android | 15 |
| 3. Android Studio | 16 |
| 4. Le SGBD MySQL | 16 |
| 5. Le SGBD SQLite | 17 |
| 6. Les Sockets | 17 |
| 6.1. Stream Sockets (TCP) | 17 |
| 6.2. Datagram Sockets (UDP) | 17 |
| 7. La Bibliothèque ZXING | 17 |
| Conclusion | 18 |
| Chapitre 3 : Analyse du système | |
| Partie1 : Etude préliminaire | |
| Introduction | 19 |
| 1. L'élaboration du cahier de charges | 19 |
| 1.1. Choix techniques | 19 |
| 1.2. Présentation du projet | 20 |
| 1.3. Recueil des besoins fonctionnels | 20 |
| 1.3.1. Les besoins fonctionnels des administrateurs (au niveau du serveur de l'application) | 20 |
| 1.3.2. Les besoins fonctionnels des agents (au niveau des Smartphones) | 20 |
| 1.4. Recueil des besoins opérationnels | 20 |
| 2. Description du contexte | 21 |
| 2.1. Identifications des acteurs | 21 |
| 2.1.1. Au niveau du serveur | 21 |

| | |
|--|----|
| 2.1.2. Au niveau du Smartphones | 21 |
| 2.2. Identifications des messages | 21 |
| 3. Le diagramme de cas d'utilisation pour les besoins fonctionnels | 21 |
| 3.1. Au niveau du serveur | 22 |
| 3.2. Au niveau de l'utilisateur Smartphone | 23 |
| 4. Description détaillée des différents cas d'utilisation | 23 |
| 4.1. Au niveau du serveur | 23 |
| 4.2. Au niveau du Smartphone | 62 |
| 5. Le modèle de domaine | 66 |
| Conclusion | 66 |
| Partie2. Capture des Besoins Techniques | |
| Introduction | 67 |
| 1. Spécification technique du point de vue matériel | 67 |
| 1.1. Configuration matérielle | 67 |
| 1.1.1. Caractéristiques techniques | 67 |
| 1.2. Spécification de l'architecture du système | 68 |
| Conclusion | 68 |
| Partie3 : Analyse | |
| Introduction | 69 |
| 1. Découpage en catégories | 69 |
| 1.1. La répartition des classes candidates en catégories | 69 |
| 1.2. Elaboration des diagrammes de classes préliminaires par catégorie | 70 |
| 1.3. Dépendance entre catégories | 71 |
| 2. Développement du modèle statique | 72 |
| 3. Développement du modèle dynamique | 73 |
| 3.1. Diagrammes de séquences | 73 |
| 3.1.1. Diagramme de séquence du cas «Scanne-code barre» | 74 |
| 3.1.2. Diagramme de séquence du cas «Transfert de code barre» | 75 |
| 3.1.3. Diagramme de séquence du cas «Synchronisation d'emplacement» | 76 |
| 3.1.4. Diagramme de séquence du cas «Ajouter bien» | 77 |

| | |
|---|----|
| Conclusion | 77 |
| Chapitre 04 : Conception | |
| Introduction | 78 |
| 1. Conception préliminaire | 78 |
| 1.1. Développement du modèle du déploiement | 78 |
| 1.2. Déploiement du modèle d'exploitation | 79 |
| 2. Conception détaillée | 81 |
| 2.1. Dictionnaire des données | 81 |
| 2.2. Les opérations | 82 |
| 2.3. Diagramme de classe détaillée | 84 |
| 3. Le modèle relationnel | 85 |
| 3.1. Les règles de passage | 85 |
| 3.2. Les règles de gestion | 86 |
| Conclusion | 87 |
| Chapitre 05 : Réalisation | |
| Introduction | 88 |
| 1. Au niveau du serveur | 88 |
| 2. Au niveau du Smartphone | 94 |
| Conclusion | 99 |
| Conclusion générale | |
| Référence | |

Liste des figures

| Chapitre1 : Généralités sur les codes-barres | |
|---|----|
| Figure 1: La structure d'un code-barres | 2 |
| Figure 2: Création d'un signal électrique | 3 |
| Figure 3: Le code EAN-13 | 4 |
| Figure 4: Le code 39 | 5 |
| Figure 5: Le code PDF 417 | 5 |
| Figure 6: Le code 16K | 6 |
| Figure 7: Le code One | 7 |
| Figure 8: Le code DataMatrix | 7 |
| Figure 9: Le QR code | 8 |
| Chapitre 2: Technologies utilisées | |
| Figure 1: Architecture du système Android | 13 |
| Figure 2: Les différentes versions du système Android | 14 |
| Figure 3: Android Studio | 16 |
| Chapitre 3: Analyse du système | |
| Figure 1: Diagramme du cas d'utilisation Au niveau de l'administrateur | 22 |
| Figure 2: Diagramme du cas d'utilisation au niveau de l'agent | 23 |
| Figure 3: Diagramme de séquence du cas d'utilisation « S'authentifier » | 24 |
| Figure 4: Diagramme de séquence du cas d'utilisation « Gestion catégorie » | 26 |
| Figure 5: Diagramme de séquence du cas d'utilisation « Ajouter catégorie » | 28 |
| Figure 6: Diagramme de séquence du cas d'utilisation « Consulter catégorie » | 29 |
| Figure 7: Diagramme de séquence du cas d'utilisation « Modifier catégorie » | 31 |
| Figure 8: Diagramme de séquence du cas d'utilisation « Supprimer catégorie » | 33 |
| Figure 9: Diagramme de séquence du cas d'utilisation « Mise à jours bien » | 35 |
| Figure 10: Diagramme de séquence du cas d'utilisation « Ajouter bien » | 37 |
| Figure 11: Diagramme de séquence du cas d'utilisation « Consulter bien » | 38 |
| Figure 12: Diagramme de séquence du cas d'utilisation « Modifier bien » | 40 |
| Figure 13: Diagramme de séquence du cas d'utilisation « Supprimer bien » | 41 |
| Figure 14: Diagramme de séquence du cas d'utilisation « Gestion Utilisateurs » | 43 |
| Figure 15: Diagramme de séquence du cas d'utilisation « Ajouter utilisateur » | 44 |

| | |
|--|----|
| Figure 16: Diagramme de séquence du cas d'utilisation « Consulter utilisateur » | 45 |
| Figure 17: Diagramme de séquence du cas d'utilisation « Modifier utilisateur » | 47 |
| Figure 18: Diagramme de séquence du cas d'utilisation « Supprimer utilisateur » | 48 |
| Figure 19: Diagramme de séquence du cas d'utilisation « Créer emplacement » | 50 |
| Figure 20: Diagramme de séquence du cas d'utilisation « Ajouter emplacement » | 52 |
| Figure 21: Diagramme de séquence du cas d'utilisation « Consulter emplacement » | 53 |
| Figure 22: Diagramme de séquence du cas d'utilisation « Modifier emplacement » | 55 |
| Figure 23: Diagramme de séquence du cas d'utilisation « Supprimer emplacement » | 57 |
| Figure 24: Diagramme de séquence du cas d'utilisation « Recherche des biens » | 59 |
| Figure 25: Diagramme de séquence du cas d'utilisation « Etat des lieux » | 61 |
| Figure 26: Diagramme de séquence du cas d'utilisation « Gestion des biens manquants » | 62 |
| Figure 27: Diagramme de séquence du cas d'utilisation « Scan automatique des biens » | 63 |
| Figure 28: Diagramme de séquence du cas d'utilisation « Transfère les biens scannés » | 64 |
| Figure 29: Diagramme de séquence du cas d'utilisation « Synchroniser l'emplacement » | 65 |
| Figure 30: Le modèle de domaine | 66 |
| Figure 31: Organisation du modèle de spécification logicielle | 68 |
| Figure 32: Découpage en catégorie de notre système | 69 |
| Figure 33: Découpage des classes de chaque catégorie | 70 |
| Figure 34: Dépendances entre les catégories | 71 |
| Figure 35: Diagramme de classe de chaque catégorie | 72 |
| Figure 36: Diagramme de séquence du cas « Scanne code barre » | 74 |
| Figure 37: Diagramme de séquence du cas « Transfert de code barre » | 75 |
| Figure 38: Diagramme de séquence du cas « Synchronisation d'emplacement » | 76 |
| Figure 39: Diagramme de séquence du cas « Ajouter bien » | 77 |
| Chapitre 4 : Conception | |
| Figure 1: Diagramme de déploiement | 79 |
| Figure 2: Définition des applications dans modèle statique | 80 |
| Figure 3: Diagramme de classe détaillé de notre système | 84 |
| Chapitre 5: Réalisation | |
| Figure 1: Fenêtre d'authentification | 88 |
| Figure 2: Espace administrateur | 89 |
| Figure 3: Fenêtre « Gérer utilisateur » | 90 |
| Figure 4: Fenêtre « Session » | 91 |

| | |
|--|----|
| Figure 5: Fenêtre « Catégorie » | 91 |
| Figure 6: Fenêtre « Bien » | 92 |
| Figure 7: Fenêtre « Bien scanné » | 93 |
| Figure 8: Fenêtre « Etat des lieux » | 93 |
| Figure 9: Interface de démarrage | 94 |
| Figure 10: Fenêtre « Créer un compte » | 94 |
| Figure 11: Interface d'authentification | 95 |
| Figure 12: Interface d'accueil | 95 |
| Figure 13: Fenêtre « Scan » | 96 |
| Figure 14: Fenêtre « Envoyer le code barre » | 96 |
| Figure 15: Fenêtre « Affichage et transfert » | 97 |
| Figure 16: Fenêtre « Paramètre » | 97 |
| Figure 17: Fenêtre « Synchronisation des emplacements » | 98 |
| Figure 18: Fenêtre « Ajouter IP » | 98 |
| Figure 19: Fenêtre « Modifier IP » | 99 |

Liste de tableaux

| | |
|--|----|
| Chapitre1: Généralités sur les codes-barres | |
| Tableau 1: Une comparaison entre les types de codes-barres | 8 |
| Chapitre 4: Conception | |
| Tableau 1: Dictionnaire de donnée avec les classes et les attributs | 82 |
| Tableau 2: Dictionnaire de donnée avec les opérations | 83 |
| Tableau 3: Les tables de la base de données | 86 |

INTRODUCTION

GENERALE

Il ne fait désormais plus aucun doute que l'informatique est la révolution la plus importante et la plus innovante qui marqua la vie de l'humanité moderne. Ses logiciels et ses méthodes de conception et de développement ont vu l'avènement des technologies quotidiennement utilisées, parmi ces technologies il y a la gestion automatique des codes-barres, ils se retrouvent dans de nombreuses sphères de l'activité économique, principalement celles qui impliquent la circulation des biens et l'identification. Parmi les principales utilisations de cette technologie, mentionnons :

- Le suivi des articles dans les magasins à rayons, les grandes surfaces, les marchés d'alimentation, etc..., afin de mieux gérer les inventaires, la facturation et la comptabilité.
- Le suivi des envois postaux, comme les colis ou les lettres recommandées.
- Le suivi des bagages sur les lignes aériennes.
- L'identification des patients et des médicaments.
- L'indexation de documents.

L'objectif de notre projet est de développer un système de gestion d'inventaire basé sur les codes-barres. Ce travail vise principalement à faciliter la gestion des biens des entreprises, en fournissant rapidement toute les informations relatives aux biens.

Nous souhaitons réaliser une application qui répond aux besoins des entreprises, en implémentant deux parties : une partie serveur (fixe) et une partie Smartphones (mobile), la partie serveur pour la gestion des biens et des utilisateurs, et pour déterminer l'emplacement des biens et la création de code barre et l'impression, et la partie Smartphone pour le scan automatique des biens et le transfert vers le serveur.

Les chapitres de ce mémoire sont organisés comme suit :

- Le chapitre 01 présente d'une façon générale les codes-barres et leurs différents types.
- Le chapitre 02 présente les différentes technologies et les outils de développement utilisés dans notre application.
- Le chapitre 03 vise à détailler la modélisation de notre application à partir des diagrammes UML, et détailler la phase d'analyse.
- Le chapitre 04 présente la phase de conception de notre application.
- Le chapitre 05 présente quelques interfaces utilisateurs.

CHAPITRE 1

Généralités sur les codes-barres



Introduction

Les codes-barres sont des acteurs discrets de la vie courante. Destinés à automatiser l'acquisition d'une information généralement numérique, ils trouvent leurs applications dans des domaines aussi variés que la gestion des prêts d'une bibliothèque, les caisses enregistreuses à lecture optique ou le contrôle de la production dans l'industrie.

Les codes-barres offrent une méthode rapide, facile et précise pour saisir des données. Une utilisation judicieuse des codes-barres peut réduire les délais nécessaires aux employés pour effectuer certaines tâches et augmenter leur efficacité. L'objectif de ce chapitre est d'étudier en détail les différents types de code-barres et choisir celui qui s'adapte le mieux à notre application.

1. Historique

L'origine des codes-barres remonte à 1949, avec l'invention d'un code-barres circulaire (NJ WOODLAND) qui ne sera pas utilisé en pratique. En 1960, SYLVANIA imagine un symbole d'identification des wagons de chemin de fer faisant appel à des barres et des espaces.

L'idée, qui est donc en l'air dès les années de l'après-guerre, est de remplacer la lecture des bandes perforées par des successions de noirs et de blanc.

La société INTERFACE MECANISMES (qui deviendra plus tard INTERMEC) commence ainsi son activité en fabricant des lecteurs de bandes de ce type destinés à remplacer les lecteurs de bandes perforées.

En 1970 est créé aux Etats-Unis un comité ayant pour but de définir une codification adaptée aux problèmes de la grande distribution : le "US SUPERMARKET AD HOC COMMITTEE ON UNIVERSAL PRODUCTCODING", ou plus simplement l'U.P.C. Ce comité se met au travail et en 1973 le code U.P.C. est adopté.

Entre temps, en 1970, PLESSEY introduit son code-barres dans les applications bibliothécaires, et, en 1971, MONARCH invente le code CODABAR (dit également Monarch) et le Docteur ALLAIS, d'INTERMEC, propose le code 2 parmi 5 entrelacé.

En 1974, sur une demande de la société BOEING, INTERMEC invente le premier code alphanumérique : le code 39. Puis, les choses s'accélérent. Le code E.A.N. (European Article Numbering), version internationale de l'U.P.C. est adopté en 1977.

La même année, le CODABAR est choisi par l'American Blood Bank Commission, et l'administration américaine commence des études ayant pour but le choix d'une symbologie dans le cadre du projet LOGMAR.

De nouvelles symbologies sont créés pour répondre à des besoins sans cesse plus variés et sophistiqués : code 128 (Computer Identics) et code 93 (INTERMEC) pour les utilisateurs cherchant un code alphanumérique dense et pouvant coder tous les caractères du jeu ASCII, et, très récemment, une nouvelle génération de codes à étages extrêmement denses: les codes 49 (INTERMEC). [1]

2. Définition du code-barres

Un code-barres est une manière graphique de représenter des données numériques ou alphanumériques sous forme de barres et d'espaces intermédiaires facilement lisible, dont l'épaisseur peut varier en fonction des données, mais aussi en fonction du type de code-barres employé, par exemple le code EAN 13, le code 128 ou le code 39.

Pour lire ces données, on emploie en général un lecteur de code-barres (appelé aussi «Douchette»), ce qui est nettement plus fiable qu'une saisie au clavier des lettres et chiffres correspondants, surtout s'il s'agit d'un code-barres avec clé de contrôle.

Il y a deux types de représentation graphique : Code à barre 1D ou 2D. Ces codes-barres sont omniprésents dans notre vie quotidienne, ils sont pré-imprimés sur tous les articles courants. [2]

3. La structure d'un code-barres

Un code-barres est une série de lignes verticales de largeur variable (appelées barres) et d'espaces. L'ensemble des barres et des espaces est appelé "éléments". Il existe différentes combinaisons de barres et d'espaces représentant différents caractères.



Figure 1 : La structure d'un code-barres.

Lors de la lecture d'un code-barres, le faisceau lumineux émis par le scanner est absorbé par les barres sombres sans être réfléchi, alors qu'il est réfléchi par les espaces clairs. A l'intérieur du scanner, un détecteur photo cellulaire reçoit la lumière réfléchie et la convertit en un signal électrique.

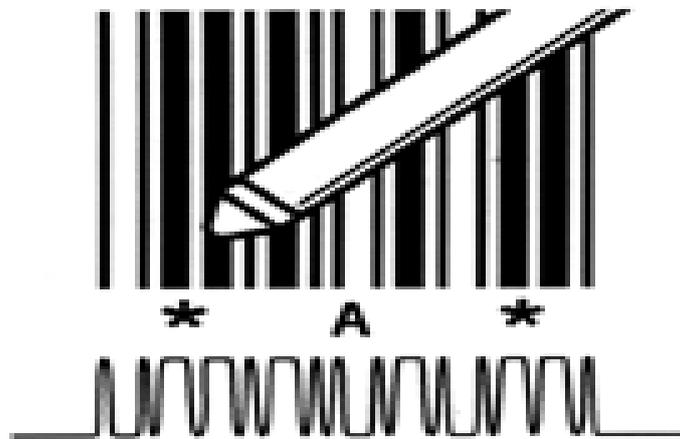


Figure 2 : Création d'un signal électrique.

Ainsi, lorsqu'un crayon optique lit un code-barres, le scanner crée un signal électrique faible pour les espaces (lumière réfléchie) et un signal électrique fort pour les barres (rien n'est réfléchi). La durée du signal électrique détermine si les éléments sont larges ou étroits. Le décodeur d'un lecteur de codes-barres peut convertir ce signal en caractères représentés par le code-barres. Les données décodées sont ensuite transmises à l'ordinateur dans un format traditionnel. [3]

4. Les différents types de codes-barres

Les codes-barres utilisent divers protocoles de codification ou symbologies, qui diffèrent en fonction des contraintes d'utilisation ou de normalisation.

Les trois types de codes-barres les plus utilisés :

- les codes-barres unidimensionnels ou linéaires.
- les codes-barres linéaires empilés.
- les codes-barres à deux dimensions.

4.1. Les codes-barres unidimensionnels ou linéaires

4.1.1. Le code EAN

Le code-barres EAN (European Article Numbering) a été développé à partir du code américain UPC pour les besoins spécifiques du commerce européen. Il existe deux variantes, l'une à 8 chiffres

et l'autre à 13 chiffres, la seconde étant la plus utilisée. Le code à barres EAN-13 se décompose ainsi :

- **Le préfixe :** qui représente le code du pays ayant délivré le numéro de participant (2 ou 3 chiffres).
- **Le numéro de participant :** qui est délivré par l'organisation EAN du pays concerné (les 4 ou 5 chiffres suivants).
- **Le numéro d'article :** du producteur de l'objet étiqueté sur 5 chiffres.
- **Le chiffre de contrôle :** ou "Check Digit", qui est calculé selon les 12 premiers chiffres qui composent le code.



Figure 3: Le code EAN-13.

Ce code est universellement utilisé dans le commerce de détail européen. Le commerce de détail américain utilise un code un peu antérieur (UPC), dont il existe plusieurs variantes.

4.1.2. Le code 39

Les caractéristiques du Code 39 sont les suivantes :

- Il est de longueur variable. Alphanumérique, il permet de codifier les 26 lettres majuscules, les 10 chiffres (0-9) ainsi que 8 caractères spéciaux (- , . , espace , * , \$, / , + , %).
- Il commence et finit toujours par le caractère « * ». Chaque caractère de code barre est composé de 9 éléments : 5 barres et 4 espaces.
- Chaque barre ou espace est "large" ou "étroit" et 3 parmi les 9 éléments sont toujours "larges". C'est d'ailleurs ce qui est à l'origine de son nom : Code 39.

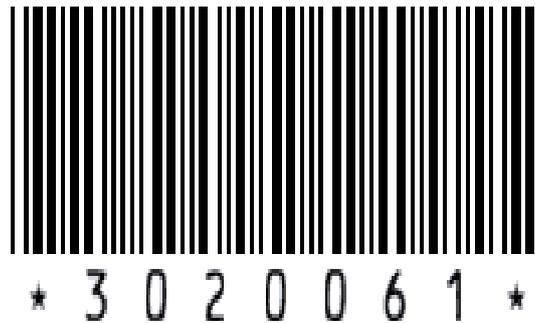


Figure 4 : Le code 39.

Ce code est utilisé pour la vente des médicaments en pharmacie, en France et dans d'autres pays européens.

Il existe aussi d'autres codes-barres linéaires tels que :

- ✓ Le **Code 93** : qui est un code 39 plus compact et plus sûr, mais moins utilisé.
- ✓ Le **Code 128** : qui comprend 3 variantes (A, B et C). Il est utilisé dans les transports et la logistique (cartons, palettes, conteneurs).
- ✓ Le **Code 2/5** : qui comprend trois variantes encore en usage (standard, entrelacé et IATA). La version IATA est utilisée pour l'étiquetage des bagages dans le transport aérien.

4.2. Les codes-barres linéaires empilés

4.2.1. Le code PDF 417

Les caractéristiques du Code-barres "PDF 417" sont les suivantes :

- Il est de longueur variable et peut comprendre jusqu'à 1.850 caractères alphanumériques ou 2.710 caractères numériques.
- Il permet d'imprimer beaucoup d'information sur très peu de surface : 180 caractères alphanumériques par cm².



Figure 5: Le code PDF 417.

La grande capacité du code PDF 417 est mise à profit lorsque des informations détaillées doivent impérativement être attachées à l'objet identifié, comme c'est le cas dans le transport des matières dangereuses par exemple.

4.2.2. Le code 16K

Le code 16K a été développé en 1989 par Ted Williams (l'inventeur des codes-barres), qui a également développé le Code 128. La structure du code 16K est basée sur celle du Code 128. Les caractéristiques du code 16K sont les suivantes :

- Il est de longueur variable et permet de codifier les 128 premiers caractères ASCII. Sa densité maximum est de 32 caractères alphanumériques, ou 65 caractères numériques, par cm².
- Il comprend 2 à 16 lignes de 5 caractères ASCII.

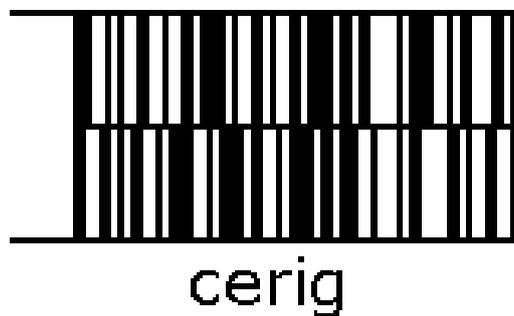


Figure 6 : Le code 16K.

Le code 16K est utilisé dans de nombreux domaines, à l'exception du commerce de détail.

4.3. Les codes-barres à deux dimensions

4.3.1. Le Code One

Les caractéristiques du code-barres Code One sont les suivantes :

- Il est de longueur variable, et peut comprendre jusqu'à 2.218 caractères alphanumériques ou 3.550 caractères numériques.
- Il permet d'imprimer beaucoup d'information sur très peu de surface : 500 caractères alphanumériques sur 1,6 cm².

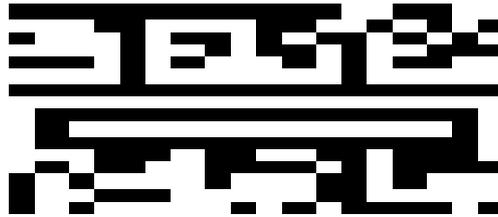


Figure 7 : Le code One.

Ce code ne semble pas être très utilisé.

4.3.2. Le code DataMatrix

Les caractéristiques du Code DataMatrix sont les suivantes :

- Il est de longueur variable, il peut comprendre jusqu'à 2.335 caractères alphanumériques ou 3.116 caractères numériques.
- Il permet d'imprimer beaucoup d'information sur très peu de surface.
- Il incorpore un système de correction d'erreur de lecture.



Figure 8 : Le code DataMatrix.

Ce code est utilisé dans l'industrie électronique pour marquer les circuits imprimés et les circuits intégrés. [4]

4.3.3. Le QR code

Le sigle « QR » dérive de « Quick Response » car le contenu peut être décodé rapidement. Les caractéristiques du code-barres QR code sont les suivantes :

- Le QR code est un code-barres à deux dimensions (code 2D ou code matriciel).
- Pouvant stocker jusqu'à 7089 caractères numériques.
- Pouvant stocker beaucoup d'informations tout en étant petit et rapide à scanner. [5]



Figure 9 : Le QR code.

5. Une comparaison entre les différents types de codes-barres

| | | Longueur variable | Caractères alphanumériques | Caractères numériques | La taille d'information | Rapidité de lecture |
|--|--------------------|-------------------|----------------------------|-----------------------|-------------------------|---------------------|
| Les codes à barres unidimensionnels ou linéaires | Le code EAN | X | X | ✓ | 13 | Rapide |
| | Le code 39 | ✓ | ✓ | X | 26 | Moyen |
| | le Code 93 | ✓ | ✓ | X | 26 | Moyen |
| | le Code 128 | ✓ | ✓ | X | 26 | Moyen |
| | le Code 2/5 | ✓ | X | ✓ | 26 | Moyen |
| Les codes à barres linéaires | Le code PDF 417 | ✓ | ✓ | ✓ | 1.850 | Rapide |
| | Le code 16K | ✓ | ✓ | ✓ | 32 | Moyen |
| Les codes à barres à deux dimensions | Le Code One | ✓ | ✓ | ✓ | 2.218 | Rapide |
| | Le code DataMatrix | ✓ | ✓ | ✓ | 2.335 | Rapide |
| | le QR code | ✓ | ✓ | ✓ | 7089 | Très rapide |

Tableau 1 : Une comparaison entre les types de codes-barres.

6. Le type de code barre choisit pour notre application

Dans notre application on va utiliser le type **QR code**. Ce code a de nombreux avantages, on cite:

- ✓ La redondance : Dans un code barre, la même information existe sur toute la hauteur des barres. Même si le code barre est détruit sur une certaine hauteur, il restera le plus souvent lisible. Cette redondance de l'information sur toute la hauteur facilite la lecture face aux défauts d'impression et aux conditions d'utilisation.
- ✓ Pouvoir stocker beaucoup d'information tout en étant petit et rapide à scanner. Le QR Code est un code-barres pouvant stocker jusqu'à 7089 caractères numériques, 4296 caractères alphanumériques (contrairement au code-barres « traditionnel » qui lui ne peut stocker que de 10 à 13 caractères) ou 2953 octets (environ 3ko).
- ✓ L'augmentation des lecteurs de codes 2D directement intégrés dans les téléphones mobiles facilite son utilisation.
- ✓ Un faible coût de revient. [6]

7. Les lecteurs de codes-barres

Un lecteur de code-barres est formé d'un décodeur et d'un scanner (un câble est également requis pour connecter le décodeur à l'ordinateur ou au terminal). La fonction de base d'un scanner est de lire un code-barres et de produire un signal électrique correspondant à la configuration des barres et des espaces. Un décodeur est généralement un boîtier séparé qui accepte les modèles numérisés de barres et d'espaces, les décode en données équivalentes avant de les transmettre immédiatement ou par lots à l'ordinateur.

7.1. Le crayon lecteur

C'est le plus simple des appareils de lecture. Le crayon effectue une lecture par passage manuel sur le code à barres. Réalisée au contact, la qualité de la lecture varie selon l'inclinaison du crayon.

Son petit prix en fait un modèle adapté aux faibles besoins de lecture, éventuellement un outil de contrôle de l'information codée.

7.2. Le lecteur CCD

La douchette CCD autorise une lecture automatique du code à barres, nul besoin de parcourir le code et chacune de ses barres. La distance de lecture est variable (jusqu'à 20 cm) et dépend des

réglages apportés au lecteur. Les modèles CCD ne comportent pas d'élément mécanique et présentent généralement une grande robustesse.

Plus économique que les modèles lasers, les lecteurs CCD sont capables de lire tous les codes à barres linéaires. Certains modèles sont capables de lire les codes empilés (PDF417).

7.3. Le lecteur laser

Construit autour d'un jeu de miroirs, le lecteur laser utilise un seul rayon lumineux généré par une diode laser. La source lumineuse est dense et précise, et autorise une lecture rapprochée ou distante de plusieurs mètres ainsi sur des objets ou documents en mouvement. Tout comme les lecteurs CCD, le lecteur laser réalise une lecture automatique du code. Nul besoin de parcourir le code dans sa longueur, un miroir motorisé le fait en réfléchissant le rayon laser de part et d'autre du code donnant ainsi l'illusion optique d'un trait continu. Certains lecteurs réalisent également ce balayage sur la hauteur du code (Raster en Anglais ou multitrames), d'autres encore démultiplient le balayage dans le but d'obtenir une grille sous laquelle le code peut être placé sous un degré d'orientation quelconque (omnidirectionnel).

8. L'utilité des codes à barres

Dans la grande distribution, le code à barres permet entre autres :

- d'entrer rapidement un produit en stock.
- d'en connaître l'origine.
- d'en faciliter le réapprovisionnement.
- d'automatiser la sortie des stocks et d'obtenir un inventaire simplifié permanent.
- d'optimiser les temps de traitement aux caisses.

Dans une usine de fabrication, le code à barres permet également :

- de connaître l'origine des matières premières.
- d'avoir en continu des informations actualisées sur l'état de production.
- d'aiguiller automatiquement un produit en fonction de ses caractéristiques, de ses défauts.
- d'identifier sa production, d'identifier les unités logistiques transmises à ses clients. [4]

Dans une entreprise, il permet :

- d'automatiser la gestion des biens.
- d'avoir une vue générale sur l'état des biens.
- déterminer plus facilement les besoins futures.

Conclusion

Les codes-barres procurent des avantages pour tous les opérateurs économiques (entreprises, producteurs, grossistes, distributeurs,..) intervenant tout le long de la chaîne commerciale et aux consommateurs. Dans les entreprises ils permettent l'élimination des erreurs d'inventaire et rendent la gestion plus efficace.

CHAPITRE 2

Technologies

Utilisées



Introduction

Dans ce chapitre nous présentons les différentes technologies utilisées pour réaliser notre application notamment la bibliothèque SDK pour le développement d'application Android et la bibliothèque ZXING pour la manipulation des codes-barres.

1. La plate-forme J2EE

J2EE (Java2 Enterprise Edition) est une extension de la plate-forme J2SE (Java Standard Edition). Cette plate-forme permet de développer des applications Web ou de bureau. Elle donne une spécification destinée aux éditeurs de logiciels qui désirent créer des Serveurs d'Applications compatibles J2EE. [7]

Les développements Java EE reposent sur un découpage en couches ou tiers, nous parlons alors d'applications multi-tiers. Trois grands tiers sont représentés :

- La couche présentation (tiers Web).
- La couche métier (tiers Métier ou tiers Business).
- La couche stockage des informations (tiers Enterprise Information System).

La spécification Java EE définit les composants suivants :

- Les composants Clients ou tiers Client sont des applications clientes (logiciel installé en local ou navigateur Web ou Applets) s'exécutant sur la machine des utilisateurs.
- Les composants Web ou tiers Web sont les technologies Servlets, Java Server Pages et Java-Server Faces.
- Les composants métier ou tiers Métier sont des composants Entreprises JavaBeans (EJB) représentant la logique métier, s'exécutant sur le serveur Java EE et dialoguant avec la couche de stockage (EIS : Enterprise Information System). [8]

2. Le système Android



Android est un système d'exploitation orienté dispositif mobiles, il s'agit donc d'un ensemble de logiciels qui sert d'interface entre le matériel (les composants du téléphone, d'une tablette...) et les logiciels applicatifs. Android est un système d'exploitation open source (disponibilité du code, importante communauté d'utilisateurs) qui offre un environnement de développement gratuit (Programmation en Java ou en langage C + Kit de développement SDK Android). [9]

2.1. L'architecture Android

L'architecture du système Android est illustrée dans la figure suivante :

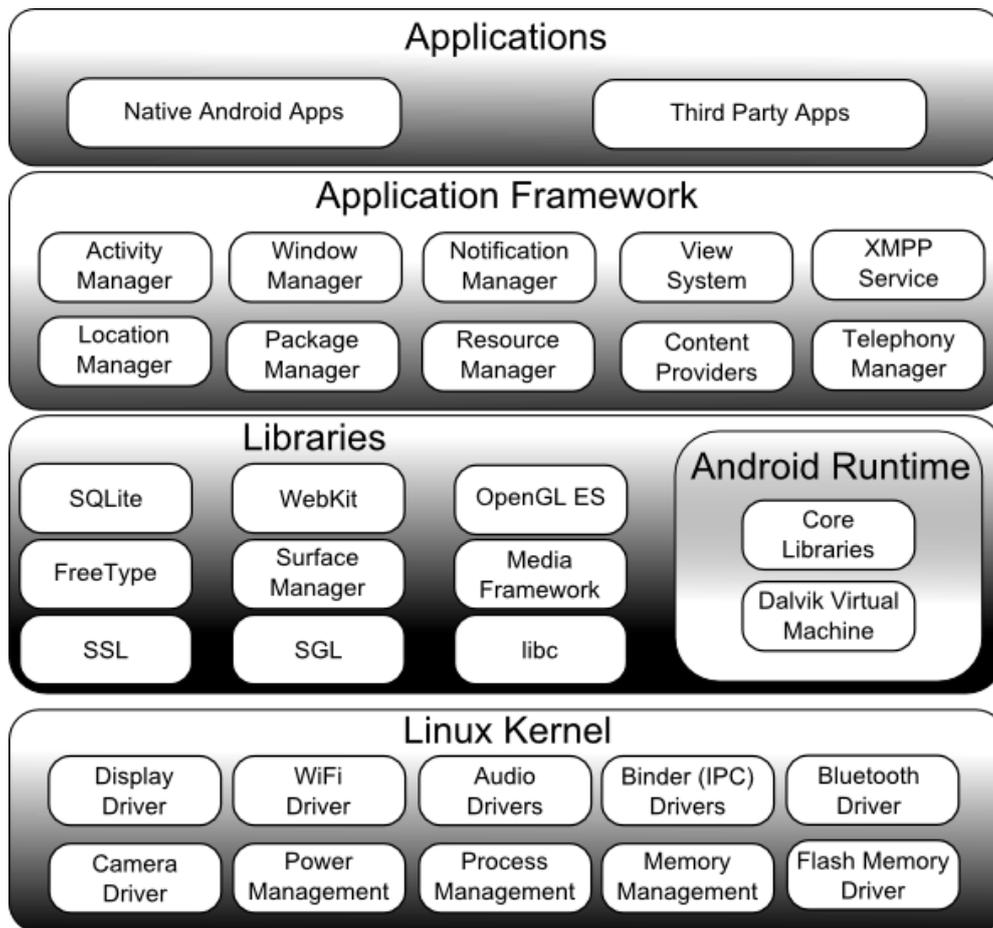


Figure 1: Architecture du système Android.

2.1.1. La couche Applications

Android fournit dans cette couche des applications natives comme un client de mail, des programmes pour envoyer des SMS, d'agenda, de navigateur web, de contacts personnels.

2.1.2. La couche "Application Framework"

En fournissant une plateforme de développement ouverte, Android offre aux développeurs la possibilité de créer des applications extrêmement riches et innovants. Les développeurs sont libres de profiter du matériel périphérique, les informations de localisation d'accès, exécutez les services d'arrière-plan, définir des alarmes, ajouter des notifications de la barre d'état, et beaucoup plus.

2.1.3. La couche "Libraries" (bibliothèques)

En interne, Android inclut un ensemble de bibliothèques C et C++ utilisées par de nombreux composants de la plateforme Android. Ces bibliothèques sont en réalité accessibles au développeur par l'intermédiaire du Framework Android. En effet, le Framework Android effectue, de façon interne, des appels à des fonctions C/C++ beaucoup plus rapides à exécuter que des méthodes Java standard. La technologie Java Native Interface (JNI) permet d'effectuer des échanges entre le code Java et le code C et C++.

2.1.4. Le noyau Android

Android repose sur un noyau Linux (version 2.6) qui gère les services du système, comme la sécurité, la gestion de la mémoire et des processus, la pile réseau et les pilotes. Il agit également comme une couche d'abstraction entre le matériel et la pile logicielle. [10]

2.2. Les versions d'Android

Les différentes versions du système Android est illustré dans la figure suivante :



Figure 2 : Les différentes versions du système Android.

2.3. Développement d'applications pour Android

Au niveau du développement d'applications, Android utilise un Framework Java et propose un environnement de développement complet comprenant un émulateur, un débogueur et un analyseur de mémoire et de performance. Pour s'exécuter sur les appareils mobiles Android, le programme en java doit être converti un exécutable compact Dalvik (.dex), un format adapté aux systèmes limités en termes de mémoire et de vitesse du processeur. Dalvik est le nom de la machine virtuelle qui roule sur les appareils mobiles Android. [11]

2.3.1. Le Sdk Android

Un SDK, pour Software Development Kit est un ensemble d'outils logiciels permettant de faciliter le développement d'un logiciel sur une plateforme donnée. Le SDK d'Android est multiplateforme, son émulateur permet de tester les applications sur plusieurs versions différentes d'Android, plusieurs tailles d'écran, et ce, même si on n'a pas d'appareil physique. [12]



2.3.2. Comment ça marche ?

Le SDK contient du code, permettant de concevoir une interface ou une partie d'une interface numérique (web, mobile, jeux, logiciels de recherches...). Ce code est conçu avec le langage de programmation correspondant au terminal (ordinateur, téléphone, tablette...) et au système de navigation ciblés.

2.3.3. Composants du Sdk Android

Le SDK Android est composé de plusieurs éléments pour aider les développeurs à créer et à maintenir des applications :

- Des API (interfaces de programmation).
- Des exemples de code.
- De la documentation.
- Des outils, parmi lesquels un émulateur permettant de couvrir quasiment toutes les étapes du cycle de développement d'une application. [13]

3. Android Studio

C'est un environnement de développement spécialisé dans le développement d'applications Android, et un outil pour gérer l'installation du SDK Android sur système. Il est basé sur IntelliJIDEA.

Android Studio permet d'avoir un IDE totalement dédié aux développements Android, pour d'autre type de développements JAVA il faut utiliser IntelliJ IDEA ou Eclipse. Cette restriction peut paraître contraignante mais elle permet de ne pas mélanger différents types de développements sur un même IDE et donc évite la surcharge de l'IDE et les erreurs de compatibilités entre les différents types de développements. [14]

Android Studio dispose d'une interface intuitive et simple d'utilisation. Son logo est présenté dans la figure suivante :



Figure 3 : Android Studio.

4. Le SGBD MySQL

MySQL est un système de gestion de base de données, SGBD. Il fait partie des serveurs de bases de données relationnelles SQL les plus utilisés au monde. Il est en concurrence avec Oracle, Informix et Microsoft SQL Server.

MySQL présente différentes caractéristiques, entre autre, on peut citer :

- Il est très performant et fortement sécurisées.
- Il est multithread et multiutilisateur.
- C'est un serveur de bases de données relationnelles libre. [15]

5. Le SGBD SQLite

SQLite est un système de base de données qui a la particularité de fonctionner sans serveur, on dit aussi "standalone" ou "base de données embarquée". On peut l'utiliser avec beaucoup de langages.



L'intérêt c'est que c'est très léger et rapide à mettre en place, on peut s'en servir aussi bien pour stocker des données dans une vraie base de données sur une application pour Smartphone (iPhone ou Android), pour une application Windows, ou sur un serveur web.

Une base de données SQLite est bien plus performante et facile à utiliser que de stocker les données dans des fichiers XML ou binaires, d'ailleurs ces performances sont même comparables aux autres SGBD fonctionnant avec un serveur comme MySQL, Microsoft SQL Server ou PostgreSQL. [16]

6. Les Sockets

Les sockets permettent de faire échanger des données entre deux applications distantes.

Différents types de sockets sont utilisés :

6.1. Stream Sockets (TCP)

- Permet d'établir une communication en mode connecté.
- Si la connexion interrompue l'application informée.

6.2. Datagram Sockets (UDP)

- Permet d'établir une communication en mode non connecté.
- Les données sont envoyées sous forme de paquets indépendants de toute connexion. Ils sont Plus rapides mais moins fiable que TCP. [17]

7. La Bibliothèque ZXING

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. Différents format sont pris en compte par la plateforme on cite :

- EAN-8 and EAN-13.
- Code 39.
- Code 128.
- QR Code.

- Data Matrix.
- PDF 417. [18]

Conclusion

Nous avons vu à travers ce chapitre les différentes technologies utilisées de notre application. Nous avons cité les différents outils utilisés lors du développement de notre système. ZXING pour la manipulation des codes-barres, JEE pour la programmation des serveurs, Android studio et la SDK Android pour la programmation des Smartphones, MySQL et SQLite pour le stockage et la manipulation des données, et enfin les sockets pour la communication entre les Smartphones et les serveurs.

CHAPITRE 3

Analyse du système



Partie1 : Etude préliminaire

Introduction

L'étude préliminaire ou (pré étude) est la première étape de notre processus de développement. Elle consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte ou des diagrammes très simples. Elle prépare les étapes plus formelles de capture des besoins fonctionnels et capture des besoins techniques.

Dans un premier temps, nous allons introduire l'étude de cas qui servira de fil conducteur tout au long du développement de notre application en donnant une version textuelle préliminaire du cahier des charges. Dans un second temps, nous commencerons par la détermination des besoins fonctionnels en considérant le système comme une boîte noire. Après avoir identifié les acteurs qui interagissent avec le système, nous développerons un premier modèle UML de niveau contacte, pour pouvoir établir précisément les fonctionnalités du système.

1. L'élaboration du cahier de charges

- Les choix techniques.
- La présentation du projet.
- Le recueil des besoins fonctionnels.
- Identification des messages.
- Identification des acteurs.
- Modélisation du contexte.

1.1. Choix techniques

Nous souhaitons utiliser une approche itérative et incrémentale, fondée sur le processus en Y (méthode 2TUP) qui utilise le langage standard de modélisation UML, nous avons choisi d'un certain nombre de techniques clés pour ce projet :

- ✓ JAVA pour la programmation de la logique applicative de ce projet dans les Plateforme JEE et SDK android.
- ✓ SQLite, MySQL pour le stockage et la gestion des données.

1.2. Présentation du projet

Le but de ce projet est le développement d'un système de gestion d'inventaires basé sur les codes-barres.

1.3. Recueil des besoins fonctionnels

Nous avons effectué une recherche pour identifier les besoins de l'application, et ceci afin de répondre à la fois aux besoins des administrateurs et des agents.

1.3.1. Les besoins fonctionnels des administrateurs (au niveau du serveur de l'application)

Ils font les actions suivantes :

- Mise à jour des biens.
- Créer des codes barre.
- Imprimer des codes barre.
- Créer des catégories.
- Déterminer l'état des lieux.
- Créer de nouveau emplacement.
- Rechercher des biens.
- Exporter les informations à d'autre logiciel.
- Gérer les utilisateurs.

1.3.2. Les besoins fonctionnels des agents (au niveau des Smartphones)

- Scanner automatiquement les biens.
- Transférer les biens scannés.
- Synchroniser les informations d'emplacements avec le serveur.

1.4. Recueil des besoins opérationnels

Ainsi l'application doit assurer certaines qualités :

- **Sécurité** : chaque utilisateur de l'application doit être authentifié.
- **Temps de réponse** : doit être acceptable.
- **Interface graphique** : doit être simple à utiliser et convivial.

2. Description du contexte

2.1. Identifications des acteurs

Dans la suite nous présenterons les différents acteurs qui interagissent avec le système.

2.1.1. Au niveau du serveur

A ce niveau nous avons une application avec un acteur administrateur. Cet acteur accède de manière sécurisée au système. Il peut :

- Mettre à jour les biens.
- Créer des codes barre.
- Imprimer des codes barre.
- Créer des catégories.
- Déterminer l'état des lieux.
- Créer de nouveaux emplacements.
- Recherche des biens.
- Exporter les informations à d'autre logiciel.
- Gérer des utilisateurs.

2.1.2. Au niveau du Smartphones

A ce niveau nous avons une application avec un acteur agent. Cet acteur accède de manière sécurisée au système. Il peut :

- Scanner automatiquement les biens.
- Transférer bien scanné.
- Synchronisation les informations d'emplacements avec le serveur.

2.2. Identifications des messages

Notre système est une collection de deux systèmes : le premier au niveau serveur et le deuxième au niveau des Smartphones. Le système interagit avec les différents acteurs à l'aide des messages échangés entre eux.

3. Le diagramme de cas d'utilisation pour les besoins fonctionnels

Après identification des cas d'utilisation et leurs acteurs, nous allons les représenter graphiquement sur un diagramme de cas d'utilisation qui représente le fonctionnement du système vis-à-vis de l'utilisateur :

3.1. Au niveau du serveur

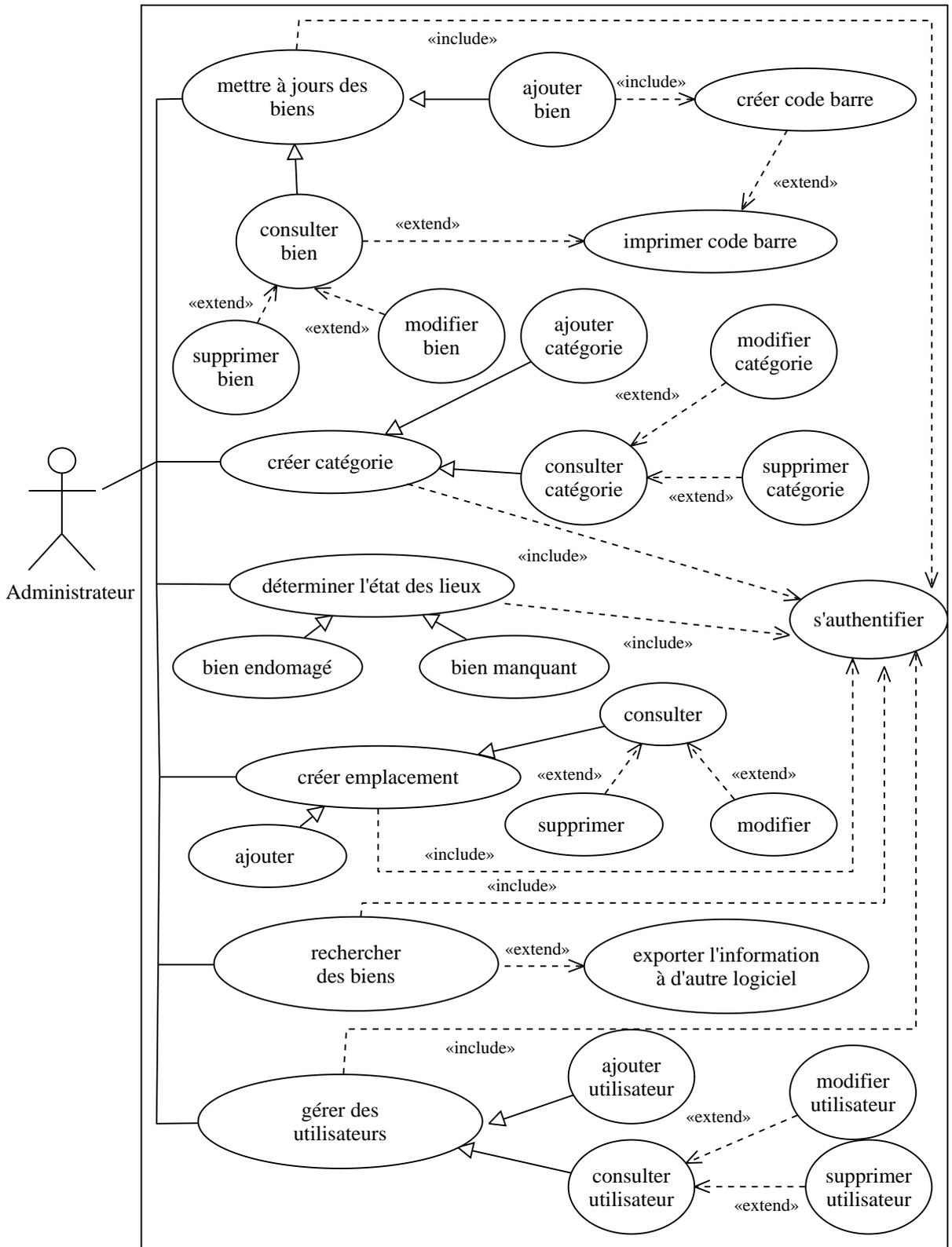


Figure 1 : Diagramme du cas d'utilisation Au niveau de l'administrateur.

3.2. Au niveau de l'utilisateur Smartphone

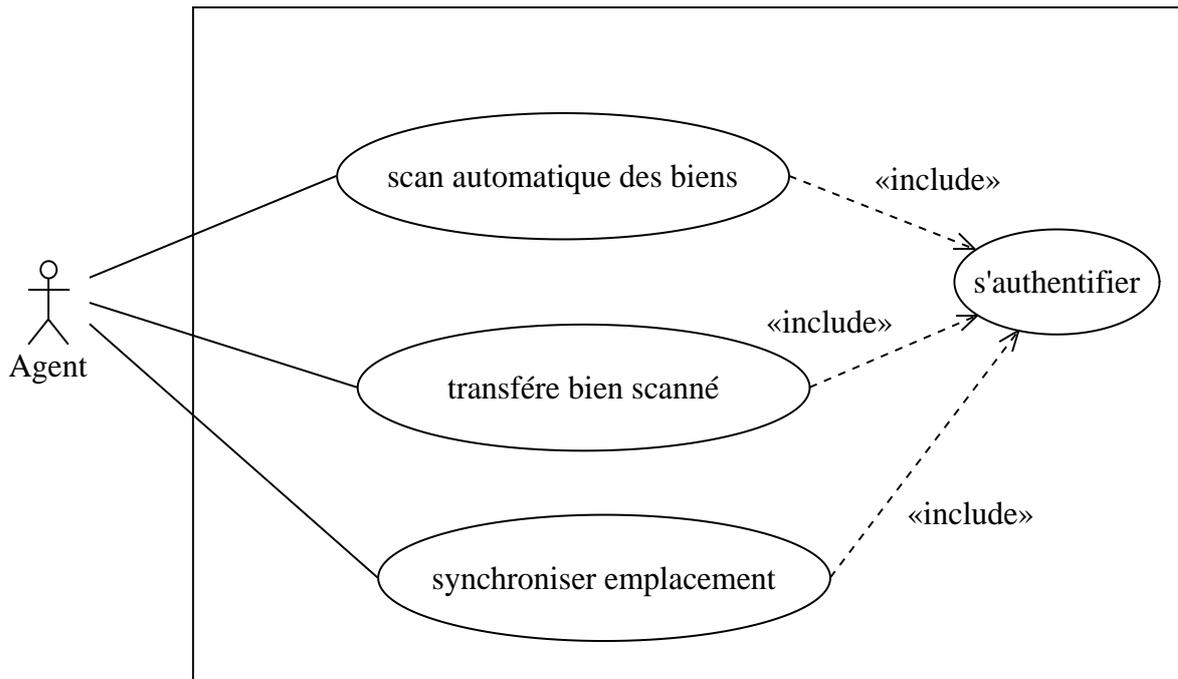


Figure 2 : Diagramme du cas d'utilisation au niveau de l'agent.

4. Description détaillée des différents cas d'utilisation

4.1. Au niveau du serveur

❖ Cas d'utilisation « S'authentifier »

Intention : l'agent désire à accéder au système avec sécurité.

Action : l'agent authentifie pour accéder au système.

Identification des besoins :

l'authentification.

Pré condition :

- L'agent est inscrit dans le système.

Scénario nominal

- L'agent saisit ses identifiants dans le formulaire d'authentification.
- Le système de niveau administrateur le dirige vers l'espace qui lui est approprié.

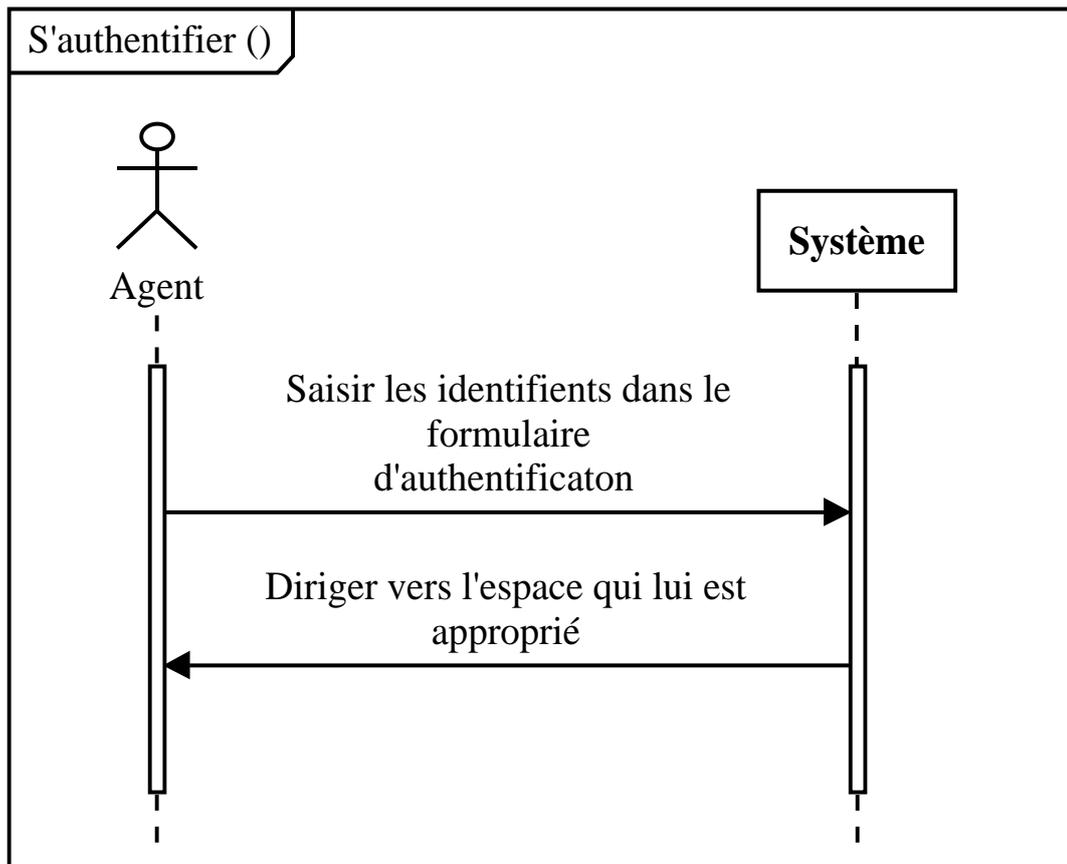


Figure 3 : Diagramme de séquence du cas d'utilisation « S'authentifier ».

❖ Cas d'utilisation « Gestion des catégories »

Sommaire d'identification

Titre : Gestion_ catégorie ().

But : Ajouter / consulter /modifier /supprimer : catégorie.

Acteur : administrateur.

Description des Enchaînements

Pré conditions : L'administrateur doit être authentifié.

- L'administrateur possède des informations concernant une catégorie.

Post-condition :

- Les informations ont été enregistrées.

Scénario nominal : Ce cas d'utilisation commence lorsque l'administrateur demande au système la gestion de catégorie.

- 1- l'administrateur accède au système.
- 2- Le système présente l'état actuel.
- 3-
 - l'administrateur choisit l'opération ajouter catégorie sous cas : ajouter catégorie.
 - l'administrateur choisit l'opération modifier catégorie sous cas : modifier catégorie.
 - l'administrateur choisit l'opération supprimer catégorie sous cas : supprimer catégorie.
 - l'administrateur choisit l'opération consulter catégorie sous cas : consulter catégorie.
- 4- Le système enregistre les modifications.

Scénario alternatif

- Lorsque l'administrateur annule l'accès au sous cas, il retourne à l'étape 2.
- Lorsque l'administrateur saisi des informations incorrectes ou incomplètes, le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 2.

Scénario exceptionnel

- Lorsque l'administrateur choisit l'option annulé le système quitte la page de création.

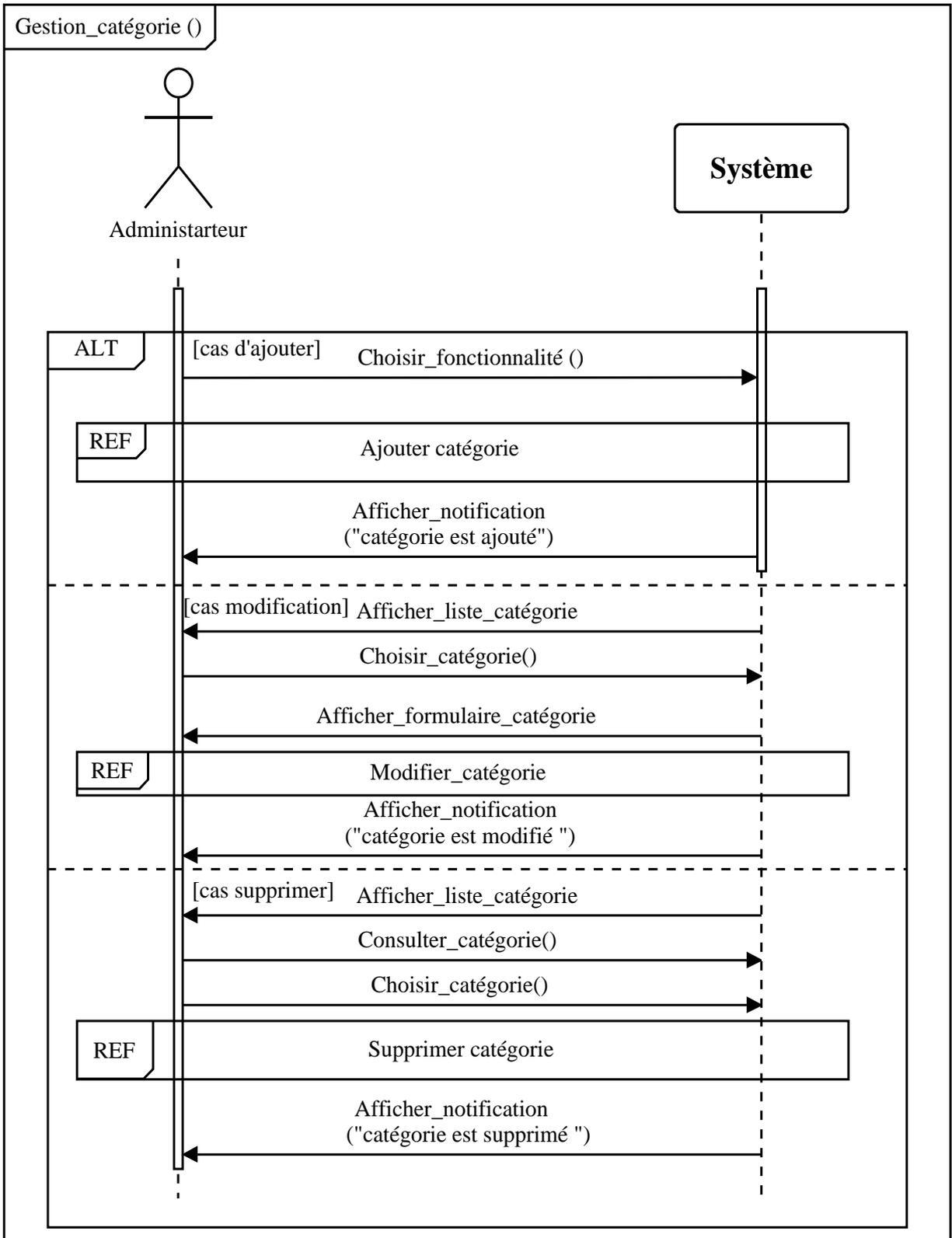


Figure 4 : Diagramme de séquence du cas d'utilisation « Gestion catégorie ».

❖ Cas d'utilisation « Ajouter catégorie »

Sommaire d'identification

Titre: Ajouter_catégorie ().

But: Ajouter une catégorie.

Résumé: Ajouter une catégorie dans la liste des catégories.

Acteur : L'administrateur.

Description des Enchaînements

Pré conditions : L'administrateur doit être authentifié.

Post-condition : L'ajout est effectué.

Scénario nominal :

Ce cas d'utilisation commence lorsque l'administrateur demande au système d'ajouter une catégorie.

- 1- le système affiche un formulaire d'ajout.
- 2- l'administrateur remplit toutes les informations.
- 3- Le système vérifie les informations saisies par l'administrateur et affiche une notification «catégorie est ajouté».

Scénario alternatif

Lorsque l'administrateur saisit des informations incorrectes ou incomplètes :

- Le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 1.

Scénario exceptionnel

Lorsque l'administrateur choisit l'option annulé :

- Le système quitte la page d'ajout.

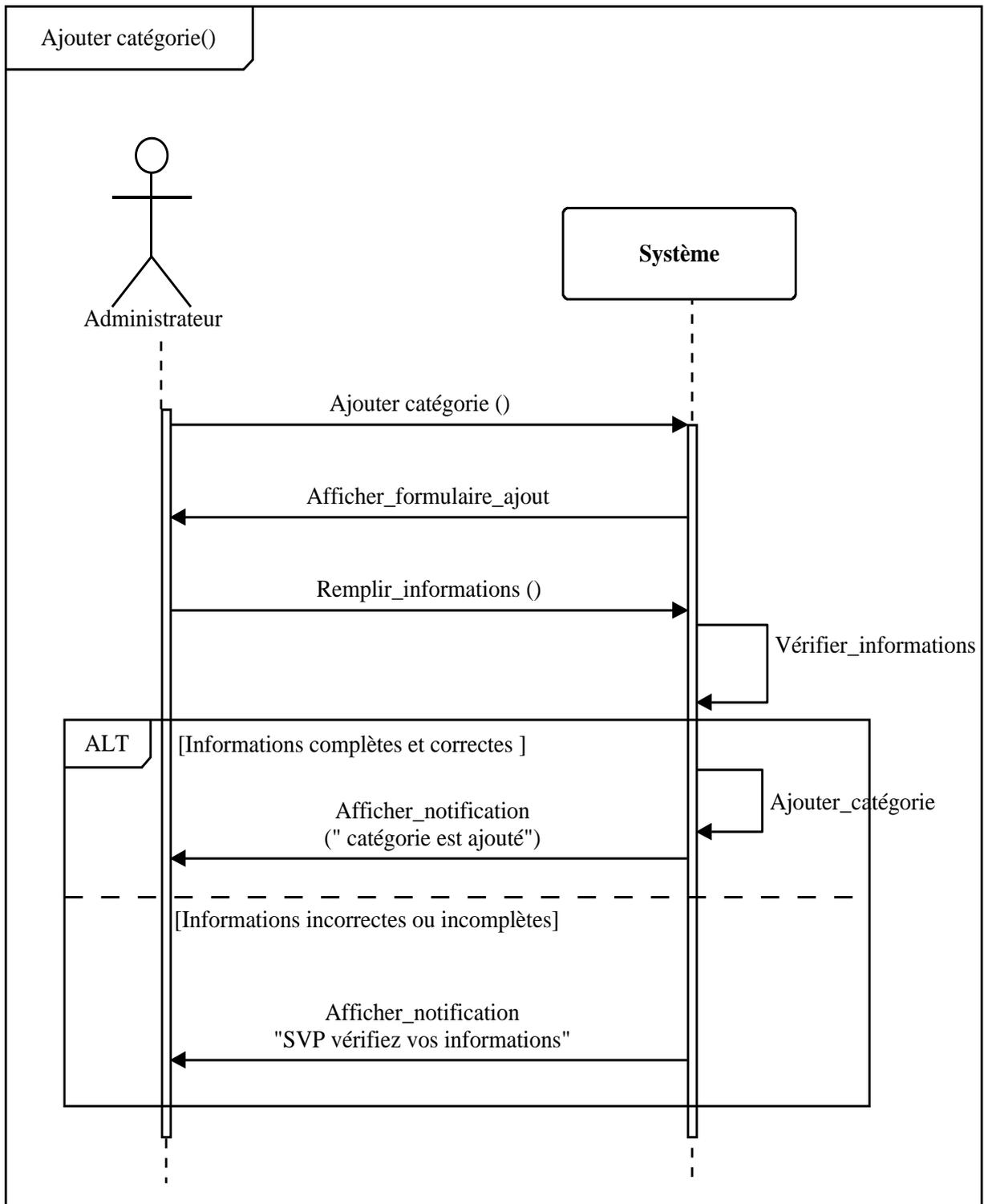


Figure 5: Diagramme de séquence du cas d'utilisation « Ajouter catégorie ».

❖ Cas d'utilisation « Consulter catégorie »

Sommaire d'identification**Titre :** Consulter_catégorie ().**But :** Afficher les catégories qui se trouvent dans la base de données.**Résumé :** Afficher la liste des catégories qui se trouvent dans la base de données.**Acteur :** Administrateur.**Description des Enchaînements****Pré condition :** l'administrateur doit être authentifié.**Post condition :** la liste des catégories est affichée.**Scénario nominal**

Ce cas d'utilisation commence lorsque l'administrateur demande au système une consultation des catégories.

- Le système affiche la liste des catégories.

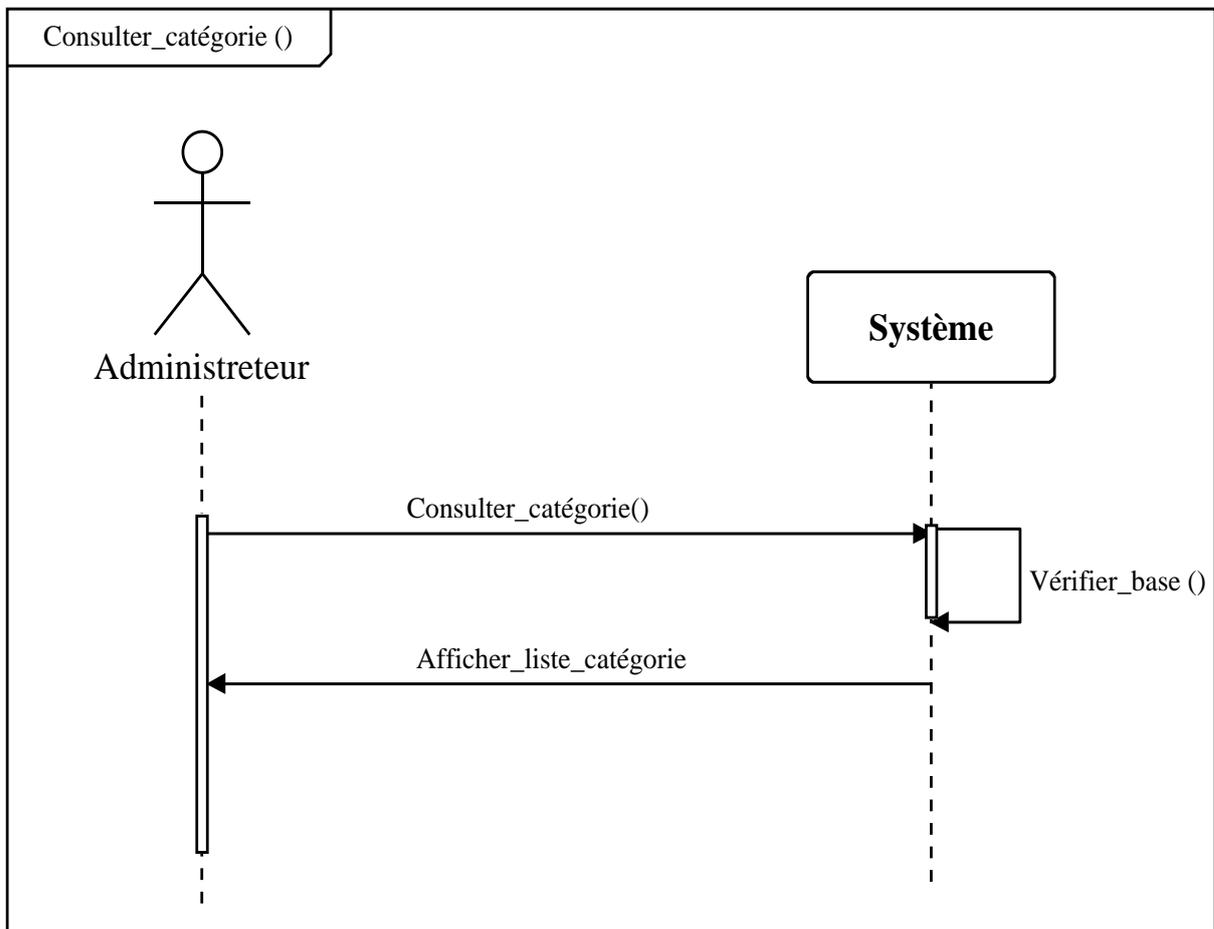


Figure 6: Diagramme de séquence du cas d'utilisation « Consulter catégorie »

❖ Cas d'utilisation « Modifier catégorie »

| |
|--|
| Sommaire d'identification |
| <p>Titre : Modifier_catégorie ().</p> <p>But : Modifier les informations d'une catégorie.</p> <p>Résumé : changer le profil d'une catégorie.</p> <p>Acteur : L'administrateur.</p> |
| Descriptions des enchainements |
| <p>Pré conditions</p> <ul style="list-style-type: none">• L'administrateur doit être authentifié.• catégorie existe déjà. <p>Post conditions : le profil catégorie est modifié.</p> <p>Scénario nominal</p> <p>Ce cas d'utilisation commence lorsque l'administrateur demande au système une modification de catégorie.</p> <ol style="list-style-type: none">1- L'administrateur consulte la liste des catégories.2- Le système affiche la liste les catégories.3- L'administrateur choisit l'un des catégories.4- Le système affiche un formulaire pour changer les informations de cette catégorie.5- L'administrateur modifie les champs qu'il veut dans le formulaire.6- Le système vérifie les informations saisis par l'administrateur et puis fait une mise à jour concernant cette catégorie dans la base de donnée. puis il affiche une notification «catégorie est modifié». <p>Scénario alternatif</p> <p>Lorsque l'administrateur remplit des informations incorrectes ou incomplètes :</p> <ul style="list-style-type: none">➤ Le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 4. <p>Scénario exceptionnel</p> <p>Lorsque l'administrateur choisit l'option annulé :</p> <ul style="list-style-type: none">➤ le système quitte la page. |

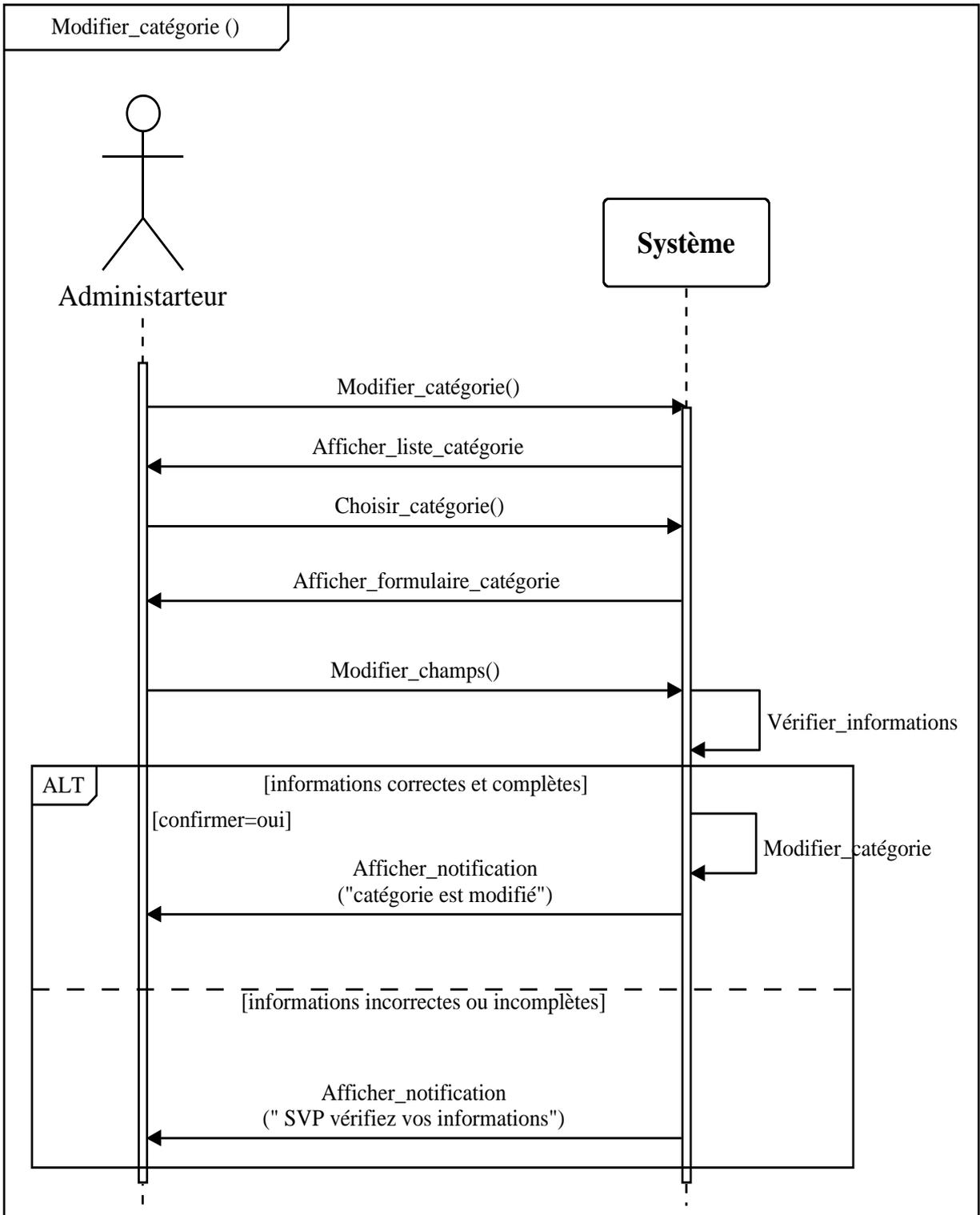


Figure 7: Diagramme de séquence du cas d'utilisation « Modifier catégorie »

❖ Cas d'utilisation « Supprimer catégorie »

| |
|--|
| Sommaire d'identification |
| Titre : Supprimer_catégorie (). But : Supprimer une catégorie. Résumé : Supprimer catégorie de la liste des usagers du système. Acteur : L'administrateur. |
| Descriptions des enchainements |
| Pré conditions <ul style="list-style-type: none">• L'administrateur doit être authentifié.• catégorie existe déjà. Post conditions : le profil catégorie est supprimé de la liste des usagers du système. Scénario nominal <p>Ce cas d'utilisation commence lorsque l'administrateur demande au système une suppression d'une catégorie.</p> <ol style="list-style-type: none">1- L'administrateur consulte la liste les catégories.2- Le système affiche la table des catégories.3- L'administrateur choisit une catégorie pour le supprimer.4- Le système demande une confirmation de la suppression.5- L'administrateur confirme la suppression.6- Le système fait une mise à jour dans la base de données, puis il affiche une notification <p>«catégorie est supprimé».</p> Scénario alternatif <ul style="list-style-type: none">➤ Lorsque l'administrateur ne confirme pas la suppression le système laisse la base de données avec son état primaire. Scénario exceptionnel : <ul style="list-style-type: none">➤ Lorsque l'administrateur choisit l'option annulé la suppression le système quitte la page. |

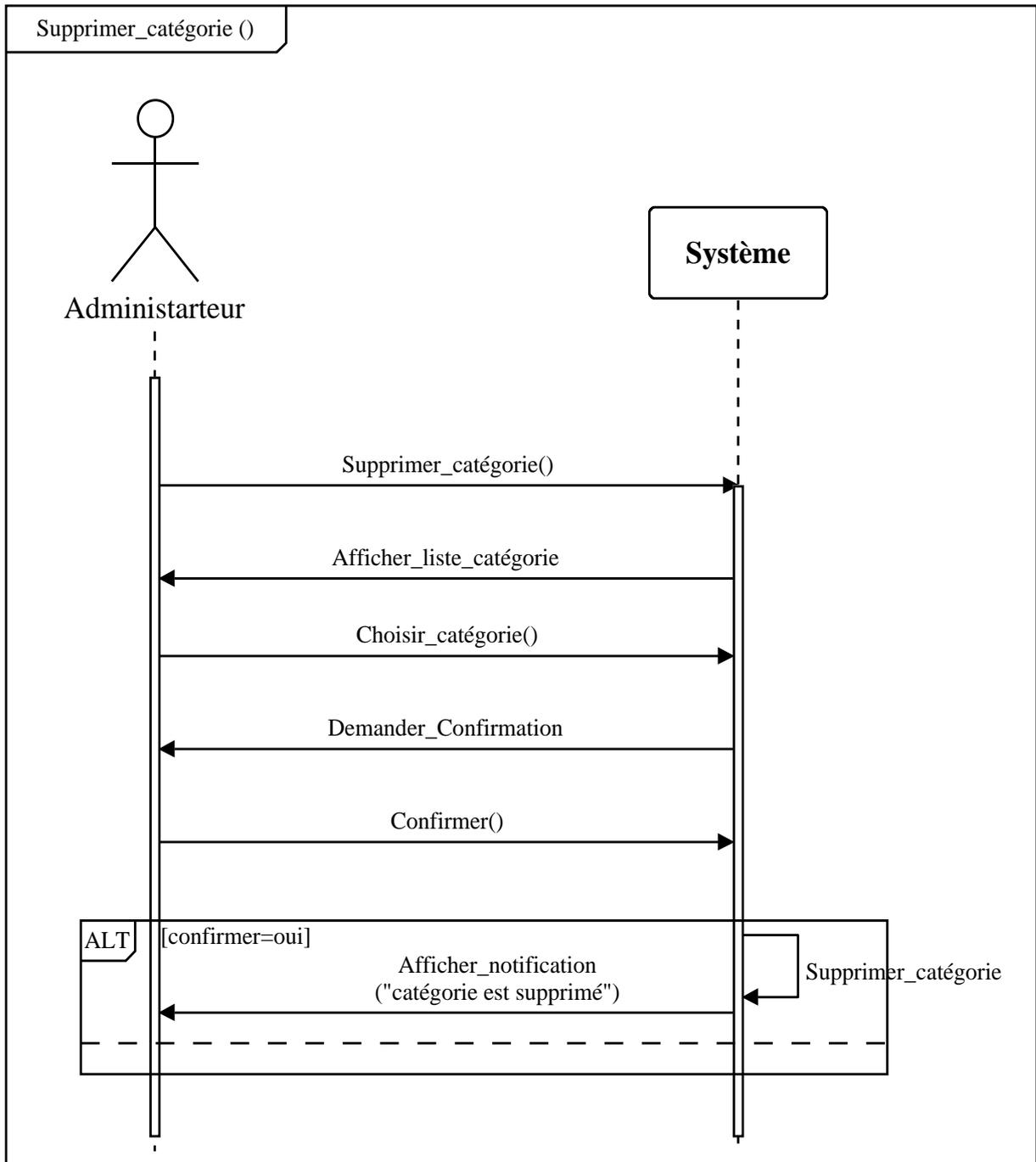


Figure 8: Diagramme de séquence du cas d'utilisation « Supprimer catégorie »

❖ Cas d'utilisation « Mise à jour des biens »

Sommaire d'identification

Titre : Mise_à_jour_Bien ().

But : Ajouter / consulter /modifier /supprimer: bien.

Acteur : administrateur.

Description des Enchaînements**Pré conditions**

- L'administrateur doit être authentifié.
- L'administrateur possède les informations concernant un bien.

Post-condition: les informations ont été enregistrées.

Scénario nominal : Ce cas d'utilisation commence lorsque l'administrateur demande au système la mise à jour des biens.

- 1- l'administrateur accède au système.
- 2- Le système présente l'état actuel.
- 3-
 - l'administrateur choisit l'opération ajouté bien sous cas : ajouter bien.
 - l'administrateur choisit l'opération modifié bien sous cas : modifier bien.
 - l'administrateur choisit l'opération supprimé bien sous cas : supprimer bien.
 - l'administrateur choisit l'opération consulté bien sous cas : consulter bien.
- 4- Le system enregistre les modifications.

Scénario alternatif

- Lorsque l'administrateur annule l'accès au sous cas, il retourne à l'étape 2.
- Lorsque l'administrateur saisi des informations incorrectes ou incomplètes. Le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 2.

Scénario exceptionnel

- Lorsque l'administrateur choisit l'option annulé le système quitte la page de création.

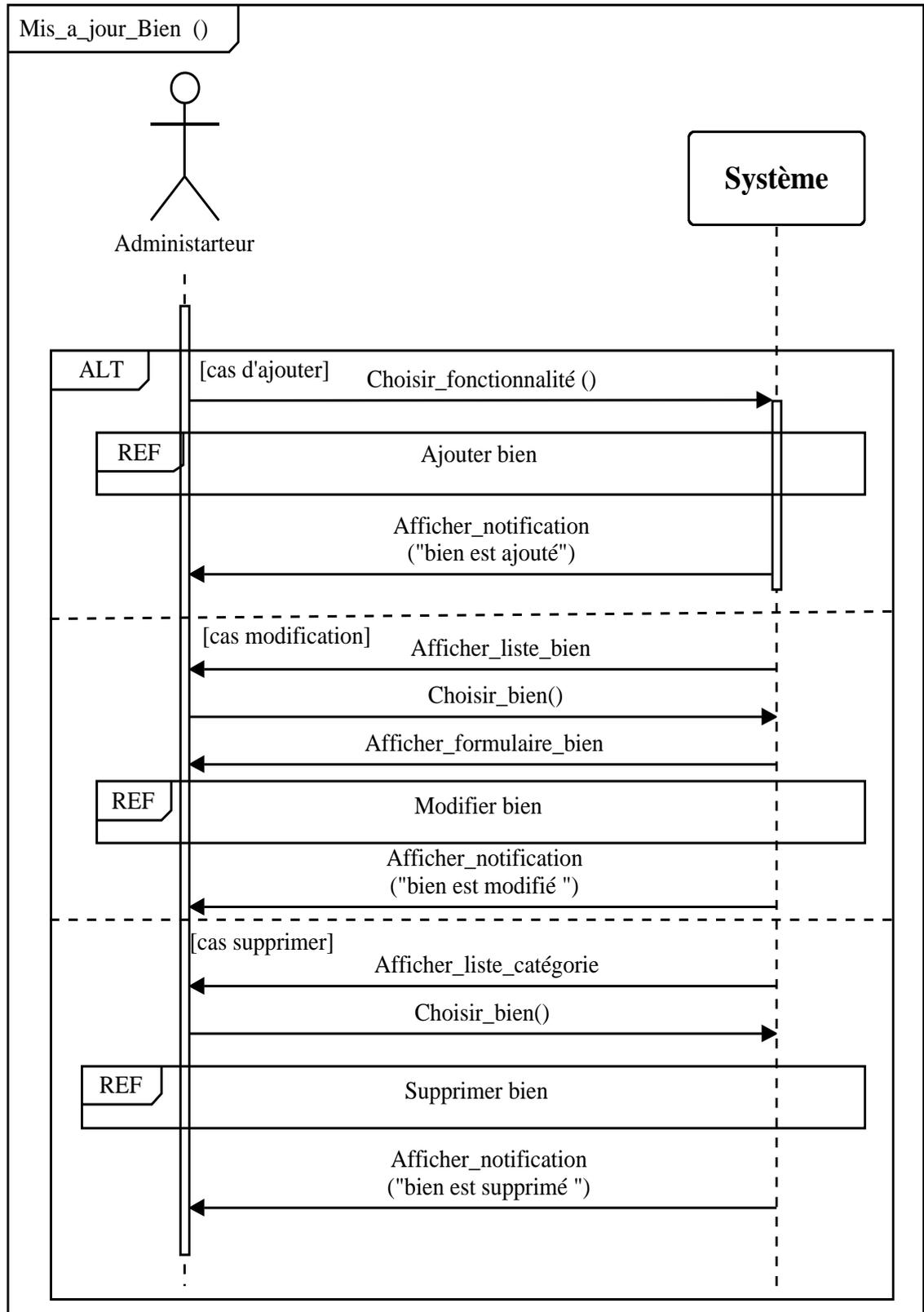


Figure 9: Diagramme de séquence du cas d'utilisation « Mise à jours bien »

❖ Cas d'utilisation « Ajouter bien »

| |
|--|
| Sommaire d'identification |
| Titre : Ajouter_bien (). But : Ajouter un bien puis créer son code barre et l'imprimer. Acteur : Administrateur. |
| Description des Enchaînements |
| Pré conditions: L'administrateur doit être authentifié. Post-condition: <ul style="list-style-type: none">• L'ajout est effectué.• Génération de code barre.• Impression de code barre. Scénario nominal : Ce cas d'utilisation commence lorsque l'administrateur demande au système d'ajouter un bien puis créer son code barre et l'imprimer. <ol style="list-style-type: none">1- le système affiche un formulaire d'ajout.2- l'administrateur remplit toutes les informations.3- Le système vérifie les informations complètes Et correctes saisies par l'administrateur et sauvegarder.5- lorsque le bien est ajouté, le système génère son code barre automatiquement.6- Le système sauvegarde le bien et ces informations dans la base de données.7- Le système affiche une notification «bien est ajouté».8- si le système veut imprimer ce code barre : L'administrateur choisit le bouton imprimer.9- le système affiche une notification « le code barre est imprimé». Scénario alternatif <ul style="list-style-type: none">➤ Lorsque l'administrateur ne confirme pas l'ajout, le système laisse la base de données avec son état primaire.➤ Lorsque l'administrateur saisit des informations incorrectes ou incomplètes, le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 1. Scénario exceptionnel <ul style="list-style-type: none">➤ Lorsque l'administrateur choisit l'option annulé, le système quitte la page de création. |

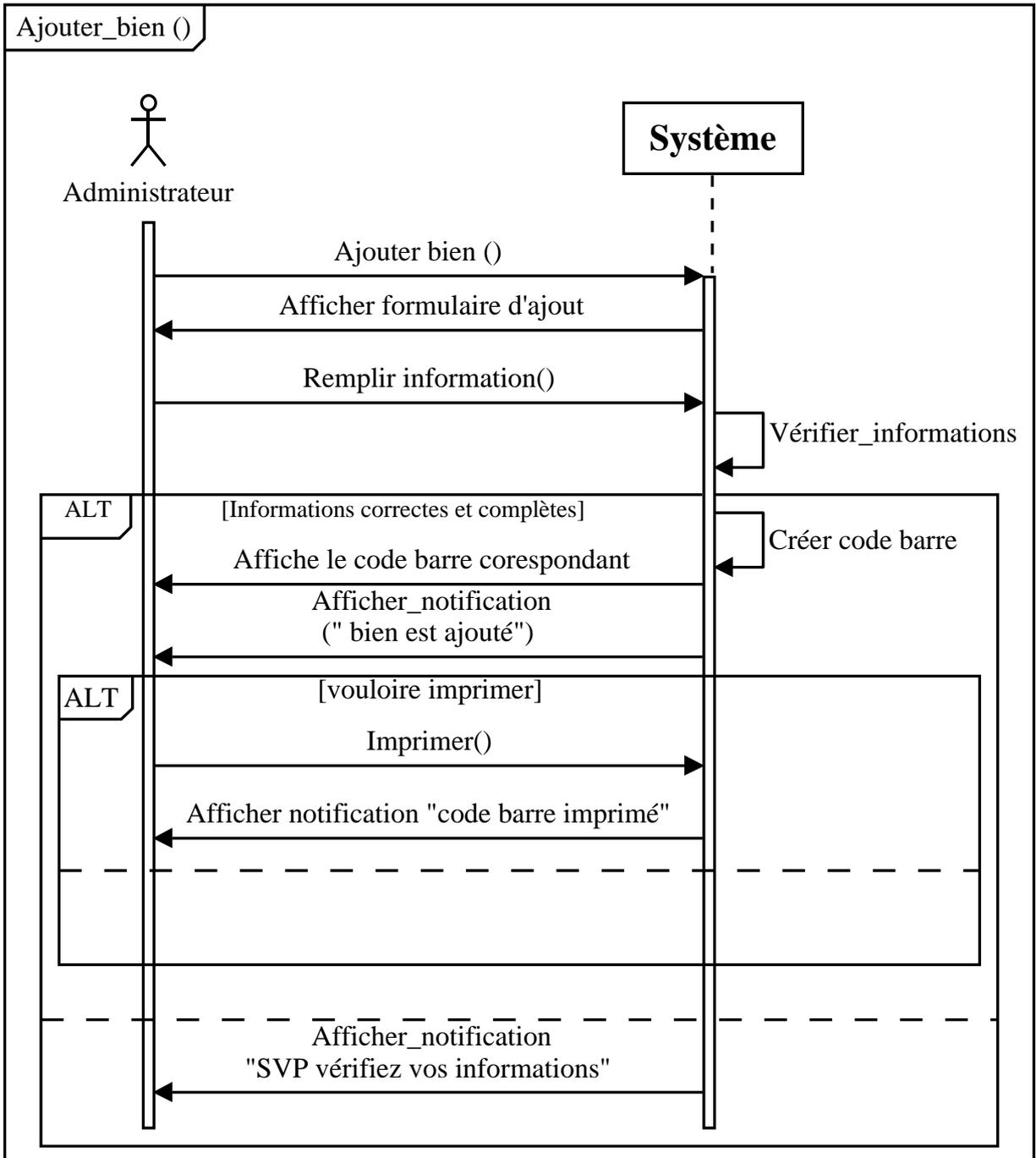
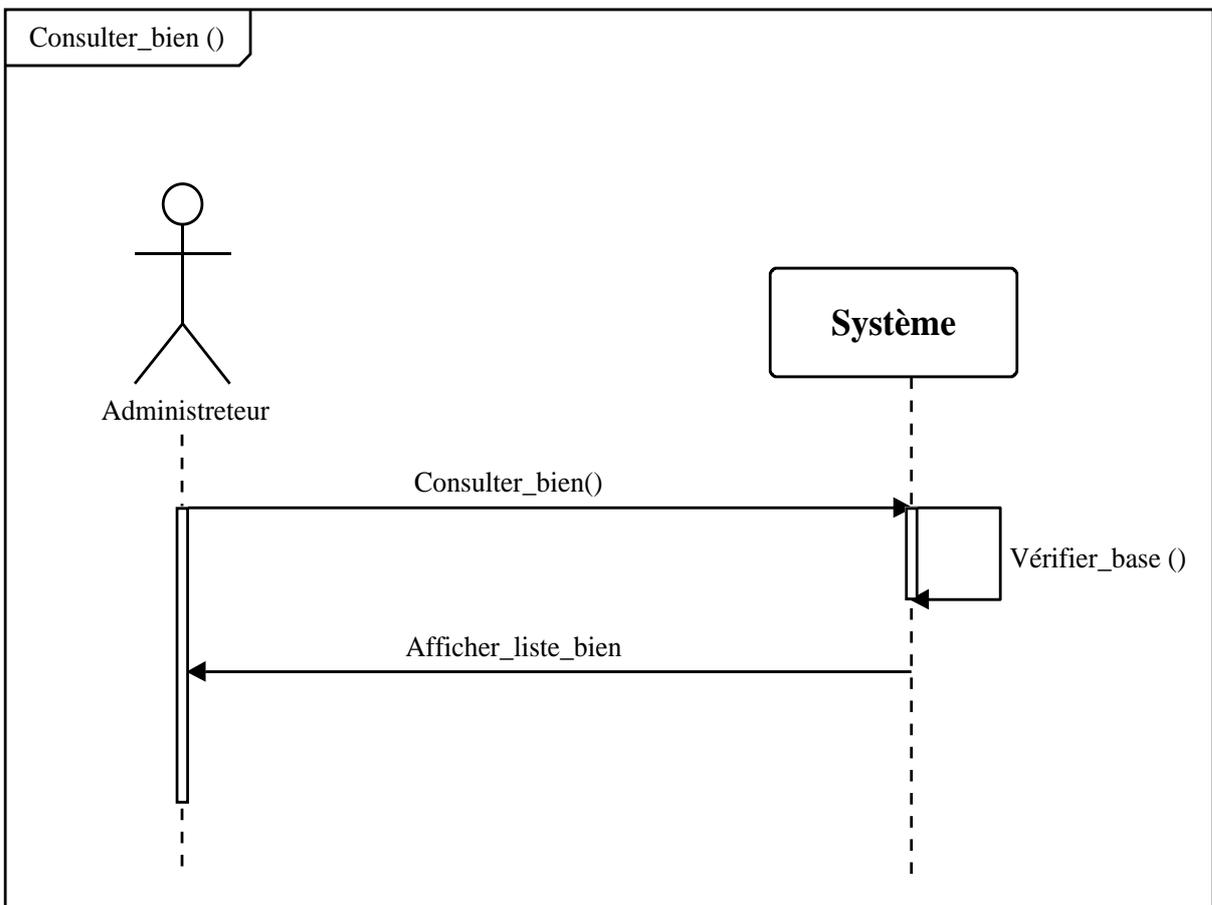


Figure 10: Diagramme de séquence du cas d'utilisation « Ajouter bien »

❖ Cas d'utilisation « Consulter bien »

Sommaire d'identification**Titre :** Consulter_bien ().**But :** Afficher les biens qui se trouvent dans la base de données.**Résumé :** Afficher la liste des biens qui se trouvent dans la base de données**Acteur :** Administrateur.**Description des Enchaînements****Pré condition :** l'administrateur doit être authentifié.**Post condition :** la liste des biens est affichée.**Scénario nominal :** Ce cas d'utilisation commence lorsque l'administrateur demande au système une consultation des biens.

- Le système affiche la liste des biens.

**Figure 11: Diagramme de séquence du cas d'utilisation « Consulter bien »**

❖ Cas d'utilisation « Modifier bien »

| |
|---|
| Sommaire d'identification |
| <p>Titre : Modifier_bien ().</p> <p>But : Modifier les informations d'un bien.</p> <p>Résumé : changer le profil d'un bien.</p> <p>Acteur : L'administrateur.</p> |
| Descriptions des enchainements |
| <p>Pré conditions</p> <ul style="list-style-type: none">• L'administrateur doit être authentifié.• Le bien existe déjà. <p>Post conditions : le profil bien est modifié.</p> <p>Scénario nominal : ce cas d'utilisation commence lorsque l'administrateur demande au système une modification du bien.</p> <ol style="list-style-type: none">1- L'administrateur consulte la liste des biens.2- Le système affiche la liste des biens.3- L'administrateur choisit l'un des biens.4- Le système affiche un formulaire pour changer les informations de ce bien.5- L'administrateur modifie les champs qu'il veut dans le formulaire.6- Le système vérifie les informations saisit par l'administrateur et puis fait une mise à jour concernant ce bien dans la base de données, puis il affiche la notification «bien est modifié». <p>Scénario alternatif</p> <ul style="list-style-type: none">➤ Lorsque l'administrateur remplit des informations incorrectes ou incomplètes, le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 4. <p>Scénario exceptionnel :</p> <ul style="list-style-type: none">➤ Lorsque l'administrateur choisit l'option annulé, le système quitte la page. |

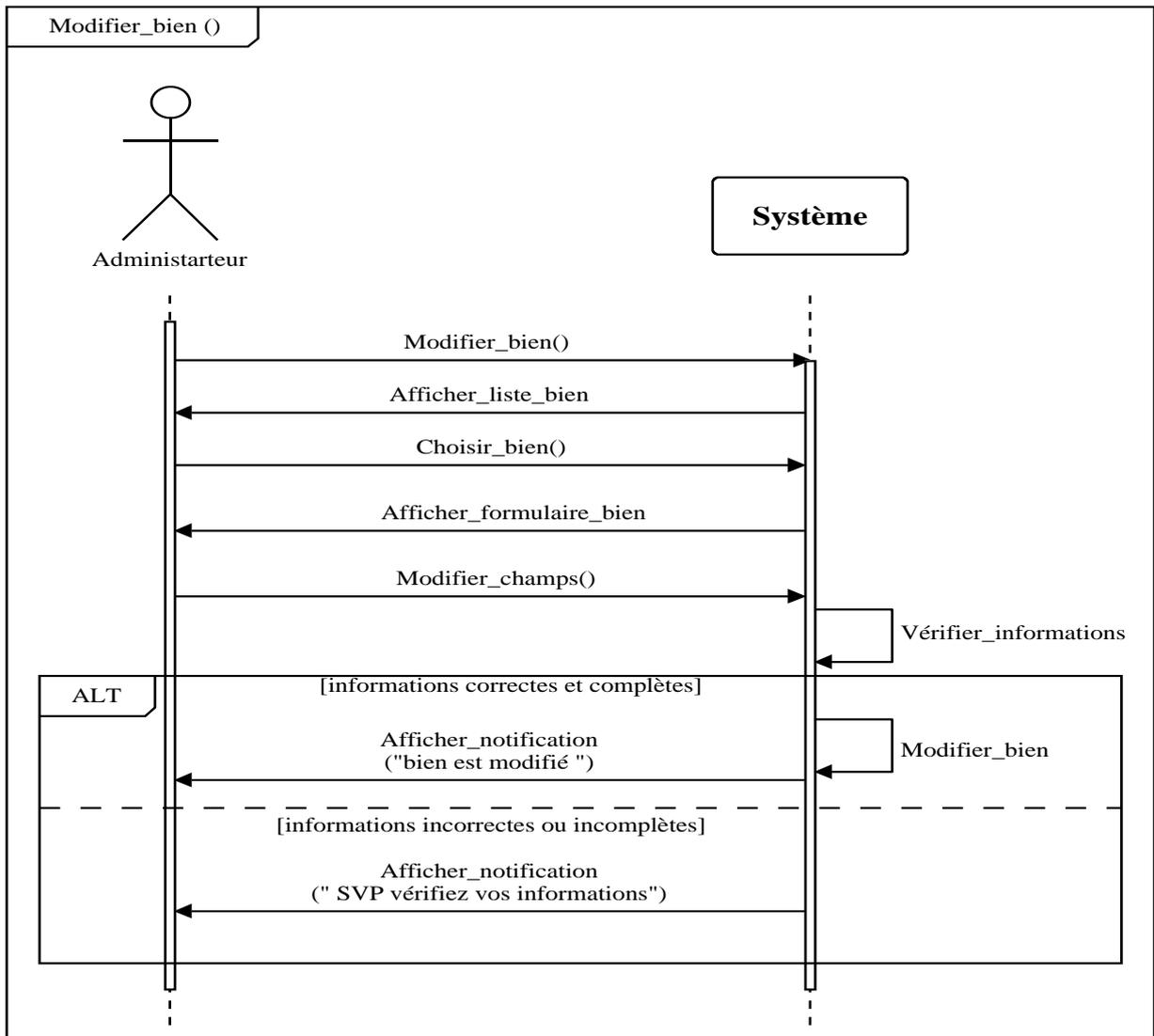


Figure 12: Diagramme de séquence du cas d'utilisation « Modifier bien »

❖ Cas d'utilisation « Supprimer bien »

Sommaire d'identification

Titre : Supprimer_bien ().

But : Supprimer un bien.

Résumé : Supprimer un bien de du système.

Acteur : L'administrateur.

Descriptions des enchainements

Pré conditions

- L'administrateur doit être authentifié.
- Le bien existe déjà.

Post conditions : Le bien est supprimé de base.

Scénario nominal : ce cas d'utilisation commence lorsque l'administrateur demande au système une suppression d'un bien.

- 1- L'administrateur consulte la liste des biens.
- 2- Le système affiche la table des biens.
- 3- L'administrateur choisit un bien pour le supprimer.
- 4- Le système demande une confirmation de la suppression.
- 5- L'administrateur confirme la suppression.
- 6- Le système fait une mise à jour dans la base de données, puis il affiche une notification «bien est supprimé».

Scénario alternatif

- Lorsque l'administrateur ne confirme pas la suppression, le système laisse la base de données avec son état primaire.

Scénario exceptionnel :

- Lorsque l'administrateur choisit l'option annulé la suppression, le système quitte la page.

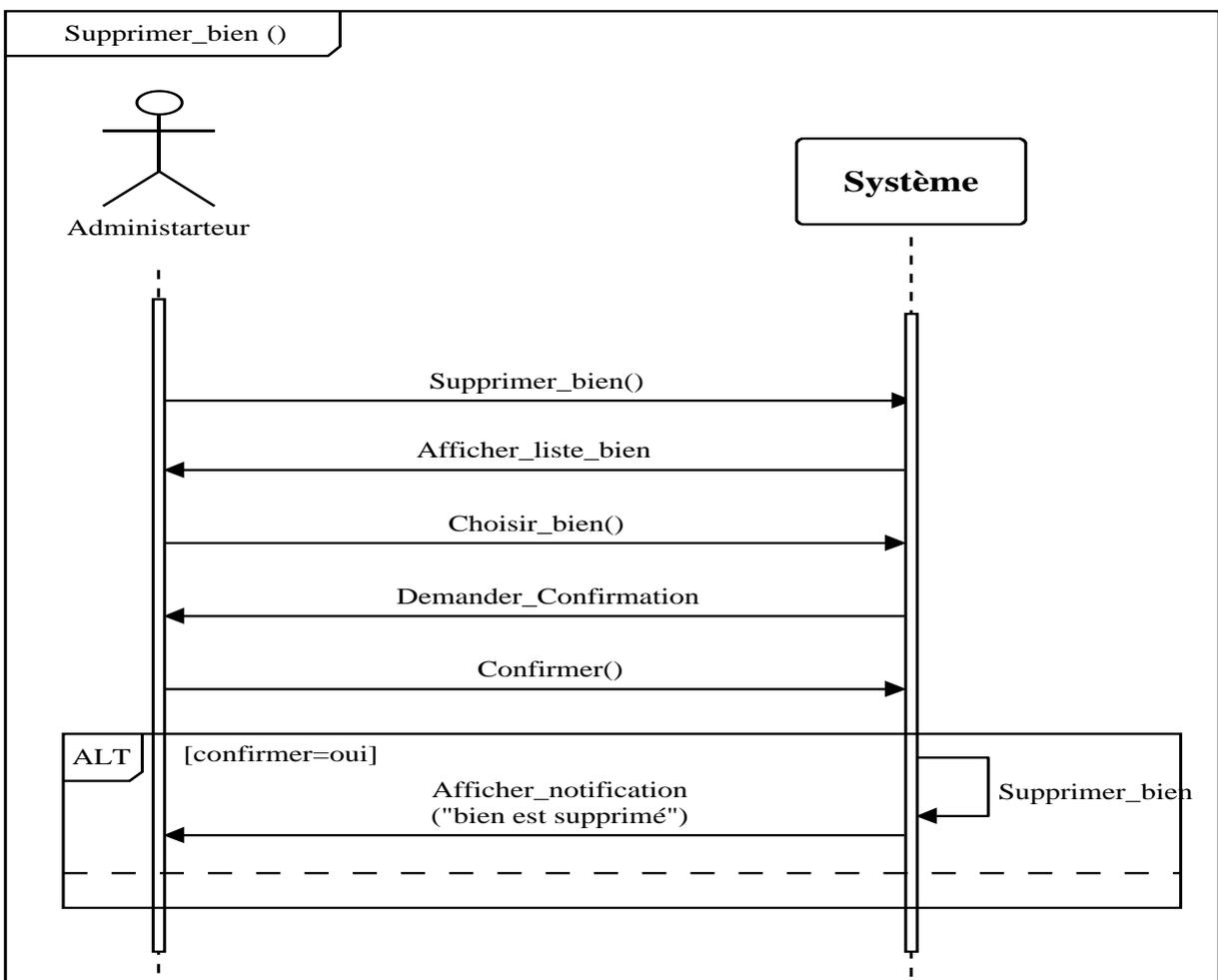


Figure 13: Diagramme de séquence du cas d'utilisation « Supprimer bien »

❖ Cas d'utilisation « Gestion Utilisateurs »

Sommaire d'identification

Titre : Gestion_Utilisateur ().

But : Ajouter /consulter /modifier/ supprimer: utilisateur.

Acteur : administrateur.

Description des Enchaînements**Pré conditions**

- L'administrateur doit être authentifié.
- L'administrateur possède les informations concernant un utilisateur.

Post-condition: les informations ont été enregistrées.

Scénario nominal : Ce cas d'utilisation commence lorsque l'administrateur demande au système la gestion de l'utilisateur.

- 1- l'administrateur accède au système.
- 2- Le système présente l'état actuel.
- 3-
 - l'administrateur choisit l'opération ajouté utilisateur sous cas : ajouter utilisateur.
 - l'administrateur choisit l'opération modifié utilisateur sous cas : modifier utilisateur.
 - l'administrateur choisit l'opération supprimé utilisateur sous cas : supprimer utilisateur.
 - l'administrateur choisit l'opération consulté utilisateur sous cas : consulter utilisateur.
- 4- Le système enregistre les modifications.

Scénario alternatif

- Lorsque l'administrateur annule l'accès au sous cas retourne à l'étape 2.

Scénario exceptionnel

- Lorsque l'administrateur choisit l'option annulé, le système quitte la page de création.

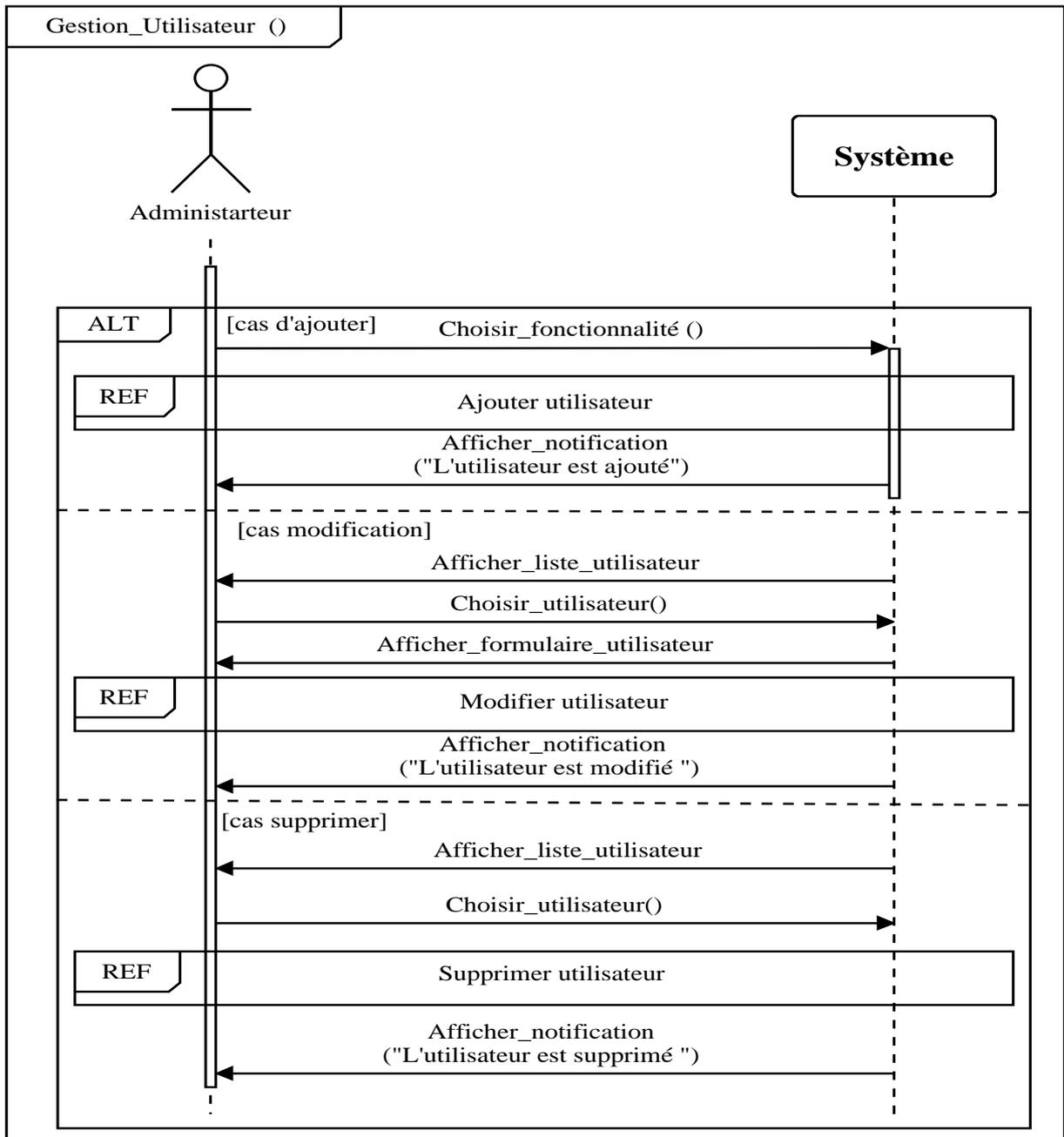


Figure 14: Diagramme de séquence du cas d'utilisation « Gestion Utilisateurs »

❖ Cas d'utilisation « Ajouter utilisateurs »

Sommaire d'identification

Titre : Ajouter_utilisateur ().

But : Ajouter un utilisateur.

Résumé : Ajouter un utilisateur dans la liste des utilisateurs de système.

Acteur : L'administrateur.

Description des Enchaînements

Pré conditions: l'administrateur doit être authentifié.

Post-condition: l'ajout est effectué.

Scénario nominal : ce cas d'utilisation commence lorsque l'administrateur demande au système d'ajouter un utilisateur.

- 1- le système affiche un formulaire d'ajout.
- 2- l'administrateur remplit toutes les informations.
- 3- Le système vérifie les informations saisies par l'administrateur puis affiche une notification «l'utilisateur est ajouté».

Scénario alternatif

- Lorsque les utilisateurs introduisent des informations incorrectes ou incomplètes, le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 1.

Scénario exceptionnel

- Lorsque l'administrateur choisit l'option annulée, le système quitte la page d'ajout.

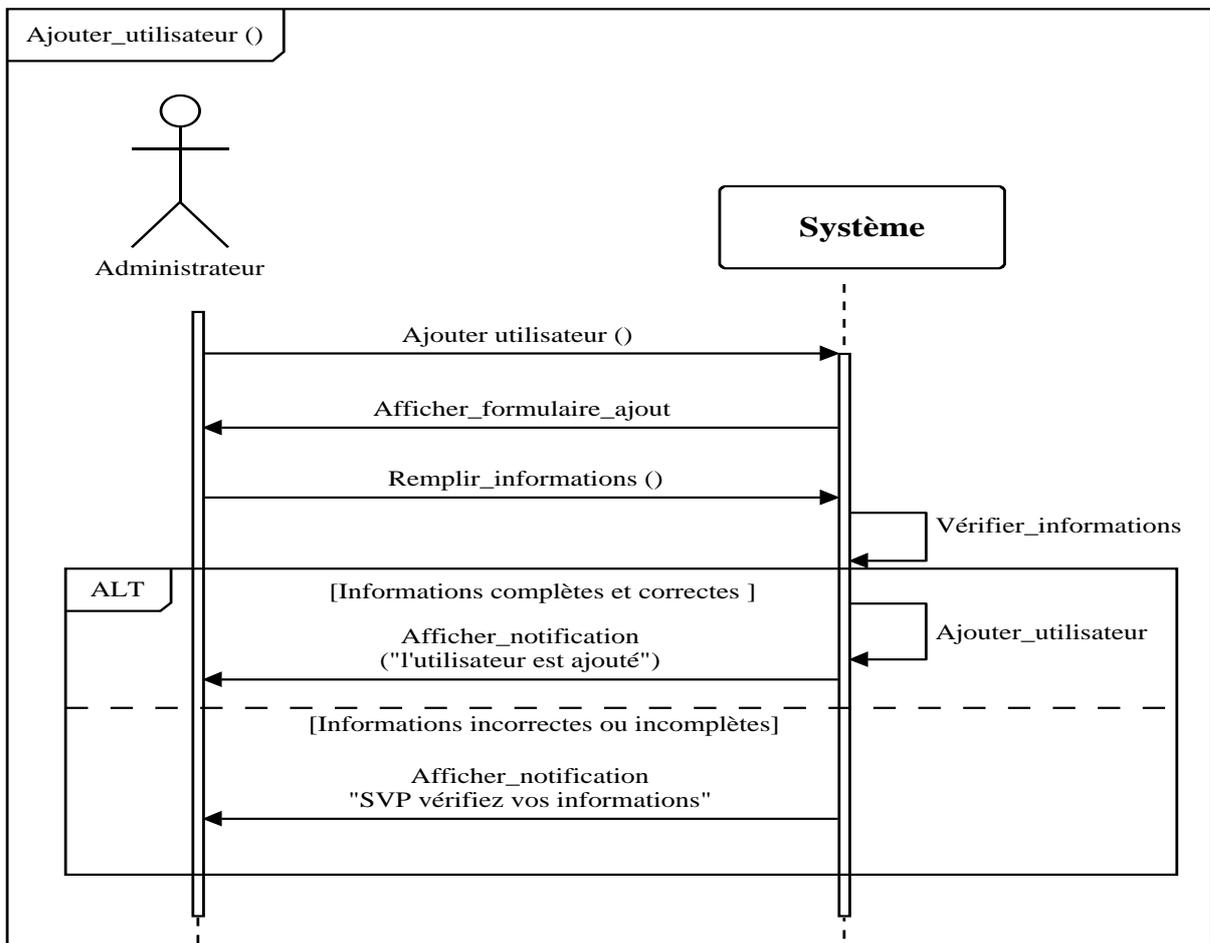


Figure 15: Diagramme séquence du cas d'utilisation « Ajouter utilisateur »

❖ Cas d'utilisation « Consulter utilisateur »

Sommaire d'identification**Titre :** Consulter_utilisateur ().**But :** Afficher les utilisateurs qui se trouvent dans la base de données.**Résumé :** Afficher la liste des utilisateurs qui se trouvent dans la base de données.**Acteur :** Administrateur.**Description des Enchaînements****Pré condition :** l'administrateur doit être authentifié.**Post condition :** la liste des utilisateurs est affichée.**Scénario nominal :** ce cas d'utilisation commence lorsque l'administrateur demande au système une consultation des utilisateurs.

- Le système affiche la liste des utilisateurs.

Scénario alternatif

- Lorsque le système ne parvient pas à accéder à la base de données, le système affiche une notification « Impossible d'afficher le contenu de la base ».

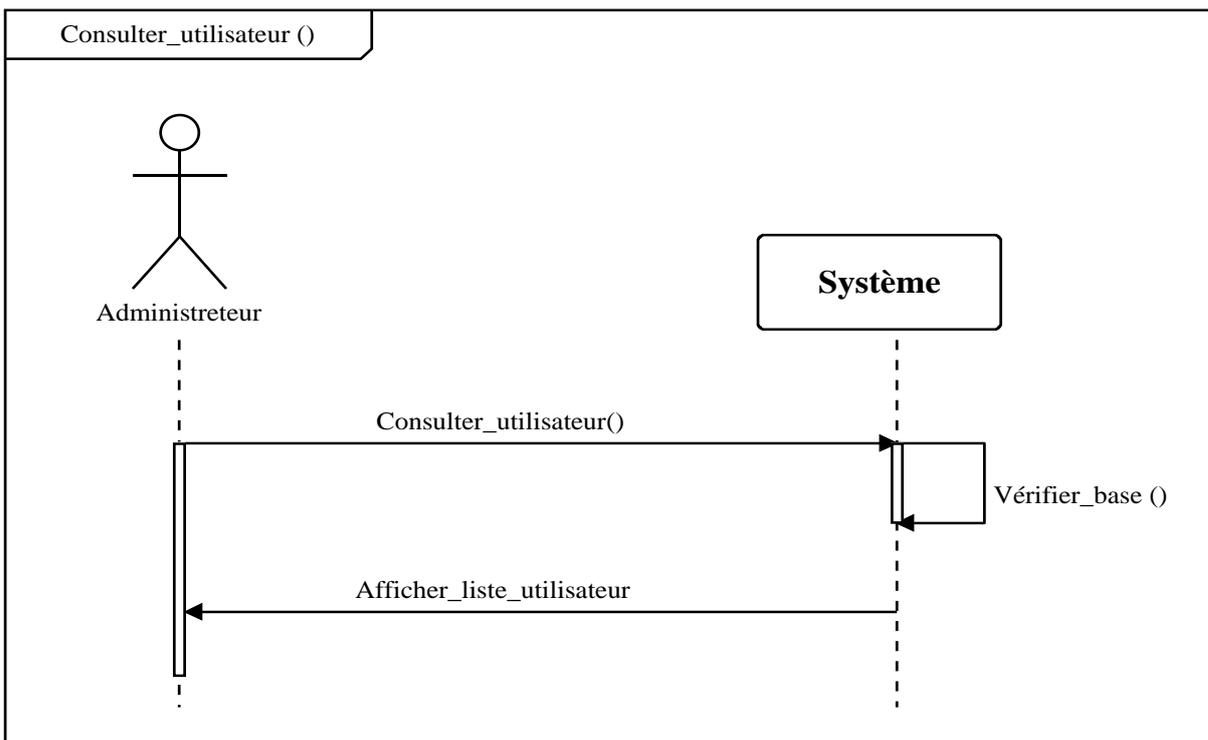


Figure 16: Diagramme de séquence du cas d'utilisation « Consulter utilisateur »

❖ Cas d'utilisation « Modifier utilisateur »

Sommaire d'identification

Titre : Modifier_utilisateur ().

But : Modifier les informations d'un utilisateur.

Résumé : Changer le profil d'un utilisateur.

Acteur : L'administrateur.

Descriptions des enchainements**Pré conditions**

- L'administrateur doit être authentifié.
- L'utilisateur existe déjà.

Post conditions : le profil de l'utilisateur est modifié.

Scénario nominal : ce cas d'utilisation commence lorsque l'administrateur demande au système une modification de l'utilisateur.

- 1- L'administrateur consulte la liste des utilisateurs.
- 2- Le système affiche la liste des utilisateurs.
- 3- L'administrateur choisit l'un des utilisateurs.
- 4- Le système affiche un formulaire pour changer les informations de cet utilisateur.
- 5- L'administrateur modifie les champs qu'il veut dans le formulaire.
- 6- Le système vérifie les informations saisies par l'administrateur et puis fait une mise à jour concernant cet utilisateur dans la base de données, puis il affiche la notification «L'utilisateur est modifié».

Scénario alternatif

- Lorsque l'administrateur remplit des informations incorrectes ou incomplètes, le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 4.

Scénario exceptionnel :

- Lorsque l'administrateur choisit l'option annulé, le système quitte la page.

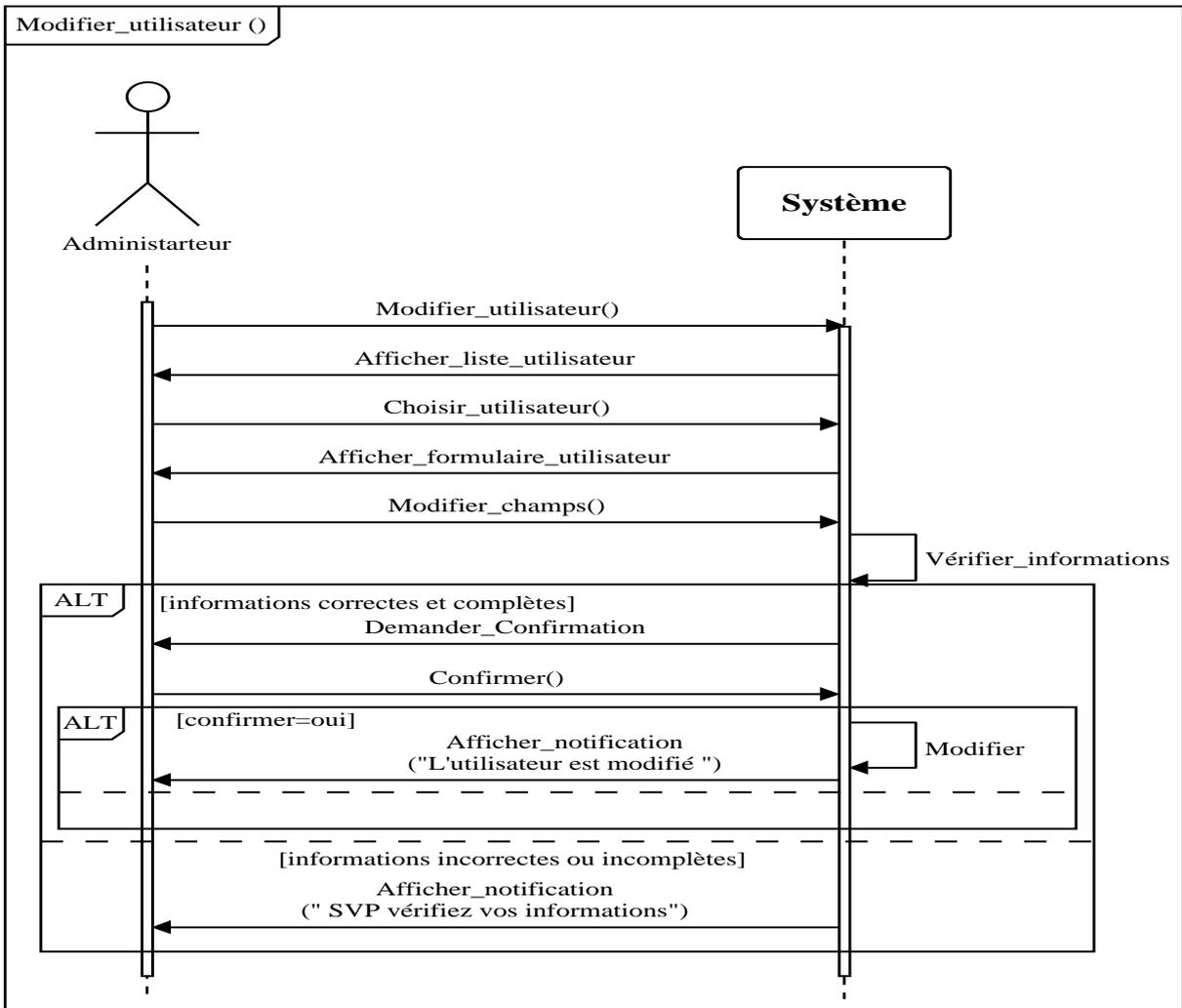


Figure 17: Diagramme de séquence du cas d'utilisation « Modifier utilisateur »

❖ Cas d'utilisation « Supprimer utilisateur »

Sommaire d'identification

Titre : Supprimer_l'utilisateur ().

But : Supprimer l'utilisateur.

Résumé : Supprimer l'utilisateur de la liste des usagers du système.

Acteur : L'administrateur.

Descriptions des enchainements

Pré conditions

- L'administrateur doit être authentifié.
- L'utilisateur existe déjà.

Post conditions : l'utilisateur est supprimé de la liste des usagers du système.

Scénario nominal : ce cas d'utilisation commence lorsque l'administrateur demande au système une suppression d'un utilisateur.

- 1- L'administrateur consulte la liste des utilisateurs.
- 2- Le système affiche la table des utilisateurs.
- 3- L'administrateur choisit un utilisateur pour le supprimer.
- 4- Le système demande une confirmation de la suppression.
- 5- L'administrateur confirme la suppression.
- 6- Le système fait une mise à jour dans la base de données, puis il affiche une notification
«l'utilisateur est supprimé».

Scénario alternatif

- Lorsque l'administrateur ne confirme pas la suppression, le système laisse la base de données avec son état primaire.

Scénario exceptionnel :

- Lorsque l'administrateur choisit l'option annulé la suppression, le système quitte la page.

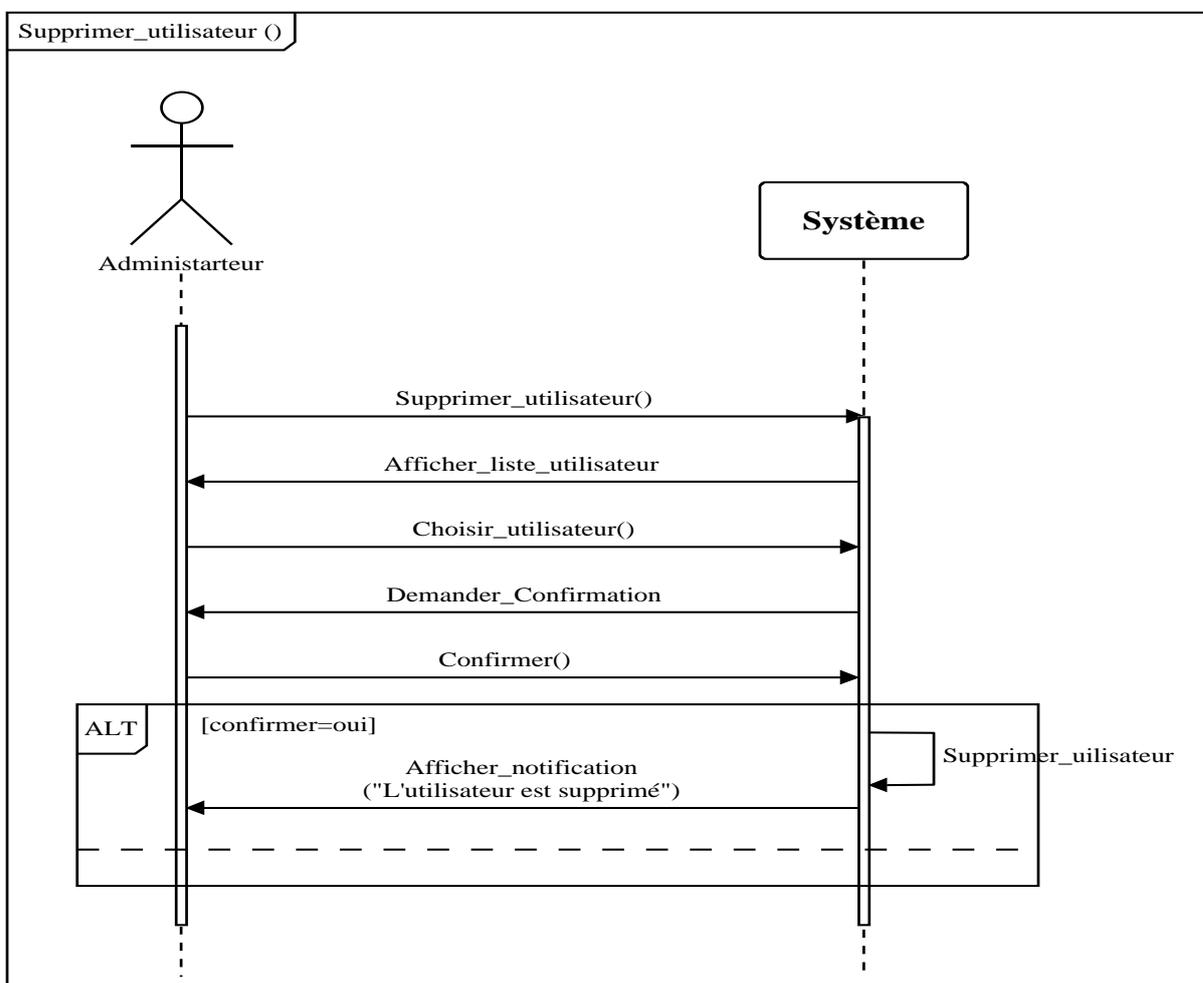


Figure 18: Diagramme de séquence du cas d'utilisation « Supprimer utilisateur »

❖ Cas d'utilisation « Créer emplacement »

Sommaire d'identification**Titre :** Créer_emplacement ().**But :** Ajouter / consulter /modifier /supprimer : emplacement.**Acteur :** administrateur.**Description des Enchaînements****Pré conditions :** L'administrateur doit être authentifié.

- L'administrateur possède les informations concernant un emplacement.

Post-condition :

- Les informations ont été enregistrées.

Scénario nominal : Ce cas d'utilisation commence lorsque l'administrateur demande au système d'emplacement.

1- l'administrateur accède au système.

2- Le système présente l'état actuel.

3-

- l'administrateur choisit l'opération ajouter emplacement sous cas : ajouter emplacement.
- l'administrateur choisit l'opération modifier emplacement sous cas : modifier emplacement.
- l'administrateur choisit l'opération supprimer emplacement sous cas : supprimer emplacement.
- l'administrateur choisit l'opération consulter emplacement sous cas : consulter emplacement.

4- Le système enregistre les modifications.

Scénario alternatif

- Lorsque l'administrateur annule l'accès au sous cas, il retourne à l'étape 2.
- Lorsque l'administrateur saisi des informations incorrectes ou incomplètes, le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 2.

Scénario exceptionnel

- Lorsque l'administrateur choisit l'option annulé le système quitte la page de création.

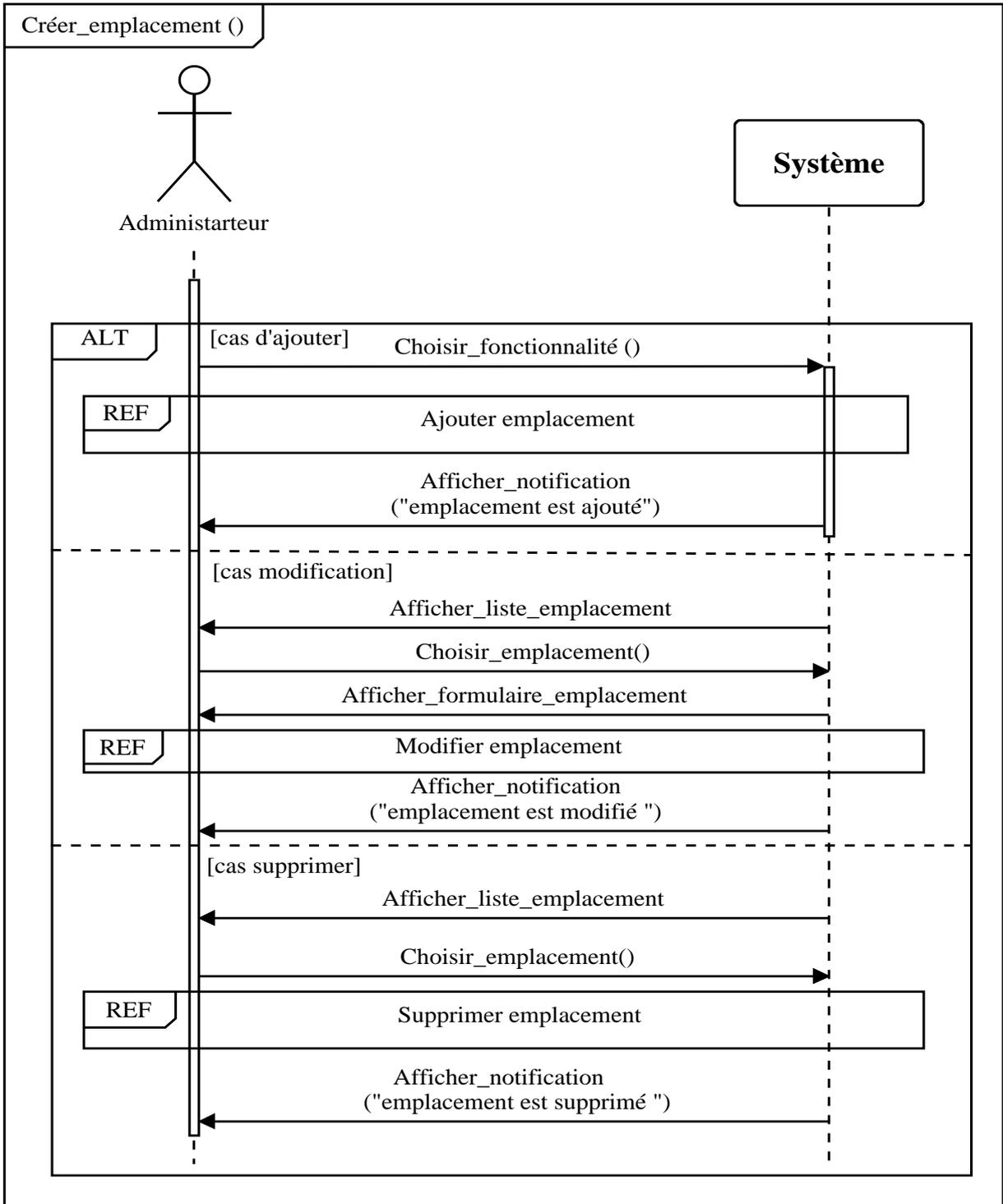


Figure 19: Diagramme de séquence du cas d'utilisation « Créer emplacement »

❖ Cas d'utilisation « Ajouter emplacement »

| |
|---------------------------------------|
| Sommaire d'identification |
| Titre: Ajouter_emplacement (). |

But: Ajouter un emplacement.

Résumé: Ajouter un emplacement dans la liste des emplacements.

Acteur : L'administrateur.

Description des Enchaînements

Pré conditions : L'administrateur doit être authentifié.

Post-condition : L'ajout est effectué.

Scénario nominal :

Ce cas d'utilisation commence lorsque l'administrateur demande au système d'ajouter un emplacement.

- 1- le système affiche un formulaire d'ajout.
- 2- l'administrateur remplit toutes les informations.
- 3- Le système vérifie les informations saisies par l'administrateur et affiche une notification «emplacement est ajouté».

Scénario alternatif

Lorsque l'administrateur saisit des informations incorrectes ou incomplètes :

- Le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 1.

Scénario exceptionnel

Lorsque l'administrateur choisit l'option annulé :

- Le système quitte la page d'ajout.

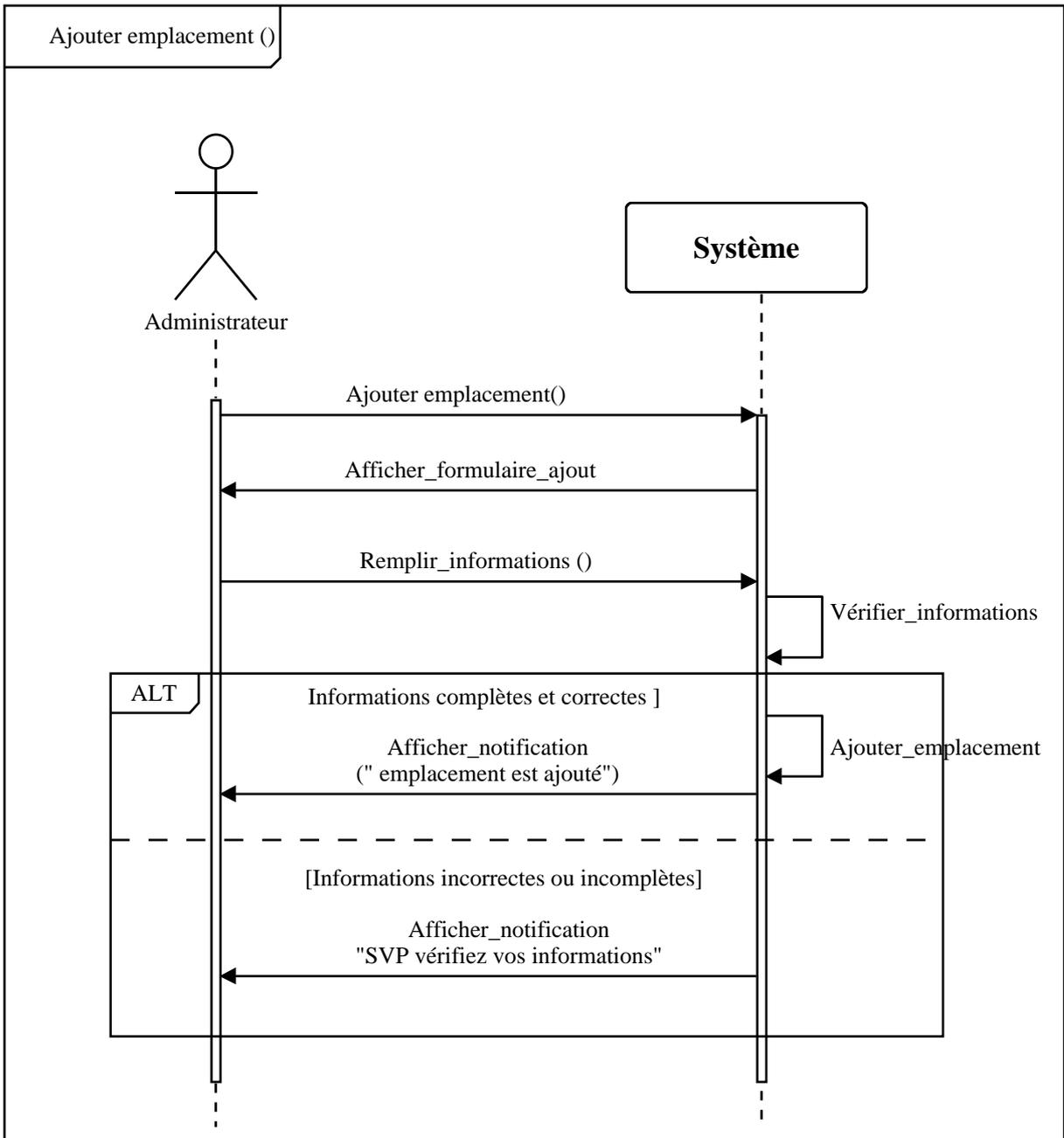


Figure 20: Diagramme de séquence du cas d'utilisation « Ajouter emplacement »

❖ Cas d'utilisation « Consulter emplacement »

Sommaire d'identification

Titre : Consulter_emplacement ().

But : Afficher les emplacements qui se trouvent dans la base de données.

Résumé : Afficher la liste des emplacements qui se trouvent dans la base de données par conséquence leurs informations.

Acteur : Administrateur.

Description des Enchaînements

Pré condition : l'administrateur doit être authentifié.

Post condition : la liste des fournisseurs est affichée.

Scénario nominal

Ce cas d'utilisation commence lorsque l'administrateur demande au système une consultation des emplacements des biens.

- Le système affiche la liste des emplacements.

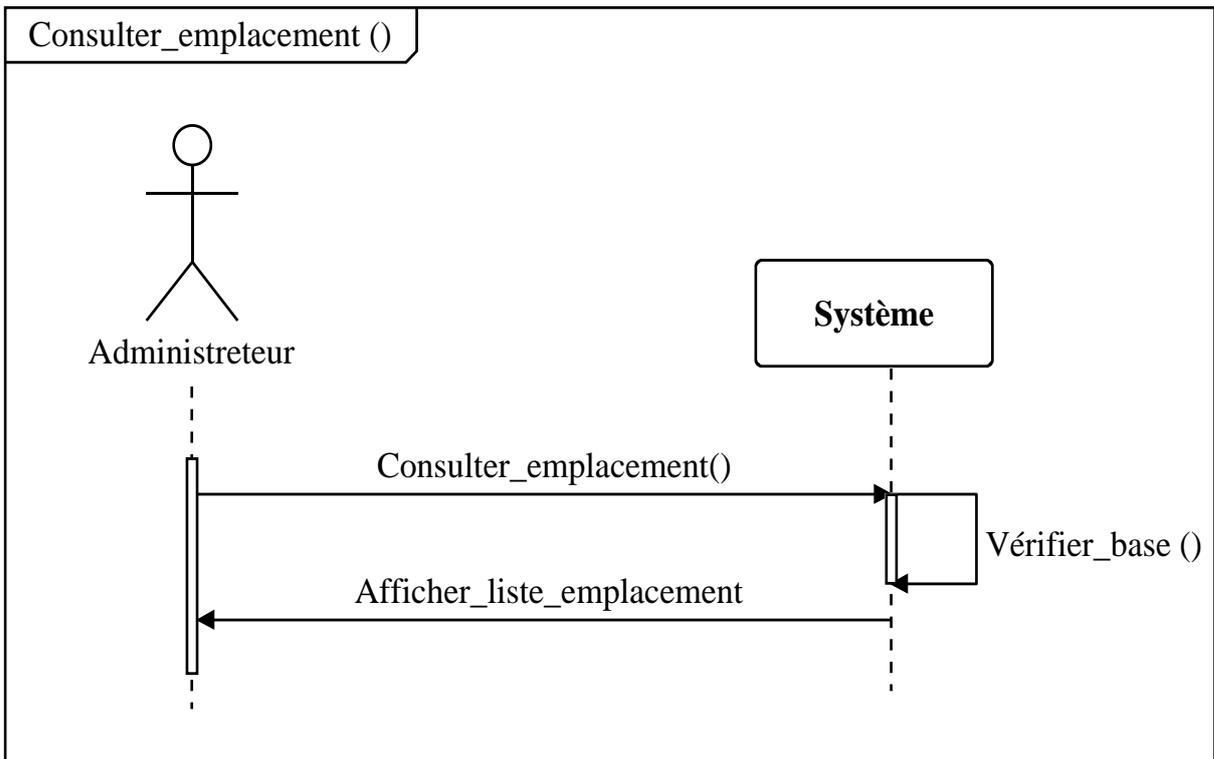


Figure 21: Diagramme de séquence du cas d'utilisation « Consulter emplacement »

❖ Cas d'utilisation « Modifier emplacement »

Sommaire d'identification

Titre : Modifier_emplacement ().

But : Modifier les informations d'un emplacement.

Résumé : changer le profil d'un emplacement.

Acteur : L'administrateur.

Descriptions des enchainements**Pré conditions**

- L'administrateur doit être authentifié.
- Emplacement existe déjà.

Post conditions : le profil emplacement est modifié.

Scénario nominal

Ce cas d'utilisation commence lorsque l'administrateur demande au système une modification de l'emplacement.

- 1- L'administrateur consulte la liste des emplacements.
- 2- Le système affiche la liste des emplacements.
- 3- L'administrateur choisit l'un des emplacements.
- 4- Le système affiche un formulaire pour changer les informations de cet emplacement.
- 5- L'administrateur modifie les champs qu'il veut dans le formulaire.
- 6- Le système vérifie les informations saisit par l'administrateur et puis fait une mise à jour concernant cet emplacement dans la base de donnée. puis il affiche une notification «emplacement est modifié».

Scénario alternatif

Lorsque l'administrateur remplit des informations incorrectes ou incomplètes :

- Le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 4.

Scénario exceptionnel

- Lorsque l'administrateur choisit l'option annulé le système quitte la page.

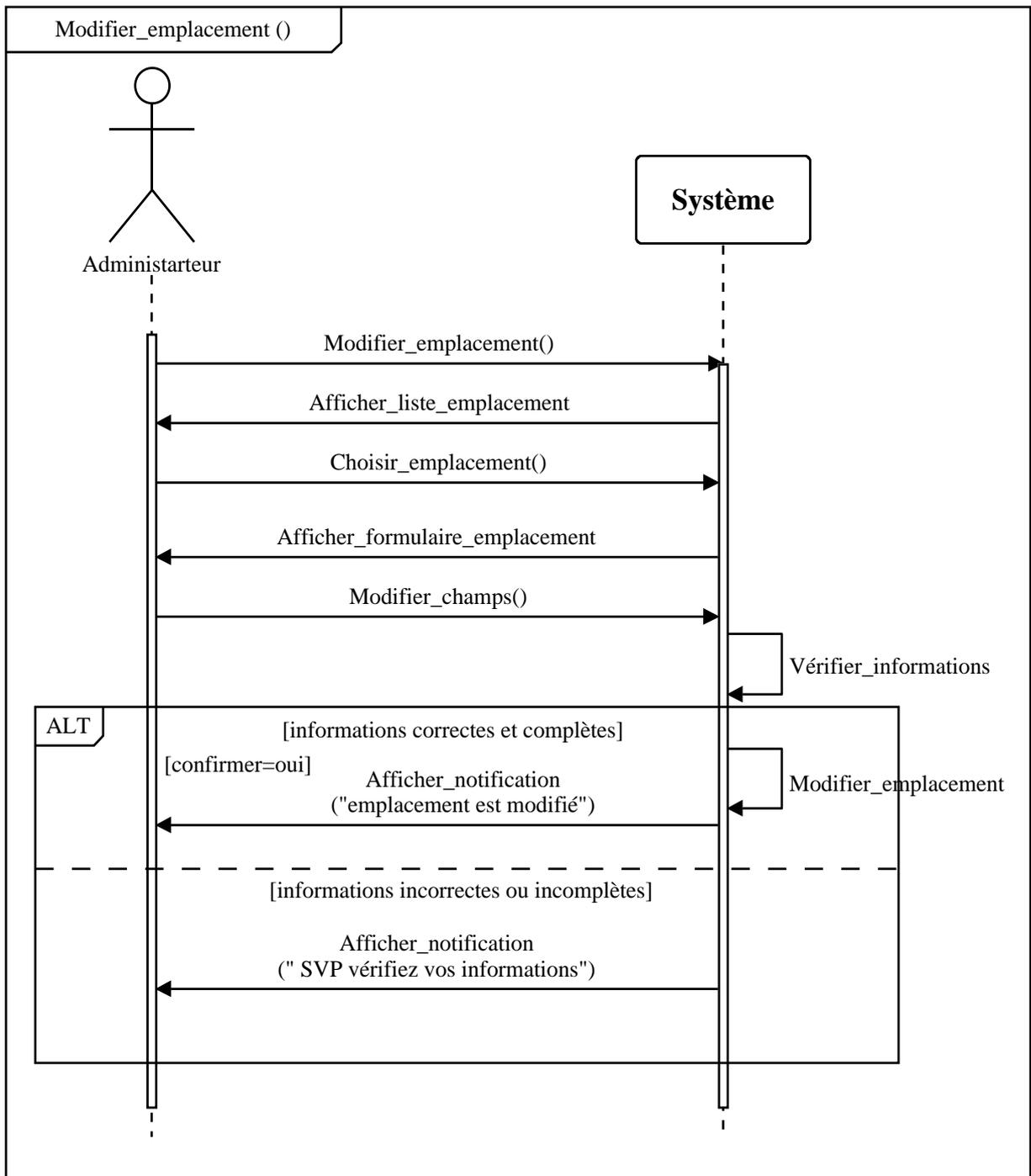


Figure 22: Diagramme de séquence du cas d'utilisation « Modifier emplacement »

❖ Cas d'utilisation « Supprimer emplacement »

Sommaire d'identification

Titre : Supprimer_emplacement ().

But : Supprimer un emplacement.

Résumé : Supprimer emplacement de la liste des usagers du système.

Acteur : L'administrateur.

Descriptions des enchainements

Pré conditions

- L'administrateur doit être authentifié.
- Emplacement existe déjà.

Post conditions : le profil emplacement est supprimé de la liste des usagers du système.

Scénario nominal

Ce cas d'utilisation commence lorsque l'administrateur demande au système une suppression d'un emplacement.

- 1- L'administrateur consulte la liste des emplacements.
- 2- Le système affiche la table des emplacements.
- 3- L'administrateur choisit un emplacement pour le supprimer.
- 4- Le système demande une confirmation de la suppression.
- 5- L'administrateur confirme la suppression.
- 6- Le système fait une mise à jour dans la base de données, puis il affiche une

notification

«emplacement est supprimé».

Scénario alternatif

- Lorsque l'administrateur ne confirme pas la suppression, le système laisse la base de données avec son état primaire.

Scénario exceptionnel :

- Lorsque l'administrateur choisit l'option annulé la suppression, le système quitte la page.

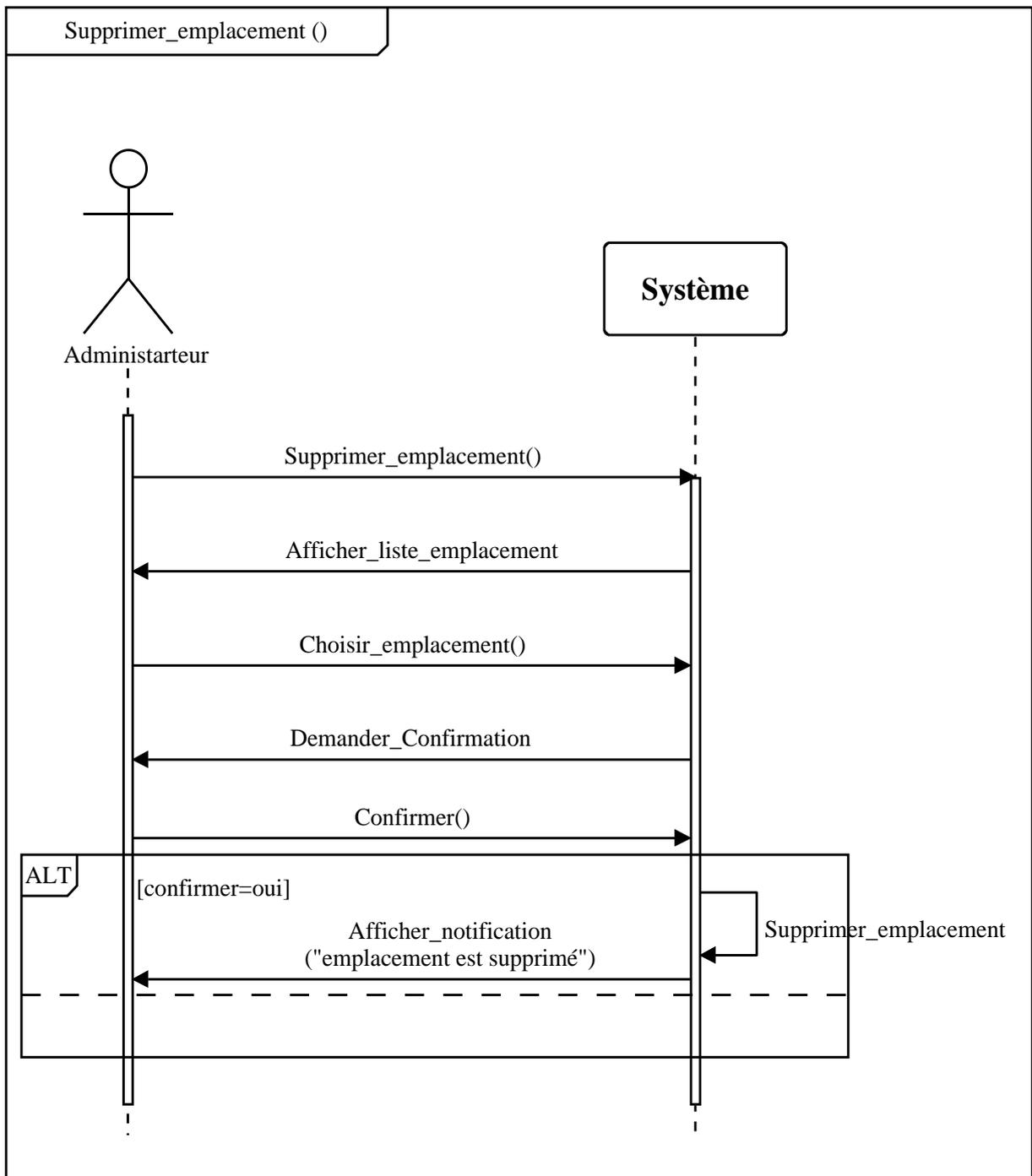


Figure 23: Diagramme de séquence du cas d'utilisation « Supprimer emplacement »

❖ Cas d'utilisation « Recherche des biens »

Sommaire d'identification

Titre: Rechercher_Bien ().

But: Trouver les informations concernant un bien puis les exporter.

Résumé: Trouver les informations concernant un bien.

Acteur: l'administrateur.

Description des Enchaînements

Pré conditions: l'administrateur doit être authentifié.

Post-condition :

- réception des informations d'un bien dans une base de données.
- Exporter les informations à d'autre logiciel.

Scénario nominal : Ce cas d'utilisation commence lorsque l'administrateur lance la recherche d'un bien dans la base de données, et il veut exporter les informations de ce bien.

- 1- L'administrateur choisit le critère de recherche puis remplit le champ de recherche valide.
- 2- Le système affiche le résultat de recherche dans le tableau des biens.
- 3- L'administrateur choisit le bouton exporter pour extraire les informations à d'autre logiciel.
- 4- Le système affiche un formulaire de sauvegarde.
- 5- L'administrateur choisit l'emplacement pour stocker sur le disque.
- 6- Le système extraire les informations des biens qui se trouve dans l'emplacement choisi et sauvegarder le fichier sur le disque.

Scénario alternatif

- en cas ou le bien n'existe pas: affiche une notification « Le bien n'existe pas» et retour à l'étape 2.

Scénario exceptionnel

- Lorsque l'administrateur choisit l'option annulé, le système quitte la page de création.

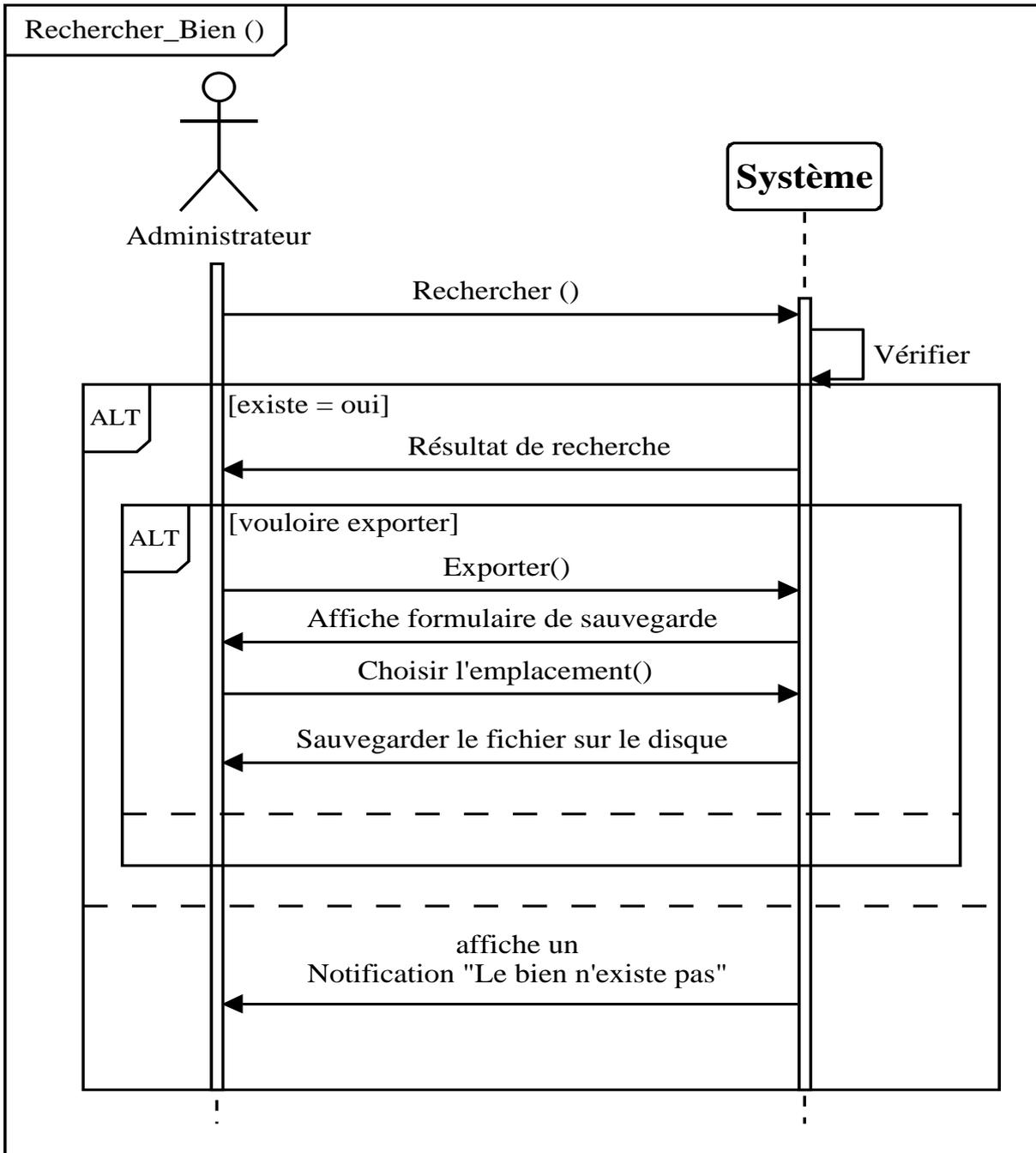


Figure 24: Diagramme de séquence du cas d'utilisation « Recherche des biens »

❖ Cas d'utilisation « Etat des lieux »

Sommaire d'identification

Titre : Etat_des_lieux ().

But : fournir des statistiques détaillées sur les biens.

Acteur: administrateur.

Description des Enchaînements

Pré conditions : L'administrateur doit être authentifié.

Post-condition :

- 1- Extraire la liste des biens endommagés.
- 2- Extraire la liste des biens à partir leurs emplacements.
- 3- Exporter les informations à d'autre logiciel.

Scénario nominal : Ce cas d'utilisation commence lorsque l'administrateur demande au système d'extraire les biens endommagés et puis extraire la liste des biens à partir leurs emplacements.

- 1- L'administrateur demande au système la liste des biens endommagés.
- 2- Le système affiche les biens endommagés
- 3- L'administrateur demande au système la liste des biens à partir de leurs emplacements.
- 4- Le système affiche les biens et leurs informations.
- 5- L'administrateur choisit le bouton exporter pour extraire les informations à d'autre logiciel.
- 6- Le système affiche un formulaire de sauvegarde.
- 7- L'administrateur choisit l'emplacement pour stocker sur le disque.
- 8- Le système extraire les informations des biens qui se trouve dans l'emplacement choisi et sauvegarder le fichier sur le disque.

Scénario exceptionnel

- ❖ Lorsque l'administrateur choisit l'option annulé, le système quitte la page de création.

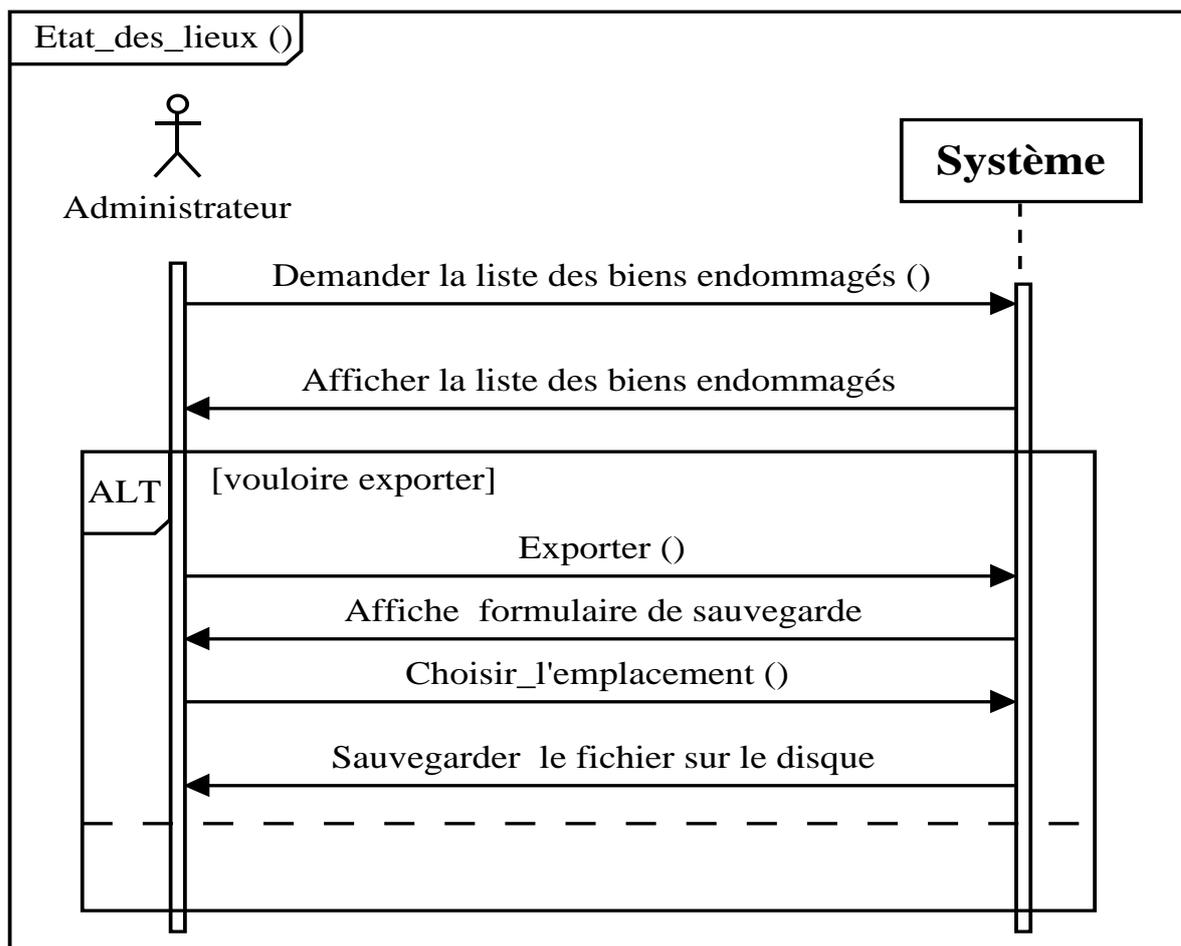


Figure 25: Diagramme de séquence du cas d'utilisation « Etat des lieux »

❖ Cas d'utilisation: « Bien manquant »

Sommaire d'identification

Titre : Bien_manquant ().

But : extraire les biens manquants.

Acteur: administrateur.

Description des Enchaînements

Pré conditions : L'administrateur doit être authentifié.

Post-condition : Extraire la liste des biens manquants.

Scénario nominal : Ce cas d'utilisation commence lorsque l'administrateur demande au système d'extraire les biens manquants.

- 1- L'administrateur demande au système la liste des biens manquants.
- 2- Le système affiche les biens manquants.
- 3- L'administrateur choisit le bouton exporter pour extraire les informations à d'autre

logiciel.

- 4- Le système affiche un formulaire de sauvegarde.
- 5- L'administrateur choisit l'emplacement pour stocker sur le disque.
- 6- Le système extrait les informations des biens qui se trouve dans l'emplacement choisi et sauvegarder le fichier sur le disque.

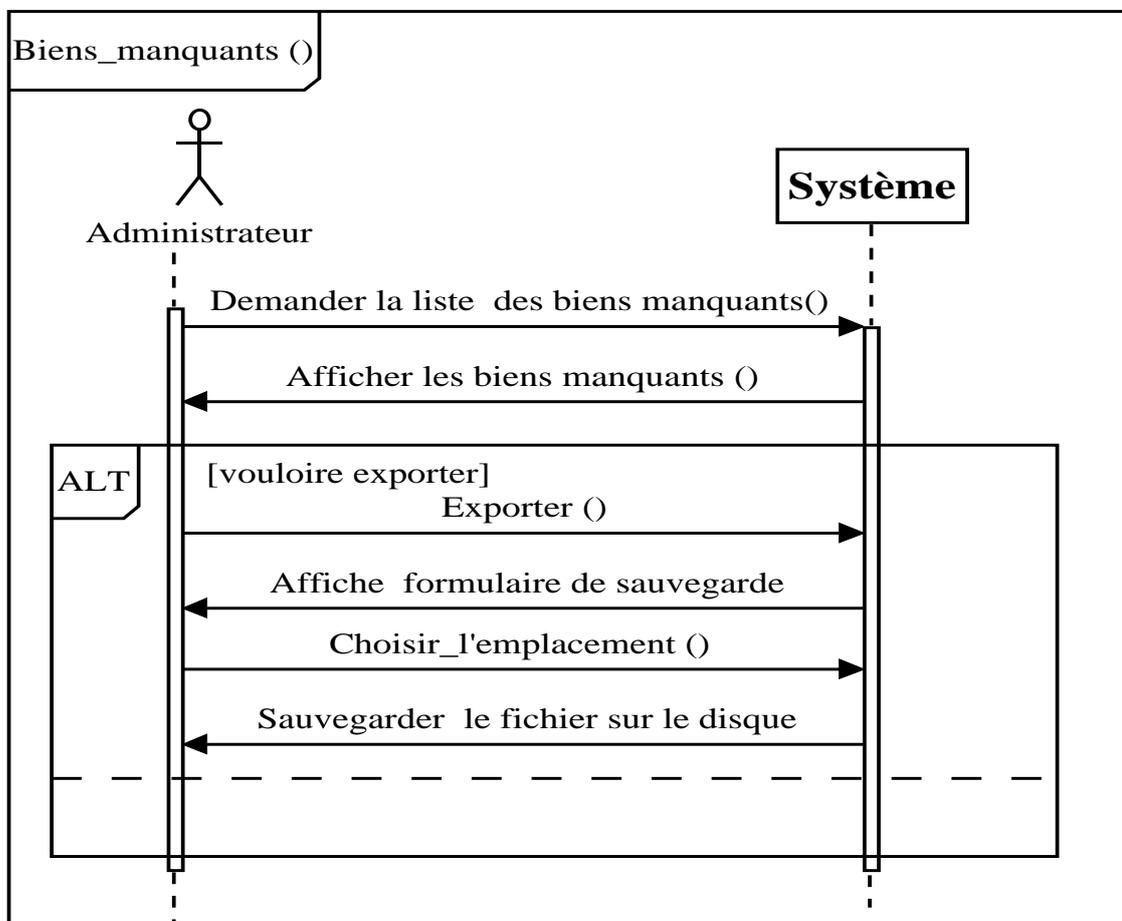


Figure 26: Diagramme de séquence du cas d'utilisation « Gestion des biens manquants »

4.2. Au niveau du Smartphone

❖ Cas d'utilisation « Scan automatique des biens»

Sommaire d'identification

Titre : Scanne ().

But : Scan automatique des biens.

Acteur : agent.

Description des Enchaînements

Pré conditions : L'agent doit être authentifié.

Post-condition : Le scanne est effectué.

Scénario nominal : ce cas d'utilisation commence lorsque l'agent demande au système un Scan automatique des biens.

1. le système Smartphone lancé le scanne et afficher le code barre.
2. L'agent sélectionné l'emplacement du bien scanné et le type de bien « Good ou Bad » et demande l'envoi.
3. le système Smartphone envoi les informations au serveur et en même temps les ajoute dans la base de donnée (SQLite au niveau Smartphone).
4. Le système serveur vérifier l'authentification et ajoute les informations dans la base de donnée (SGBD au niveau serveur).

Scénario exceptionnel

- Lorsque l'agent choisit l'option annulé, le système quitte la page création.

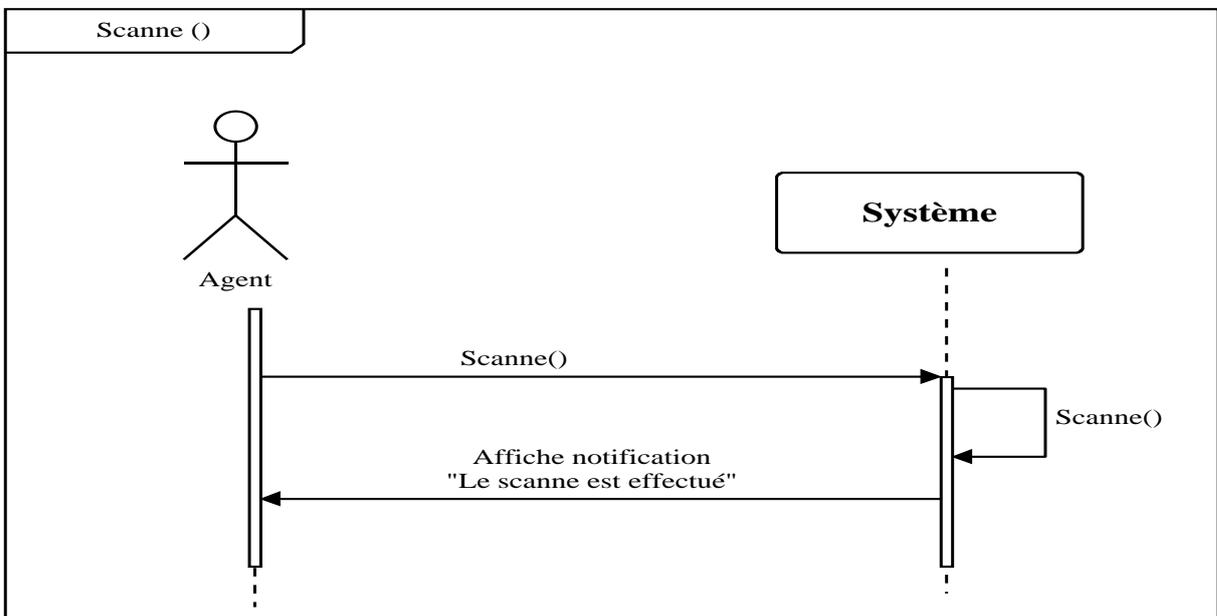


Figure 27: Diagramme de séquence du cas d'utilisation « Scan automatique des biens »

❖ Cas d'utilisation « Transfert des biens scannés »

Sommaire d'identification

Titre : Transfère_biens_scannés ().

But : transfère les biens scannés.

Acteur : agent.

Description des Enchaînements

Pré conditions : L'agent doit être authentifié.

Post-condition : le transfert est effectué.

Scénario nominal : ce cas d'utilisation commence lorsque l'agent demande au système un transfert des biens scannés.

- 1- le système Smartphone sélectionne les codes-barres non envoyé et lance le transfert vers le système serveur.
- 2- Le système serveur vérifie l'authentification et ajoute les informations dans la base de donnée (SGBD au niveau serveur).

Scénario exceptionnel

- Lorsque l'agent choisit l'option annulé, le système quitte la page de transfert.

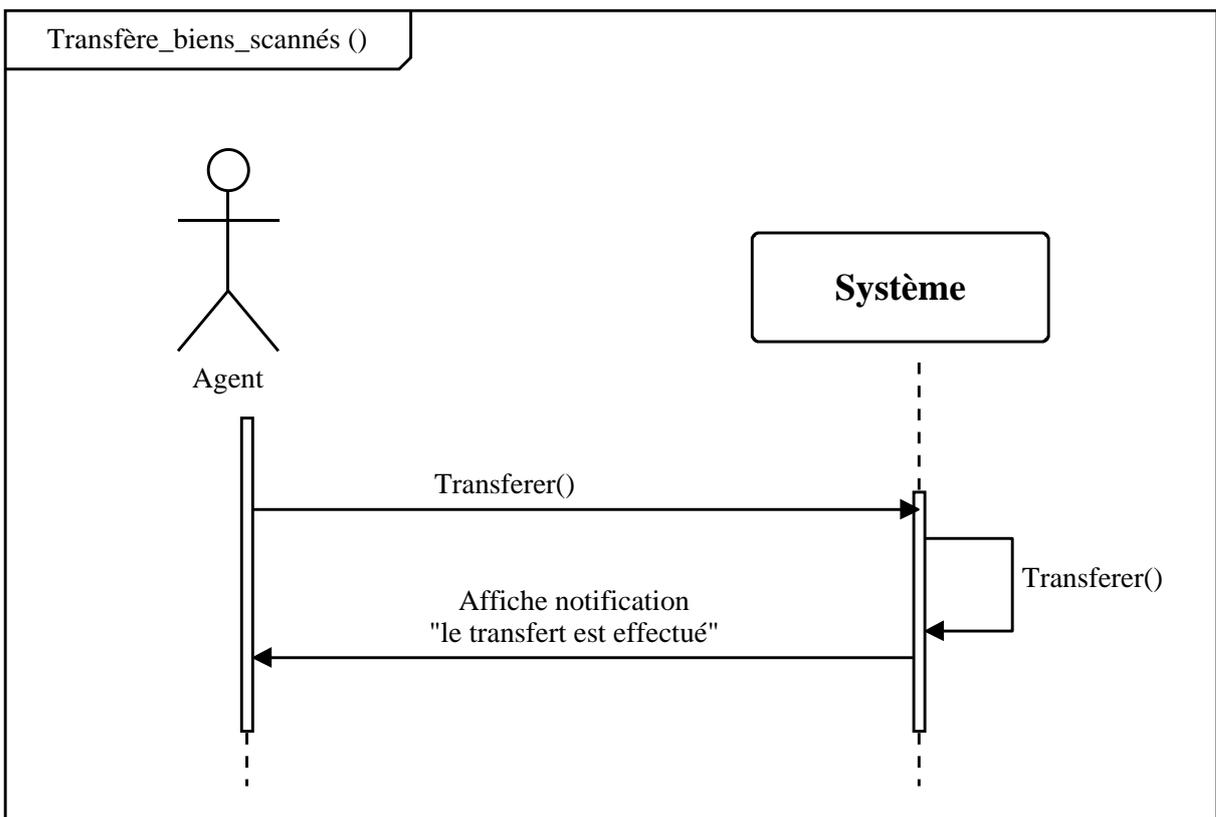


Figure 28: Diagramme de séquence du cas d'utilisation « Transfert des biens scannés »

❖ Cas d'utilisation « Synchroniser emplacement »

Sommaire d'identification

Titre : Synchroniser_emplacement ().

But : synchroniser les emplacements.

Acteur : agent.

Description des Enchaînements

Pré conditions : L'agent doit être authentifié.

Post-condition : Le transfert est effectué.

Scénario nominal : ce cas d'utilisation commence lorsque l'agent demande au système une synchronisation des emplacements.

1. le système Smartphone demande au serveur les emplacements.
2. Le système serveur vérifie l'authentification et sélectionne les emplacements dans la base de données (SGBD au niveau serveur) et l'envoi au Smartphone.
3. le système Smartphone ajoute les emplacements dans la base de données (SQLite).

Scénario exceptionnel

- Lorsque l'agent choisit l'option annulé, le système quitte la page de synchronisation.

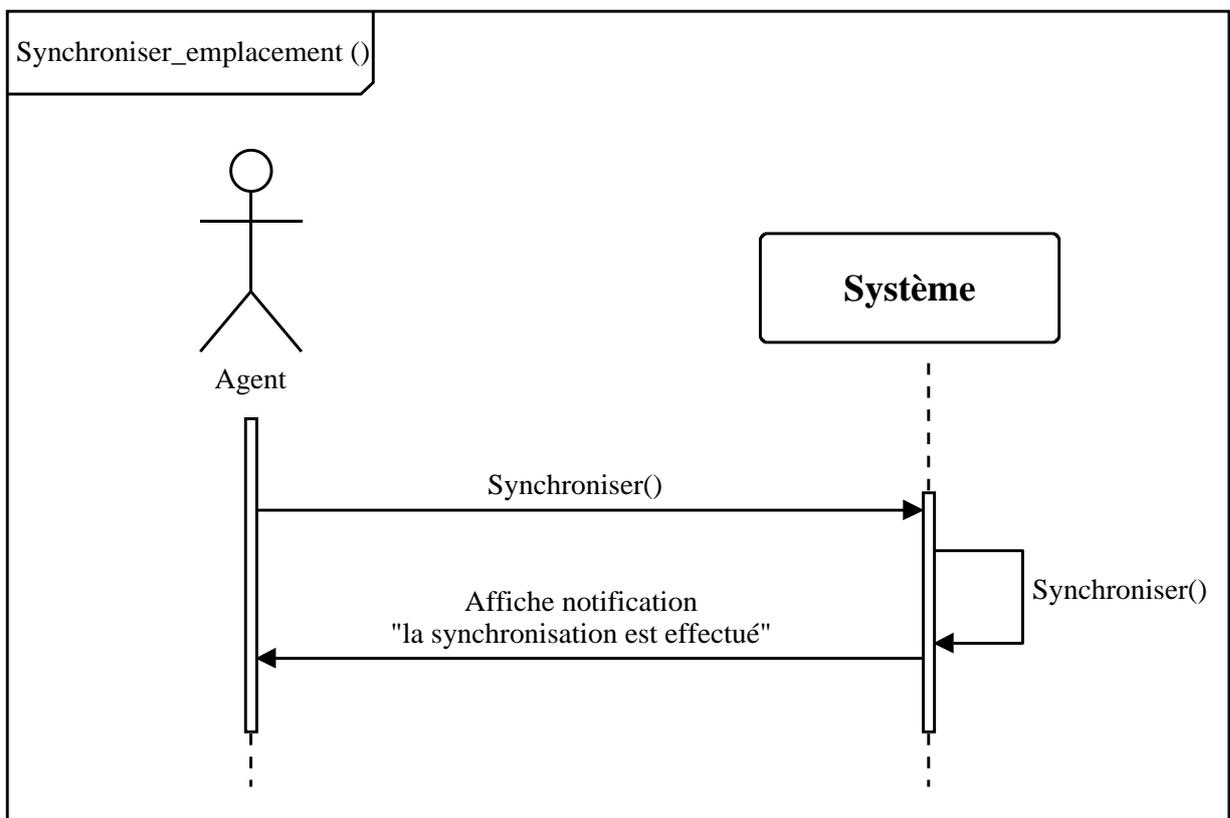


Figure 29: Diagramme de séquence du cas d'utilisation « Synchroniser l'emplacement »

5. Le modèle de domaine

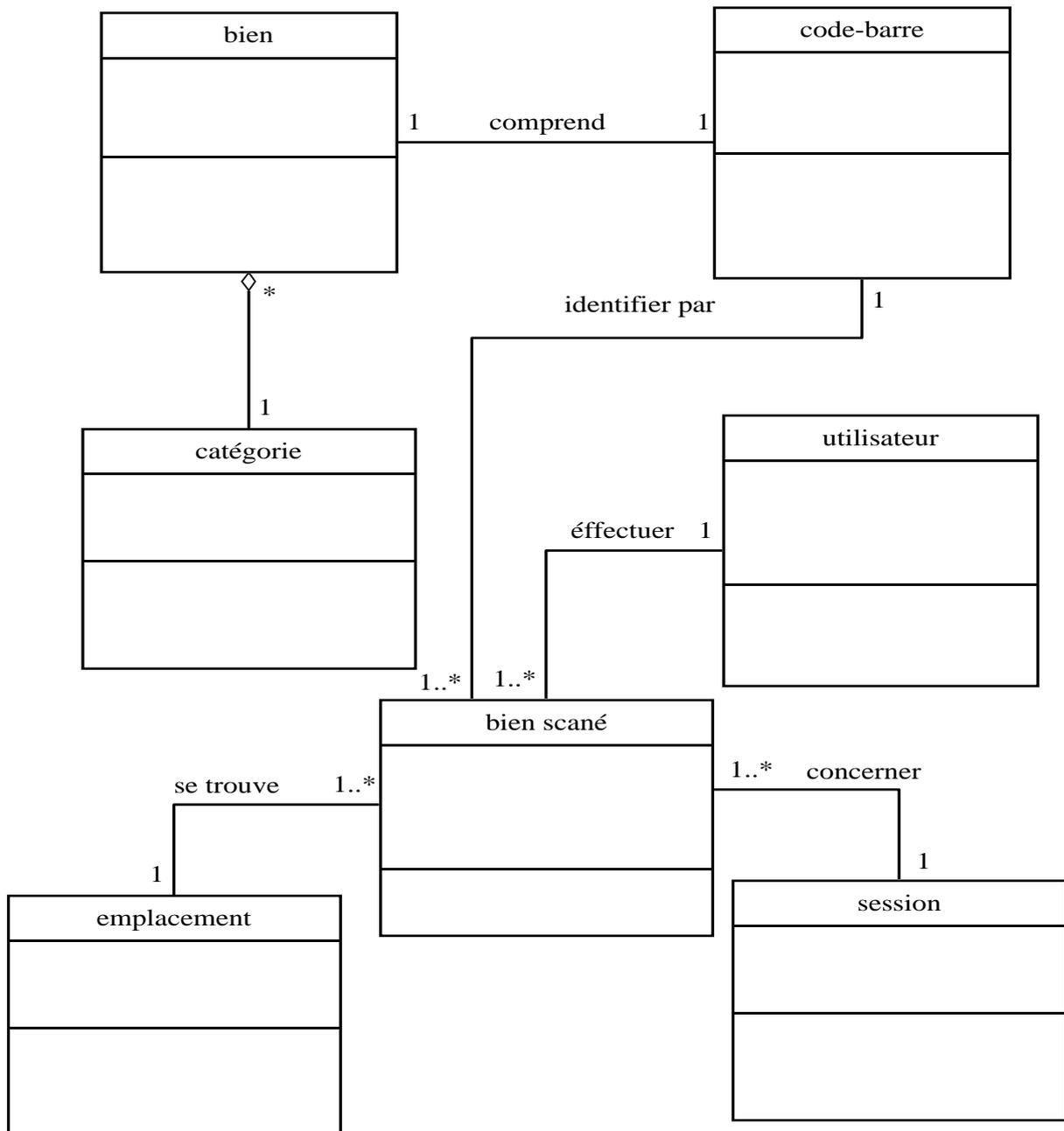


Figure 30: Le modèle de domaine.

Conclusion

Durant cette partie nous avons recueilli toutes les informations d'aspect fonctionnel qui représente différents actions (scan automatique des biens, génération de code-barres, gestion des utilisateurs...) afin de pouvoir fixer les principales fonctionnalités que doit disposer notre future Système, bien que ceci n'est qu'une première étape pour la mise en place totale de notre Système, Dans la prochaine partie nous allons commencer la capture des besoins techniques.

Partie2. Capture des Besoins Techniques

Introduction

On va s'intéresser à la branche droite du cycle en Y qui est « la capture des besoins techniques » en couvrant les contraintes qui ne traitent pas la description applicative. Nous choisissons lors de cette phase l'environnement de travail ainsi que l'architecture globale utilisée pour notre système. La capture des besoins techniques se présente selon un point de vue :

- ✓ La spécification technique du point de vue matériel.

1. Spécification technique du point de vue matériel

1.1. Configuration matérielle

1.1.1. Caractéristiques techniques

❖ Au niveau du serveur

Nous avons utilisé un ordinateur : dont les caractéristiques suivantes :

- Processeur : Intel(R) CORE (TM) i5 CPU @ 1.8 GHz.
- Mémoire installée (RAM) : 4,00 Go (3,85 G utilisable).
- Type du système : Système d'exploitation 64 bits, processeur x64.
- Débit 100 Mbit/s.
- Fast Ethernet.
- Protocole TCP/IP.

❖ Au niveau du Smartphone

- Smartphone avec système Android.
- Débit 100 Mbit/s.
- Fast Ethernet.
- Protocole TCP/IP.

1.2. Spécification de l'architecture du système



Figure 31: Organisation du modèle de spécification logicielle.

Conclusion

Dans cette partie, l'architecture physique a été choisie selon l'environnement adopté (nous avons choisi un ordinateur et un Smartphone puis les différentes caractéristiques de réseau), dans la prochaine partie nous allons présenter La phase d'analyse.

Partie3 : Analyse

Introduction

La phase d'analyse succède à la capture des besoins fonctionnels et un a pour but d'insérer une première phase sur le but de l'étape d'analyse. Elle est constituée de trois activités :

- Le découpage en catégories.
- Le développement du modèle statique.
- Le développement du modèle dynamique.

1. Découpage en catégories

Cette phase utilise la notion de package pour définir des catégories de classes d'analyse et découper le modèle UML en blocs logiques les plus indépendants possibles.

- ❖ **Définition** : une catégorie consiste en un regroupement logique de classes à forte Cohérence interne et faible couplage externe.

Le découpage en catégories se fait en trois étapes :

1.1. La répartition des classes candidates en catégories

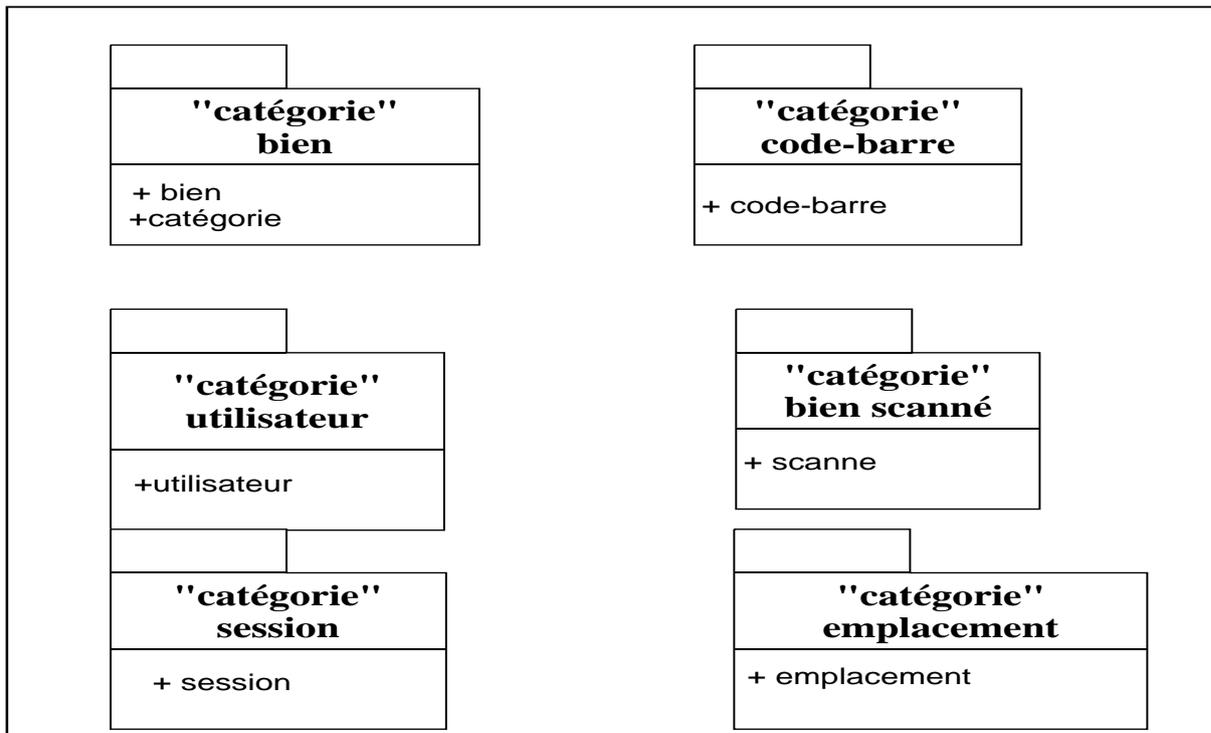


Figure 32: Découpage en catégorie de notre système.

1.2. Elaboration des diagrammes de classes préliminaires par catégorie

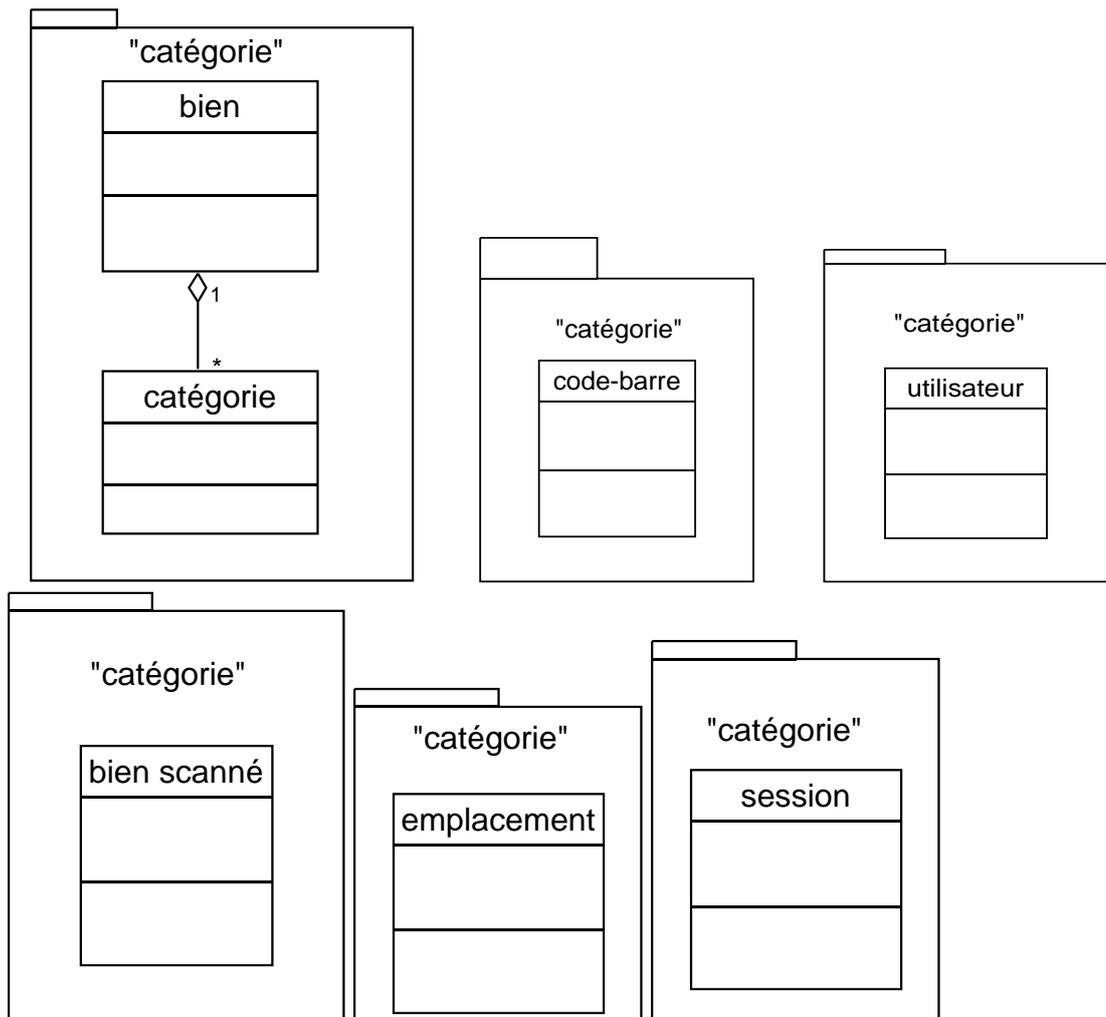


Figure 33. Découpage des classes de chaque catégorie.

1.3. Dépendance entre catégories

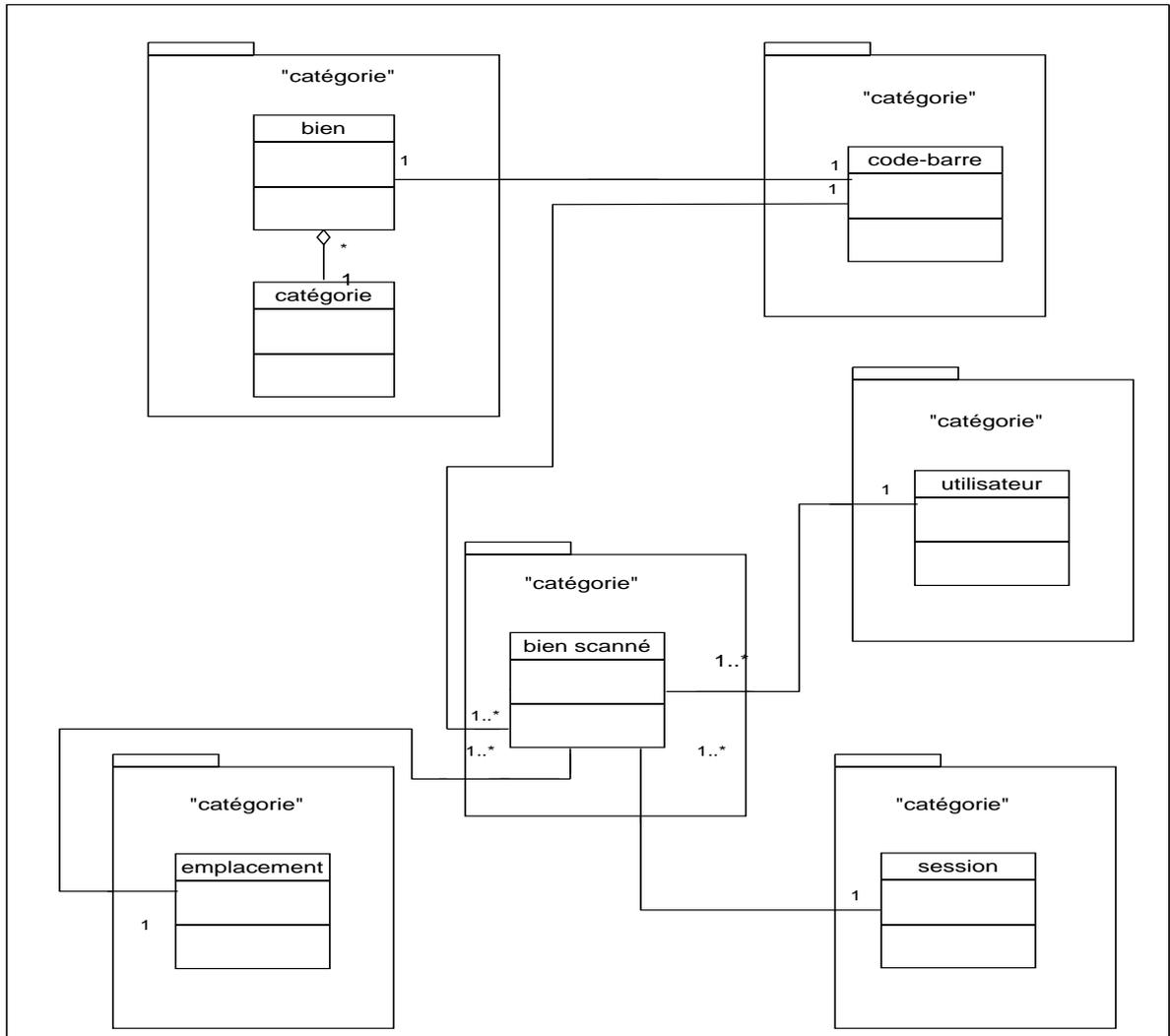


Figure 34: Dépendances entre les catégories.

2. Développement du modèle statique

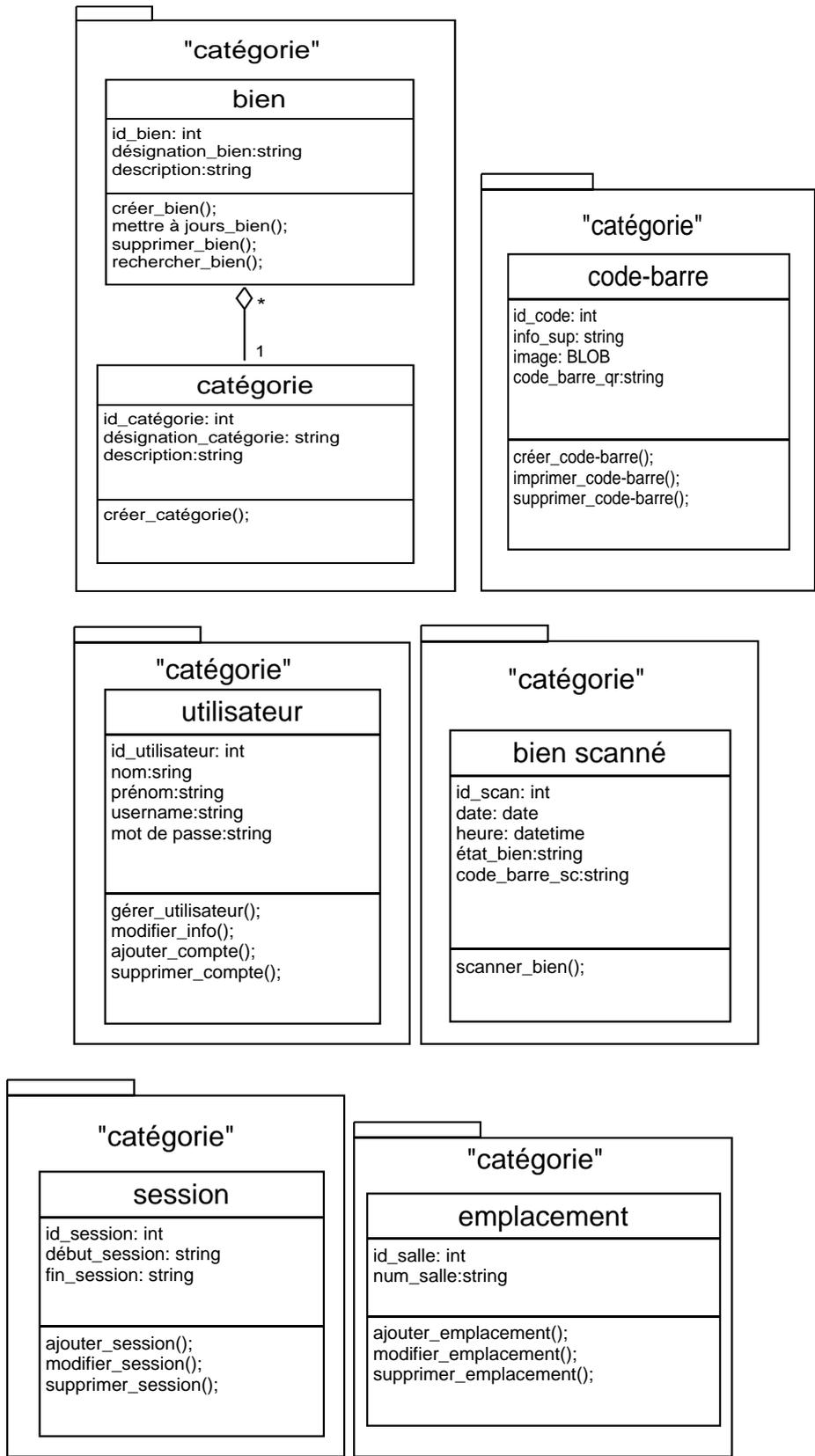


Figure 35: Diagramme de classe de chaque catégorie.

3. Développement du modèle dynamique

Le développement du modèle dynamique est la troisième activité de l'étape d'analyse. Cette activité est en relation avec l'activité de modélisation statique. Lors de cette étape, nous décrivons les différentes interactions entre les objets de notre application. En effet, nous avons utilisés le modèle dynamique : le diagramme de séquence détaillé.

3.1. Diagrammes de séquences

Le diagramme de séquence est un diagramme d'interaction entre les objets, qui met l'accent sur le classement des messages par ordre chronologique durant l'exécution du système. Un diagramme de séquence est un tableau dans lequel les objets sont rangés sur l'axe des abscisses et des messages par ordre d'apparition sur l'axe des ordonnées. Il est utilisé pour représenter certains aspects dynamiques d'un système : dans le contexte d'une opération, d'un système, d'un sous-système, d'un cas d'utilisation (un scénario d'un cas d'utilisation) selon un point de vue temporel.

En effet dans cette phase, et après identification des cas d'utilisation, nous représentons à l'aide des diagrammes de séquences les différents cas d'utilisation.

3.1.1. Diagramme de séquence du cas « Scanne code barre »

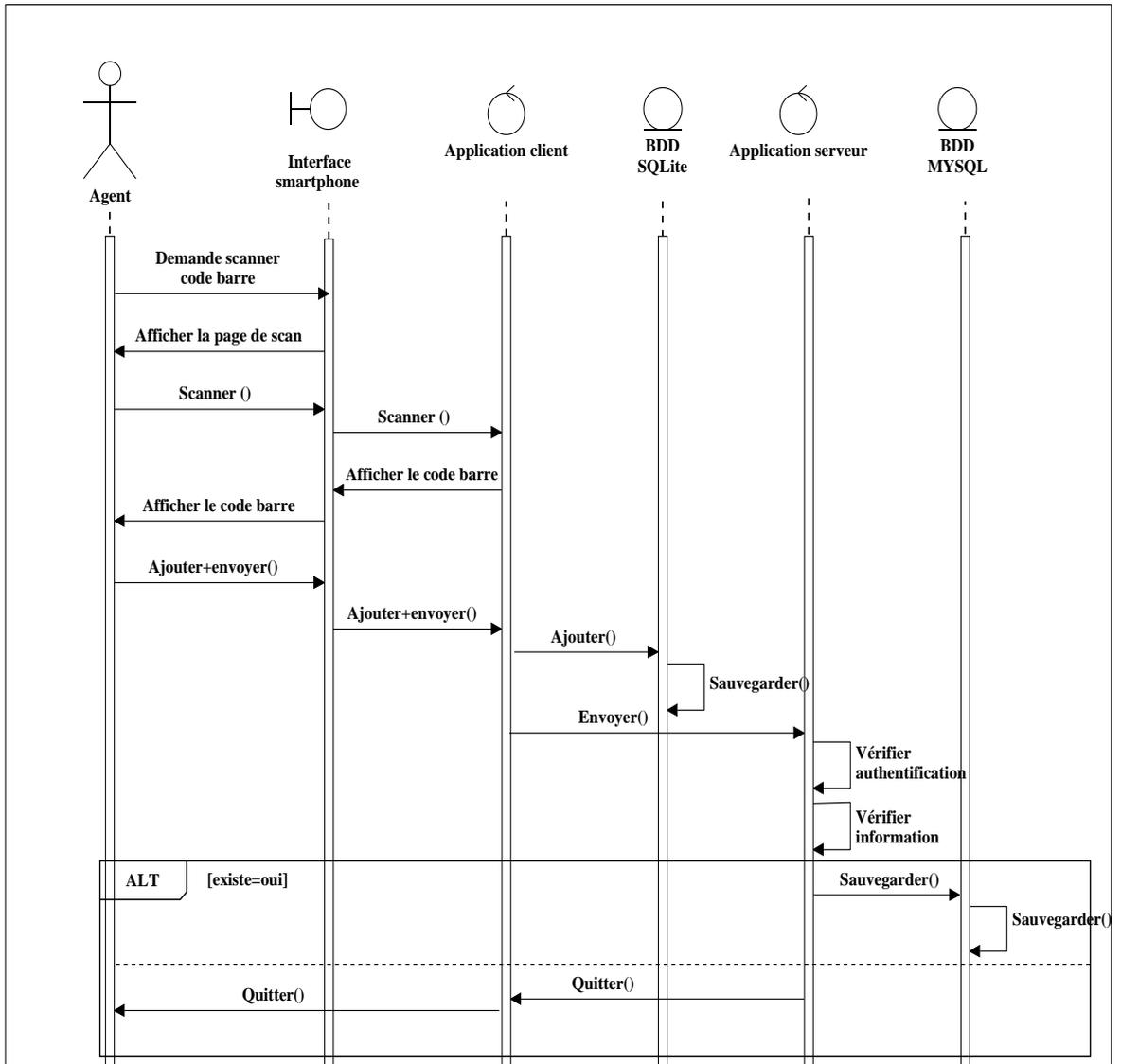


Figure 36: Diagramme de séquence du cas « Scanne code barre »

3.1.2 Diagramme de séquence du cas « Transfert de code barre »

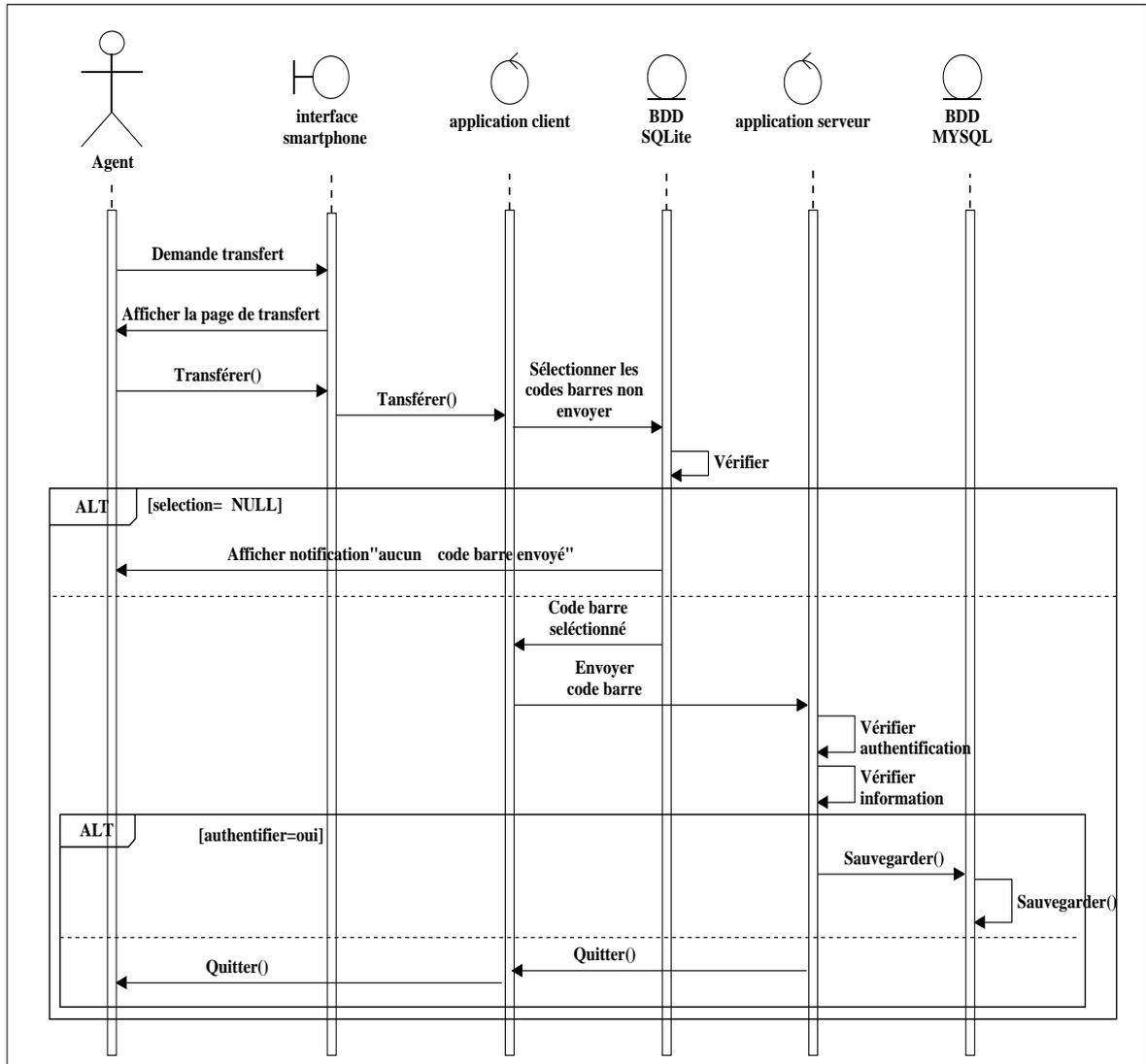


Figure 37: Diagramme de séquence du cas « Transfert de code barre »

3.1.3. Diagramme de séquence du cas « Synchronisation d'emplacement »

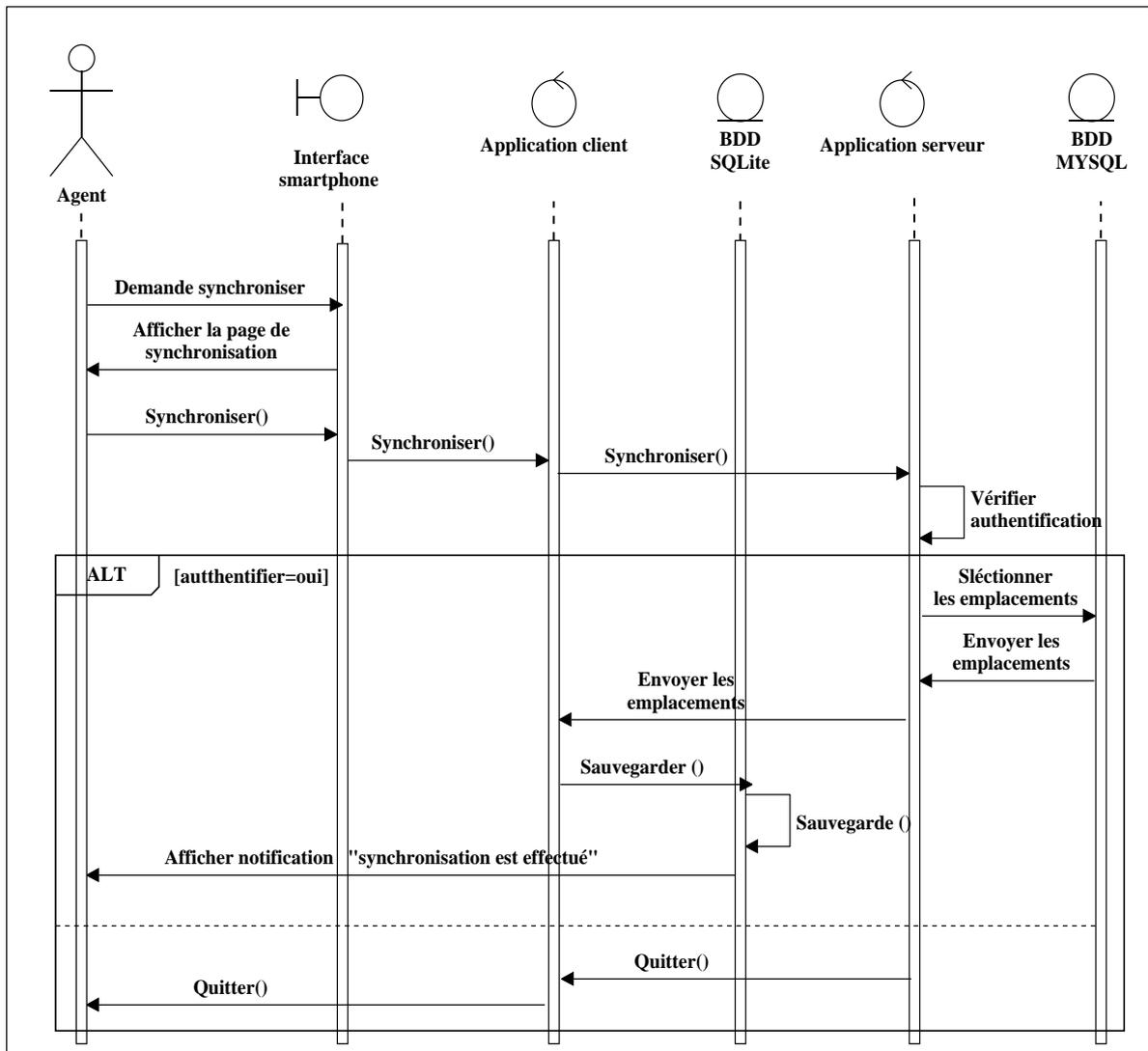


Figure 38: Diagramme de séquence du cas « Synchronisation d'emplacement »

3.1.4. Diagramme de séquence du cas « Ajouter bien »

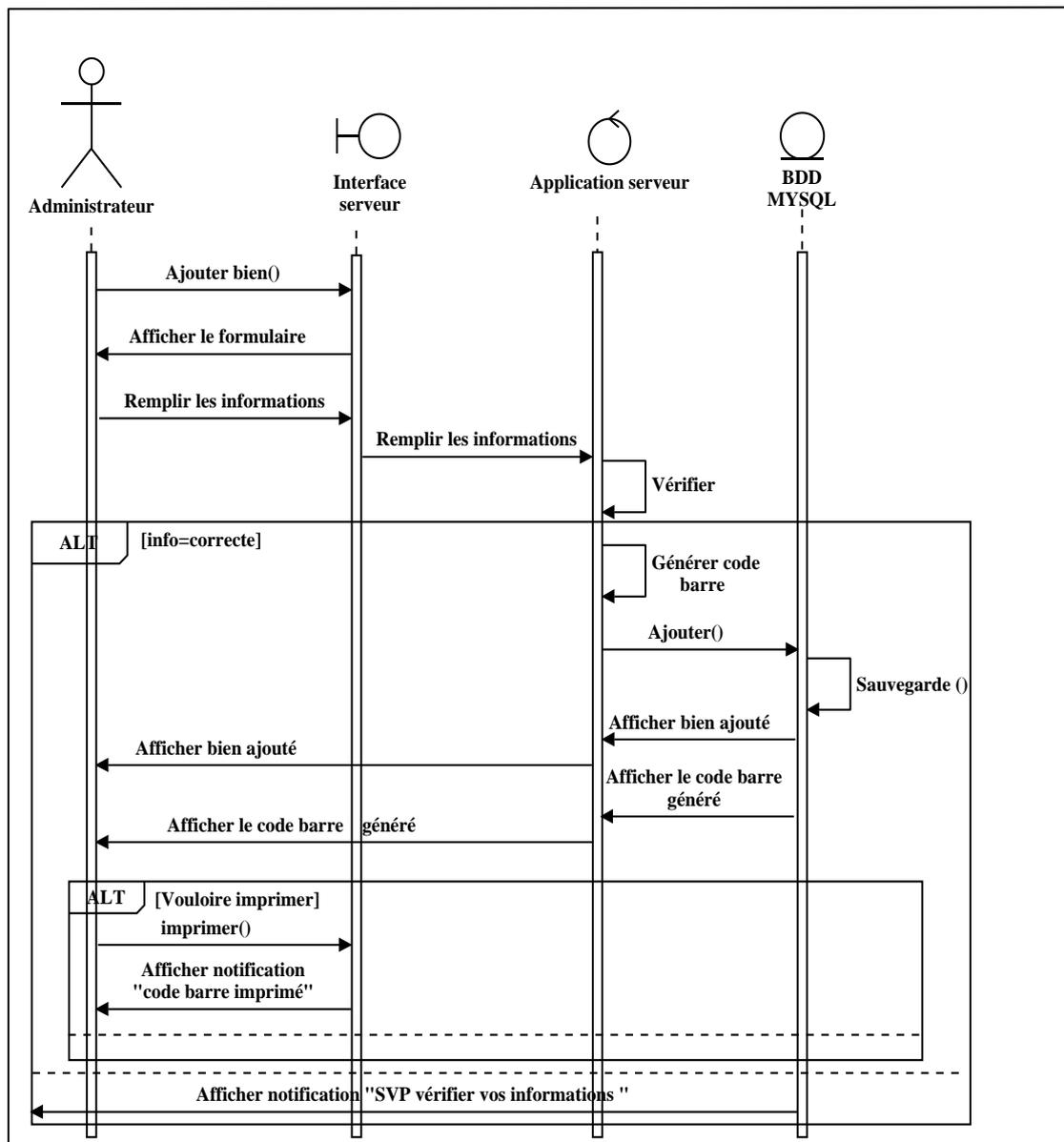


Figure 39: Diagramme de séquence du cas « Ajouter bien »

Conclusion

Au cours de cette partie, nous avons présenté l'étape d'analyse qui nous a permis de passer d'une structuration fonctionnelle via les cas d'utilisations et les packages à une structuration objet via les classes et les catégories, Dans le prochain chapitre nous allons suivre la conception préliminaire.

CHAPITRE 4

Conception



Introduction

Ce chapitre contient la phase de conception, nous allons décrire le fonctionnement de notre système, on a 2 parties : la conception préliminaire et la conception détaillée.

La conception préliminaire est certainement l'étape la plus délicate du processus 2TUP, C'est en effet à cette occasion que s'effectue la fusion des études fonctionnelles et techniques. Cette étape adapter la conception aux spécifications fournies par l'analyse et intégrer les fonctions métier et applicatives du système dans l'architecture technique.

La conception détaillée qui vient juste après est une activité qui s'inscrit dans l'organisation définie par la conception préliminaire. Il est ainsi possible de confier les catégories à des personnes différentes, qui pourront travailler indépendamment les unes des autres.

1. Conception préliminaire

1.1. Développement du modèle du déploiement

- Notre Diagramme de déploiement

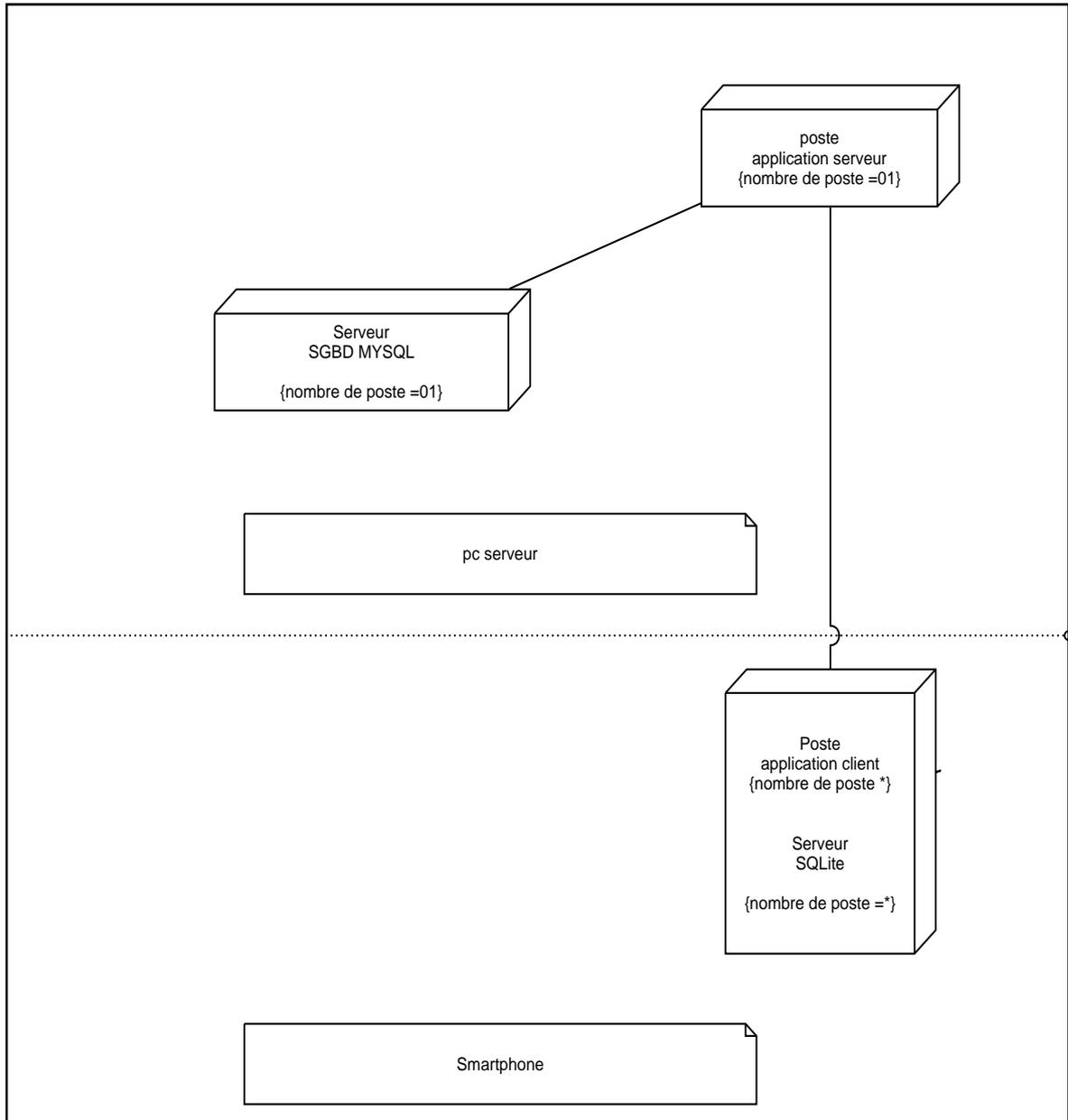


Figure 1 : Diagramme de déploiement.

1.2. Déploiement du modèle d'exploitation

Le modèle d'exploitation va définir les applications installées sur les postes de travail, les composants métier déployés sur les serveurs et les instances de base de données implantées sur les serveurs également.

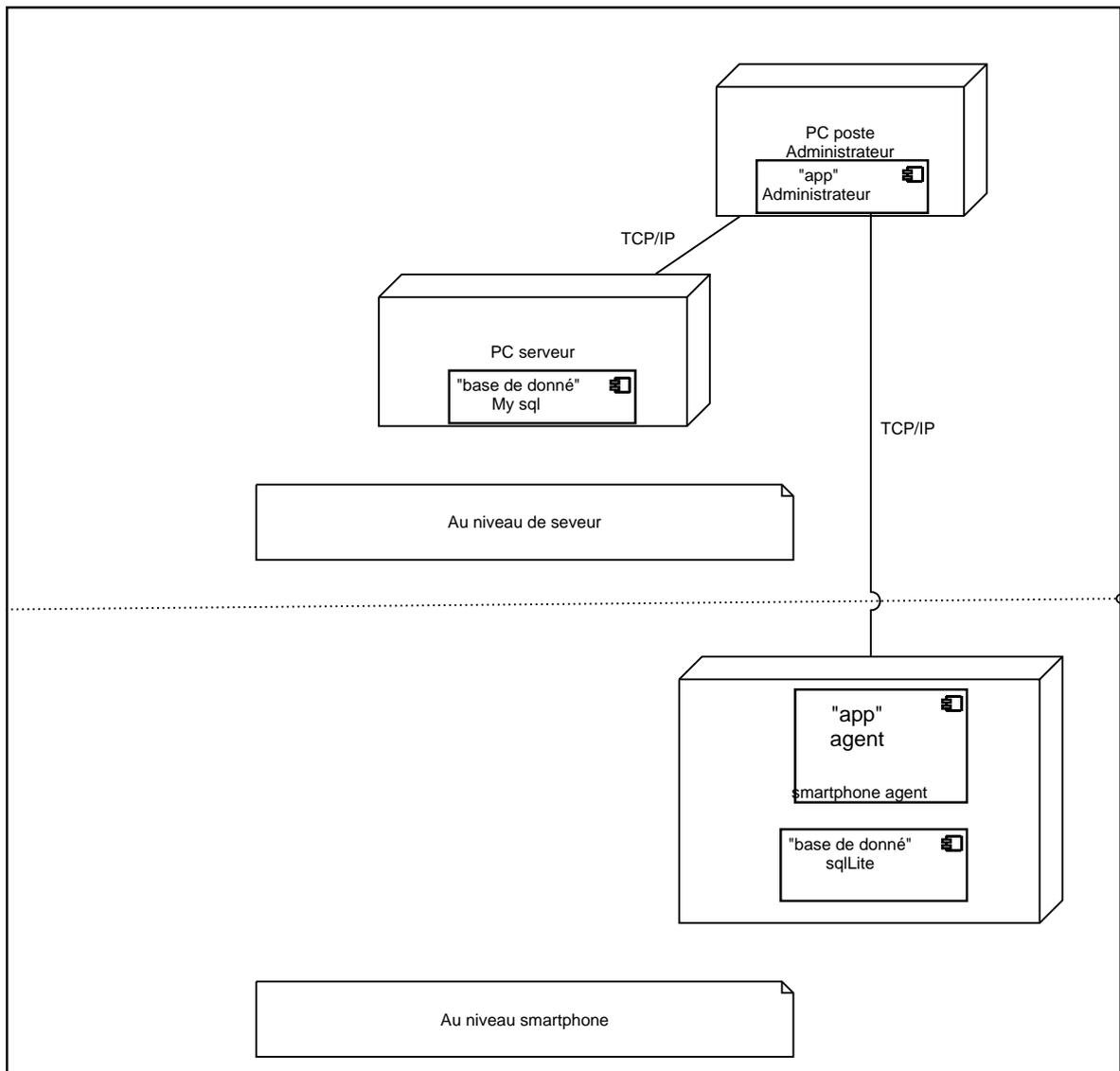


Figure 2: D efinition des applications dans mod ele statique.

2. Conception détaillée

2.1. Dictionnaire des données

❖ Les classes et les attributs

| | Description | Code | Type |
|-------------|---|-----------------------|---------------------|
| Bien | L'identifiant de bien | Id_bien | Entière |
| | Désignation de bien | Désignation_bien | Chaîne de caractère |
| | Description de bien | Description_bien | Chaîne de caractère |
| Code-barre | Identifiant de code barre | Id_code | Entière |
| | Les informations supplémentaires sur le code-barres | Info_sup | Chaîne de caractère |
| | L'image de code barre | Image | BLOB |
| | Le code barre | Code-barre-qr | Chaîne de caractère |
| catégorie | L'identifiant de catégorie | Id-catégorie | Entière |
| | Désignation de catégorie | Désignation_bien | Chaîne de caractère |
| | Description de catégorie | Description_catégorie | Chaîne de caractère |
| Utilisateur | L'identifiant de l'utilisateur | Id_utilisateur | Entière |
| | Le nom de l'utilisateur | Nom | Chaîne de caractère |
| | Le prénom de l'utilisateur | Prénom | Chaîne de caractère |
| | Username de l'utilisateur | Username | Chaîne de caractère |
| | Mot de passe de l'utilisateur | Mot de passe | Chaîne de caractère |
| Bien scanné | L'identifiant de bien scanné | Id_scan | Entière |
| | La Date de scan | Date | Date |

| | | | |
|-------------|--------------------------------|---------------|---------------------|
| | L'Heure de scan | Heure | Datetime |
| | L'état de bien | Etat_bien | Chaîne de caractère |
| | Le Code barre | Code_barre_sc | Chaîne de caractère |
| emplacement | l'identifiant de l'emplacement | Id_salle | Entière |
| | Le numéro de salle | Num_salle | Chaîne de caractère |
| Session | L'identifiant de session | Id_session | Entière |
| | Le début de session | Début_session | Chaîne de caractère |
| | La fin de session | fin_session | Chaîne de caractère |

Tableau 1: Dictionnaire de donnée avec les classes et les attributs.

2.2. Les opérations

| Classe | Opération | Description |
|-------------|--------------------------|--|
| Bien | Créer_bien (). | Créer un bien |
| | Mettre_à_jours_bien (). | Mettre à jours un bien |
| | Supprimer_bien (). | Supprimer un bien |
| | Rechercher_bien (). | Recherche |
| Code-barres | Créer_code-barre (). | Créer code barre |
| | Imprimer_code-barre (). | Imprimer code barre |
| | Supprimer_code-barre (). | Supprimer code barre |
| Catégorie | Créer_catégorie (). | Créer une catégorie |
| | Modifier_catégorie (). | Modifier une catégorie |
| | Supprimer_catégorie (). | Supprimer une catégorie |
| | Consulter_catégorie (). | Consulter une catégorie |
| Utilisateur | Gérer_utilisateur (). | Ajouter, modifier, supprimer, consulter utilisateur. |

| | | |
|-------------|---------------------------|-------------------------|
| Bien scanné | Scanner_bien (). | Scanne un bien |
| Session | Ajouter_session (). | Ajouter une session |
| | Modifier_session (). | Modifier une session |
| | Supprimer_session (). | Supprimer une session |
| | Consulter_session (). | Consulter une session |
| Emplacement | Ajouter_emplacement (). | Ajouter l'emplacement |
| | Modifier_emplacement (). | Modifier l'emplacement |
| | Supprimer_emplacement (). | Supprimer l'emplacement |
| | Consulter_emplacement (). | Consulter l'emplacement |

Tableau 2:Dictionnaire de donnée avec les opérations.

2.3. Diagramme de classe détaillée

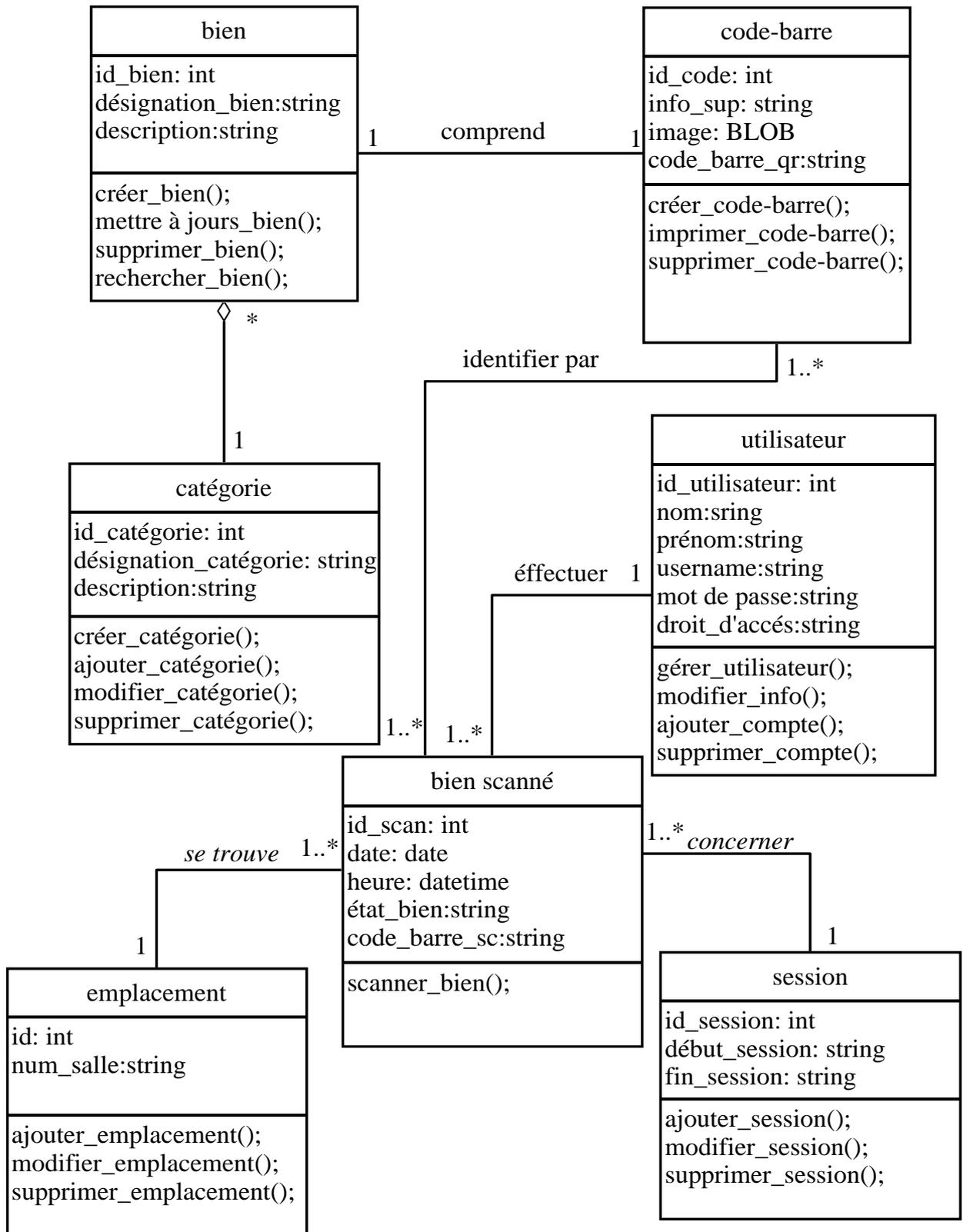


Figure 3 : Diagramme de classe détaillé de notre système.

3. Le modèle relationnel

A partir de la description conceptuelle que nous avons effectuée, on peut réaliser le modèle relationnel vu que le système d'information ne peut pas le manipuler directement, et cela en utilisant des règles de passage de l'UML vers le relationnel.

Quelques notions essentielles :

- **Domaine:** c'est l'ensemble de valeurs d'un attribut.
- **Relation:** c'est un sous-ensemble du produit cartésien d'une liste de domaines.

C'est en fait un tableau à deux dimensions dont les colonnes correspondent aux Domaines et dont les lignes contiennent des tuples. On associe un nom à chaque Colonne.

- **Attribut:** c'est une colonne d'une relation, caractérisé par un nom.
- **Tuples:** c'est la liste des valeurs d'une ligne d'une relation.
- **Cardinalité:** elle permet de définir les conditions de participation d'une entité à une relation. Toutefois, une entité peut participer à plusieurs relations.
- **L'arité:** est le nombre d'attributs d'une relation.
- **Clé:** on distingue deux types de clés:
 - **Clé primaire:** ensemble d'attributs dont les valeurs permettent de distinguer les nuplets les uns des autres (notion d'identifiant).
 - **Clé étrangère:** attribut qui est clé primaire d'une autre entité.

3.1. Les règles de passage

La réalisation du modèle relationnel peut être faite à partir de la description conceptuelle. Ce passage de modèle doit respecter un certain nombre de règles.

- **Règle1:** Toute entité devient une relation ayant pour clé primaire son identifiant.

Chaque propriété se transforme en attribut.

- **Règle2:** Toute association hiérarchique (de type $[1, n]$) se traduit par une clé étrangère.
- **Règle3:** Toute association non hiérarchique (de type $[n, n]$ ou de dimension >2) devient une relation.
- **Règle4:** Les entités n'ayant que leur identifiant comme attribut ne deviennent pas des relations, mais des attributs dans les autres relations liées.
- **Règle5:** cas de héritage, transformer chaque sous classe en une relation, la clé primaire de la super classe devient clé primaire de chaque sous classe.
- **Règle6:** cas de composition, la clé primaire de la classe composée devient clé étrangère de la classe composant.

- **Règle7:** cas d'agrégation, le même principe que la Règle 2.

3.2. Les règles de gestion

- ✓ Un bien comprend à un code Barre.
- ✓ Un utilisateur peut scanner plusieurs biens.
- ✓ Une catégorie peut contenir plusieurs biens.
- ✓ Dans un emplacement on peut trouver plusieurs biens.
- ✓ Une session concerne plusieurs biens scannés.

❖ Les tables de la base de données

En se basant sur les règles ci-dessus, nous avons converti les classes entités et leurs associations, à des tables dans la base données. Les tables générées sont :

| Nom de la table | Les attributs |
|-----------------|--|
| Catégorie | <u>id catégorie</u> , désignation_catégorie, description. |
| Bien | #id_catégorie, <u>id bien</u> , désignation_bien, description. |
| Code-barres | #id_bien, <u>id code</u> , info_sup, image, code_barre_qr. |
| Utilisateur | <u>id utilisateur</u> , nom, prénom, username, mot de passe. |
| bien scanné | #id_code, #id_utilisateur, #id_emplacement, #id_session, <u>id scan</u> , date, heure, état_bien, code_barre_sc. |
| Emplacement | <u>id emplacement</u> , num_salle. |
| Session | <u>id session</u> , début_session, fin_session. |

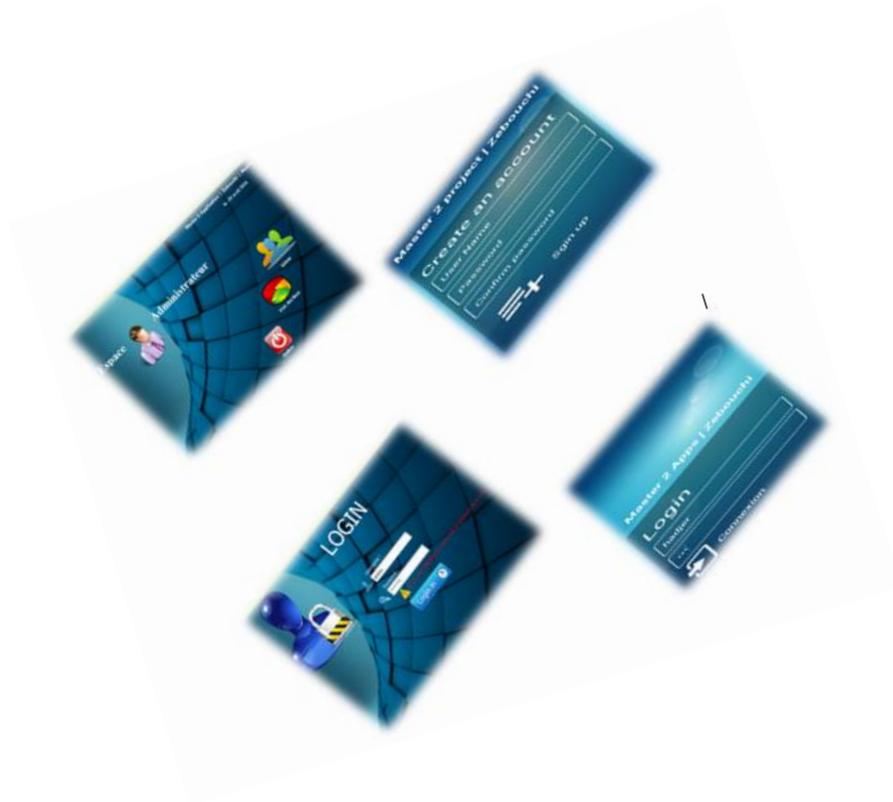
Tableau 3 : Les tables de la base de données

Conclusion

La conception nous a permis de documenter précisément le code qui va être produit. Nous regrettons de n'avoir pas eu le temps de spécifier de façon détaillée les opérations de chaque classe. Dans le prochain chapitre nous allons présenter la partie de réalisation.

CHAPITRE 5

Réalisation



Introduction

Ce chapitre est consacré à la phase d'implémentation. Cette dernière phase a pour but de générer un code exécutable qui répond aux besoins utilisateur. Nous allons présenter les interfaces principales de notre application.

1. Au niveau du serveur

❖ L'interface d'authentification

Cette interface permet aux utilisateurs d'accéder à l'application pour l'exploitation les services offerts par celle-ci. Si le mot de passe est incorrect, le système affiche une notification pour déterminer l'erreur.



Figure 1 : Fenêtre d'authentification.

❖ Espace administrateur

Cette espace n'est validée que si les champs de la fenêtre précédente sont remplis correctement par les informations du compte de l'administrateur. Elle affiche à l'administrateur quelques liens offerts par l'application.



Figure 2 : Espace administrateur.

❖ Fenêtre « Gérer utilisateur »

Cette fenêtre présente la gestion des inventaires.

Gestion d'inventaire

Master 2 Application | Zebouchi | Maiche

Gestion d'inventaire : Accueil Gérer Etat des lieux Quitter

| Utilisateur | Catégorie | Emplacement | Session | Bien scanné | Bien | |
|-------------|-----------|-------------|--------------------|-------------|-------------|----------------|
| id_uti | nom | prenom | adress | pseudo | mot_de_pass | droit_dacces |
| 14 | zebouchi | abdou | abdou.hd@gmail.com | abdou | abdou1992 | administrateur |
| 15 | maiche | hajer | hajer@gmail.com | hajer | hajer1992 | agent |
| 18 | housssam | houssame | kakjd@hdhd.com | bibouli | bibouli10 | agent |
| 19 | seraoi | wadjih | wajih@gdgd.com | wajih | wajih1991 | agent |
| 20 | adlkj | fkjddd | a@ee.com | azazklalz | aalkaeae | administrateur |

Nom :

Prenom :

Adresse : (ex: examle@xx.xx)

Pseudo :

Mot de passe :

Droit d'accès : administrateur

Recherche

Recherche par : id_uti

Recherche

Actualiser

Ajouter Modifier Supprimer

Figure 3 : Fenêtre « Gérer utilisateur ».

❖ Fenêtre « Session »

Cette fenêtre permet aux utilisateurs de déterminer le début et la fin de session.



Figure 4 : Fenêtre «Session».

❖ Fenêtre « Catégorie »

Cette fenêtre permet d'ajouter, modifier, supprimer une catégorie.



Figure 5 : Fenêtre « Catégorie ».

❖ Fenêtre « Bien »

Permet à l'administrateur d'ajouter, modifier, supprimer un bien et générer automatiquement son code barre QR, et permet aussi d'imprimer ce code barre.

| id_bien | désignation | désignation_bien | description_bien | code_barre_qr | info_sup | image |
|---------|-------------|--------------------|------------------|----------------------------|-------------------------------|------------------------|
| 125 | table | hp | hp e150 | 125 19A | 19A | C:/Users/abdoun/Doc... |
| 126 | table | rtffff | rrr | 126 r | r | C:/Users/abdoun/Doc... |
| 120 | pc | Dell | inspiron 15 | 120 core i5 ram 4GO ... | core i5 ram 4GO mem... | C:/Users/abdoun/Doc... |
| 121 | pc | Toshiba | a 13 | 121 core i3 ram 4GO ... | core i3 ram 4GO mem... | C:/Users/abdoun/Doc... |
| 122 | pc | Acer | Em145 plus | 122 core i3 ram 4GO ... | core i3 ram 4GO mem... | C:/Users/abdoun/Doc... |
| 114 | téléphone | Samsung | Galaxy S7 | 114 ram 4 Mémoire 32... | ram 4 Mémoire 32 Go | C:/Users/abdoun/Doc... |
| 116 | téléphone | LG | G5 | 116 RAM 3Go Mémoir... | RAM 3Go Mémoire 32 ... | C:/Users/abdoun/Doc... |
| 117 | téléphone | Xperia™ Z5 Premium | Z5 Premium | 117 RAM 3 Go Mémoir... | RAM 3 Go Mémoire 32 ... | C:/Users/abdoun/Doc... |
| 118 | téléphone | iPhone | 6 Plus | 118 Capacité16 Go 64... | Capacité16 Go 64 Go | C:/Users/abdoun/Doc... |
| 123 | imprimante | Canon | cn 10 | 123 debut 2 f/s lazer n... | debut 2 f/s lazer noir et ... | C:/Users/abdoun/Doc... |

Figure 6 : Fenêtre « Bien ».

❖ Fenêtre « Bien scanné »

Cette fenêtre pour l'affichage des biens scannés par l'utilisateur.



Figure 7 : Fenêtre « Bien scanné ».

❖ Fenêtre « Etat des lieux »

A partir de cette fenêtre l'administrateur peut déterminer les biens endommagés et les biens manquants, et déterminer aussi l'emplacement de chaque bien.

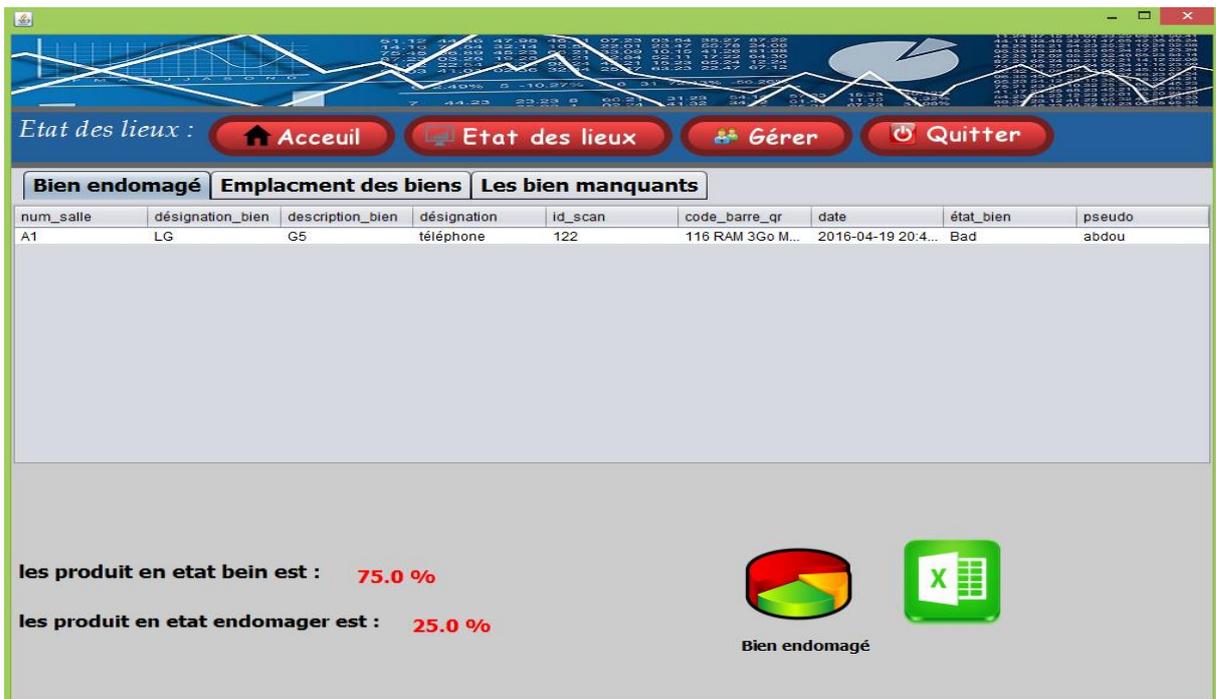


Figure 8 : Fenêtre « Etat des lieux ».

2. Au niveau du Smartphone

❖ Interface de démarrage

Cette interface présente notre application mobile.

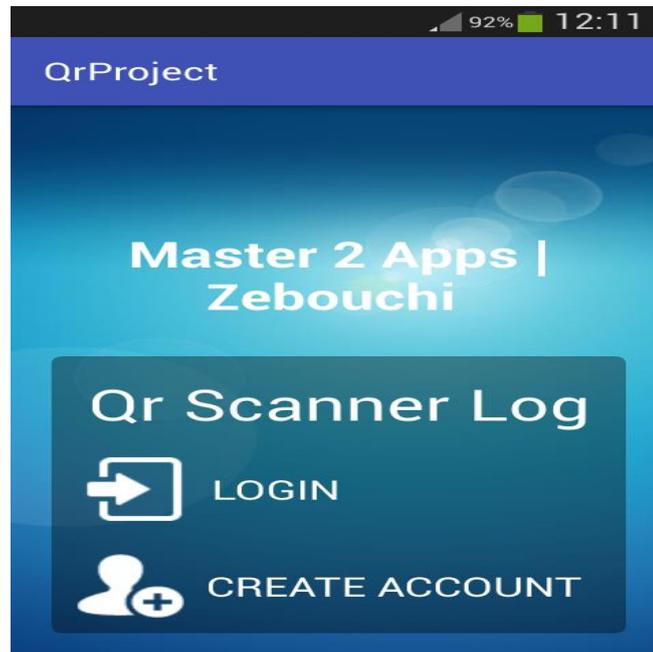


Figure 9 : Interface de démarrage.

❖ Fenêtre « Créer un compte »

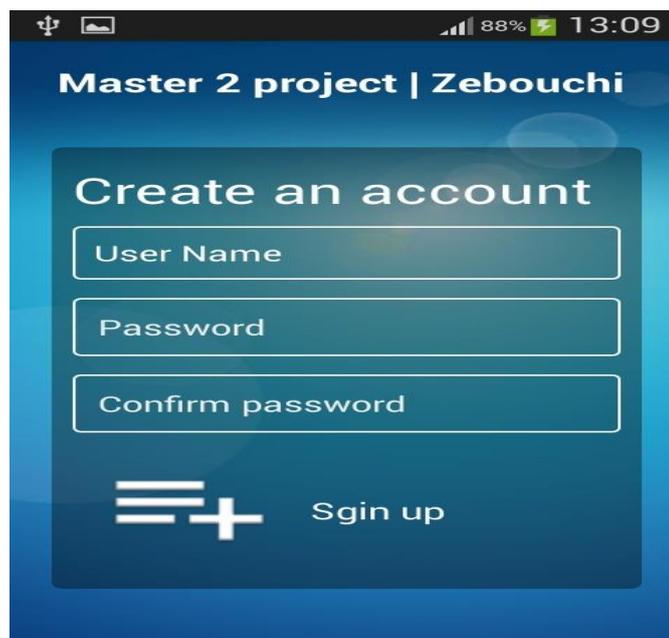


Figure 10 : Fenêtre « Créer un compte ».

❖ Interface d'authentification

Cette fenêtre permet à l'utilisateur d'accéder à notre application, l'utilisateur doit remplir les champs de façon correcte pour passer à la page d'accueil.

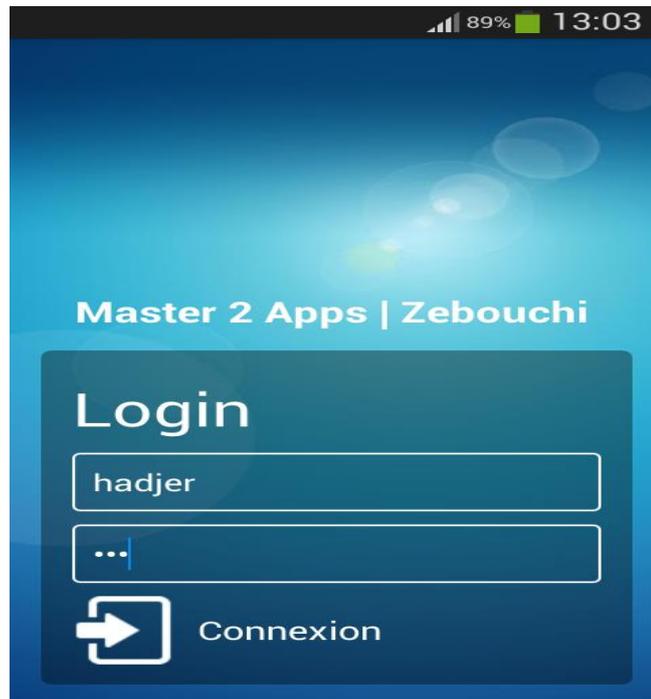


Figure 11 : Interface d'authentification.

❖ L'interface d'accueil



Figure 12 : Interface d'accueil.

❖ Fenêtre « Scan »

Cette fenêtre pour scanner le code barre.

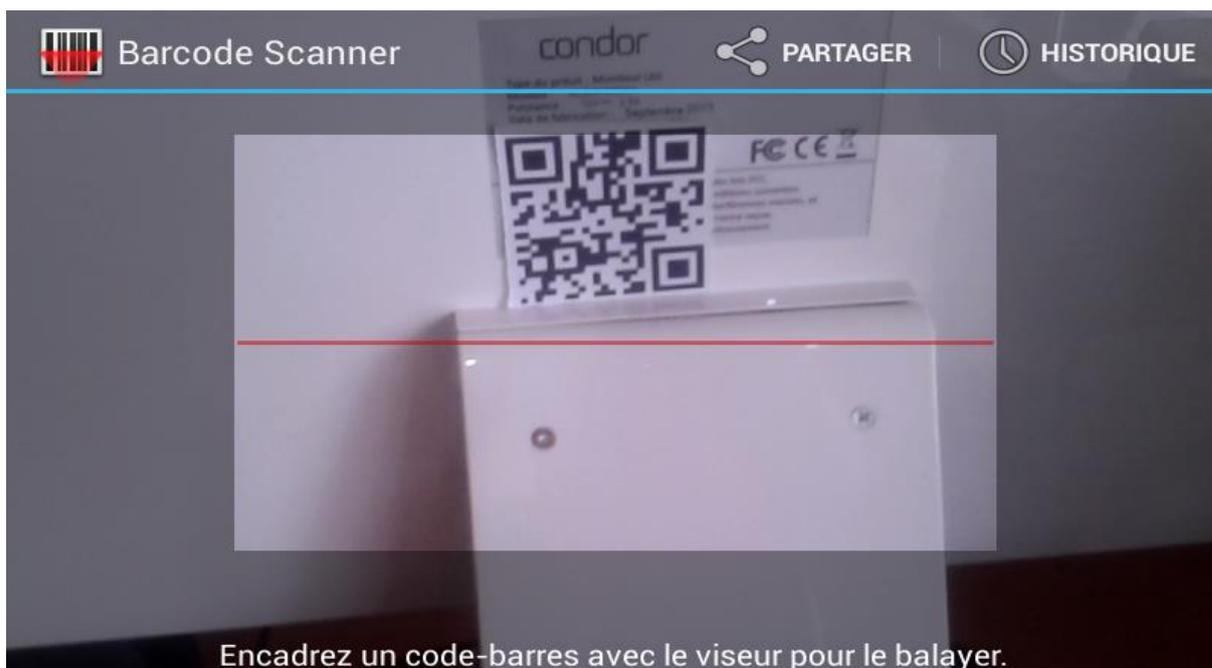


Figure 13 : Fenêtre « Scan ».

❖ Fenêtre « Envoyer le code barre »

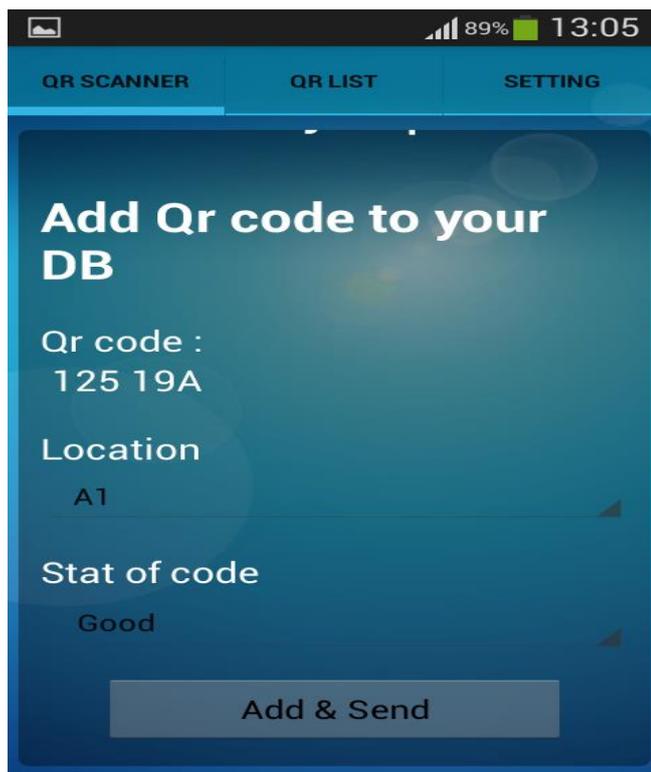


Figure 14 : Fenêtre « Envoyer le code barre ».

❖ Fenêtre « Affichage et transfert »

Cette fenêtre permet aux utilisateurs d'afficher les codes-barres scannés et l'envoyer via le serveur.

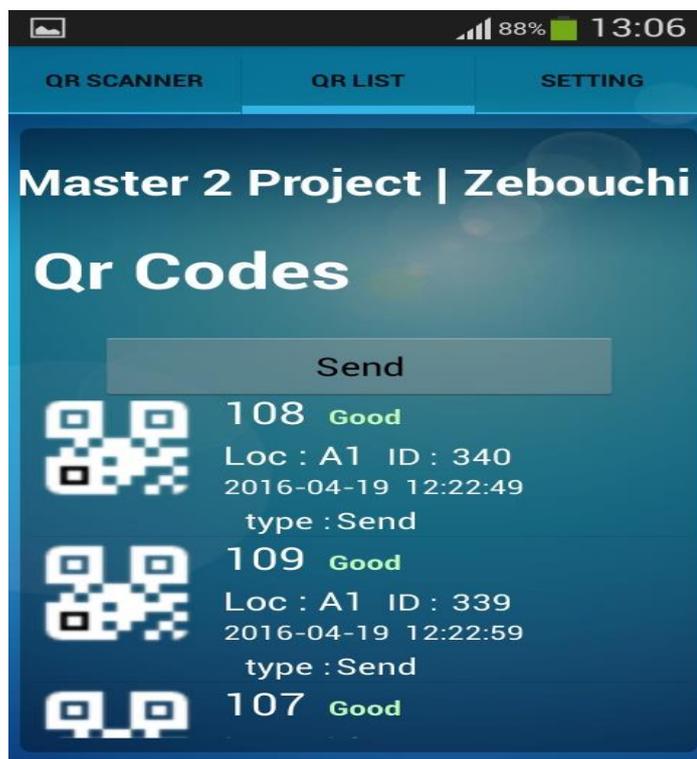


Figure 15 : Fenêtre « Affichage et transfert ».

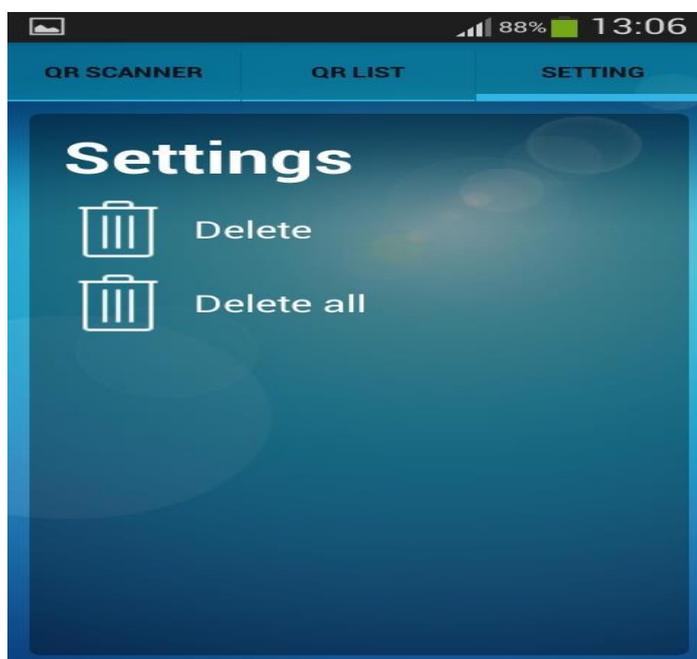
❖ Fenêtre « Paramètre »

Figure 16 : Fenêtre « Paramètre ».

❖ Fenêtre « Synchronisation des emplacements »

Cette fenêtre permet aux utilisateurs de synchroniser les emplacements.



Figure 17 : Fenêtre « Synchronisation des emplacements ».

❖ Fenêtre « Ajouter IP »

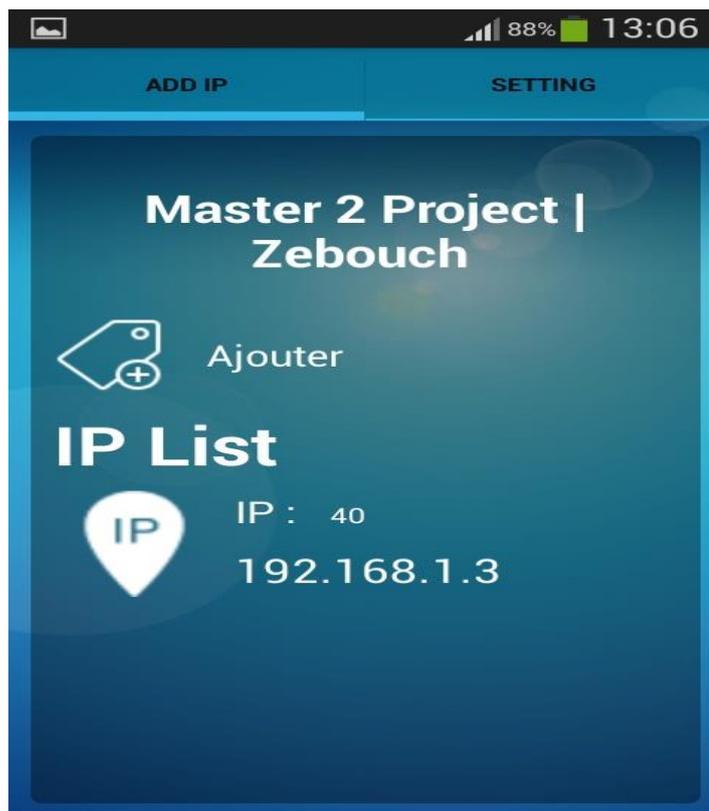


Figure 18 : Fenêtre « Ajouter IP ».

❖ Fenêtre « Modifier IP »

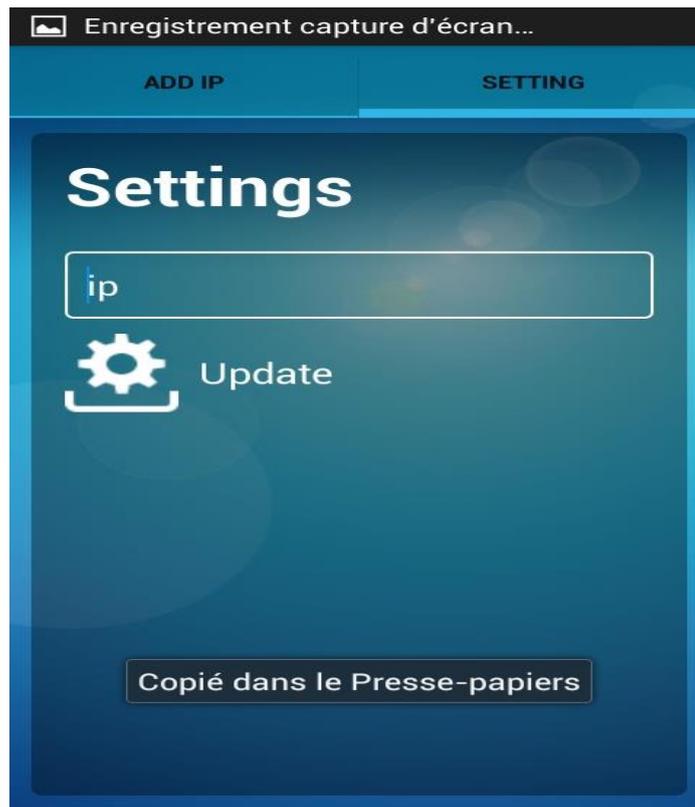


Figure 19 : Fenêtre « Modifier IP ».

Conclusion

La phase d'implémentation a été concrétisée par la réalisation de l'application, en respectant la modélisation qu'on a présentée dans le chapitre précédent. Les tests effectués, durant l'étape finale, confirment que la solution proposée répond aux besoins de la gestion des inventaires, néanmoins notre étude couvre juste une partie des problèmes existant actuellement, avec la possibilité d'effectuer des extensions afin d'enrichir les services offerts par notre application.

CONCLUSION GENERALE

Dans notre étude nous avons développé un système informatique de gestion d'inventaires basé sur les codes-barres. Dans ce contexte, nous avons cherché à développer une application fiable, flexible et évolutive répond aux besoins des utilisateurs. Notre application est indispensable pour améliorer la productivité des entreprise, elle est rapides, pratiques et répond parfaitement aux différents besoins des entreprises dans différents domaines.

Pour la conception, nous avons eu recours à la méthode 2TUP. Cette approche nous a permis de bien comprendre la problématique et de bien modéliser les objectifs à atteindre.

Pour la réalisation, de côté serveur nous avons utilisé JAVA comme langage de programmation et MySQL comme système de gestion de base de données, et de côté Smartphone, nous avons utilisé ANDROID STUDIO pour le développement et SQLite comme système de gestion de base de données.

Le plus grand bénéfice tiré de cette expérience est que nous avons eu la chance d'enrichir nos connaissances dans des domaines variés comme: la manipulation des codes-barres, et la programmation des équipements mobiles avec système ANDROID, le langage JAVA., les SGBD MySQL et SQLite.

Pour le système produit et bien que nous avons pu couvrir les besoins initial de notre projet, il y a beaucoup de fonctionnalités que nous espérons intégrer comme par exemple la localisation GPS des biens qui sera très bénéfique dans la gestion des parcs automobiles.

Référence

| | |
|------|---|
| [1] | http://www.codebarres.com/frprimer.pdf . |
| [2] | http://amcubat.be/docpmb/codes-barres-concepts-base . |
| [3] | http://www.codebarres.com/frprimer.pdf . |
| [4] | http://d1n7iqsz6ob2ad.cloudfront.net/document/pdf/5385cd6e5483b.pdf |
| [5] | http://www.qrdresscode.com/pages/Definition_QR_code-3409034.html |
| [6] | http://www.code2d.ch/fr/avantages . |
| [7] | http://www.dicodunet.com/definitions/developpement/j2ee.html . |
| [8] | http://jlafosse.developpez.com/java/developpement-n-tiers/plate-forme-java-EE/ . |
| [9] | Programmation sousAndroid, Gauthier Picard,Ecole Nationale Supérieure des Mines, 2012 |
| [10] | http://blog.erlem.fr/android/31-architecture-generale-d-android |
| [11] | Ferhat Yasser, LaaouarDjaafar ; Réalisation d'une application mobile de gestion de la vie estudiantine pour les Smartphones Android, Centre Universitaire de Mila 2013. |
| [12] | http://igm.univ-mlv.fr/~dr/XPOSE2011/SDKAndroid/sdk.html . |
| [13] | http://www.mobizel.com/2015/04/definition-cest-quoi-un-sdk/ . |
| [14] | http://igm.univ-mlv.fr/~dr/XPOSE2014/lesconsignes/android.html . |
| [15] | http://cyberzoide.developpez.com/php4/mysql/ |
| [16] | http://www.finalclap.com/faq/180-sqlite-definition . |
| [17] | Les Sockets, Karima Boudaoud, université Nice, IUT GTR-Sophia Antipolis-. |
| [18] | http://jayelco-android.blogspot.com/2009/11/scan-bibliotheque-zxing.html . |
| [19] | DI GALLO Frédéric ,Methodologie des systèmes d'information – UML, 2000-2001. |