



# **UNIVERSITE DE FIANARANTSOA**

## **ECOLE NATIONALE D'INFORMATIQUE**

**MEMOIRE DE FIN D'ETUDES POUR L'OBTENTION DU  
DIPLOME DE MASTER PROFESSIONNELLE EN  
INFORMATIQUE**

**Parcours : Génie logiciel et Base de Données**

**Intitulé :**

### **CONCEPTION ET REALISATION D'UNE APPLICATION DE PRISE DE COMMANDE MOBILE**

**Présenté le 21 Mars 2017 par :  
RANDRIAMBOLA Toky Nantenaina**

**Membres du Jury :**

**Président du jury : Monsieur RAMAMONJISOA Bertin Olivier, Professeur titulaire  
Examineur : Monsieur RAZAFIMANDIMBY Josvah Paul, Professeur titulaire**

**Rapporteurs : Monsieur RATIARSON Venot, Maître de conférences  
Monsieur ANDRIANDRANTONAVALONA Rado, Directeur du pôle  
JAVA**



# CURRICULUM VITAE

RANDRIAMBOLA

Toky Nantenaina

Né le 16 Décembre 1994 à Antananarivo

Lot 140 AB Ambohipanja

Antanananarivo 103

Adresse e-mail : randriambolat@gmail.com

Téléphone : 0343131468

Célibataire



## FORMATIONS ET DIPLOMES

- 2015-2016** : Deuxième année en master professionnel à l'Ecole Nationale d'Informatique, Parcours : Génie logiciel et base de données
- 2014-2015** : Première année en master professionnel à l'Ecole Nationale d'Informatique, Parcours : Génie logiciel et base de données
- 2013-2014** : Troisième année en licence professionnelle à l'Ecole Nationale d'Informatique, Parcours : Génie logiciel et base de données  
Diplôme obtenue : Licence professionnelle
- 2012-2013** : Deuxième année en licence professionnelle à l'Ecole Nationale d'Informatique, Parcours : Génie logiciel et base de données.
- 2011-2012** : Première année en licence professionnelle à l'Ecole Nationale d'Informatique, Parcours : Génie logiciel et base de données.
- 2010-2011** : Terminale au Collège Saint Joseph Mahamasina.  
Diplôme obtenu : Baccalauréat série C, mention : Bien.

## EXPERIENCES PROFESSIONNELLES

- 2015-2016** : - Création d'une application de résolution d'un jeu de taquin avec JAVA.
- 2014-2015** : - Création d'une application web de gestion de scolarité illustrant une étude sur les outils de reporting open source, utilisant JasperReport avec JSF, BootsFaces, Hibernate, MySQL et Glassfish.  
- Création d'une application web de gestion de stock avec JSF, PrimeFaces, Hibernate et MySQL utilisant le server Glassfish.
- 2013-2014** : - Stage au sein de l'Ingenosya Madagascar
  - Durée : 3 mois
  - Thème : Conception et réalisation d'une application mobile d'enquête sur terrain pour le FID (MISFID)

- Outils : Eclipse adt, JAVA, SQLite, JSON.
- Création d'une application de gestion des notes avec JAVA et MySQL.

- 2012-2013 :**
- Stage au sein de l'IGE+XAO Madagascar
    - Durée : 3 mois
    - Thème : Réalisation d'outils de contrôle xml.
    - Outils : Visual Studio 2010, C#.NET, xml.
  - Création d'une application web « gestion des commandes » avec PHP et MySQL.
  - Création d'une application « gestion des commandes » avec Visual Basic et MS ACCESS.
- 2011-2012 :**
- Gestion des cours au sein de l'Ecole Nationale d'Informatique sous Debian 6 en langage de programmation C.
  - Création d'un site web via HTML5 – CSS3.

### COMPETENCES EN INFORMATIQUE

<b>Systèmes d'exploitation</b>	Linux, Windows.														
<b>Langages de programmation</b>	C/C++, SQL, Visual Basic, Shell, C#.NET, BATCH, XML, JAVA														
<b>Notation</b>	UML.														
<b>Framework</b>	JSF, Hibernate, .NET Framework, Qt.														
<b>Technologies web</b>	HTML5-CSS3, JavaScript, ASP.NET, JSP, PHP.														
<b>Développement Mobile</b>	Android.														
<b>Système de gestion de base de données</b>	MySQL, MS ACCESS, SQLite.														
<b>Méthode de conception</b>	MERISE.														
<b>Multimédia</b>	Adobe Photoshop CC, Adobe Illustrator CC, 3DSMAX 2012, FL Studio 10.														
<b>Outils bureautiques</b>	Microsoft PowerPoint, Microsoft Word, Microsoft Excel.														

### COMPETENCES LINGUISTIQUES

Aptitude Langue	A comprendre à l'audition					A lire					A écrire et à rédiger					A parler et à communiquer				
	TB	B	AB	P	NS	TB	B	AB	P	NS	TB	B	AB	P	NS	TB	B	AB	P	NS
Français	X					X					X					X				

Anglais			X				X						X				X		
---------	--	--	---	--	--	--	---	--	--	--	--	--	---	--	--	--	---	--	--

## ACTIVITE ET DIVERS

- Tuteur au sein du club Multimédia de l'Ecole Nationale d'Informatique.
- Multimédia.
- Jeux vidéo.
- Natation.

# REMERCIEMENT

Ce mémoire n'aurait pas pu être achevé sans l'aide de certaines personnes ayant bonne volonté. Sur ce, j'adresse mes sincères remerciements à toutes ces personnes liées à la réalisation de ce projet. En particulier :

- Monsieur RAMAMONJISOA Bertin Olivier, Directeur de l'Ecole Nationale d'Informatique qui m'a donné l'opportunité de partir en stage ;
- Monsieur PASCAL Vila, Directeur de la société NETAPSYS Madagascar qui m'a permis d'effectuer ce stage au sien de sa société ;
- Monsieur ANDRIANDRANTONAVALONA Rado, Directeur du pôle JAVA au sein de NETAPSYS Madagascar et aussi mon encadreur professionnel durant mon stage pour ses conseils lors de la réalisation du projet ;
- Monsieur RATIARSON Venot, responsable de Parcours de l'Ecole Nationale d'Informatique et aussi mon encadreur pédagogique pour son appuie lors de l'élaboration de ce présent mémoire.
- Monsieur MAHATODY Thomas responsable de Mention de l'Ecole Nationale d'Informatique, étant notre encadreur, pour son étroite collaboration ;
- Les membres du jury pour leur assistance à cette soutenance ;
- Tous les Enseignants de l'Ecole Nationale d'Informatique pour leurs précieuses assistances pédagogiques et les travaux pratiques ;
- Tous les personnels de Netapsys Madagascar, qui ont acceptés de travailler avec moi, et grâce à qui, j'ai pu collecter les renseignements essentiels pour la réalisation du rapport ;
- Finalement, je tiens à remercier mes parents, les membres de ma famille et mes amis, de m'avoir soutenu que ce soit moralement, matériellement et surtout financièrement.

# LISTE DES FIGURES

Figure 1 : Organigramme de l'ENI.....	6
Figure 2 : Chiffres clés de la croissance des groupes Netapsys et Sodifrance. ....	18
Figure 3 : Organigramme de la société Netapsys .....	21
Figure 4 : Représentation d'un ORM entre les modèles et la base.....	37
Figure 5 : Représentation d'un acteur.....	44
Figure 6 : Représentation d'un cas d'utilisation. ....	44
Figure 7 : exemple d'un cas d'utilisation.....	46
Figure 8 : Diagramme des cas d'utilisations.....	47
Figure 9 : Diagramme des séquences systèmes d'authentification.....	49
Figure 10 : Diagramme de séquence système de prise de commande. ....	50
Figure 11 : Diagramme de séquence système de création d'un point de vente. ....	50
Figure 12 : Message asynchrone.....	52
Figure 13 : messages synchrone .....	53
Figure 14 : Diagramme de séquence d'authentification. ....	54
Figure 15 : Diagramme de séquence de prise de commande.....	55
Figure 16 : Diagramme de séquence de prise de commande.....	56
Figure 17 : Structure d'une classe.....	58
Figure 18 : Association entre deux classes .....	58
Figure 19 : Exemple d'une relation d'agrégation .....	59
Figure 20 : Exemple de composition .....	60
Figure 21 : Exemple d'une généralisation .....	61
Figure 22 : Diagramme de classe de prise de commande.....	61
Figure 23 : Diagramme de classe de création d'un point de vente. ....	62
Figure 24 : Diagramme de classe de conception globale.....	63
Figure 25 : Représentation d'un nœud et d'une instance de nœud.....	63
Figure 26 : Diagramme de déploiement de l'application.....	64
Figure 27 : Téléchargement de JDK. ....	67
Figure 28 : Ecran de démarrage d'Android Studio.....	67
Figure 29 : Téléchargement d'Android Studio. ....	68
Figure 30 : Ecran de développement avec Android Studio. ....	69
Figure 31 : Ecran de gestion du SDK avec Android Studio. ....	69

Figure 32 : Architecture d'Android .....	70
Figure 33 : Schéma de compilation et d'exécution d'Android .....	70
Figure 34 : Ecran d'authentification de l'application .....	81
Figure 35 : Ecran d'accueil de l'application .....	82
Figure 36 : Ecran de routing de l'application .....	83
Figure 37 : Ecran de synthèse des commandes.....	84
Figure 38 : Ecran de commande .....	85
Figure 39 : Ecran de prise de command.....	86
Figure 40 : Ecran de bon de commandes .....	87
Figure 41 : Ecran de modification de la commande de produit .....	88
Figure 42 : Ecran de création d'un point de vente .....	89
Figure 43 : Ecran d'annulation d'un point de vente. ....	90



# LISTE DES TABLEAUX

Tableau 1 Organisation du système de formation pédagogique de l'Ecole.....	8
Tableau 2 : Architecture des études correspondant au système LMD.....	9
Tableau 3 : Liste des formations existantes à l'ENI .....	10
Tableau 4 : Débouchés professionnels éventuels des diplômés.....	16
Tableau 5 : Exemple des clients de la société.....	24
Tableau 6 : La liste des ordinateurs courant au sein du service informatique du client .....	30
Tableau 7 : Comparaison entre Android Studio et Eclipse ADT.....	38
Tableau 8 : Dictionnaire des données .....	41
Tableau 9 : Priorisation des cas d'utilisations.....	48
Tableau 10 : Liste des multiplicités .....	59

# LISTE DES ABREVIATIONS

ACID : Atomicité Cohérence, Isolation et Durabilité

ADT: Android Development Tools

AVD: Android Virtual Device

BNGRC : Bureau National de Gestion des Risques et des Catastrophes

BTS : Brevet de Technicien Supérieur

CARI : Colloque Africain sur la Recherche en Informatique.

CCNA : Cisco Networking Academy.

CITEF : Conférence Internationale des Ecoles de formation d'ingénieurs et Techniciens d'Expression Française.

CMS: Content Management System

CNH : Commission Nationale d'Habilitation

COFAV : Corridor forestier de Fandriana jusqu'à Vondrozo.

D.E.A : Diplôme d'Etude Approfondi.

DTS : Diplôme de Technicien Supérieur

DUT : Diplôme Universitaire de Technicien

ENI : Ecole Nationale d'Informatique.

ESPA : Ecole Supérieure Polytechnique d'Antananarivo.

FID : Fond d'Intervention pour le Développement

FPPSM : Forêts, Parcs et Pauvreté dans le Sud de Madagascar.

IDE: Integrated Development Environment.

INPG : Institut National Polytechnique de Grenoble.

IRD : Institut de Recherche pour le Développement.

IREMIA : Institut de REcherche en Mathématiques et Informatique Appliquées.

JDK : Java Development Kit

JEE : Java Enterprise Edition

JIRAMA: JIro sy RAno MAlagasy.

JRE: Java Runtime Environment

JSON: JavaScript Object Notation

LMD: License Master Doctorat

MVC: Model View Controller

ORM: Object Relational Mapping

PDV : Point de vente

PRESUP : Programme de Renforcement de l'Enseignement Supérieur.

RAM: Random Access Memory.

REST : REpresentational State Transfer

RH: Ressources Humaines

SDK : Software Development Kit

SGBD : Système de Gestion de Base de Données.

SQL : Structured Query Language.

SSII : Société de Services en Ingénierie Informatique

TIC : Technologie de l'Information et de la Communication

TMA: Tierce Maintenance Applicative

UML: Unified Modeling Language

UPST : Université Paul Sabatier de Toulouse.

URI : Uniform Resource Identifier

WWF: World Wildlife Fund

XML: eXtensible Markup Language

# SOMMAIRE

<b>CURRICULUM VITAE.....</b>	<b>I</b>
<b>REMERCIEMENT.....</b>	<b>IV</b>
<b>LISTE DES FIGURES .....</b>	<b>V</b>
<b>LISTE DES TABLEAUX.....</b>	<b>VII</b>
<b>LISTE DES ABREVIATIONS.....</b>	<b>VIII</b>
<b>SOMMAIRE.....</b>	<b>XI</b>
<b>INTRODUCTION GENERALE .....</b>	<b>1</b>
<b>PARTIE I : PRESENTATION .....</b>	<b>2</b>
<i>Chapitre 1 :           Présentation de l'école nationale d'informatique.....</i>	<i>3</i>
1.1.           Informations d'ordre général .....	3
1.2.           Missions et historique .....	3
1.3.           Organigramme institutionnel de l'ENI .....	5
1.4.           Domaines de spécialisation.....	7
1.5.           Architecture des formations pédagogiques .....	8
1.6.           RELATIONS DE L'ENI AVEC LES ENTREPRISES ET LES ORGANISMES.....	11
1.7.           PARTENARIAT AU NIVEAU INTERNATIONAL .....	12
1.8.           DEBOUCHES PROFESSIONNELS DES DIPLÔMES .....	14
1.9.           RESSOURCES HUMAINES .....	16
<i>Chapitre 2 :           Présentation de NETAPSYS .....</i>	<i>18</i>
2.1.           Présentation du Groupe Netapsys .....	18
2.2.           Netapsys Madagascar.....	19
2.3.           Organigramme .....	21
2.4.           Les valeurs du Groupe .....	21
2.5.           Technologies et outils maîtrisés .....	22
2.6.           Industrialisation des savoir-faire.....	22
2.7.           Partage et diffusion des connaissances .....	23
2.8.           Outils de pilotage projets .....	23
2.9.           Références Clients.....	24

<i>Chapitre 3 :</i>	<i>Description du projet.....</i>	<i>27</i>
3.1.	Formulation.....	27
3.2.	Objectif et besoin de l'utilisateur.....	27
3.3.	Moyens nécessaires pour la réalisation du projet.....	28
3.4.	Résultats attendus.....	28
<b>PARTIE II : ANALYSE ET CONCEPTION.....</b>		<b>29</b>
<i>Chapitre 4 :</i>	<i>Analyse préalable .....</i>	<i>30</i>
4.1.	Analyse de l'existant.....	30
4.2.	Critique de l'existant.....	31
4.3.	Conception avant-projet.....	31
<i>Chapitre 5 :</i>	<i>Analyse conceptuelle .....</i>	<i>41</i>
5.1.	Spécification des besoins .....	41
5.2.	Modélisation du domaine.....	48
<i>Chapitre 6 :</i>	<i>Conception détaillée.....</i>	<i>51</i>
6.1.	Architecture du système MVC.....	51
6.2.	Diagramme de séquence de conception pour chaque cas d'utilisation	52
6.3.	Diagramme de classe de conception pour chaque cas d'utilisation .....	57
6.4.	Diagramme de classe de conception globale .....	63
6.5.	Diagramme de déploiement .....	63
<b>PARTIE III : REALISATION .....</b>		<b>65</b>
<i>Chapitre 7 :</i>	<i>Mise en place de l'environnement de développement .....</i>	<i>66</i>
7.1.	Installation des outils .....	66
7.2.	Architecture et système d'exécution d'Android .....	70
<i>Chapitre 8 :</i>	<i>Développement de l'application.....</i>	<i>71</i>
8.1.	Création de la base de données .....	71
8.2.	Codage de l'application .....	75
8.3.	Présentation de l'application.....	80
<b>CONCLUSION .....</b>		<b>X</b>
<b>REFERENCE BIBLIOGRAPHIQUES.....</b>		<b>XI</b>
<b>REFERENCE WEBOGRAPHIQUES.....</b>		<b>XI</b>
<b>GLOSSAIRE .....</b>		<b>XII</b>
<b>TABLE DES MATIERES.....</b>		<b>XIV</b>

<b>RESUME .....</b>	<b>XVIII</b>
<b>ABSTRACT.....</b>	<b>XVIII</b>

# INTRODUCTION GENERALE

Actuellement, la technologie devient une nécessité dans presque tous les domaines du monde professionnel. De ce fait, l'informatique qui est le traitement automatique des informations joue un rôle capital dans l'évolution du monde moderne.

NETAPSYS Madagascar est une société de service en ingénierie informatique spécialisée dans la conception, le développement et la maintenance de projets digitaux, applicatifs et mobiles. Elle est une référence dans toute l'île grâce aux qualités des produits et à sa vitesse de production.

Afin de répondre au besoin de leurs clients, NETAPSYS Madagascar m'a confié comme projet de stage « Conception et réalisation d'une application mobile de prise de commande ».

L'objectif de ce projet est de concevoir et réaliser une application mobile visant à aider, l'une des clients de NETAPSYS à faciliter le processus de vente de leur produit au sein des épiceries.

Pour la réalisation de ce projet, on a dû faire appel à un langage de modélisation, un langage de programmation, une méthode de conception, un Système de Gestion de Base de Donnée, un IDE de développement et un Système d'exploitation.

Ainsi pour présenter le rapport de ce projet, on va entamer par la présentation de l'Ecole Nationale d'Informatique (ENI) suivie de celle de NETAPSYS Madagascar, ensuite la présentation du projet, l'analyse et conception et en dernier lieu la réalisation du projet.



# **PARTIE I : PRESENTATION**

# **Chapitre 1 : Présentation de l'école nationale d'informatique**

## **1.1. Informations d'ordre général**

L'Ecole Nationale d'Informatique, en abrégé ENI, est un établissement d'enseignement supérieur rattaché académiquement et administrativement à l'Université de Fianarantsoa.

Le siège de l'Ecole se trouve à Tanambao- Antaninarenina à Fianarantsoa.

L'adresse pour la prise de contact avec l'Ecole est la suivante :

Ecole Nationale d'Informatique (ENI) Tanambao, Fianarantsoa. Le numéro de sa boîte postale est 1487 avec le code postal 301. Téléphone : 020 75 508 01. Son adresse électronique est la suivante : [eni@univ-fianar.mg](mailto:eni@univ-fianar.mg). Site Web : [www.eni@univ-fianar.mg/eni](http://www.eni@univ-fianar.mg/eni)

## **1.2. Missions et historique**

L'ENI se positionne sur l'échiquier socio-éducatif malgache comme étant le plus puissant secteur de diffusion et de vulgarisation des connaissances et des technologies informatiques.

Cette Ecole Supérieure peut être considérée aujourd'hui comme la vitrine et la pépinière des élites informaticiennes du pays.

L'Ecole s'est constituée de façon progressive au sein du Centre Universitaire Régional (CUR) de Fianarantsoa.

De façon formelle, l'ENI était constituée et créée au sein du (CUR) par le décret N° 83-185 du 24 Mai 1983, comme étant le seul établissement Universitaire Professionnalisé au niveau national, destiné à former des techniciens et des Ingénieurs de haut niveau, aptes à répondre aux besoins et exigences d'Informatisation des entreprises, des sociétés et des organes implantés à Madagascar.

L'ENI a pour conséquent pour mission de former des spécialistes informaticiens compétents et opérationnels de différents niveaux notamment :

- En fournissant à des étudiants des connaissances de base en informatique ;
- En leur transmettant le savoir-faire requis, à travers la professionnalisation des formations dispensées et en essayant une meilleure adéquation des formations par rapport aux besoins évolutifs des sociétés et des entreprises.
- En initiant les étudiants aux activités de recherche dans les différents domaines des Technologies de l'information et de la communication (TIC).

L'implantation de cette Ecole Supérieure de technologie de pointe dans un pays en développement et dans une Province (ou Faritany) à tissu économique et industriel faiblement développé ne l'a pourtant pas défavorisée, ni empêchée de former des spécialistes informaticiens de bon niveau, qui sont recherchés par les entreprises, les sociétés et les organismes publics et privés sur le marché de l'emploi.

La filière de formation d'Analystes Programmeurs a été mise en place à l'Ecole en 1983, et a été gelée par la suite en 1996, tandis que la filière de formation d'ingénieurs a été ouverte à l'Ecole en 1986.

Dans le cadre du Programme de renforcement en l'Enseignement Supérieur (PRESUP), la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes des informatiques a été mise en place en 1986 grâce à l'appui matériel et financier de la Mission Française de coopération auprès de l'Ambassade de France à Madagascar.

Une formation pour l'obtention de la certification CCNA et / ou NETWORK +. Appelée « CISCO Networking Academy » a été créée à l'Ecole en 2002-2003 grâce au partenariat avec CISCO SYSTEM et l'Ecole Supérieure Polytechnique d'Antananarivo (ESPA). Cependant, cette formation n'avait pas duré longtemps.

Une formation de troisième cycle a été ouverte à l'Ecole a été ouverte à l'Ecole depuis l'année 2003 – 2004 grâce à la coopération académique et scientifique entre l'Université de Fianarantsoa pour le compte de l'ENI et l'Université Paul Sabatier de Toulouse (UPST).

Cette filière avait pour objectif de former certains étudiants à la recherche dans les différents domaines de l'Informatique, et notamment pour préparer la relève des Enseignants-Chercheurs qui étaient en poste.

Pendant l'année 2007-2008, la formation en vue de l'obtention du diplôme de Licence Professionnelle en Informatique a été mise en place à l'ENI avec les deux options suivantes de formation :

- Génie Logiciel et base de Données.
- Administration des Système et réseaux.

La mise en place à l'Ecole de ces deux options de formation devait répondre au besoin de basculement vers le système Licence – Master – Doctorat (LMD).

Mais la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes Informatiques a été gelée en 2009.

En vue de surmonter les difficultés de limitation de l'effectif des étudiants accueillis à l'Ecole, notamment à cause du manque d'infrastructures, un système de « Formation Hybride » a été mise en place à partir de l'année 2010. Il s'agit en effet d'un système de formation semi-présentielle et à distance avec l'utilisation de la visioconférence pour la formation à distance.

Le système de formation hybride a été ainsi créé à Fianarantsoa ainsi qu'Université de Toliara.

### **1.3. Organigramme institutionnel de l'ENI**

Cet organigramme de l'Ecole est inspiré des dispositions du décret N° 83-185 du 23 Mai 1983.

L'ENI est administrée par un conseil d'Ecole, et dirigée par un directeur nommé par un décret adopté en conseil des Ministres.

Le Collège des enseignants regroupant tous les enseignants-chercheurs de l'Ecole est chargé de résoudre les problèmes liés à l'organisation pédagogique des enseignements ainsi que à l'élaboration des emplois du temps.

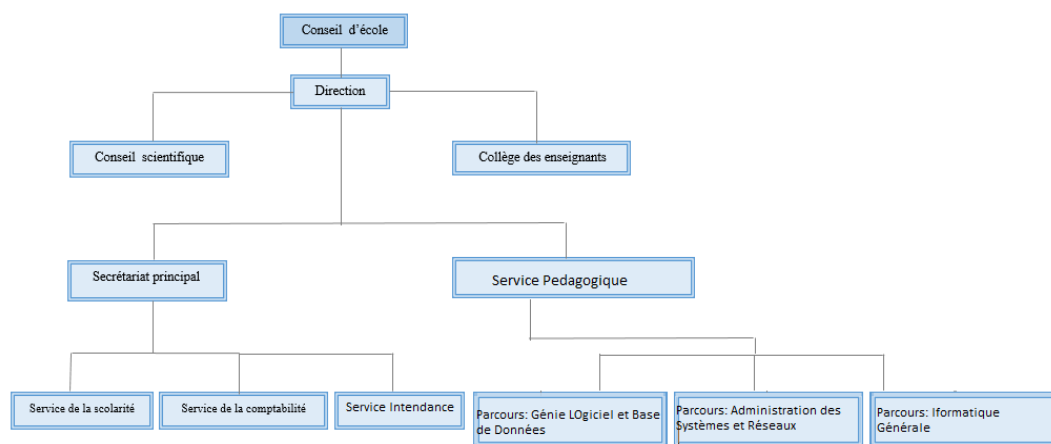
Le Conseil Scientifique propose les orientations pédagogiques et scientifiques de l'établissement, en tenant compte notamment de l'évolution du marché de travail et de l'adéquation des formations dispensées par rapport aux besoins des entreprises.

Trois départements de formation caractérisent l'organigramme :

- Le département de formation théorique à l'intérieur de l'Ecole ;
- Le département de formation pratique pour la coordination et la supervision des stages en entreprise et des voyages d'études ;

Le département de formation doctorale pour l'organisation de la formation de 3ème cycle.

La figure 1 présente l'organigramme actuel de l'Ecole.



**Figure 1 : Organigramme de l'ENI**

Sur cet organigramme, l'Ecole placée sous la tutelle académique et administrative de l'Université de Fianarantsoa, et dirigée par un Directeur élu par les Enseignants – Chercheurs permanents de l'Etablissement et nommé par un décret pris en Conseil des ministres pour un mandat de 3 ans.

Le Conseil de l'Ecole est l'organe délibérant de l'Ecole.

Le Collège des Enseignants propose et coordonne les programmes d'activités pédagogiques.

Le Conseil scientifique coordonne les programmes de recherche à mettre en œuvre à l'Ecole.

Le Secrétariat principal coordonne les activités des services administratifs (Scolarité, Comptabilité, et Intendance).

Conformément aux textes en vigueur régissant les Etablissements malgaches d'Enseignement Supérieur, qui sont barrés sur le système LMD, les Départements de Formation pédagogique ont été ainsi remplacés par des Mentions et des parcours. Et les chefs des Départements ont été ainsi remplacés par des responsables des mentions et les responsables des parcours.

Un administrateur des Réseaux et Systèmes gère le système d'information de l'Ecole et celui de l'Université.

#### **1.4. Domaines de spécialisation**

Les activités de formation et de recherche organisées à l'ENI portent sur les domaines suivants :

- Génie logiciel et Base de Données ;
- Administration des Systèmes et Réseaux ;
- Informatique Générale
- Modélisation informatique et mathématique des Systèmes complexes.

D'une manière plus générale, les programmes des formations sont basés sur l'informatique de gestion et sur l'informatique des Systèmes et Réseaux. Et les modules de formation intègrent aussi bien des éléments d'Informatique fondamentale que des éléments d'Informatique appliquée.

Le tableau 1 décrit l'organisation du système de formation pédagogique de l'Ecole.

**Tableau 1 Organisation du système de formation pédagogique de l'Ecole**

<b>Formation théorique</b>	<b>Formation pratique</b>
<ul style="list-style-type: none"><li>- Enseignement théorique</li><li>- Travaux dirigés</li><li>- Travaux pratiques</li></ul>	<ul style="list-style-type: none"><li>- Etude de cas</li><li>- Travaux de réalisation</li><li>- Projets / Projets tutorés</li><li>- Voyage d'études</li><li>- Stages</li></ul>

## **1.5. Architecture des formations pédagogiques**

Le recrutement des étudiants à l'ENI se fait uniquement par voie de concours d'envergure nationale en première année.

Les offres de formation organisées à l'Ecole ont été validées par la Commission Nationale d'Habilitation (CNH) auprès du Ministère de l'Enseignement Supérieur et de la Recherche Scientifique selon les dispositions de l'Arrêté N°31.174/2012-MENS en date du 05 Décembre 2012.

Au sein de l'ENI, il existe une seule mention (INFORMATIQUE) et trois parcours :

- Génie logiciel et Base de Données ;
- Administration des Systèmes et Réseaux ;
- Informatique Générale

L'architecture des études à trois niveaux conformément au système Licence- Master- Doctoral (LMD) permet les comparaisons et les équivalences académiques des diplômes au niveau international.

- L = Licence (Bac + 3) = L1, L2, L3 = 6 semestres S1 à S6
- M = Master (Bac + 5) = M1, M2 = 4 semestres S7 à S10

Le diplôme de licence est obtenu en 3 années des études après Baccalauréat. Et le diplôme de Master est obtenu en 2 ans après obtenu du diplôme de LICENCE.

Le MASTER PROFESSIONNEL est un diplôme destiné à la recherche emploi au terme des études.

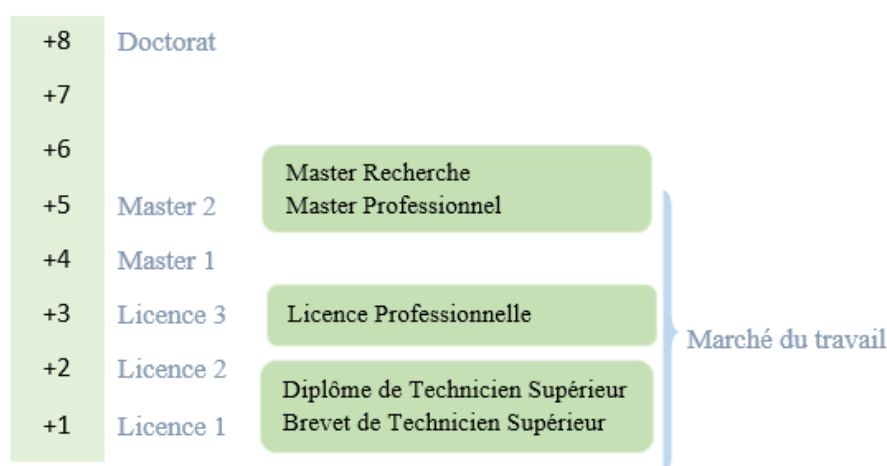
Le MASTER RECHERCHE est un diplôme qui remplace l'ancien Diplôme d'Etudes Approfondies (DEA), et qui permet de s'inscrire directement dans une Ecole Doctorale au terme des études.

➤ D = Doctorat (Bac +8)

Le Doctorat est un diplôme qu'on peut obtenir en 3 ans après l'obtention du diplôme de MASTER RECHERCHE.

Le tableau 2 présente l'architecture des études correspondant au système LMD.

**Tableau 2 : Architecture des études correspondant au système LMD**



DTS : Diplôme de Technicien Supérieur

BTS : Brevet de Technicien Supérieur

DUT : Diplôme Universitaire de Technicien

La licence peut avoir une vocation générale ou professionnelle.

Le master peut avoir une vocation professionnelle ou de recherche.

Le tableau 3 présente la liste des formations existantes à l'ENI



**Tableau 3 : Liste des formations existantes à l'ENI**

	FORMATION EN	
	LICENCE PROFESSIONNELLE ET HYBRIDE	MASTER
Condition d'admission	Par voie de concours Formation Professionnelle : 100 candidats Formation hybride : 150 candidats	
Condition d'accès	Bac de série C, D ou Technique	Etre titulaire de licence professionnelle
Durée de formation	3 années	2 années
Diplôme à délivrer	Diplôme de Licence Professionnelle en Informatique	Diplôme de Master Professionnel Diplôme de Master Recherche

L'accès en première année de MASTER se fait automatiquement pour les étudiants de l'Ecole qui ont obtenu le diplôme de Licence Professionnelle.

Le Master Recherche permet à son titulaire de poursuivre directement des études en doctorat et de s'inscrire directement dans une Ecole Doctorale.

Les Ecoles Doctorales jouissent d'une autonomie de gestion par rapport aux Etablissements de formation universitaire.

Il convient de signaler que par arrêté ministériel N° 21.626/2012 – MESupRES publié le 9 Août 2012 par la Commission National d'habilitation (CNH), l'Ecole Doctorale « Modélisation – Informatique » a été habilitée pour l'Université de Fianarantsoa. »

Depuis l'année universitaire 2010-2011, l'ENI s'est mise à organiser des formations hybrides en informatique dans les différentes régions (Fianarantsoa, Toliara) en raison de

l'insuffisance de la capacité d'accueil des infrastructures logistiques. En effet, le système de formation hybride semi - présentielle utilise la visioconférence pour la formation à distance.

Bien qu'il n'existe pas encore au niveau international de reconnaissance écrite et formelle des diplômes délivrés par l'ENI, les étudiants diplômés de l'Ecole sont plutôt bien accueillis dans les instituts universitaires étrangères (CANADA, Suisse, France...)

## **1.6. RELATIONS DE L'ENI AVEC LES ENTREPRISES ET LES ORGANISMES**

Les stages effectués chaque année par les étudiants mettent l'Ecole en rapport permanent avec plus de 300 entreprises et organismes publics, semi-publics et privés, nationaux et internationaux.

L'Ecole dispose ainsi d'un réseau d'entreprises, de sociétés et d'organismes publics et privés qui sont des partenaires par l'accueil en stage de ses étudiants, et éventuellement pour le recrutement après l'obtention des diplômes par ces derniers.

Les compétences que l'Ecole cherche à développer chez ses étudiants sont l'adaptabilité, le sens de la responsabilité, du travail en équipe, le goût de l'expérimentation et l'innovation.

En effet, la vocation de l'ENI est de former des techniciens supérieurs de niveau LICENCE et des ingénieurs de type généraliste de niveau MASTER avec des qualités scientifiques, techniques et humaines reconnues, capables d'évoluer professionnellement dans des secteurs d'activité variés intégrant l'informatique.

Les stages en milieu professionnel permettent de favoriser une meilleure adéquation entre les formations à l'Ecole et les besoins évolutifs du marché de l'emploi.

Les principaux débouchés professionnels des diplômés de l'Ecole concernent les domaines suivants :

- L'informatique de gestion d'entreprise
- Les technologies de l'information et de la communication (TIC)
- La sécurité informatique des réseaux
- L'administration des réseaux et des systèmes

- Les services bancaires et financiers, notamment le Mobile Banking
- Les télécommunications et la téléphonie mobile
- Les Big Data
- Le commerce, la vente et l'achat, le Marketing
- L'ingénierie informatique appliquée
- L'écologie et le développement durable

Parmi les sociétés, entreprises et organismes partenaires de l'Ecole, on peut citer : ACCENTURE Mauritius, Air Madagascar, Ambre Associates, Airtel, Agence Universitaire de la Francophonie ( AUF) , B2B, Banque Centrale, BFG-SG, BIANCO, BLUELINE, Bureau national de gestion des Risques et des catastrophes (BNGRC), CEDII-Fianarantsoa, Data Consulting, Central Test, Centre National Antiacridien, CNRE, CHU, CNRIT, COLAS, Direction Générale des Douanes, DLC, DTS/Moov, FID, FTM, GNOSYS, IBONIA, INGENOSYA, INSTAT, IOGA, JIRAMA, JOUVE, MADADEV, MAEP, MEF, MEN, MESupRES, MFB, MIC, MNINTER, Min des postes/Télécommunications et du Développement Numérique, NETAPSYS MADAGASCAR, Ny Havana, Madagascar National Parks, OMNITEC, ORANGE, OTME, PRACCESS, QMM Fort-Dauphin, SMMC, SNEDADRS Antsirabe, Sénat, Société d'Exploitation du Port de Toamasina (SEPT), SOFTWELL, Strategy Consulting, TELMA, VIVETEC, Société LAZAN'I BETSILEO, WWF ...

L'organisation de stage en entreprise continue non seulement à renforcer la professionnalisation des formations dispensées, mais elle continue surtout à accroître de façon exceptionnelle les opportunités d'embauche pour les diplômés de l'Ecole.

## **1.7. PARTENARIAT AU NIVEAU INTERNATIONAL**

Entre 1966 et 1999, l'ENI avait bénéficié de l'assistance technique et financière de la Mission Française de Coopération et d'action culturelle dans le cadre du Programme de Renforcement de l'Enseignement Supérieur (PRESUP) consacré à l'Ecole a notamment porté sur :

- Une dotation en logiciels, micro-ordinateurs, équipements de laboratoire de maintenance et de matériels didactiques

- La réactualisation des programmes de formation assortie du renouvellement du fonds de la bibliothèque
- L'appui à la formation des formateurs
- L'affectation à l'Ecole d'Assistants techniques français

De 2000 à 2004, l'ENI avait fait partie des membres du bureau de la Conférence Internationale des Ecoles de formation d'Ingénieurs et Technicien d'Expression Française (CITEF).

Les Enseignants-Chercheurs de l'Ecole participent régulièrement aux activités organisées dans le cadre du Colloque Africain sur la Recherche en Informatique (CARI).

L'ENI avait également signé un accord de coopération interuniversitaire avec l'Institut de Recherche en Mathématiques et Informatique Appliquées (IREMIA) de l'Université de la Réunion, l'Université de Rennes 1, l'INSA de Rennes, l'Institut National Polytechnique de Grenoble (INPG).

A partir du mois de Juillet 2001, l'ENI avait abrité le Centre de Réseau Opérationnel (Network Operating Center) du point d'accès à Internet de l'Ecole ainsi que de l'Université de Fianarantsoa. Grâce à ce projet américain qui a été financé par l'USAID Madagascar, l'ENI de l'Université de Fianarantsoa avait été dotées d'une ligne spécialisée d'accès permanent au réseau Internet.

L'ENI avait de même noué des relations de coopération avec l'Institut de Recherche pour le Développement (IRD).

L'objet du projet de coopération avait porté sur la modélisation environnementale du Corridor forestier de Fandriana jusqu'à Vondrozo (COFAV). Dans ce cadre, un atelier scientifique international avait été organisé à l'ENI en Septembre 2008. Cet atelier scientifique avait eu pour thème de modélisation des paysages.

Et dans le cadre du programme scientifique PARRUR, l'IRD avait financé depuis 2010 le projet intitulé « Forêts, Parcs et Pauvreté dans le Sud de Madagascar » (FPPSM). Des étudiants en DEA et des Doctorants issus de l'ENI avaient participé à ce Programme.

Par ailleurs, depuis toujours la même année 2010, l'ENI de Fianarantsoa avait été sélectionnée pour faire partie des organismes partenaires de l'Université de Savoie dans le cadre du projet TICEVAL relatif à la certification des compétences en TIC ;

Le projet TICEVAL avait été financé par le Fonds Francophone des Inforoutes pour la période allant de 2010 à 2012, et il avait eu pour objectif de généraliser la certification des compétences en Informatique et Internet du type C2i2e et C2imi.

Dans le cadre du projet TICEVAL, une convention de coopération avec l'Université de Savoie avait été signée par les deux parties concernées. La mise en œuvre de la Convention de Coopération avait permis d'envoyer des étudiants de l'ENI à Chambéry pour poursuivre des études supérieures en Informatique.

Enfin et non des moindres, l'ENI avait signé en Septembre 2009 un protocole de collaboration scientifique avec l'ESIROI – STIM de l'Université de la Réunion.

Comme l'ENI constitue une pépinière incubatrice de technologie de pointe, d'emplois et d'entreprises, elle peut très bien servir d'instrument efficace pour renforcer la croissance économique du pays, et pour lutter contre la Pauvreté.

De même que le statut de l'Ecole devrait permettre de renforcer la position concurrentielle de la Grande Ile sur l'orbite de la modélisation grâce au développement des nouvelles technologies.

## **1.8. DEBOUCHES PROFESSIONNELS DES DIPLOMÉS**

Le chômage des jeunes diplômés universitaires fait partie des maux qui gangrènent Madagascar. L'environnement socio-politique du pays depuis 2008 jusqu'à ce jour a fait que le chômage des diplômés est devenu massif par rapport aux établissements de formation supérieure existants.

Cependant, les formations proposées par l'Ecole permettent aux diplômés d'être immédiatement opérationnels sur le marché du travail avec la connaissance d'un métier complet lié à l'informatique aux TIC.

L'Ecole apporte à ses étudiants un savoir-faire et un savoir-être qui les accompagnent tout au long de leur vie professionnelle. Elle a une vocation professionnalisant.

Les diplômés en LICENCE et en MASTER issus de l'ENI peuvent faire carrière dans différents secteurs.

L'Ecole bénéficie aujourd'hui de 34 années d'expériences pédagogiques et de reconnaissance auprès des sociétés, des entreprises et des organismes. C'est une Ecole Supérieure de référence en matière informatique.

Par conséquent, en raison de fait que l'équipe pédagogique de l'Ecole est expérimentée, les enseignants-chercheurs et les autres formateurs de l'Ecole sont dotés d'une grande expérience dans l'enseignement et dans le milieu professionnel.

L'Ecole est fière de collaborer de façon régulière avec un nombre croissant d'entreprises, de sociétés et d'organismes publics et privés à travers les stages des étudiants. Les formations dispensées à l'Ecole sont ainsi orientées vers le besoin et les attentes des entreprises et des sociétés.

L'Ecole fournit à ses étudiants de niveau LICENCE et MASTER des compétences professionnelles et métiers indispensables pour les intégrer sur le marché du travail.

L'Ecole s'efforce de proposer à ses étudiants une double compétence à la fois technologique et managériale combinant l'informatique de gestion ainsi que l'administration des réseaux et systèmes.

D'une manière générale, les diplômés de l'ENI n'éprouvent pas de difficultés particulières à être recrutés au terme de leurs études. Cependant, l'ENI recommande à ses diplômés de promouvoir l'entrepreneuriat en TIC et de créer des cybercafés, des SSII ou des bureaux d'études.

Le tableau 4 montre les débouchés professionnels éventuels des diplômés

**Tableau 4 : Débouchés professionnels éventuels des diplômés**

LICENCE	<ul style="list-style-type: none"> <li>- Analyste</li> <li>- Programmeur</li> <li>- Administrateur de site web/de portail web</li> <li>- Assistant Informatique et internet</li> <li>- Chef de projet web ou multimédia</li> <li>- Développeur Informatique ou multimédia</li> <li>- Intégrateur web ou web designer</li> <li>- Hot liner/Hébergeur Internet</li> <li>- Agent de référencement</li> <li>- Technicien/Supérieur de help desk sur Informatique</li> <li>- Responsable de sécurité web</li> <li>- Administrateur de réseau</li> <li>- Administrateur de cybercafé</li> </ul>
MASTER	<ul style="list-style-type: none"> <li>- Administrateur de réseau et système</li> <li>- Architecture de système d'information</li> <li>- Développeur d'application /web /java/Python/ IOS /Android</li> <li>- Ingénieur réseau</li> <li>- Webmaster /web designer</li> <li>- Concepteur Réalisateur d'applications</li> <li>- Directeur du système de formation</li> <li>- Directeur de projet informatique</li> <li>- Chef de projet informatique</li> <li>- Responsable de sécurité informatique</li> <li>- Consultant fonctionnel ou freelance</li> <li>- Administrateur de cybercafé</li> </ul>

## 1.9. RESSOURCES HUMAINES

- Directeur de l'Ecole : Professeur RAMAMONJISOA Bertin Olivier
- Responsable de Mention : Docteur MAHATODY Thomas
- Responsable de Parcours « Génie Logiciel et Base de Données » : Docteur RATIARSON Venot
- Responsable de Parcours « Administration Systèmes et Réseaux » : Monsieur SIAKA

- Responsable de Parcours « Informatique Générale » : Docteur RAKOTOASIMBAHOAKA Cyprien Robert
- Nombre d'Enseignants permanents : 13 dont deux (02) Professeurs Titulaires, six (06) Maîtres de Conférences et cinq (05) Assistants d'Enseignement Supérieur et de Recherche
- Nombre d'Enseignants vacataires : 10
- Personnel Administratif : 23



## Chapitre 2 : Présentation de NETAPSYS

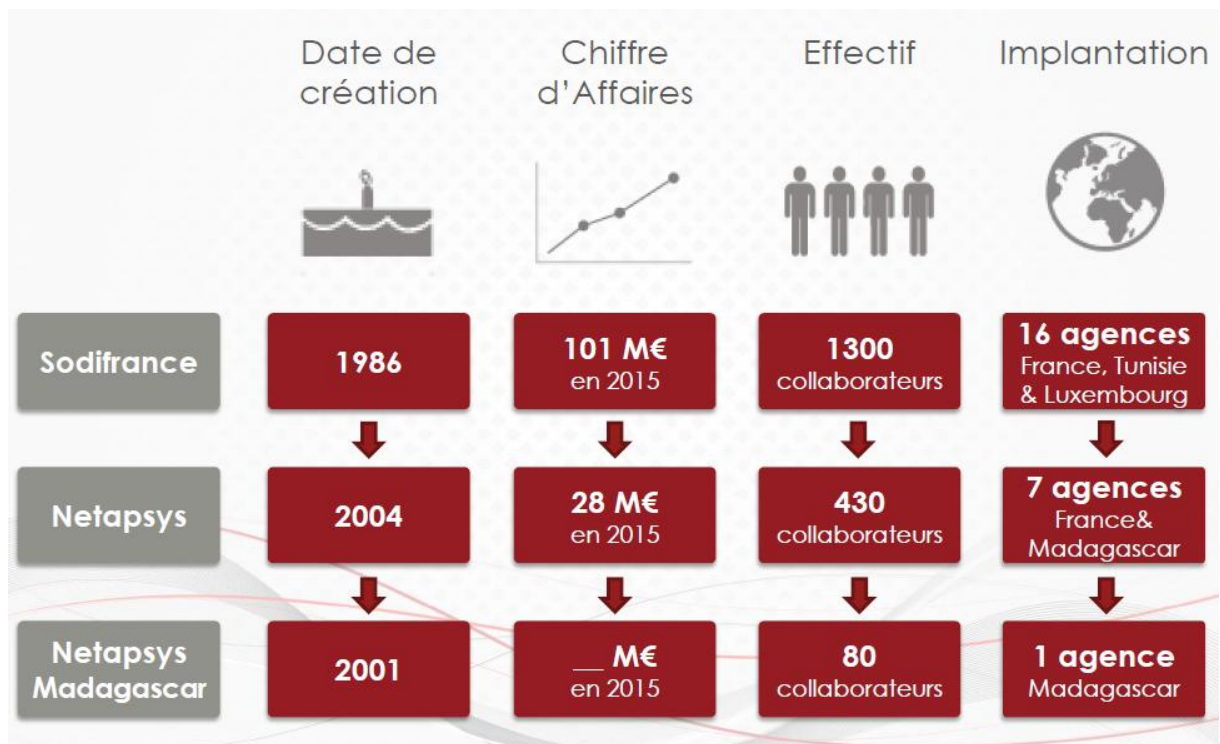
### 2.1. Présentation du Groupe Netapsys

Netapsys est une société d'ingénierie informatique spécialiste des nouvelles technologies, des solutions de Business Intelligence, de Gestion de contenu, Collaboratives, d'E-business, d'E-commerce et de Mobilité.

Netapsys conçoit, développe et maintient des applications informatiques et des systèmes d'information sur-mesure, basés sur des technologies de pointe.

Ses fondateurs, Yoann Hébert et Jérémy Rousselle, experts en technologies objet et en systèmes d'information, dirigent la société depuis sa création.

La figure 2 montre quelques chiffres de croissance



**Figure 2 : Chiffres clés de la croissance des groupes Netapsys et Sodifrance.**

Netapsys qui est une société spécialiste de la transformation digitale (applications et SI sur-mesure) a rejoint le Groupe Sodifrance en 2015, ce dernier étant spécialiste de la transformation et modernisation des systèmes mainframe ; créant ainsi la complémentarité et

la synergie des 2 marques. Netapsys est une Entreprise de Services en Numérique spécialisée dans :

- Les nouvelles technologies : .NET, Java JEE, LAMP, ... essentiellement issues du « monde internet »
- La conduite d'opérations en mode projet : engagement forfaitaire sur les délais, les livrables, les coûts, ...
- Le maintien en conditions opérationnelles : Tierce Maintenance Applicative(TMA), Infogérance et accompagnement technique.

## **2.2. Netapsys Madagascar**

Netapsys Madagascar, basée à Antananarivo, Porte B201 Immeuble ARO Ampefiloha, est spécialisée dans la conception, le développement et la maintenance de projets digitaux, applicatifs et mobiles en technologie Web Open Source LAMP et JAVA, en méthodologies agiles.

Dirigée par Pascal Vila et forte de 90 collaborateurs, Netapsys Madagascar occupe aujourd'hui une position de premier plan sur les métiers de l'ingénierie logicielle dans l'Océan Indien. Elle intervient pour des clients locaux, dans le domaine des télécommunications, de la finance, dans l'administration ou encore l'agroalimentaire mais également sur des projets français, en direct ou avec les agences françaises du groupe.

Elle est aujourd'hui constitué de 6 pôles à savoir : pôle PHP, pôle CMS, pôle Studio, Pôle java, pôle admin et le pôle Infrastructure ; basés sur :

- Formation

Netapsys a mis en place un programme de formation complet pour aider ses collaborateurs à s'adapter aux nouveaux défis technologiques. Les formations sont basées sur des workshops, MOOC, de la recherche, technical breakfast, ...

- Certification

Netapsys propose à ses ingénieurs de passer différents niveaux de certification. Ainsi ses ingénieurs peuvent évoluer et devenir des experts dans leur domaine.

- Autonomie et Responsabilités

Netapsys encourage chez ses collaborateurs les qualités d'autonomie, de prise d'initiatives afin d'augmenter l'agilité de l'entreprise.

➤ Carrière

Les collaborateurs ont des perspectives d'évolution géographique et professionnelle intéressantes, nombre d'entre eux ont pu évoluer avec le temps en passant de « Ingénieur d'Etudes et de Développement » à « Chef de Projet » et « Directeur de Pôle » par exemple.

➤ Innovation

Netapsys est à la pointe de la technologie. Ils se forment continuellement pour toujours rester au top, apprendre des nouvelles technologies pour satisfaire tous les besoins des clients.

Etre dynamiques : Devfest, DevoXx, Atlassian Tour, livres blancs, blog technique, séminaires...

➤ Multiculturel

Au sein de l'Agence Netapsys Madagascar, privilège d'une double culture franco-malgache. Un échange dans les deux sens de visions professionnelles au départ différentes pour créer une vision commune où chacun apprend de l'autre.

➤ Solidarité & Convivialité

Mettre un point d'honneur à travailler dans une bonne ambiance, à s'aider les uns les autres. À mettre en place des activités pour souder les équipes.

➤ Sport

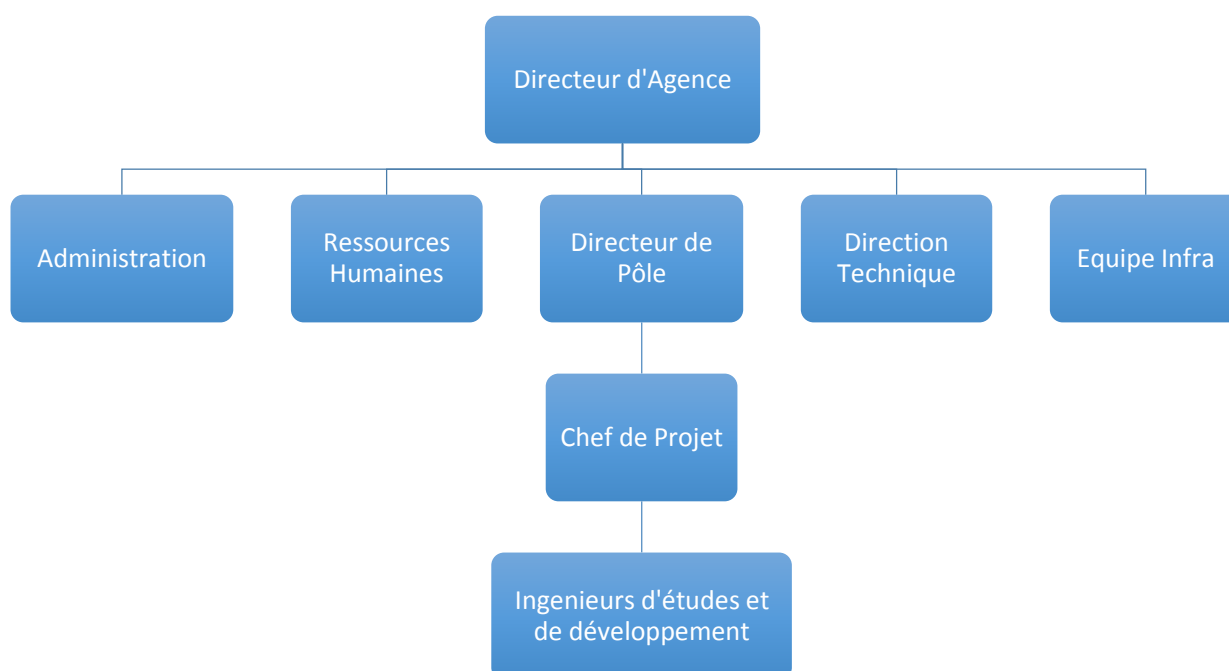
Netapsys participe à plusieurs tournois sportifs. Mais aussi, toujours dans cet esprit de convivialité, permet à ses collaborateurs de jouer au futsal tous les jeudis. Les mardis et vendredis c'est pétanque

➤ Bien vivre

Afin que les collaborateurs se sentent bien dans l'entreprise, ils ont droit bien évidemment à la CNAPS et l'OSTIE, mais aussi à une prime quotidienne de panier repas. Les horaires sont flexibles selon les préférences des collaborateurs.

## 2.3. Organigramme

La figure 3 présente l'organigramme de la société Netapsys.



**Figure 3 : Organigramme de la société Netapsys**

## 2.4. Les valeurs du Groupe

### ➤ Culture d'engagement

Bâtie sur deux principes fondamentaux, l'**expertise technologique** et la **maîtrise de la conduite d'opérations en mode projet**, Netapsys se démarque par sa capacité à s'engager auprès de ses clients.

- **Engagement à proposer la solution** (fonctionnelle, technique et organisationnelle) la plus adaptée au contexte du projet et à la stratégie du client.
- **Engagement clair sur les résultats** : exhaustivité des livrables ; qualité des travaux (en accord avec les normes du client, l'état de l'art du marché et la politique interne de Netapsys d'industrialisation des développements) et respect des délais.

### ➤ Démarche d'amélioration continue

Le partage des connaissances et le développement des compétences techniques sont au cœur de la culture Netapsys. L'ambition, portée par l'ensemble des équipes, est de devenir

une référence française sur les nouvelles technologies. Pour atteindre cet objectif, Netapsys s'inscrit dans une démarche d'amélioration continue et s'appuie sur :

- **Une montée en compétence continue de ses équipes** (formation continue, technical breakfast, politique de certification ...)
- **Un développement de son outillage logiciel** (intégration continue, mesure permanente de la qualité des développements, gestion de projet, documentation...)
- **Des méthodes de travail inspirées des méthodologies agiles**

➤ **Solidarité, convivialité, développement durable**

Netapsys recherche une performance qui n'est pas uniquement financière, mais aussi humaine. Cela se traduit par la mise en place de pratiques fondées sur des valeurs éthiques et une politique RH visant à mettre en adéquation les envies personnelles et les projets de la société par :

- La dynamique collective, qui se traduit par le partage des résultats (accord d'intéressement ou participation), la mise en place de standup meeting permettant de favoriser l'échange et la communication ainsi que l'animation de technical breakfasts, 2 à 3 fois par semaine, durant lesquels un collaborateur présente à l'ensemble de la société un sujet technique.
- L'attention particulière accordée à la convivialité (budgets dédiés à la vie des pôles, séminaires, soirées, sponsoring d'évènements sportifs type marathons relais ou ludiques). L'équipe féminine Netapsys a participé en 2012 à La parisienne, une course pour soutenir la lutte contre le cancer du sein.

## **2.5. Technologies et outils maîtrisés**

Pour réaliser des projets conformes aux attentes des clients, la société propose à ses clients les technologies et les outils de dernière génération.

## **2.6. Industrialisation des savoir-faire**

- Utilisation de Framework;
- Mutualisation des développements;
- Partage des méthodes et bonnes pratiques ;

- Normalisation des recettes;
- Documents génériques de spécifications;
- Méthodes: UML, Merise, AGILES.

## **2.7. Partage et diffusion des connaissances**

- Intranet collaboratif de partage des méthodes, process et documentations techniques
- Échange et partage de liens et sources d'information ;
- Recensement exhaustif des modules et librairies utilisés ;
- Formations continues : 100 heures de formation dispensées chaque année.

## **2.8. Outils de pilotage projets**

- TinyPM : une plateforme collaborative de gestion de projets qui permet de centralisé l'ensemble des informations relatives à un projet et permet un suivi en temps réel de l'avancement du projet.
- Une Plateforme de gestion des recettes projet : Netapsys a développé sous technologie Flex son propre outil de recettes, afin de coordonner au mieux cette phase cruciale avant livraison ;
- JIRA : un système de suivi de bugs, un système de gestion des incidents, et un système de gestion de projets, gestions des temps consacrés pour chaque sous tâche développé par Atlassian Software Systems ;
- Projector : un système de gestion de projet, permettant d'imputer le temps dans un projet
- Mantis Bug Tracker : plateforme de gestion d'événements Open Source.
- GIT : Netapsys utilise cet outil de gestionnaire des versions (ou gestionnaire des sources) pour assurer le développement en équipe.

## 2.9. Références Clients

La Société a une clientèle prestigieuse tant sur le marché local qu'à l'étranger.


Le tableau 5 présente une liste (non exhaustive) de cette clientèle avec les projets réalisés et leurs domaines :

**Tableau 5 : Exemple des clients de la société**

Clients	Projet	Domaine
 France	Refonte du réseau social de l'entreprise	TRANSPORT
 France   La Réunion	Gestion demande de crédits  Outil de contrôle des risques   Suivi des commerciaux  Aide à la vente de crédit/épargne  App mobile classement des agences	BANQUE
 France	Mise en place du site team Europcar	SPORT

    	<p>Site orange.mg Service d'actualités Vidéo streaming</p> <p>Interconnexion du SI de Telma et de Blueline pour vendre de l'Airtime</p> <p>Mvola Gestion de clients Application de covoiturage Création du SI Comores</p>	<p>TELE-OPERATEURS</p>
  <p>France</p>	<p>Site d'émissions (ONPQDC, nouvelles écritures)</p>	<p>MEDIA</p>
  <p>France</p>    <p>Madagascar</p>	<p>App de génération des notices de présentation pour tous les salons automobiles du monde</p>  <p>App tablette fête de la bière Gestion documentaire pour la norme Iso 9001</p>	<p>INDUSTRIE</p>



 <p>France</p>	<p>Outils de production : Vente et publication de rapports de notation sur le marché de l'informatique</p>	<p>SERVICE</p>
---	--	----------------

# Chapitre 3 : Description du projet

## 3.1. Formulation

NETAPSYS Madagascar est une référence en matière de vente de logiciel dans toute l'île ; de ce fait, nombreux sont les sociétés, entreprises ou organismes qui sollicitent l'aide de NETAPSYS à fin d'accroître leurs champs d'action, booster leurs productions et leurs chiffre d'affaire.

Comme le cas de ce projet, une société demande à NETAPSYS Madagascar de faire un système d'information pour qu'elle puisse améliorer leur processus de vente et exploité pleinement les possibilités offertes par la technologie mais à cause d'une clause de confidentialité la société ne peut pas être divulgué.

Durant le stage, le projet consiste donc à concevoir puis réaliser une application mobile pour leur de prise de commande.

## 3.2. Objectif et besoin de l'utilisateur

L'objectif de du projet est de concevoir et réaliser une application mobile permettant la prise de commande faite par les marchands détaillant auprès d'une société qui est le client de NETAPSYS Madagascar et ainsi facilitant leurs tâches pour le stockage de ces données et facilite aussi la circulation des informations en temps réel. Pour aboutir à ce projet on a dû effectuer une enquête sur les employés de NETAPSYS qui ont une contacte directe avec le client et pris en charge toutes les demandes de ces derniers.

Et c'est à partir de cette enquête que nous avons pu déterminer les problèmes du client et en déduire les solutions afin de résoudre ces derniers.

Les besoins de l'utilisateur sont :

- Authentification
- Consultation de la liste des points de ventes du « routing ».
- Prise de commande et envoie via une connexion internet.
- Consultation de l'évolution du « routing ».
- Stockage des données avec un système embarqué.

- Création d'un point de vente.
- Clôture de la journée.

### **3.3. Moyens nécessaires pour la réalisation du projet**

**Moyens personnels :** Quatre programmeurs :

- Un chef de projet
- Deux développeurs pour la partie serveur
- Le stagiaire
- Un développeur mobile

**Moyens matériels :**

- Deux ordinateurs pour le développement de l'application mobile
  - Ordinateur de bureau : intel Core i3 3200 2.67Ghz, RAM 8 Go.
  - Ordinateur de bureau : intel Core i5 2500 2.74Ghz, RAM 8 Go.
- Un ordinateur utilisé comme serveur de test
- Deux tablettes utilisées comme appareils de test
  - ARCHOS

**Moyens logiciels :** Des logiciels qui facilitent les communications et le développement en groupe.

- Skype
- GIT
- JIRA

### **3.4. Résultats attendus**

L'utilisation de cette application au sein du client devra donc répondre à ces critères :

- Facile à utiliser
- Fiable dans les résultats.
- Accélération du processus de vente de ses produits.
- Sécurité de donnée assurée.

## **PARTIE II : ANALYSE ET CONCEPTION**

# Chapitre 4 : Analyse préalable

## 4.1. Analyse de l'existant

### 1.1.1. Organisation actuelle

Actuellement pour la prise de commande, le client utilise des documents en papier.

- Facile à utiliser par tous les agents responsables de la prise de commande.
- Facile à lire et plus pratique pour les marchands qui effectue les commandes.
- Pièce justificative de passage des agents.

### 4.1.1. Inventaire des moyens matériels et logiciels

Après enquête auprès des employés au sein de NETAPSYS et après avoir lu leur rapport nous avons pu identifier les moyens suivants :

#### Moyens matériels :

- Des ordinateurs

Le tableau 6 montre la liste des ordinateurs courant au sein du service informatique du client

**Tableau 6 : La liste des ordinateurs courant au sein du service informatique du client**

Type	Marques	Processeurs	RAM	Disque dur
Desktop	HP, DELL	>=2Ghz, Core de Duo	1 à 8Go	160Go à 1To
Laptop	TOSHIBA, HP COMPAQ	>=2Ghz, Core de Duo	1 à 8Go	160Go à 1To

- Connexion internet

#### Moyens logiciels :

- Windows XP, 7, 8 32 ou 64bit

Système d'exploitation appartenant à Microsoft.

- Suite Microsoft office 2007 à 2013

Outils bureautiques avec licence.

- Kaspersky

Antivirus professionnel avec licence utilisateur.

## **4.2. Critique de l'existant**

Certes, l'utilisation des documents en papier est plus facile pour les agents, elle représente quand même des failles.

- Facile à perdre
- Susceptible d'être endommagé
- Une grande perte de temps entre la prise de commande et sa préparation
- Non modifiable

## **4.3. Conception avant-projet**

### **4.3.1. Proposition des solutions**

Pour faire face à ces blocages, nous avons envisagé deux solutions :

#### **Solution 1 : Création d'une application web de vente en ligne.**

Cette première solution est très intéressante du fait qu'elle ne limite plus la prise de commande qu'au passage des commerciaux mais que chaque magasin ou épicerie peut effectuer des commandes à tout moment. Cela réduirait aussi le nombre d'effectif et le temps perdu à attendre durant lequel les commerciaux prennent la commande.

#### **Solution 2 : Création d'une application mobile de prise de commande.**

L'avantage de cette deuxième solution est le fait que les données sont plus en sécurité et que ça ne chamboule en rien leur processus de vente. La transmission des données se fait aussi en temps réel, donc le temps d'attente pour la préparation est donc réduit, amenant la possibilité que celui qui prépare pourra donc finir sa journée avec celui qui passe les commandes.

Vue que les clients potentiels visés par notre client sont des épiceries, donc ceux qui vendent des produits de première nécessité, la solution deux a donc été choisie du fait que le prix de la connexion pourrait être un blocage pour les marchands, et aussi le fait qu'il est possible qu'un marchand ne sache pas utiliser le web.

### **4.3.2. Outils utilisés**

#### **4.3.2.1 UML**

UML (Unified Modeling Language) est un langage de modélisation graphique à base de pictogramme utilisé pour le développement des logiciels que ce soit « desktop », web ou mobile.

De ce fait, pour bien mener à terme le développement de notre application suivant une norme et aussi pour minimiser le temps de développement nous avons dû faire appel à ce langage. L'utilisation de celle-ci facilite aussi les futures améliorations lors des maintenances vues qu'il est utilisé par les informaticiens pour réaliser des travaux d'experts.

##### *a. Présentation d'UML*

UML n'est pas une méthode (i.e. une description normative des étapes de la modélisation) : ses auteurs ont en effet estimé qu'il n'était pas opportun de définir une méthode en raison de la diversité des cas particuliers. Ils ont préféré se borner à définir un langage graphique qui permet de représenter et de communiquer les divers aspects d'un système d'information. [1]

UML est caractérisé par un travail d'expert qui utilise l'approche orientée objet et suivant une norme. Sa notation limite les ambiguïtés et les incompréhensions. Il est aussi indépendant du langage de programmation et peut être utilisé pour modéliser n'importe quel système.

UML 2.0 comporte ainsi treize types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent en deux grands groupes :

#### **Diagrammes structurels ou diagrammes statiques (UML Structure) [1]**

- Diagramme de classes (Class diagram)[1]
- Diagramme d'objets (Object diagram) [1]

- Diagramme de composants (Component diagram) [1]
- Diagramme de déploiement (Deployment diagram) [1]
- Diagramme de paquetages (Package diagram) [1]
- Diagramme de structures composites (Composite structure diagram) [1]

### **Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior)**

[1]

- Diagramme de cas d'utilisation (Use case diagram) [1]
- Diagramme d'activités (Activity diagram) [1]
- Diagramme d'états-transitions (State machine diagram) [1]
- **Diagrammes d'interaction (Interaction diagram)** [1]
  - Diagramme de séquence (Sequence diagram) [1]
  - Diagramme de communication (Communication diagram) [1]
  - Diagramme global d'interaction (Interaction overview diagram) [1]
  - Diagramme de temps (Timing diagram) [1]

#### **4.3.2.2 La méthode AGILE notamment SCRUM**

La réussite d'un projet ne dépend pas seulement de la compétence du développeur ni des algorithmes ou de la modélisation. Elle dépend surtout de la gestion du projet donc de la méthodologie pour aboutir à un système d'information respectant à la fois le délai et la qualité.

Lors de développement d'une application, il se peut que l'application soit mal testée, que plusieurs fonctionnalités ne sont même pas utilisées ou que le développement met plus de temps que prévu d'où les retards de livraison et de dette technique. Afin, de minimiser ces défauts, la méthode AGILE notamment « SCRUM » adopte une approche assez particulière.

##### *a. Définition*

Une méthode agile est une approche itérative et incrémentale pour le développement de logiciel, réalisé de manière très collaborative par des équipes responsabilisées, appliquant un cérémonial minimal, qui produisent, dans un délai contraint, un logiciel de grande qualité répondant aux besoins changeants des utilisateurs [2].

Scrum sert à développer des produits, en quelques mois tout au plus. Les fonctionnalités souhaitées sont collectées dans le backlog de produit et classées par priorité.



C'est le Product Owner qui est responsable de la bonne tenue de ce backlog. Scrum fait partie des approches itératives et incrémentales dont le modèle de cycle de développement est basé sur une phase qui se répète plusieurs fois successivement. C'est la notion d'itération, appelée sprint avec Scrum. Tous les sprints se déroulent selon le même schéma et on y fait à chaque fois les mêmes types de travaux [2].

### *b. Principes*

Le principe de la méthode SCRUM est d'itérer une même action successivement afin d'aboutir à une application livrable par sprint. De ce fait un sprint est donc composé par le développement et les tests en même temps afin de pouvoir effectuer une livraison après chaque sprint.

D'ailleurs, dans la méthodologie SCRUM il n'y a plus la notion d'hierarchie dans l'équipe, plus de chef de projet et c'est l'équipe elle-même qui s'organise pour le définir le rôle de chacun dans le processus de développement.

Pour ce qui est de l'estimation de charge, on procède à des *plannings poker* durant laquelle on pourra déterminer les compétences de chacun par rapport aux fonctionnalités du projet et ainsi faciliter la distribution des tâches pour aboutir à un résultat optimale. Après le *planning poker* on obtient non la durée de développement mais les points d'efforts associés à un sprint qui est un laps de temps prédéfini à l'avance.

### *c. Vocabulaires de SCRUM*

- **Sprint** est le terme utilisé pour itération. Un sprint est un bloc de temps fixé aboutissant à créer un incrément de produit potentiellement livrable.
- Le **Product Owner** est la personne qui représente le « métier ». Il apporte sa vision à l'équipe et définit les priorités de façon à obtenir un produit ayant le maximum de valeur.
- Le **ScrumMaster** ne gère pas des ressources interchangeables, il guide les femmes et les hommes de l'équipe. Son rôle essentiel est de les faire progresser collectivement pour la réussite du projet.
- Le **backlog** de produit est la liste des futures réalisations de l'équipe. C'est l'élément pivot d'un développement avec Scrum en ce qui concerne le contenu du produit et la gestion de projet.

- Un **release** est la période de temps, constituée de sprints, utilisée pour planifier à moyen terme. Il correspond aussi à une version livrable de l'application qui va être livré au client suite la réalisation d'une ou plusieurs sprint.

#### **4.3.2.3 Le système d'exploitation Windows**

C'est un système d'exploitation appartenant à Microsoft et qui est de nos jours le plus utilisé dans le monde.

- Caractéristiques de Windows :
  - La plus grande bibliothèque de programmes et applications [3].
  - Certains jeux commerciaux fonctionnent uniquement avec Windows et DirectX [3].
  - Presque tous les matériels à pilotes sont compatibles avec Windows [3].
  - Le plus largement utilisé [3].

#### **4.3.2.4 Le langage JAVA**

Le langage Java est un langage de programmation informatique orienté objet créé par James Godling et Patrick Naughton, employé de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld. La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java. [4]

- Caractéristiques de JAVA :
  - Orienté objet
  - Les applications faites en Java sont particulièrement faciles à porter sur plusieurs systèmes d'exploitation comme UNIX, Windows, Mac OS et aussi sur certains appareils mobiles grâce à JRE (Java Runtime Environnement)
  - Peut être développé sur plusieurs plateformes
  - Possède aussi de nombreux Frameworks
  - C'est aussi un langage de programmation très utilisé pour le développement d'application mobile notamment Android
  - Possède des collections qui sont très pratique pour le développement d'une application
  - C'est un langage qui reprend une grande partie la syntaxe du langage C++

#### **4.3.2.5 Le SGBD SQLite**

SQLite est une bibliothèque écrite en C qui propose un moteur de base de données relationnelle accessible par le langage SQL. SQLite implémente en grande partie le standard SQL-92 et des propriétés ACID.

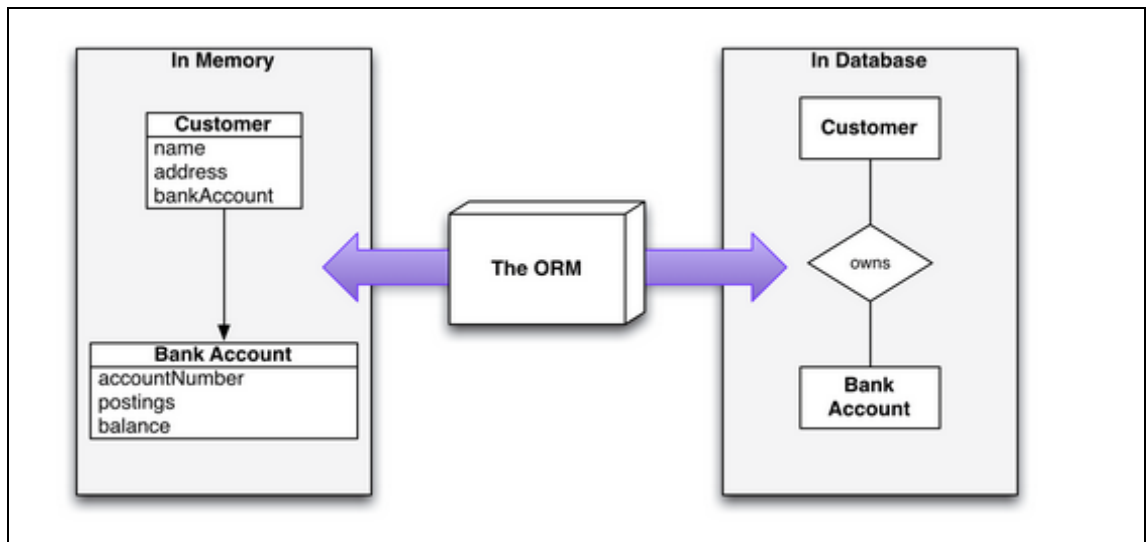
➤ **Caractéristiques de SQLite :**

- Directement intégrée aux programmes contrairement à MySQL qui reproduit le schéma habituel client-serveur.
- L'intégralité de la base est stockée dans un fichier indépendant de la plateforme.
- Multiplateforme.
- Licence Open source.
- Moteur de base de données le plus distribué au monde grâce à son utilisation dans de nombreux logiciels grand public comme Firefox, Skype, Google Gears et bien d'autre encore.
- Légèreté.
- Idéale pour les systèmes embarqués.

#### **4.3.2.6 L'ORM Activeandroid**

Activeandroid est un Object Relational Mapper (ORM) pour Android facilitant les interactions avec SQLite. Comme toute ORM elle permet de construire les tables de la base par l'intermédiaire de model java sans écrire de script SQL et aussi d'effectuer l'insertion, mise à jour et suppression des données sans aucun script.

La figure 4 montre la représentation d'un ORM entre les modèles et la base



**Figure 4 : Représentation d'un ORM entre les modèles et la base**

#### **4.3.2.7 L'IDE Android Studio**

Pour le développement d'application pour Android, Google a montré la première version beta d'Android Studio durant le Google I/O en 2013 et a sorti la première version stable 1.0 en décembre 2014 et par la suite a fait d'Android Studio l'IDE officiel pour le développement d'application Android. De ce fait, la documentation officielle pour le développement d'Android fait actuellement référence à Android Studio qui était au par avant basé sur Eclipse ADT.

Le tableau 7 présente la comparaison entre Android Studio et Eclipse ADT

**Tableau 7 : Comparaison entre Android Studio et Eclipse ADT**

Caractéristique	ADT (Eclipse)	Android Studio
Facilité d'installation	Moyen	Simple
Langue	Nombreuse	Anglais
Performance	Peut être lourd	Rapide
Système de build	Ant	Gradle
Génération de variante et de multiple APK	Non	Oui
Android Code completion et refactoring	Base	Avancé
Editeur d'interface graphique	Oui	Oui
Signature d'APK et gestion de Keystore	Oui	Oui
Support NDK	Oui	A venir

[5]

Android Studio est un IDE développement d'application Android basé sur l'IDE de JetBrains, IntelliJ IDEA. Malgré le fait que le SDK d'Android n'est pas intégré à l'Android Studio, il reste néant moins l'IDE le plus adapté au développement d'application Android au niveau professionnel. Une autre différence par rapport à Eclipse est l'utilisation de Gradle qui apporte des fonctionnalités intéressantes.

- Permet la construction de projet. [5]
- Combine Ant et Maven. [5]
- Permet d'écrire des tâches de construction utilisant le langage Groovy. [5]
- Facilité d'intégrer des plugins. [5]
- Gestion automatique des dépendances. [5]

#### 4.3.2.8 GIT

GIT est un logiciel de gestion de version décentralisé créé par Linus Torvalds en 2005. Il permet de conserver l'historique des modifications apportées dans le développement d'un logiciel. Il est très important de savoir qui a modifié quoi et à quel moment et pourquoi dans le développement en groupe. GIT est l'outil utilisé au sein de NETAPSYS Madagascar comme logiciel de gestion de version. En tant que logiciel de gestion de version décentralisé, GIT permet de :

- Travailler de manière désynchronisée par rapport aux autres membres du projet.
- Travailler en mode hors-ligne.
- Donner aux développeurs une grande liberté, puisque leurs modifications ne peuvent affecter que leurs dépôts.

#### 4.3.2.9 Web service REST Client

REST (REpresentational State Transfer) est un style d'architecture pour concevoir un service web, utilisant au maximum les possibilités de HTTP. Ses principales caractéristiques sont les suivantes : [6]

- **application client/serveur** : le transport sur le réseau est assuré par HTTP (essence même d'un web service) [6]
- **interface uniforme** : tout élément offert à la manipulation par l'application est nommé ressource et est identifié de manière unique par les Identifiants de Ressource Uniforme (URI ci-après) suivant le schéma : [6]

`http_URL = "http:" "//" host [":" port]`

Deux types de schéma d'URI sont distingués :

- **URI member** qui désigne une seule ressource.
- **URI collection** qui désigne une liste de ressources de même type.
- **La sémantique** des messages du client au serveur est **celle de HTTP** : [6]
  - **GET URI** : récupérer la représentation d'une ressource (URI member), ou d'une liste de ressources (URI collection)
  - **POST URI** (collection) : ajouter une ressource à une liste de ressources existantes, donc création de ressource.
  - **PUT URI** (member) : modifier une ressource existante ou créer une nouvelle ressource.

- **DELETE URI** : destruction d'une (URI member) ou plusieurs (URI collection) ressources.
- **les réponses** du serveur aux clients utilisent également les messages HTTP, en particulier les codes d'erreur suivants :
  - 200 Ok
  - 404 Ressource non trouvée sur le serveur
- **représentation des ressources** : le format avec lequel une ressource est représentée est obtenu par négociation grâce aux entêtes HTTP. Le client indique ce qu'il souhaite comme format de représentation avec l'entête "Accept", le client et le serveur utilisent l'entête "Content-type" pour décrire le format de représentation de la ressource incluse dans le message. Les formats les plus fréquemment rencontrés sont XML et JSON. [6]

# Chapitre 5 : Analyse conceptuelle

## 5.1. Spécification des besoins

### 5.1.1. Règles de fonctionnalité

- Une connexion internet est requise pour l'utilisation de l'application.
- Le commercial ne pourra pas terminer sa journée tant qu'il n'a pas visité ou annulé tous les points de ventes.
- Aucune action possible après clôture de la journée.
- Le commercial peut prendre plusieurs bons de commande sur un seul point de vente.

### 5.1.2. Dictionnaire des données

Le dictionnaire des données est un tableau regroupant tous les données nécessaire à la création d'une base de données d'une application.

Le tableau 8 montre le dictionnaire des données.

**Tableau 8 : Dictionnaire des données**

Nom	Signification	Type	Taille
abreviation	Contient l'abréviation du nom de l'article	A	10
adresse	L'adresse du point de vente	AN	20
category	Libellé de la catégorie de point de vente	A	10
contact_name	Le nom de la personne qui gère le point de vente	A	50
date	Date de la prise d'une commande	AN	10
description	La description du point de vente	A	30
firstName	Le prénom de l'utilisateur connecté à l'application	A	50
id_animation	Identifiant de l'animation de l'article	N	5
id_article	Identifiant de l'article	N	5



id_command	Identifiant de la commande effectuée par l'utilisateur sur un point de vente	N	5
id_pdv	Identifiant du point de vente	N	5
id_produit_category	Identifiant de la catégorie de produit	N	5
id_produit_family	Identifiant de la famille de produit	N	5
id_quartier	Identifiant du quartier dans lequel se trouve un point de vente	N	5
id_statut	Identifiant du statut d'un point de vente	N	2
id_user	Identifiant de l'utilisateur connecté à l'application	N	5
is_available	Sert à savoir si le produit est toujours disponible	N	1
label_animation	Contient le label de l'animation de l'article	A	10
label_produit_category	Libellé de la catégorie de produit	A	30
label_produit_family	Libellé de la famille de produit	A	30
Label_quartier	Le nom du quartier dans lequel se trouve un point de vente	A	20
label_statut	Libellé du statut d'un point de vente	A	10
lastName	Le nom de l'utilisateur connecté à l'application	A	30
login	Contient le login de l'utilisateur connecté à l'application	AN	40
name	Le nom de l'article	A	20
ordre	Contient l'ordre de l'article	N	4
packaging	Contient le système de paquage de l'article	A	10

password	Contient le mot de passe de l'utilisateur connecté à l'application	AN	50
pdv_name	Le nom du point de vente	A	20
phone_number	Le numéro de téléphone de l'utilisateur connecté à l'application	AN	10
photo	Contient l'Url du photo de l'utilisateur connecté à l'application	AN	50
product_buying_price	Contient le prix d'achat de l'article	N	10
product_reference	Contient la référence de l'article	AN	10
product_selling_price	Contient le prix de vente de l'article	N	10
quantité	C'est la quantité commandé pour un article	N	4
total_prix	Le prix total de la commande	N	10
url_image_produit	Contient l'url de l'image de l'article	AN	50

AN : Alphanumérique

N : Numérique

A : Alphabétique



### 5.1.3. Diagramme des cas d'utilisations

Les cas d'utilisation sont une technique de description du système étudié privilégiant le point de vue de l'utilisateur. Il s'agit de la solution UML pour représenter le modèle conceptuel. Les cas d'utilisation décrivent sous la forme d'actions et de réactions, le comportement d'un système du point de vue d'un utilisateur. Les cas d'utilisation servent à structurer les besoins des utilisateurs et les objectifs correspondants du système. Il est à noter que les cas d'utilisations ne s'enchainent pas, puisqu'il n'y a aucune représentation temporelle dans un diagramme de cas d'utilisation.

Les éléments constitutifs des cas d'utilisation :

- **Acteur** : entité externe qui agit sur le système ; Le terme acteur ne désigne pas seulement les utilisateurs humains mais également les autres systèmes. Les acteurs sont des classificateurs qui représentent des rôles au travers d'une certaine utilisation (cas) et non pas des personnes physiques. Ce sont des acteurs types.
- **Cas d'utilisation** : ensemble d'actions réalisées par le système en réponse à une action d'un acteur.
  - Les cas d'utilisation peuvent être structurés,
  - Les cas d'utilisation peuvent être organisés en paquetages,
  - L'ensemble des cas d'utilisation décrit les objectifs du système.

#### 5.1.3.1 Formalisme

Dans le diagramme de cas d'utilisation, l'acteur est représenté par un bonhomme. [1]

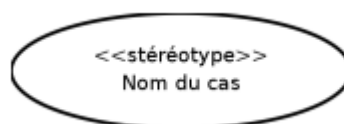
La figure 5 illustre la représentation d'un acteur.



**Figure 5 : Représentation d'un acteur**

Pour ce qui est d'un cas d'utilisation, il est représenté par une ellipse contenant le nom du cas d'utilisation.

La figure 6 montre la représentation d'un cas d'utilisation



**Figure 6 : Représentation d'un cas d'utilisation.**

### 5.1.3.2 Relations entre cas d'utilisation

Il existe principalement deux types de relations :

- Les dépendances stéréotypées, qui sont explicitées par un stéréotype (les plus utilisés sont l'inclusion et l'extension), [1]
- La généralisation/spécialisation. [1]

Une dépendance se représente par une flèche avec un trait pointillé et la génération par une flèche avec un trait pleins dont la pointe est un triangle fermé désignant le cas le plus général.

#### *a. Relation d'inclusion*

Un cas A inclut un cas B si le comportement décrit par le cas A inclut le comportement du cas B : le cas A dépend de B. Lorsque A est sollicité, B l'est obligatoirement, comme une partie de A. Cette dépendance est symbolisée par le stéréotype « include ». Cette relation permet aussi de décomposer un cas complexe en sous cas plus simple. [1]

#### *b. Relation d'extension*

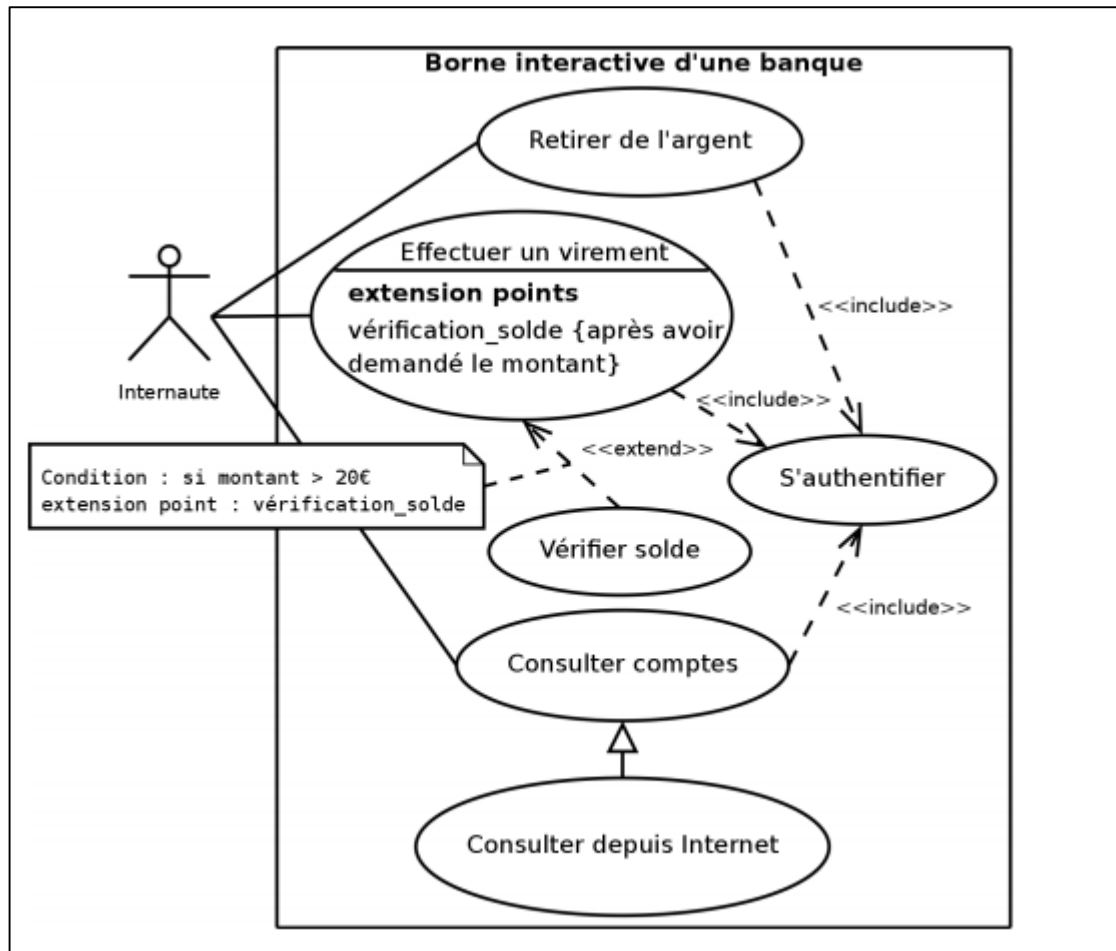
On dit qu'un cas d'utilisation A étend un cas d'utilisation B lorsque le cas d'utilisation A peut être appelé au cours de l'exécution du cas d'utilisation B. Exécuter B peut éventuellement entraîner l'exécution de A: contrairement à l'inclusion, l'extension est optionnelle. Cette dépendance est symbolisée par le stéréotype « extend ». [1]

L'extension peut intervenir à un point précis du cas étendu. Ce point s'appelle le point d'extension. Il porte un nom, qui figure dans un compartiment du cas étendu sous la rubrique point d'extension, et est éventuellement associé à une contrainte indiquant le moment où l'extension intervient. Une extension est souvent soumise à condition. Graphiquement, la condition est exprimée sous la forme d'une note.

#### *c. Relation de généralisation*

Un cas A est une généralisation d'un cas B si B est un cas particulier de A.

La figure 7 démontre un exemple de cas d'utilisation

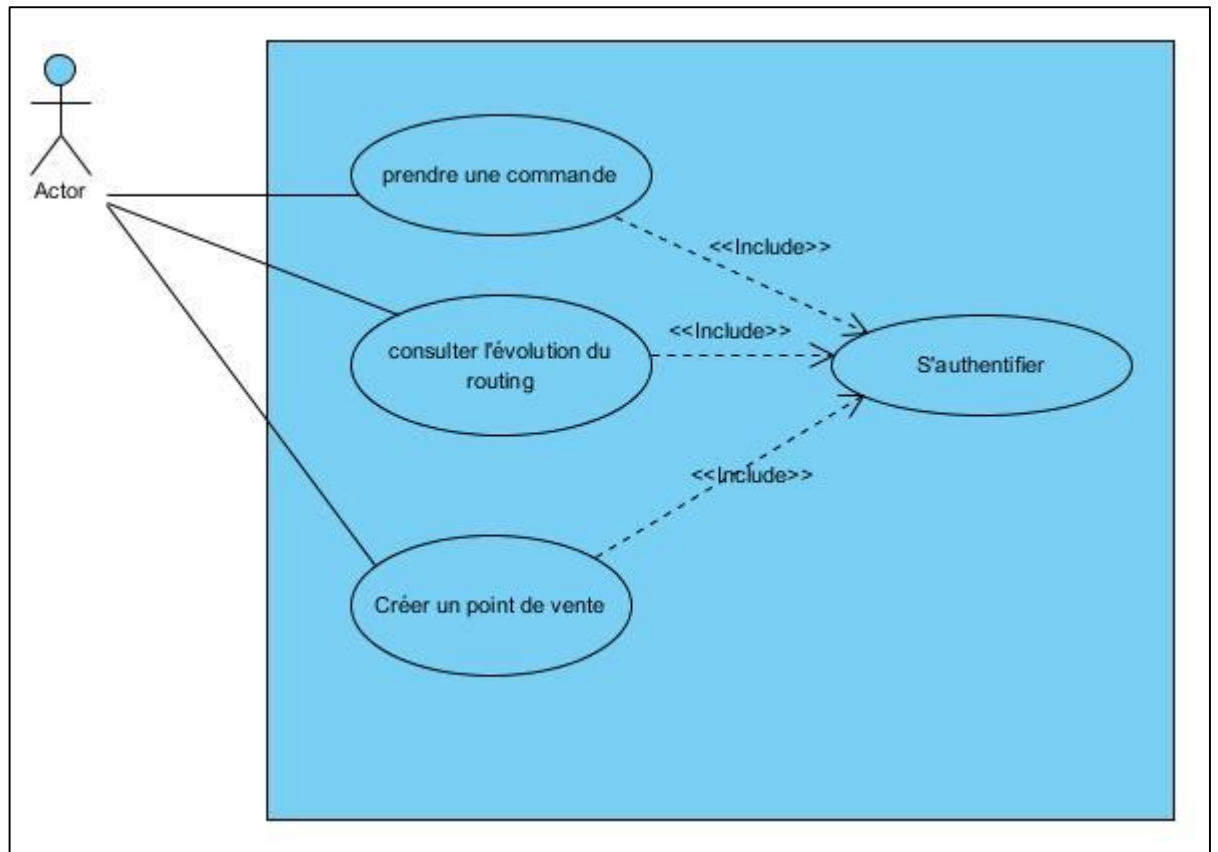


**Figure 7 : exemple d'un cas d'utilisation**

#### 5.1.4. Cas d'utilisation de l'application

Suite aux fonctionnalités demandées par le client, on a pu aboutir à un diagramme de cas d'utilisation.

La figure 8 représente le diagramme des cas d'utilisations.



**Figure 8 : Diagramme des cas d'utilisations.**

#### **5.1.5. Priorisation des cas d'utilisations**

La priorité des cas d'utilisation est conforme au besoin de l'utilisateur et de la fonctionnalité de l'application. Ce qui veut dire que ceux qui ont le plus de priorité qui garantissent le fonctionnement minimal du logiciel.

- Indice 1 : Cas d'utilisation les plus prioritaires.
- Indice 2 : Cas d'utilisation prioritaires.
- Indice 3 : Cas d'utilisation les moins prioritaires.

Le tableau 9 montre la priorisation des cas d'utilisation

**Tableau 9 : Priorisation des cas d'utilisations**

Nom du cas d'utilisation	Priorité
S'authentifier	1
Prendre une commande	1
Créer un point de vente	2
Consulter l'évolution du routing	3

## **5.2. Modélisation du domaine**

### **5.2.1. Diagramme des séquences systèmes**

Le diagramme des séquences systèmes est une représentation temporelle d'interaction entre l'acteur et le système sans se soucier des processus internes du système.

Le diagramme de séquence « système » décrit, pour un cas d'utilisation, ou pour un scénario d'un cas d'utilisation, les échanges entre l'utilisateur et le système (c'est-à-dire le logiciel à réaliser) et plus généralement les échanges entre le système et tous les acteurs du système. [7]

On peut choisir de se limiter au diagramme de séquence système du scénario nominal pour éviter d'alourdir le diagramme. Les scénarios alternatifs pourront être présentés dans un diagramme d'activité.

#### ➤ Syntaxe des diagrammes de séquence « système »:

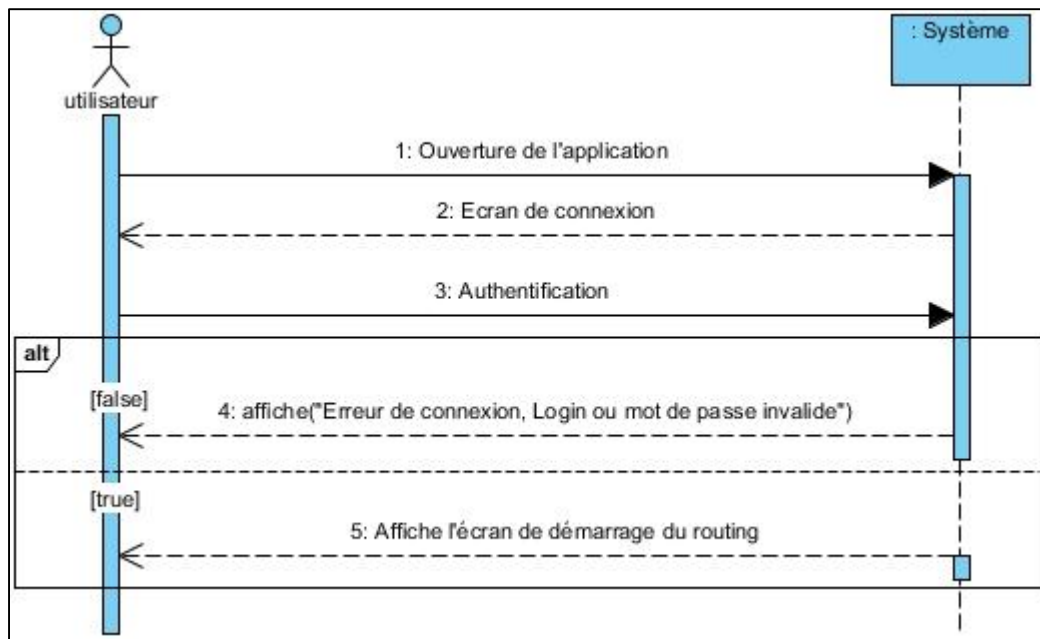
Préconditions et post-conditions décrivent l'état du système avant et après l'usage du cas d'utilisation.

—————▶ Message synchrone : une fois le message envoyé, l'expéditeur est bloqué jusqu'à ce que le destinataire accepte le message. [7]

—————> Message asynchrone : le message envoyé n'interrompt pas l'exécution de l'expéditeur. [7]

—————> Caractère synchrone ou asynchrone est indéterminé. Attention, cette représentation peut aussi être utilisée pour les messages asynchrones. [7]

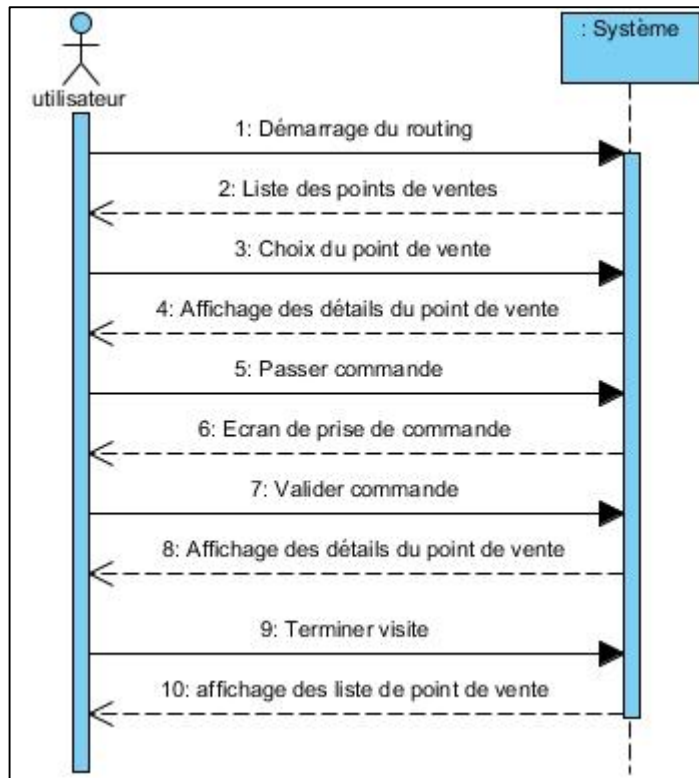
La figure 9 représente le diagramme des séquences systèmes d'authentification



**Figure 9 : Diagramme des séquences systèmes d'authentification**

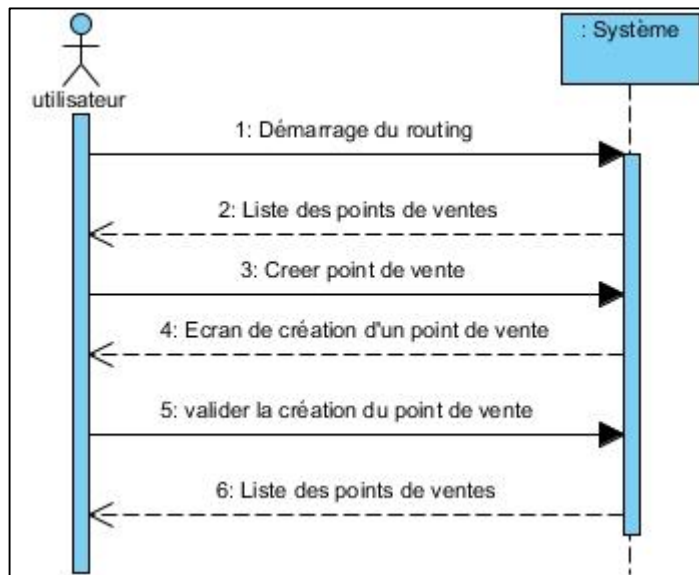
La figure 10 montre le diagramme de séquence système de prise de commande.





**Figure 10 : Diagramme de séquence système de prise de commande.**

La figure 11 illustre le diagramme de séquence système de création d'un point de vente



**Figure 11 : Diagramme de séquence système de création d'un point de vente.**

# Chapitre 6 : Conception détaillée

## 6.1. Architecture du système MVC

Le patron MVC est issu de travaux de Trygve Reenskaug en 1978-1979. Son but principal était de proposer une solution générale aux problèmes d'utilisateurs manipulant des données volumineuses et complexes. D'abord appelé Model-View-Editor (Modèle-vue-éditeur), Reenskaug le renomme Modèle-vue-contrôleur après avoir discuté notamment avec Adele Goldberg.

Le but de MVC est de séparer les couches d'une application (en 3 couches distinctes, au minimum, et le plus souvent). On va donc distinguer :

- Le modèle,
- La vue,
- Le contrôleur.

**Le modèle** : C'est le point le plus important dans l'architecture MVC car c'est le cœur même de l'application. C'est celui qui fait les traitements de données et les interactions avec la base. La base de données est un composant du modèle.

**Vue** : C'est l'interface même de l'application, c'est donc ce que l'utilisateur voit et manipule à sa guise. Elle présente tout simplement les résultats renvoyés par le modèle et envoie les événements aux contrôleurs.

**Contrôleur** : c'est celui qui gère les événements et enclenchées par l'utilisateur sur la vue. Il joue aussi le rôle d'intermédiaire entre le modèle et la vue. A préciser qu'il ne modifie pas la base de données, il se contente juste de demander au modèle de les modifier. C'est aussi le même principe pour la vue, il demande à la vue de ce modifier toute seule en cas de besoins.

## 6.2. Diagramme de séquence de conception pour chaque cas d'utilisation

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique. Ainsi, contrairement au diagramme de communication, le temps y est représenté explicitement par une dimension (la dimension verticale) et s'écoule de haut en bas.

### ➤ Représentation des lignes de vie

Une ligne de vie se représente par un rectangle, auquel est accrochée une ligne verticale pointillée, contenant une étiquette dont la syntaxe est : [`<nom_du_rôle>`]:[`<Nom_du_type>`]

Au moins un des deux noms doit être spécifié dans l'étiquette, les deux points (:) sont, quant à eux, obligatoire.

### ➤ Représentation des messages

Un message définit une communication particulière entre des lignes de vie. Plusieurs types de messages existent, les plus communs sont :

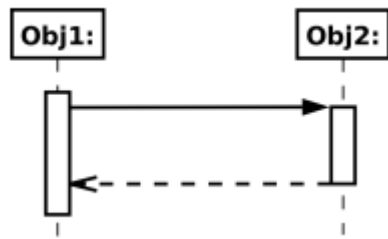
- l'envoi d'un signal ;
- l'invocation d'une opération ;
- la création ou la destruction d'une instance. [1]

La Figure 12 représente un message asynchrone



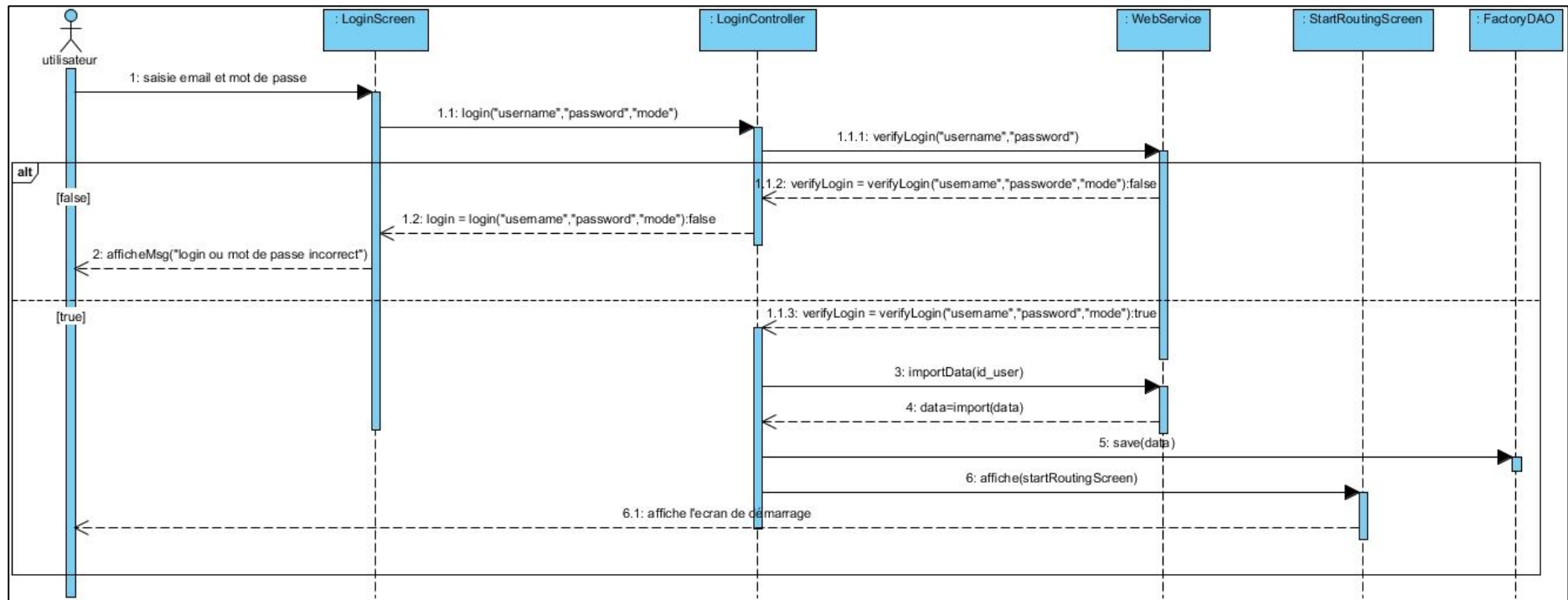
**Figure 12 : Message asynchrone.**

La Figure 13 illustre un message synchrone



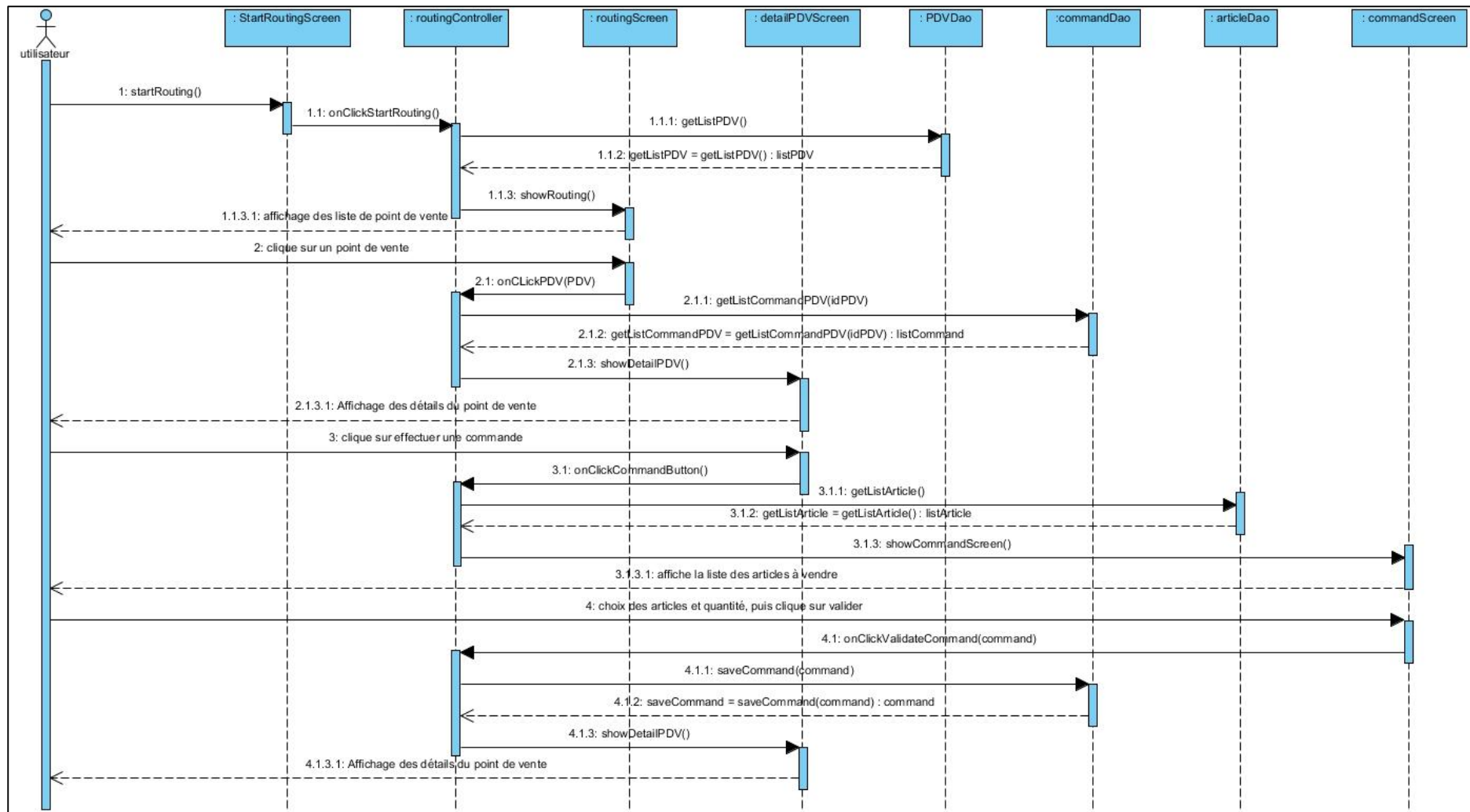
**Figure 13 : messages synchrone**

La Figure 14 montre le diagramme de séquence d'authentification.



**Figure 14 : Diagramme de séquence d'authentification.**

La figure 15 représente le diagramme de séquence de prise de commande



**Figure 15 : Diagramme de séquence de prise de commande.**

La figure 16 illustre le diagramme de séquence de création d'un point de vente

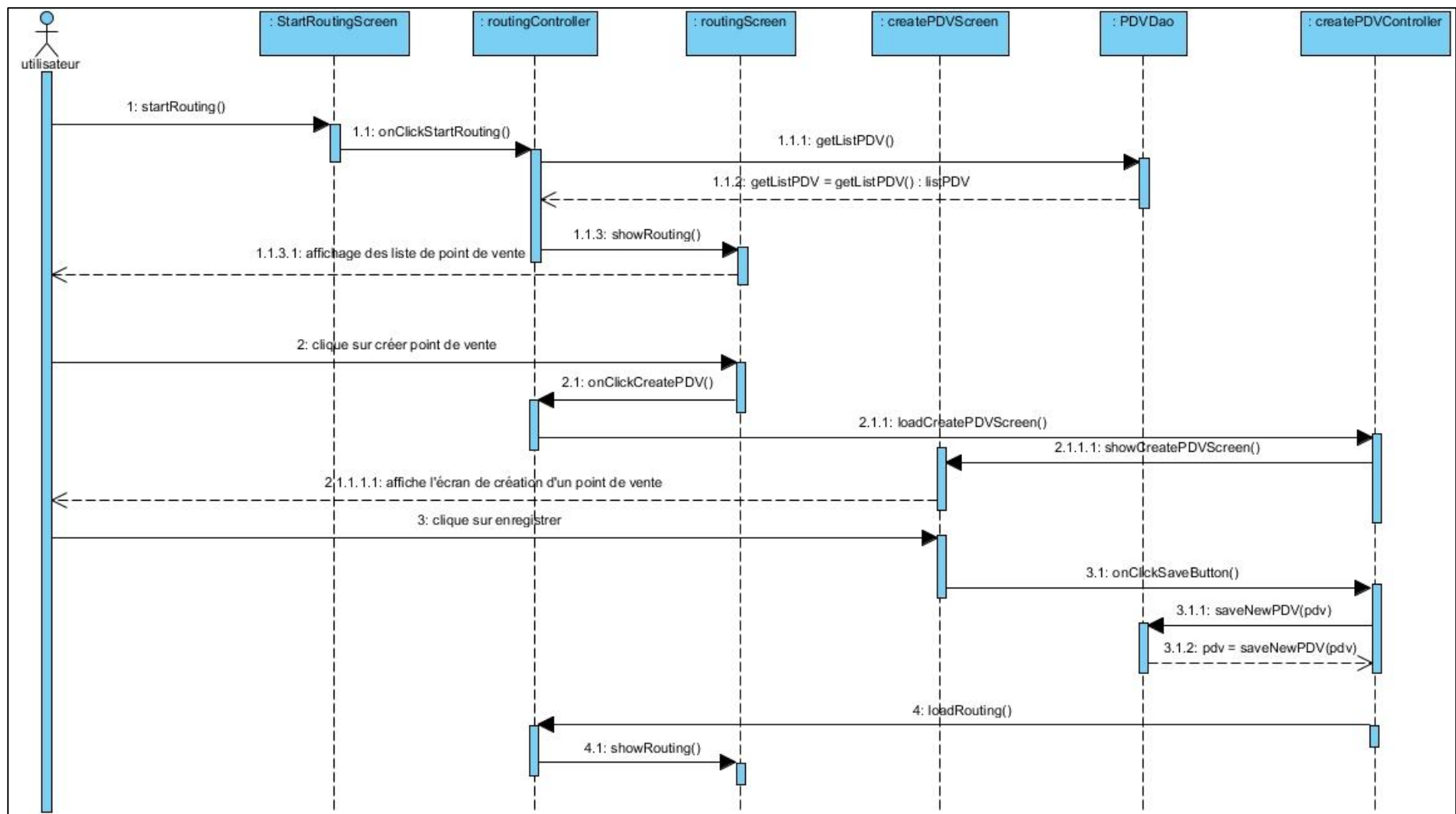


Figure 16 : Diagramme de séquence de prise de commande.

### 6.3. Diagramme de classe de conception pour chaque cas d'utilisation

Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation.

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes. Une classe permet de décrire un ensemble d'objets (attributs et comportement), tandis qu'une relation ou association permet de faire apparaître des liens entre ces objets. On peut donc dire : [8]

- Un objet est une instance de classe,
- Un lien est une instance de relation

#### 6.3.1. Formalisme

##### 6.3.1.1 Les classes

Une classe correspond à un concept global d'information et se compose d'un ensemble d'informations élémentaires, appelées attributs de la classe qui servent à la décrire. UML définit trois niveaux de visibilité pour les attributs et les opérations :

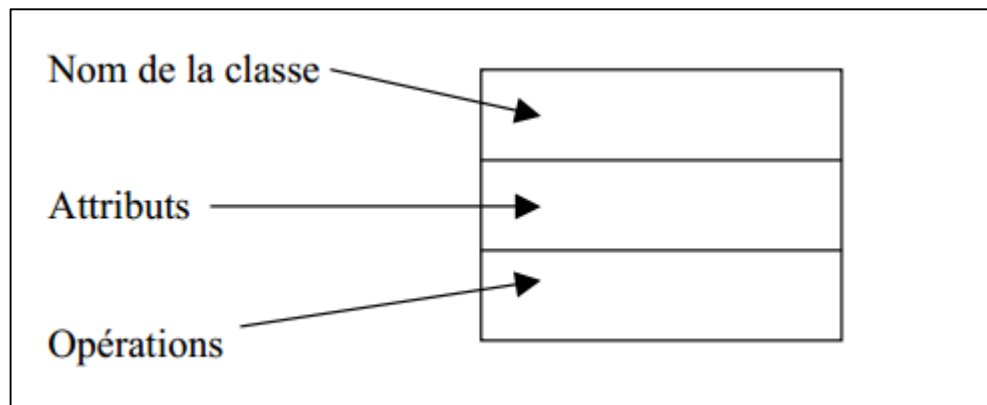
- Public qui rend l'élément visible à tous les clients de la classe,
- Protégé qui rend l'élément visible aux sous classes de la classe,
- Privé qui rend l'élément visible à la classe seule.

##### *a. Définition*

- **Classe**: description abstraite d'un ensemble d'objets qui partagent les mêmes propriétés (attributs et associations) et comportements (opérations et états). [8]
- **Attribut**: donnée déclarée au niveau d'une classe, éventuellement typée, à laquelle chacun des objets de cette classe donne une valeur. Un attribut peut posséder une multiplicité et une valeur initiale. Un attribut dérivé («/» ) est un attribut dont la valeur peut être déduite d'autres informations disponibles dans le modèle. [8]
- **Opération**: élément de comportement des objets, défini de manière globale dans leur classe. Une opération peut déclarer des paramètres (eux-mêmes typés) ainsi qu'un type de retour. [8]



La figure 17 montre la structure d'une classe

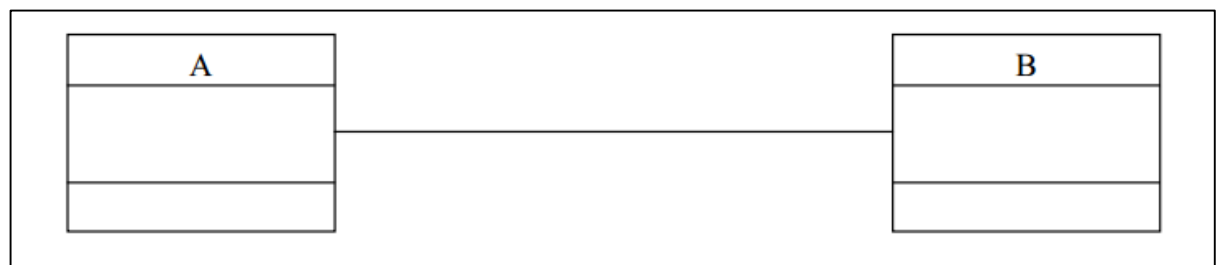


**Figure 17 : Structure d'une classe**

#### **6.3.1.2 Les associations**

Une association représente une relation structurelle entre classes d'objets. La plupart des associations sont binaires, c'est à dire qu'elles connectent deux classes. On représente une association en traçant une ligne entre les classes associées. [8]

La figure 18 représente l'association entre deux classes



**Figure 18 : Association entre deux classes**

##### *a. Multiplicité des associations*

Chaque rôle peut porter une multiplicité montrant combien d'objets de la classe considérée (celle qui joue ce rôle) peuvent être liés à une instance de l'autre classe par l'association. La multiplicité est représentée sous la forme d'un couple de cardinalités. [8]

Le tableau 10 présente la liste des multiplicités

**Tableau 10 : Liste des multiplicités**

1..1 noté 1	Un et un seul
0..1	Zéro ou un
0..* noté *	De Zéro à n
1..*	De un à n
n..m	De n à m

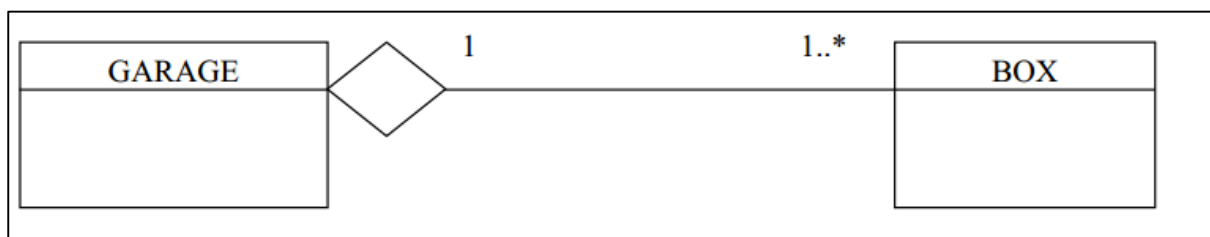
***b. Agrégation***

Une agrégation est un type particulier d'association. Elle traduit la volonté de renforcer la dépendance entre les classes. C'est une association non symétrique dans laquelle une des extrémités joue un rôle prédominant par rapport à l'autre extrémité.

Les critères suivants impliquent une agrégation :

- une classe fait partie d'une autre classe,
- une action sur une classe implique une action sur une autre classe,
- les objets d'une classe sont subordonnés aux objets d'une autre classe. [8]

La figure 19 illustre un exemple d'une relation d'agrégation

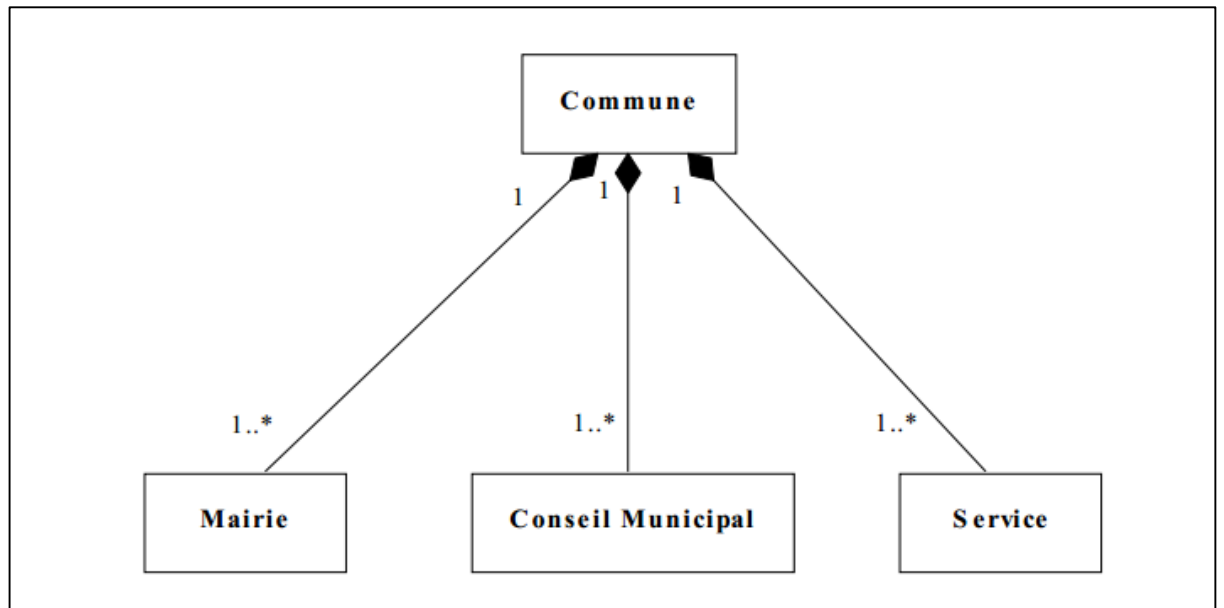


**Figure 19 : Exemple d'une relation d'agrégation**

***c. Composition***

La composition est un cas particulier d'agrégation dans laquelle la vie des composants est liée à celle de l'agrégat. Dans la composition, l'agrégat ne peut être multiple. La composition se représente par un losange noir. Une composition est une association contraignante : la suppression d'un objet agrégat entraîne la suppression des objets agrégés. [8]

La figure 20 montre un exemple de composition

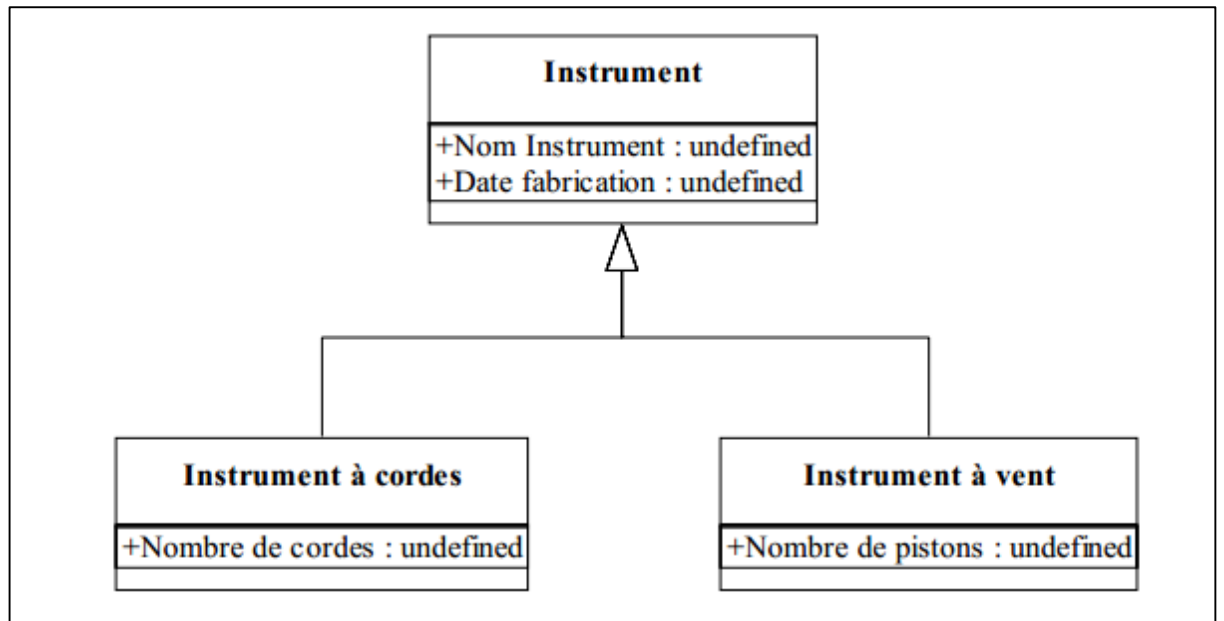


**Figure 20 : Exemple de composition**

#### *d. Généralisation*

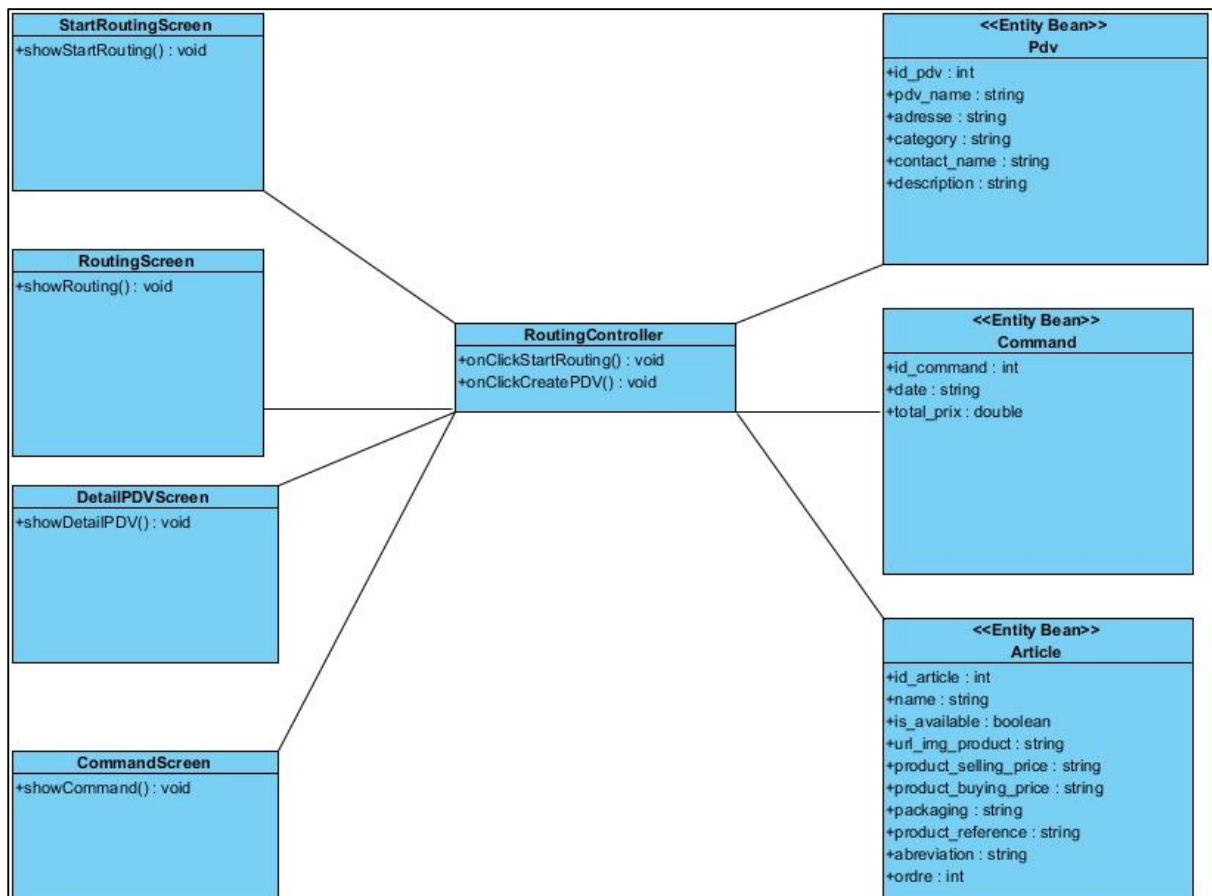
UML emploie le terme de généralisation pour désigner la relation de classification entre un élément plus général et un élément plus spécifique. La relation de généralisation signifie « est un » ou « est une sorte de ».

La figure 21 représente un exemple d'une généralisation



**Figure 21 : Exemple d'une généralisation**

La Figure 22 montre le diagramme de classe de prise de commande



**Figure 22 : Diagramme de classe de prise de commande.**

La figure 23 présente le diagramme de classe de création d'un point de vente

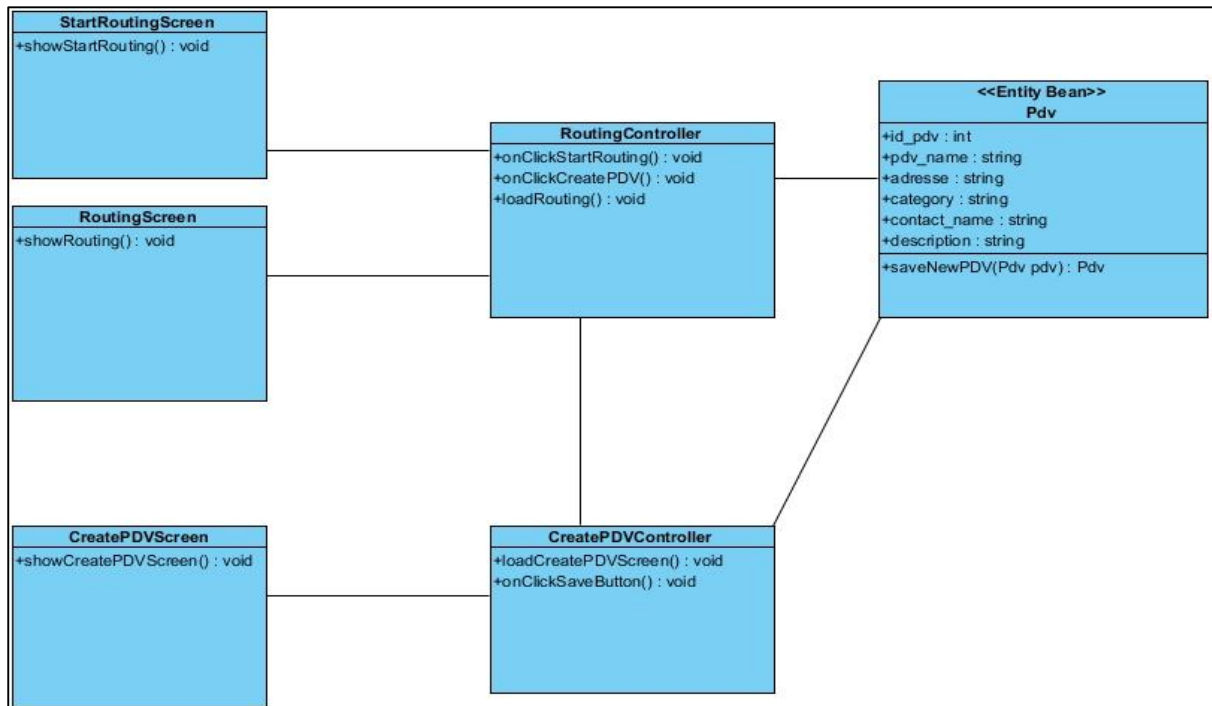


Figure 23 : Diagramme de classe de création d'un point de vente.

## 6.4. Diagramme de classe de conception globale

La figure 24 démontre le diagramme de classe de conception global de l'application

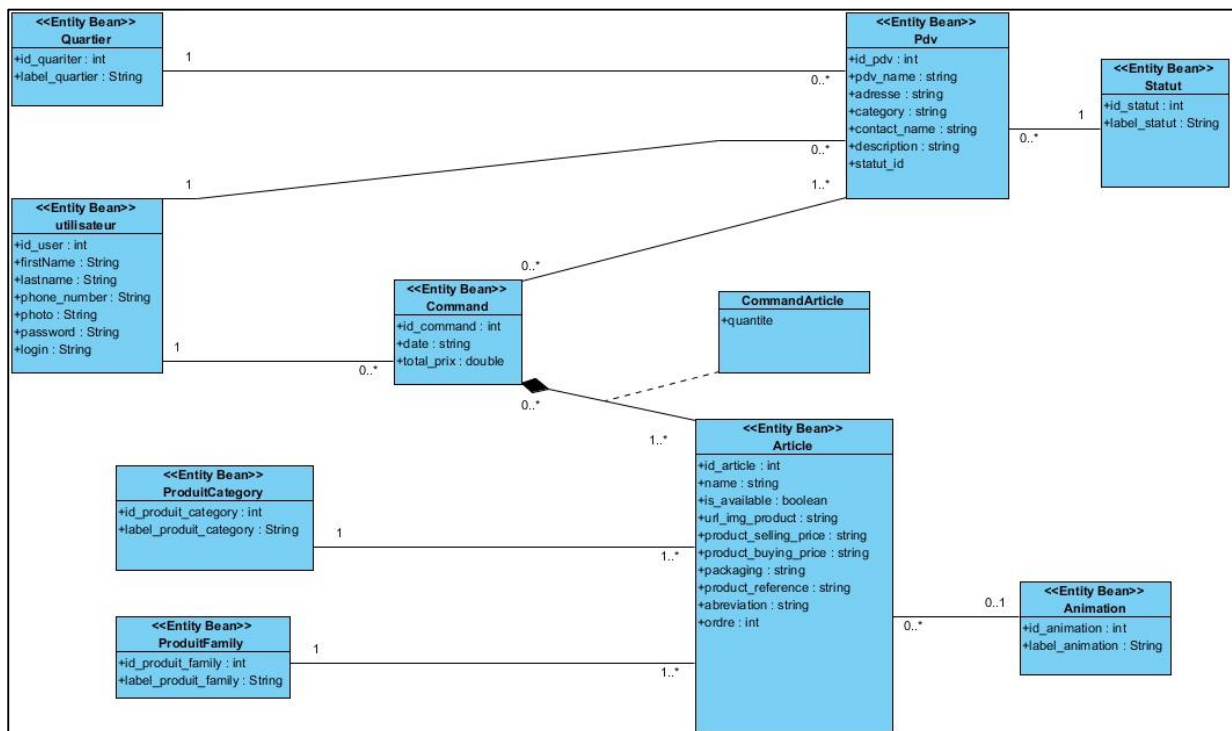


Figure 24 : Diagramme de classe de conception globale.

## 6.5. Diagramme de déploiement

Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. Chaque ressource étant matérialisée par un nœud, le diagramme de déploiement précise comment les composants sont répartis sur les nœuds et quelles sont les connexions entre les composants ou les nœuds. Les diagrammes de déploiement existent sous deux formes : spécification et instance. [1]

La figure 25 montre la représentation d'un nœud et d'une instance de nœud



Figure 25 : Représentation d'un nœud et d'une instance de nœud

Dans notre cas on a ici une application mobile alimenté par un serveur par le billet de web service.

La figure 26 illustre le diagramme de déploiement de l'application



**Figure 26 : Diagramme de déploiement de l'application**

## **PARTIE III : REALISATION**



# Chapitre 7 : Mise en place de l'environnement de développement

## 7.1. Installation des outils

Pour la réalisation de ce projet nous avons eu recours à des outils.

- Java Development Kit (JDK) et Java Runtime Environment (JRE)
- Android Studio
- Le SDK d'Android

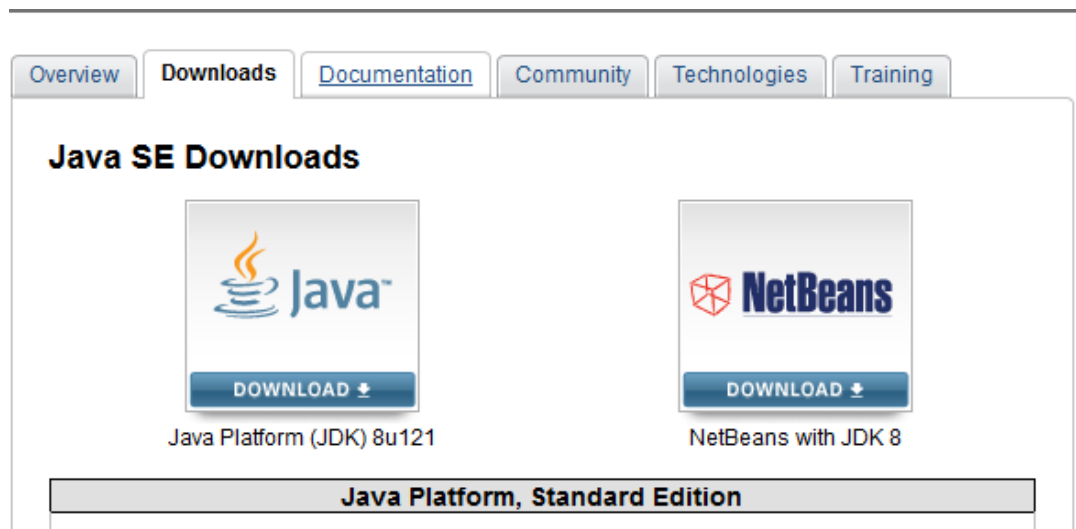
### 7.1.1. Java Development Kit (JDK) et Java Runtime Environment (JRE)

Le Java Development Kit (JDK) désigne un ensemble de bibliothèque logicielle de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peuvent être compilé, transformé en bytecode destiné à la machine virtuelle Java.

Le Java Runtime Environment (JRE) est une famille de logiciels qui permet l'exécution des programmes écrits en langage de programmation Java.

Par défaut, JDK et JRE n'est pas installé sur Windows alors que l'exécution et la compilation du code Java est nécessaire pour le développement de notre application. Il faut donc les télécharger sur le site d'oracle ou directement sur le lien : <http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>.

La Figure 27 représente la capture d'écran du téléchargement de JDK.



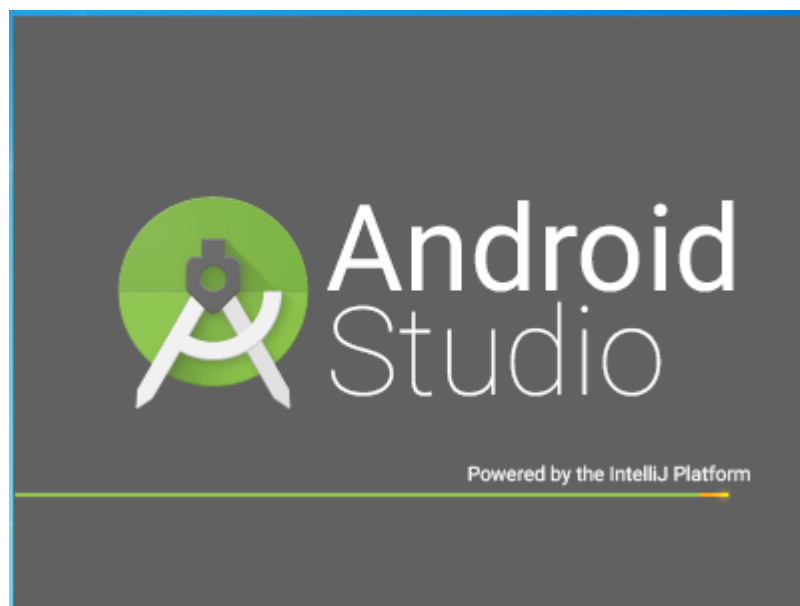
**Figure 27 : Téléchargement de JDK.**

Pour notre projet JDK seul est suffisante du fait qu'il contient déjà le JRE.

### **7.1.2. Android Studio**

Pour le développement de l'application, nous avons eu recours à l'IDE Android Studio qui est l'IDE recommandé par Google pour le développement d'application Android actuellement.

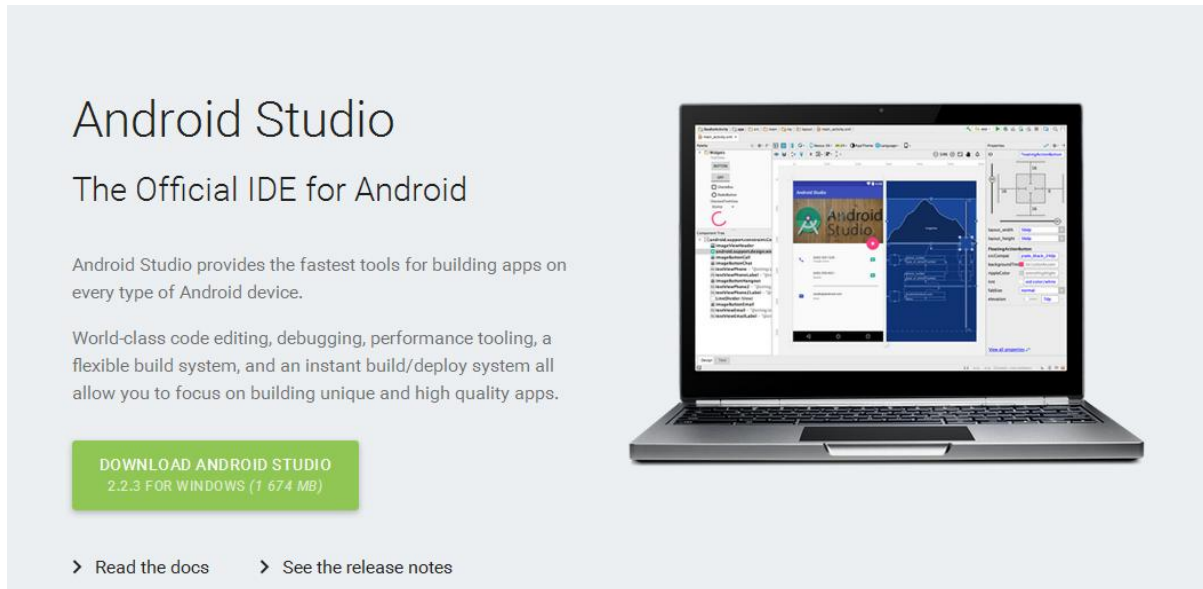
La figure 28 montre l'écran de démarrage d'Android Studio



**Figure 28 : Ecran de démarrage d'Android Studio.**

Pour installer Android Studio, il suffit juste d'aller sur le site officiel <https://developer.android.com/studio/index.html> et de télécharger l'application Android Studio puis suivre les procédures pour ce qui est de l'installation et des mises à jour du SDK.

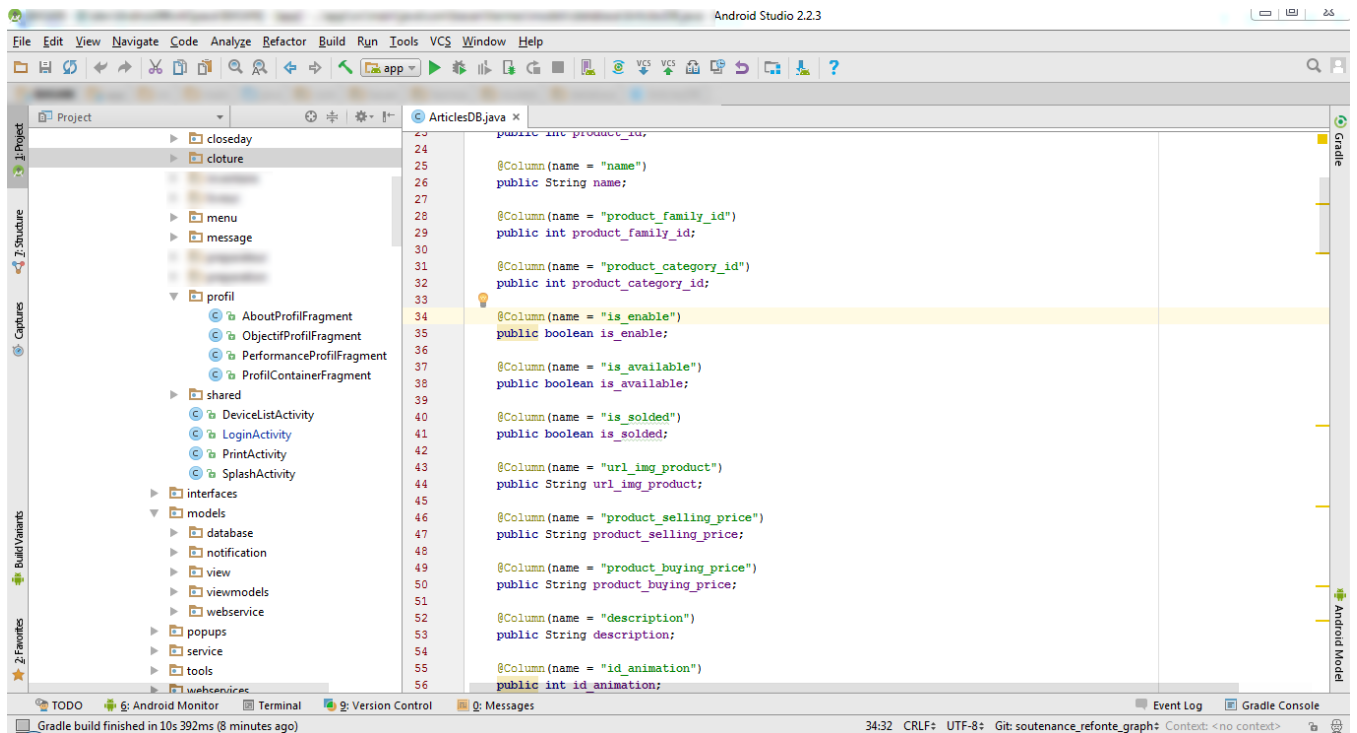
La Figure 29 illustre le téléchargement d'Android Studio.



**Figure 29 : Téléchargement d'Android Studio.**

Pour le développement de l'application, voici l'écran d'Android Studio.

La figure 30 représente l'écran pour le développement avec Android Studio

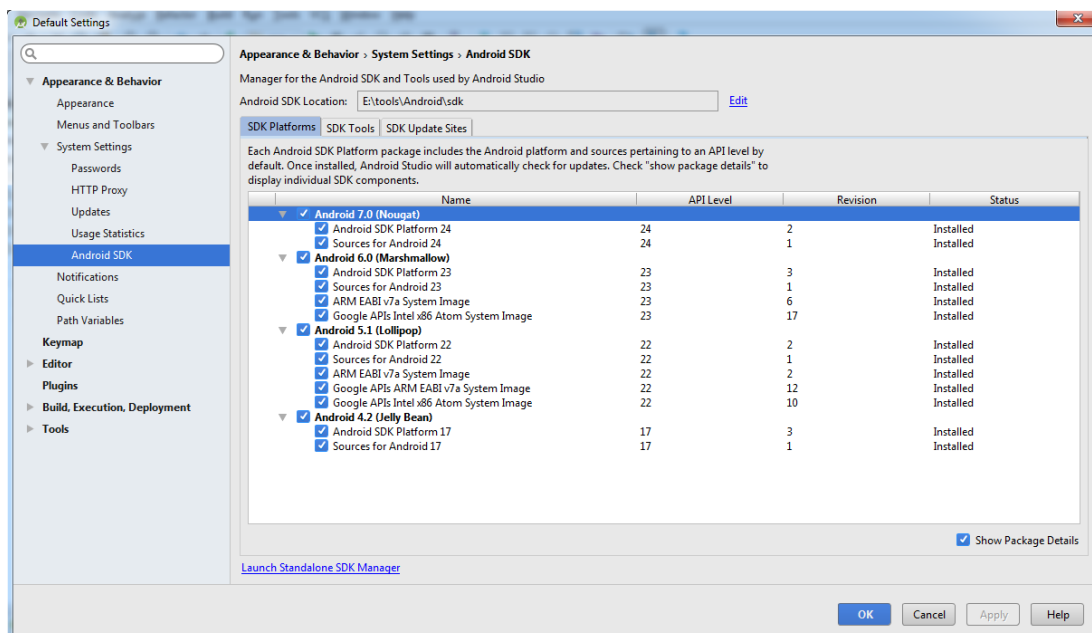


**Figure 30 : Ecran de développement avec Android Studio.**

### 7.1.3. Le SDK d'Android

Software Development Kit (SDK) d'Android est un ensemble complet d'outils de développement.

La figure 31 montre l'interface gestion du SDK avec Android Studio

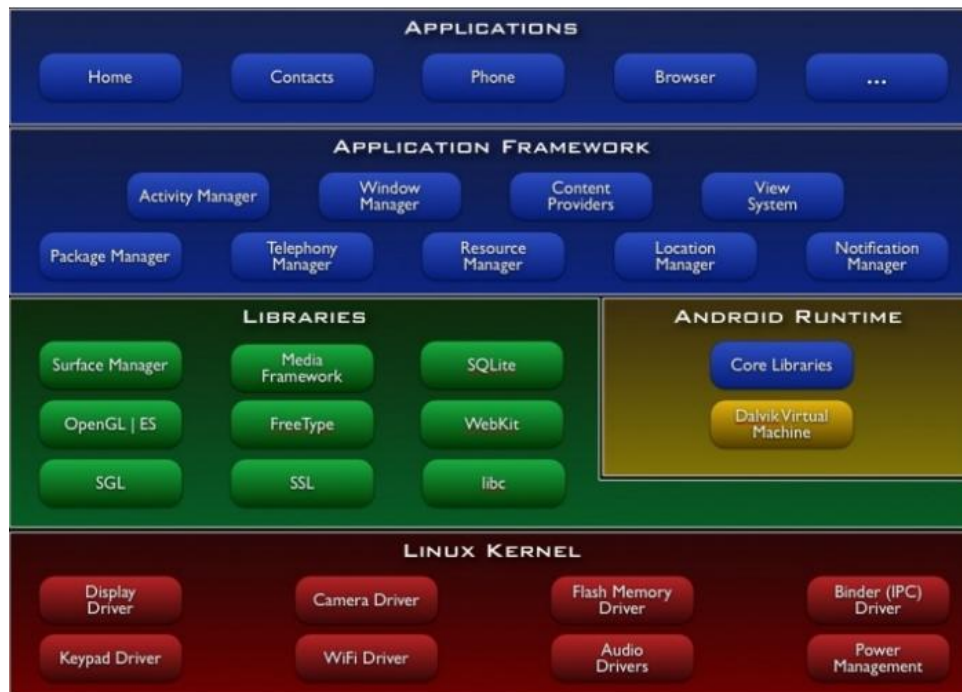


**Figure 31 : Ecran de gestion du SDK avec Android Studio.**

## 7.2. Architecture et système d'exécution d'Android

Comme dit précédemment Android est un système open source utilisant le noyau Linux. Il a été lancé par une startup du même nom rachetée par Google en 2005.

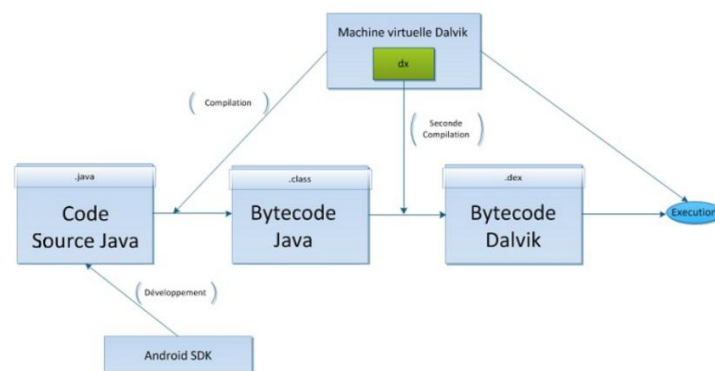
La figure 32 montre l'architecture d'Android.



**Figure 32 : Architecture d'Android**

Pour la compilation d'une application Android on passe par deux compilations successives à commencer par celle du langage JAVA.

La figure 33 illustre le schéma de compilation et d'exécution d'Android.



**Figure 33 : Schéma de compilation et d'exécution d'Android**

# Chapitre 8 : Développement l'application

de

## 8.1. Création de la base de données

Pour la réalisation du projet nous avons utilisé SQLite comme SGBD et Activeandroid comme ORM, de ce fait, la création des tables de la base ce fait donc par la création de modèle d'où l'utilisation de l'architecture MVC pour le développement de l'application.

Voici le code de la création d'une table article qui est un modèle avec le langage JAVA.

```
@Table(name = "ArticlesDB")
public class ArticlesDB extends Model
{
    @Column(name = "product_id", unique = true, onUniqueConflict =
Column.ConflictAction.REPLACE)
    public int article_id;

    @Column(name = "name")
    public String name;

    @Column(name = "product_family_id")
    public int product_family_id;

    @Column(name = "product_category_id")
    public int product_category_id;

    @Column(name = "is_enable")
    public boolean is_enable;

    @Column(name = "is_available")
    public boolean is_available;

    @Column(name = "is_solded")
    public boolean is_solded;

    @Column(name = "url_img_product")
    public String url_img_product;

    @Column(name = "product_selling_price")
    public String product_selling_price;

    @Column(name = "product_buying_price")
    public String product_buying_price;

    @Column(name = "description")
    public String description;

    @Column(name = "id_animation")
```

```

public int id_animation;

@Column(name = "packaging")
public String packaging;

@Column(name = "product_reference")
public String product_reference;

@Column(name = "abbreviation")
public String abbreviation;

@Column(name = "ordre")
public int ordre;

public ArticlesDB()
{
    super();
}

public ArticlesDB(
    int product_id,
    String name,
    int product_family_id,
    int product_category_id,
    boolean is_enable,
    boolean is_available,
    boolean is_solded,
    String url_img_product,
    String product_selling_price,
    String product_buying_price,
    String description,
    int id_animation,
    String packaging,
    String product_reference,
    String abbreviation,
    int ordre
)
{
    super();
    this.product_id = product_id;
    this.name = name;
    this.product_family_id = product_family_id;
    this.product_category_id = product_category_id;
    this.is_enable = is_enable;
    this.is_available = is_available;
    this.is_solded = is_solded;
    this.url_img_product = url_img_product;
    this.product_selling_price = product_selling_price;
    this.product_buying_price = product_buying_price;
    this.description = description;
    this.id_animation = id_animation;
    this.packaging = packaging;
    this.product_reference = product_reference;
    this.abbreviation = abbreviation;
    this.ordre = ordre;
}

/*
 * Recupération de tous les articles disponibles
 * */
public static List<ArticlesDB> getAllProducts()

```

```

{
    return new Select()
        .from(ArticlesDB.class)
        .where("is_enable = 1")
        .orderBy("ordre ASC")
        .execute();
}

/*
 * Recupération des arcticles en promo is_solded = true
 * */
public static List<ArticlesDB> getOffersProduct()
{
    return new Select()
        .from(ArticlesDB.class)
        .where("is_solded = 1 AND is_enable = 1")
        .orderBy("ordre ASC")
        .execute();
}

/*
 * Recupération d'un produit par id
 * */
public static ArticlesDB getProductById(int id)
{
    return new Select()
        .from(ArticlesDB.class)
        .where("product_id = ?", id)
        .executeSingle();
}

/*
 * Efface tous les enregs
 * */
public static void clearProducts()
{
    ActiveAndroid.execSQL("DELETE FROM ArticlesDB;");
}

/*
 * Renvois les articles par filter
 * */
public interface ISearchProductsByFilterListener
{
    void searchProductsByFilterResponse(boolean result,
List<ArticlesDB> content);
}

public static void getArticlesByFilters(SearchProductFilter filtre,
final ISearchProductsByFilterListener listener)
{
    final SearchProductFilter filterObj = filtre;
    final ISearchProductsByFilterListener ecouteur = listener;
    new AsyncTask<Void, Void, List>()
    {
        @Override
        protected List doInBackground(Void... params)
        {
            String queryFilter =
queryFilterCreator(filterObj.getProductName(),
                    filterObj.getIdProductCategory(),
                    filterObj.getIdProductFamily(),

```



```

        filterObj.getProductPackaging());

List<ArticlesDB> listeArticles;

if (!GenericTools.isNullOrEmpty(queryFilter)) {
    listeArticles = new Select()
        .from(ArticlesDB.class)
        .where(queryFilter)
        .orderBy("name ASC")
        .execute();
} else {
    listeArticles = new Select()
        .from(ArticlesDB.class)
        .orderBy("name ASC")
        .execute();
}
return listeArticles;
}

@Override
protected void onPostExecute(List list)
{
    super.onPostExecute(list);
    if (ecouteur != null) {
        ecouteur.searchProductsByFilterResponse(true, list);
    }
}

}.execute();
}

/*
 * Construction des filtres
 */
private static String queryFilterCreator(String product_name, int
product_category_id, int product_family_id, String packaging)
{
    String queryFilter = "is_enable = 1";

    if (product_category_id != 0) {
        if (!GenericTools.isNullOrEmpty(queryFilter)) {
            queryFilter = queryFilter + " AND ";
        }
        queryFilter = "product_category_id =" + product_category_id;
    }
    if (product_family_id != 0) {
        if (!GenericTools.isNullOrEmpty(queryFilter)) {
            queryFilter = queryFilter + " AND ";
        }
        queryFilter = queryFilter + "product_family_id =" +
product_family_id;
    }
    if (!GenericTools.isNullOrEmpty(packaging)) {
        if (!GenericTools.isNullOrEmpty(queryFilter)) {
            queryFilter = queryFilter + " AND ";
        }
        queryFilter = queryFilter + "packaging =" + packaging + "'";
    }
    if (!GenericTools.isNullOrEmpty(product_name)) {
        if (!GenericTools.isNullOrEmpty(queryFilter)) {
            queryFilter = queryFilter + " AND ";
        }
    }
}

```

```

    }
    queryFilter = queryFilter + "name LIKE '%" + product_name +
"%'";
}
return queryFilter;
}
}

```

## 8.2. Codage de l'application

Vu que l'on a utilisé MVC comme architecture de codage de l'application :

Voici un exemple de code XML d'une vue en Android :

```

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

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/gray_background">

    <include layout="@layout/content_loader"
        android:visibility="gone"/>

    <LinearLayout
        android:orientation="vertical" android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/gray_background"
        android:id="@+id/content_about_commande_pdv"
        android:visibility="gone">

        <RelativeLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:visibility="visible"
            android:id="@+id/aboutCommandeContainer">

            <LinearLayout
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:visibility="visible"
                android:id="@+id/containerCommandeList"
                android:background="@color/gray_background"
                android:layout_alignParentLeft="false"
                android:layout_marginLeft="0dp"
                android:layout_alignParentTop="true"
                android:layout_marginTop="0dp"
                android:layout_centerHorizontal="true"
                android:layout_above="@+id/containerAction">

                <RelativeLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="16dp"

```

```

        android:layout_marginBottom="16dp">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="EFFECTUER UNE COMMANDE"
            android:id="@+id/btnEffectuerCommande"
            android:layout_centerHorizontal="true"
            android:background="@drawable/rounded_red_button"
            android:textColor="@color/blue_soutenance"
            android:textAppearance="@style/OpenSansRegular"
            android:textSize="24sp"
            android:padding="15dp" />
    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp"
        android:background="@color/black">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

            android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="COMMANDE(S) PASSEE(S)"
            android:id="@+id/textView49"
            android:textSize="16sp"
            android:textColor="@color/white"
            android:layout_marginLeft="8dp"
            android:layout_marginTop="8dp"
            android:layout_marginBottom="8dp" />
    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"

        android:background="@color/transparent"
        android:layout_marginBottom="10dp">

        <ListView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/listViewAboutCommande"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:layout_marginBottom="10dp"
            android:background="@color/transparent"
            android:dividerHeight="8dp"
            android:divider="@color/transparent"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:layout_marginTop="5dp" />

    </RelativeLayout>
</LinearLayout>

```

```

<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="700dp"
    android:visibility="gone"
    android:id="@+id/containerDeatilCommande"
    android:layout_alignParentLeft="true"
    android:layout_marginLeft="0dp"
    android:layout_alignParentTop="true"
    android:layout_marginTop="0dp">

</FrameLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="84dp"
    android:layout_centerHorizontal="true"
    android:id="@+id/containerAction"
    android:background="@drawable/bg_end_visit"
    android:layout_alignParentLeft="false"
    android:layout_marginLeft="0dp"
    android:layout_alignParentBottom="true"
    android:visibility="gone">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TERMINER LA VISITE"
        android:id="@+id/btnVisiteTerminer"
        android:background="@drawable/rounded_red_button"
        android:textColor="@color/blue_soutenance"
        android:textAppearance="@style/OpenSansRegular"
        android:textSize="20sp"
        android:padding="15dp"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="10dp"
        android:layout_marginTop="10dp"
        android:layout_alignParentBottom="true" />

</RelativeLayout>

</RelativeLayout>

</LinearLayout>

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/container_fragment_about_commande_pdv"
    android:background="@android:color/transparent">

</FrameLayout>

</FrameLayout>

```

Voici en dernier le code Java d'un adapter avec Android.

```

public class PDVListAdapter extends BaseAdapter
{

```

```

private Context context;
private LayoutInflater inflater;
private ArrayList<PointDeVenteObj> pos_list;
private boolean state;

public PDVListAdapter(Context context, ArrayList<PointDeVenteObj>
pos_list)
{
    this.context = context;
    this.pos_list = pos_list;
    inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
}

@Override
public int getCount()
{
    return this.pos_list.size();
}

@Override
public PointDeVenteObj getItem(int position)
{
    return this.pos_list.get(position);
}

@Override
public long getItemId(int position)
{
    return position;
}

@Override
public View getView(final int position, View convertView, ViewGroup
parent)
{
    final ViewPDV holder;
    final PointDeVenteObj PDVItem = this.pos_list.get(position);
    if (PDVItem != null) {
        if (convertView == null) {
            holder = new ViewPDV();
            convertView = inflater.inflate(R.layout.item_pdv_a_visiter,
null);

            holder.itemContainer = (LinearLayout)
convertView.findViewById(R.id.itemContainer);
            holder.containerShape = (RelativeLayout)
convertView.findViewById(R.id.containerShape);
            holder.aboutContainer = (RelativeLayout)
convertView.findViewById(R.id.aboutContainer);
            holder.txtNumPDV = (TextView)
convertView.findViewById(R.id.txtNumPDV);
            holder.txtCategory = (TextView)
convertView.findViewById(R.id.txtCurrentPDVCategory);
            holder.txtQuartier = (TextView)
convertView.findViewById(R.id.txtQuartier);

            holder.labelCategory = (TextView)
convertView.findViewById(R.id.txtProductRef);
            holder.labelQuartier = (TextView)
convertView.findViewById(R.id.txtStockActuel);

```

```

        holder.txtPDVName = (TextView)
convertView.findViewById(R.id.txtCurrentPDVName);
        convertView.setTag(holder);
    } else {
        holder = (ViewPDV) convertView.getTag();
    }

    holder.txtNumPDV.setText("'" + PDVItem.getOrdre());
    if (!GenericTools.isNullOrEmpty(PDVItem.getCategory())) {
        holder.txtCategory.setText(PDVItem.getCategory());
    } else {

holder.txtCategory.setText(context.getResources().getString(R.string.not_de
fined));
    }
    if (PDVItem.getQuartier() != null &&
!GenericTools.isNullOrEmpty(PDVItem.getQuartier().nom_quartier)) {

holder.txtQuartier.setText(PDVItem.getQuartier().nom_quartier);
    } else {

holder.txtQuartier.setText(context.getResources().getString(R.string.not_de
fined));
    }
    if (!GenericTools.isNullOrEmpty(PDVItem.getName())) {
        holder.txtPDVName.setText(PDVItem.getName().toUpperCase());
    } else {

holder.txtPDVName.setText(context.getResources().getString(R.string.not_def
ined));
    }

    // Non selectionné par défaut

holder.containerShape.setBackgroundResource(R.drawable.shape_enable);

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {

holder.txtNumPDV.setTextColor(context.getResources().getColor(R.color.white
, context.getTheme()));

holder.itemContainer.setBackgroundColor(context.getResources().getColor(R.c
olor.white, context.getTheme()));

holder.labelCategory.setTextColor(context.getResources().getColor(R.color.b
lack, context.getTheme()));

holder.txtCategory.setTextColor(context.getResources().getColor(R.color.gra
y, context.getTheme()));

holder.txtQuartier.setTextColor(context.getResources().getColor(R.color.gra
y, context.getTheme()));

holder.labelQuartier.setTextColor(context.getResources().getColor(R.color.b
lack, context.getTheme()));

holder.txtPDVName.setTextColor(context.getResources().getColor(R.color.red,
context.getTheme()));
    } else {

```

```

holder.txtNumPDV.setTextColor(context.getResources().getColor(R.color.white));

holder.itemContainer.setBackgroundColor(context.getResources().getColor(R.color.white));

holder.labelCategory.setTextColor(context.getResources().getColor(R.color.black));

holder.txtCategory.setTextColor(context.getResources().getColor(R.color.gray));

holder.txtQuartier.setTextColor(context.getResources().getColor(R.color.gray));

holder.labelQuartier.setTextColor(context.getResources().getColor(R.color.black));

holder.txtPDVName.setTextColor(context.getResources().getColor(R.color.red));
    }
    }
    return convertView;
}

public static class ViewPDV
{
    public RelativeLayout containerShape;
    public LinearLayout itemContainer;
    public TextView txtNumPDV;
    public RelativeLayout aboutContainer;
    public TextView txtAboutTitle;

    public TextView txtPDVName;
    public TextView labelCategory;
    public TextView txtCategory;
    public TextView txtQuartier;
    public TextView labelQuartier;
}

public void setState(boolean state)
{
    this.state = state;
}
}

```

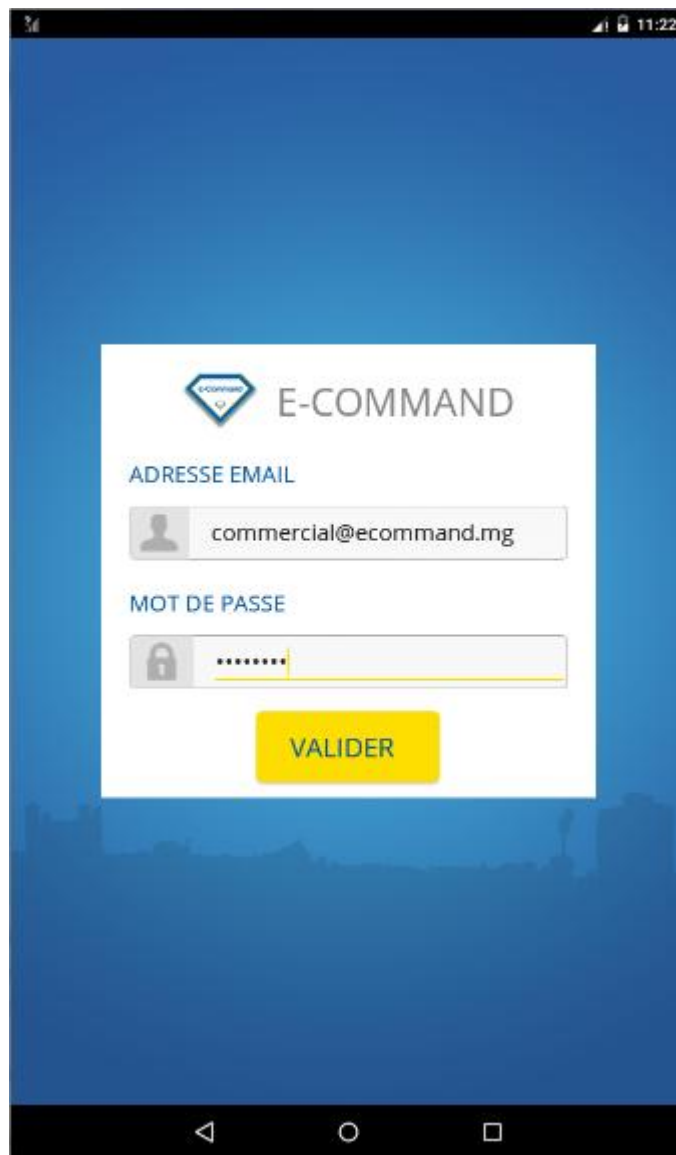
### 8.3. Présentation de l'application

L'application développée consiste à prendre des commandes au sein des épiceries, de ce fait on va le présenter dans ce paragraphe.

➤ L'interface de l'application

➤ L'utilisation de l'application

La figure 34 montre l'écran d'authentification de l'application.

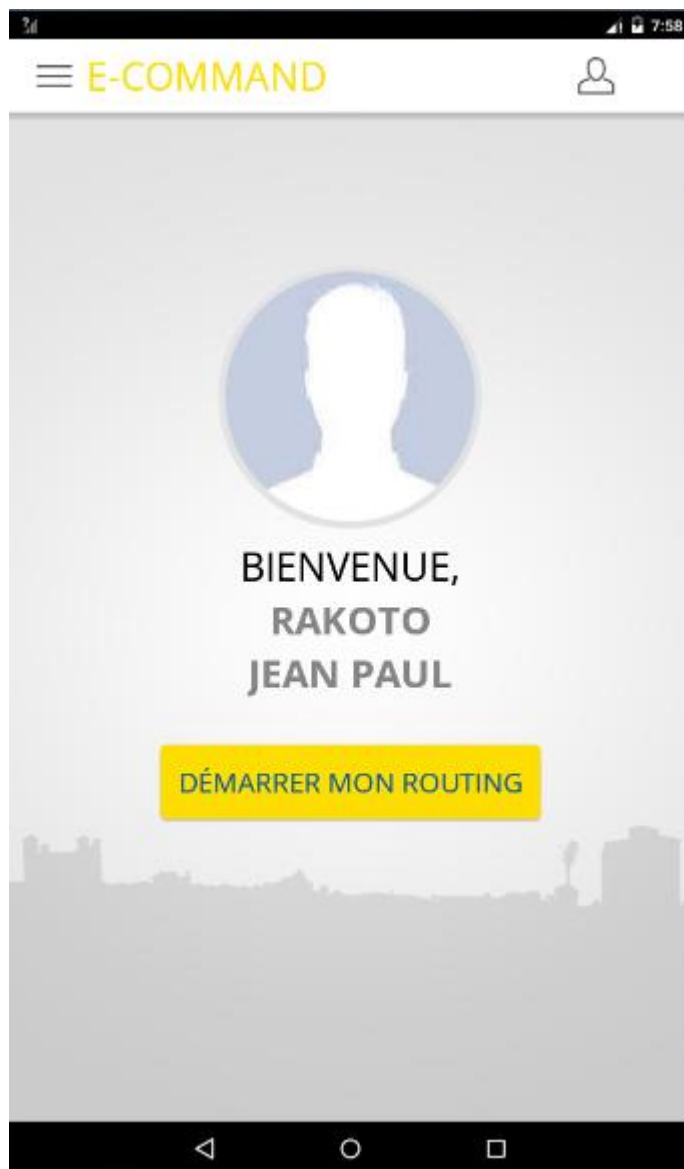


**Figure 34 : Ecran d'authentification de l'application**

L'authentification est essentielle pour la sécurité des données. La connexion doit s'effectuer avec une connexion de donnée.

La figure 35 présente l'écran d'accueil de l'application

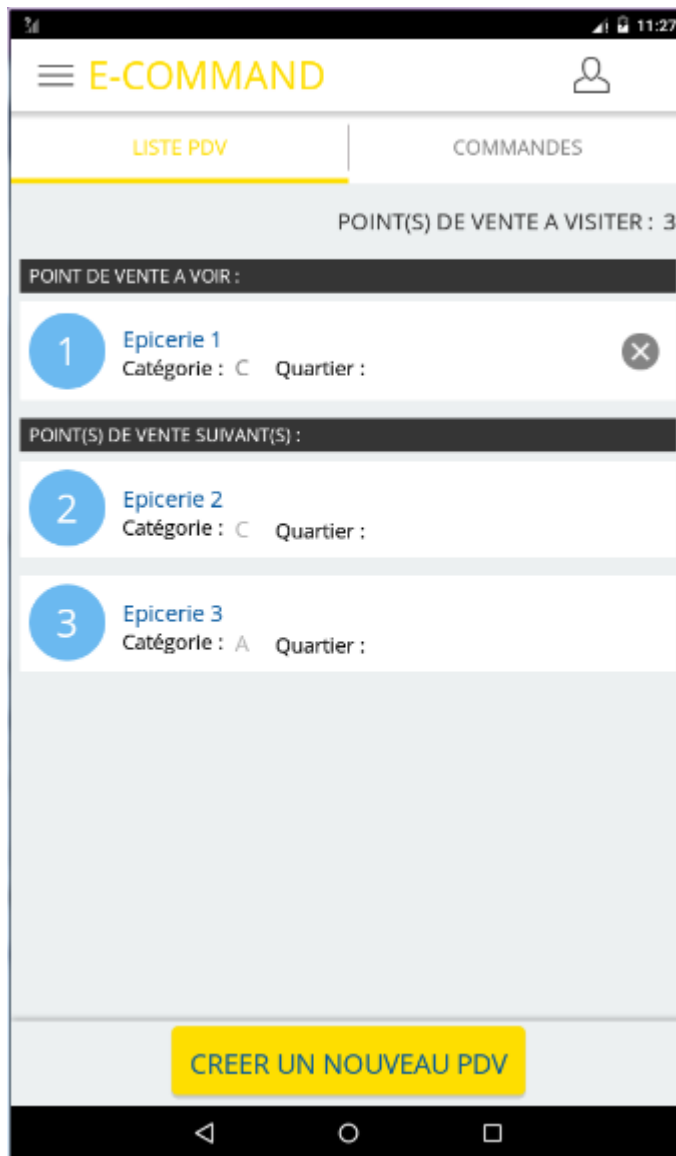




**Figure 35 : Ecran d'accueil de l'application**

Après l'authentification on arrive sur la page d'accueil et en cliquant sur le bouton « DEMARRER MON ROUTING » on accèdera à l'écran de routing.

La figure 36 illustre l'écran de routing de l'application



**Figure 36 : Ecran de routing de l'application**

Après avoir cliqué sur le bouton de démarrage de la routing on arrive sur l'écran de routing.

- La clique sur le point de vente à voir affichera les détails du point de vente et aussi la liste des commandes faites sur le point de vente.
- La clique sur la croix à droite dans point de vente à voir accèdera un dialogue d'annulation d'un point de vente.
- La clique sur le bouton « CREER UN NOUVEAU PDV » affichera l'écran de création d'un point de vente.

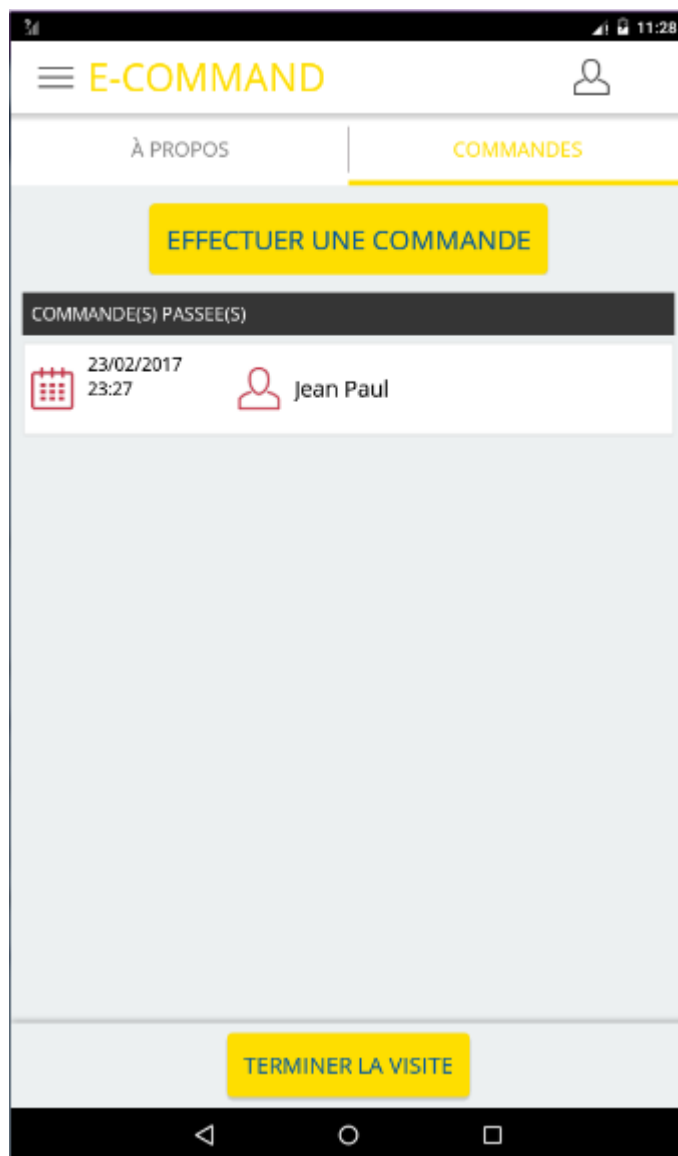
La figure 37 montre la synthèse des commandes



**Figure 37 : Ecran de synthèse des commandes.**

L'écran de synthèse des commandes montre l'évolution de la routing.

La figure 38 représente l'écran de commande de l'application

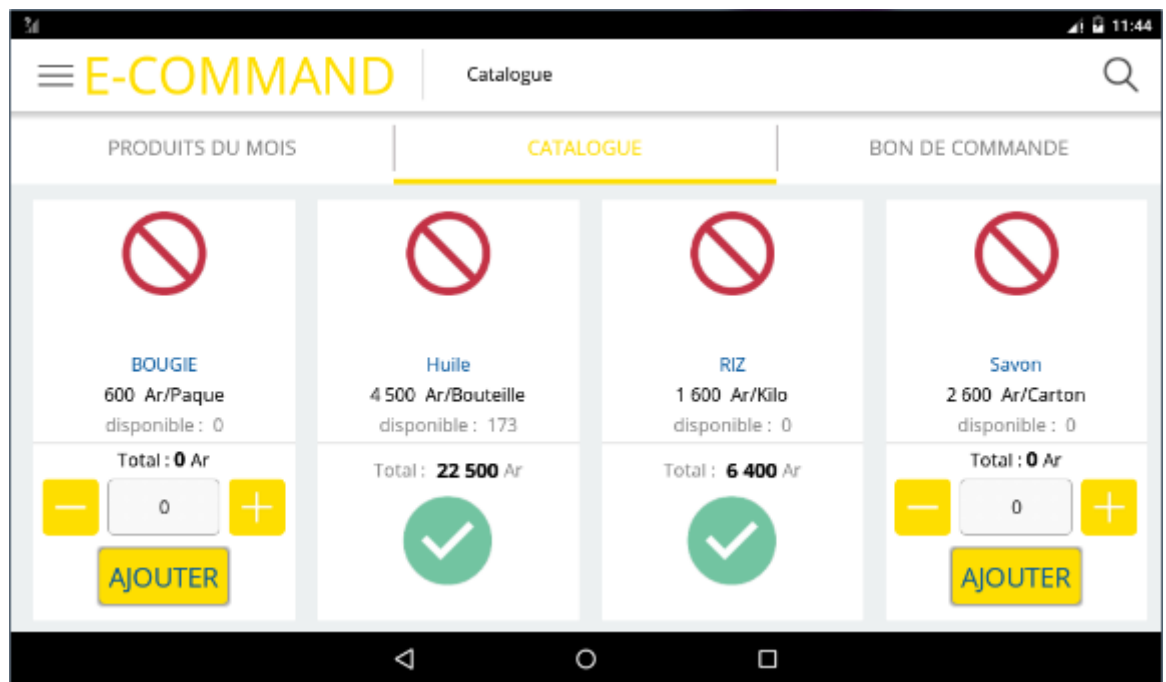


**Figure 38 : Ecran de commande**

L'écran de commande est obtenu après avoir cliqué sur le point de vente à voir. Cet écran présente la liste des commandes effectuées sur le point de vente et aussi la possibilité d'ajouter une commande ou de terminer la visite du point de vente.

- Le bouton « EFFECTUER UNE COMMANDE » permet d'afficher l'écran de prise de commande.
- Le bouton « TERMINER LA VISITE » terminera la visite du point de vente et renvoie sur l'écran de routing.

La figure 39 montre l'écran de liste des produits et quantité commandés.

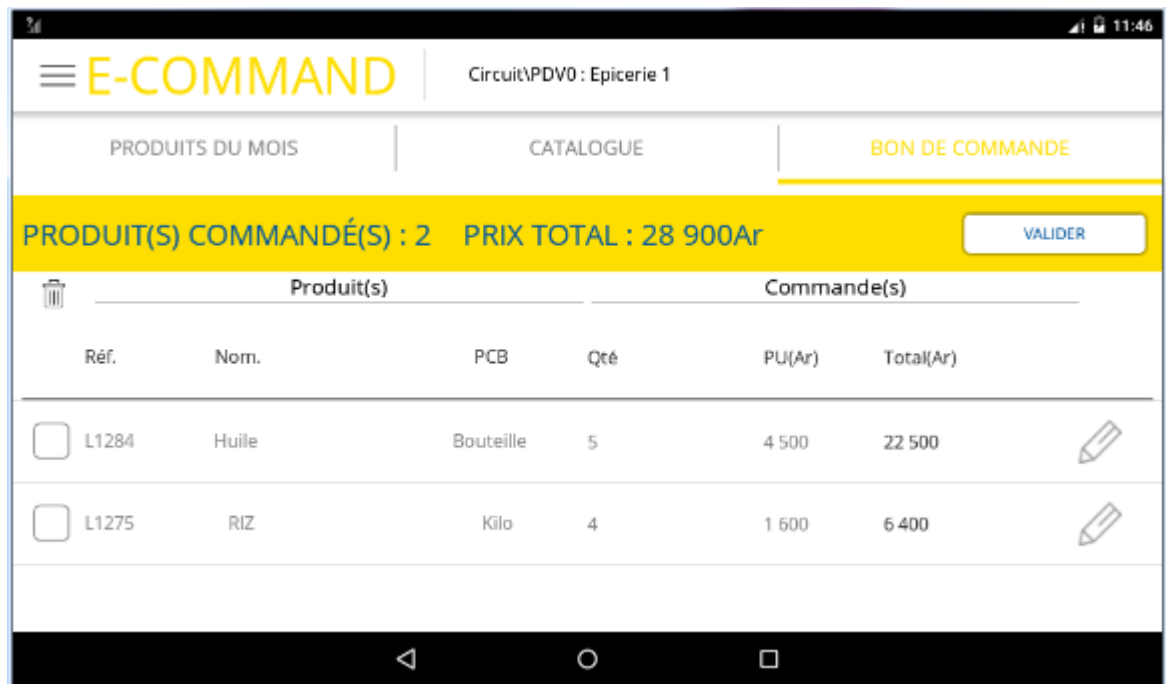


**Figure 39 : Ecran de prise de command**

L'écran de prise de commande est obtenu après avoir cliqué sur « EFFECTUER UNE COMMANDE ». Cet écran affiche la liste des produits qui pourront être commandé par le point de vente. C'est dans cet écran aussi que l'on renseigne la quantité à commander pour chaque produit.

- Le bouton ajouter sert à ajouter le produit dans la liste des produits à commander après avoir renseigner la quantité.

La figure 40 représente l'écran de bon de commande



**Figure 40 : Ecran de bon de commandes**

Cet écran s'affiche après avoir cliqué sur l'onglet bon de commande de l'écran prise de commande. Sur cet écran on peut modifier ou supprimer un produit commandé.

- Le bouton valider enregistrera le bon de commande et enverra directement la commande au client.
- Le bouton en forme de crayon affichera l'écran de modification de la quantité de produit à commander pour un produit choisi.

La figure 41 montre l'écran de modification de la commande d'un produit



**Figure 41 : Ecran de modification de la commande de produit**

Sur cet écran on peut modifier la quantité du produit à commander. Le bouton modifier la commande enregistrera cette modification.

La figure 42 illustre l'écran de création d'un point de vente

The screenshot shows a mobile application interface titled "E-COMMAND". At the top, there is a hamburger menu icon and a user profile icon. Below the title bar, the section "INFORMATIONS GENERALES" is displayed. The form contains the following elements:

- NOM DU POINT DE VENTE**: A text input field labeled "PDV NAME".
- REPERE**: A large text area labeled "Décrivez le point de vente".
- CATEGORIE**: A dropdown menu labeled "Sélectionnez la catégorie du point de vente".
- QUARTIER**: A dropdown menu labeled "Sélectionnez le quartier".

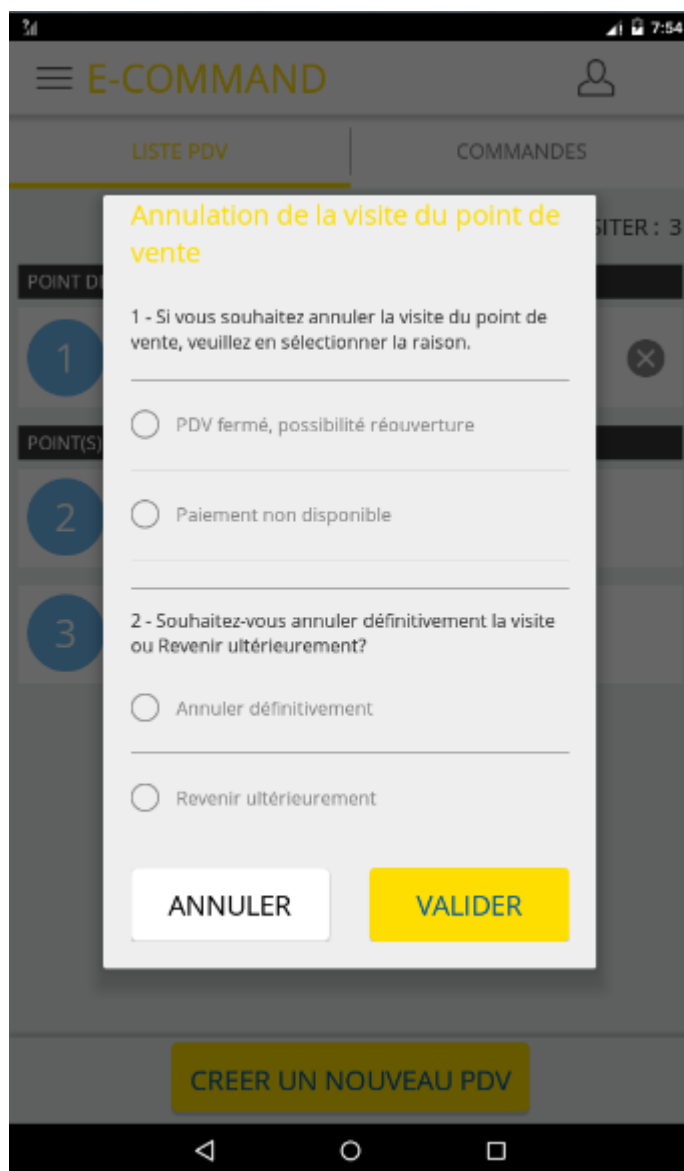
At the bottom of the form, there is a prominent yellow button with the text "CREEZ NOUVEAU PDV". The bottom of the screen shows the standard Android navigation bar.

**Figure 42 : Ecran de création d'un point de vente**

Dans cet écran on renseigne les informations du nouveau point de vente. Le bouton « CREEZ NOUVEAU PDV » enregistrera le nouveau point de vente et enverra les informations sur le point de vente au client.

La figure 43 représente l'écran d'annulation d'un point de vente





**Figure 43 : Ecran d’annulation d’un point de vente.**

Dans cet écran on peut choisir la raison et le type d’annulation d’un point de vente.

# CONCLUSION

La société NETAPSYS MADAGASCAR est une Société de Services en Ingénierie Informatique (SSII) qui occupe une grande partie du marché en termes de vente de logiciel à Madagascar. Actuellement, l'un des projets en cours de NETAPSYS se porte sur la mise en place d'un système d'information pour la prise de commande.

L'objectif de ce projet est de concevoir et réaliser une application mobile de prise de commande est une application Android qui porte le nom d'E-COMMAND.

Pour la réalisation de ce projet, on a utilisé le langage de modélisation UML, la méthode SCRUM, le système d'exploitation Windows, le langage de programmation JAVA, le SGBD SQLite, l'ORM Activeandroid, l'IDE Android Studio et le SDK d'Android.

L'application permet de prendre des commandes auprès des épiceries et est sécurisée par un système d'authentification. Elle peut aussi créer des points de vente, annuler et visiter. Elle possède une base de données embarqué qui stocke les données récupérés sur l'évolution de la routing et elle envoie aussi ces données par webservice à un serveur.

Il est vrai que l'application possède une grande variété de fonctionnalité, il est toujours envisageable d'ajouter plus de fonctionnalité comme l'impression des non de commandes.

Ce stage nous a permis de découvrir l'univers de développement d'une application mobile notamment Android et aussi une grande expérience concernant les méthodes de travail dans le monde professionnel.

## **REFERENCE BIBLIOGRAPHIQUES**

- [1].Laurent Audibert, 2007-2008, UML 2 , A78 pages
- [2].Claude Aubry, 2010-2011, SCRUM Le guide pratique de la méthode agile la plus populaire, 2<sup>ème</sup> édition, 317 pages
- [7].Bertrand LIAUDET, Edition 2015, UML 2 – Diagramme de séquence « système ».9 pages
- [8].DI GALLO Frédéric, 2000-2001, Méthodologie des systèmes d’information – UML, 59 pages

## **REFERENCE WEBOGRAPHIQUES**

- [3].<http://www.esiee.fr/~mahmoudr/engineering/sys/sys04.pdf>, 23 Janvier 2017
- [4].[http://fr.wikipedia.org/wiki/Java\\_%28langage%29](http://fr.wikipedia.org/wiki/Java_%28langage%29) , 23 Janvier 2017
- [5].[http://www.android-dev.fr/presentation\\_de\\_android\\_studio\\_et\\_comparatif\\_avec\\_adt](http://www.android-dev.fr/presentation_de_android_studio_et_comparatif_avec_adt), 25 Janvier 2017
- [6].<https://www.projet-plume.org/ressource/les-services-web-de-type-rest>, 25 Janvier 2017

# GLOSSAIRE

**Android Virtual Device (AVD) :** C'est un émulateur de système Android appartenant à SDK destiné pour les tests. Dans notre cas, on a utilisé Genymotion et des appareils de test réel.

**Base de données :** c'est le terme utiliser pour dire que c'est le lieu de stockage des informations dont l'application a besoin ou bien que l'utilisateur à fait entré dans l'application. Elle est gérée par un Système de Gestion de Base de Données qui est dans notre cas SQLite.

**Commerciale :** désigne la personne qui va prendre les commandes auprès de point de vente, elle peut aussi créer un point de vente.

**MVC :** Modèle View Controller, c'est une architecture utilisé pour le développement d'une application qui sépare les codes sources du modèle qui est relative à la base de données, les view qui est l'interface entre l'utilisateur et l'application et les controllers qui prend en compte les actions faites par l'utilisateur et aussi les fonctionnalité de l'application.

**ORM :** Object Relational Mapping, technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle en définissant des correspondances entre cette base de données et les objets du langage utilisé. En d'autre terme, les tables de la base de donnée sont gérées à partir des modèles qui sont des class de dans le code source de l'application.

**Point de vente :** désigne les entités que les commerciaux vont visiter et y prendre des commandes, plus précisément les vendeurs détaillants.

**Release :** c'est le mot désignant la livraison d'une partie de l'application. C'est aussi le temps définit par plusieurs sprint pour effectuer une livraison auprès du client.

**Routing :** Mot désignant le chemin que le commercial doit faire. Il présente la liste des points de vente suivant l'ordre de visite.

**SCRUM :** qui veut dire « mêlée » en français comme lors d'un jeu de rugby, et en ce qui concerne l'informatique est une méthode de conduite de projet basé sur la méthode AGILE.

**Software Development Kit (SDK) :** ensemble d'outils permettant aux développeurs de créer des applications de type défini comme le développement d'application iOS ou Android. Dans

ce rapport nous utilisons le SDK d'Android, ce qui veut dire que les outils utilisés dans notre cas sont destinés pour Android.

**Sprint** : c'est le mot désignant le temps que l'on définit à l'avance pour avoir une application livrable dans la méthode SCRUM.

**UML** : Unified Modeling Language, c'est une langage de modélisation qui sert à modéliser les besoins de l'utilisateur en diagramme pour faciliter le développement de l'application.

# TABLE DES MATIERES

<b>CURRICULUM VITAE.....</b>	<b>I</b>
<b>REMERCIEMENT.....</b>	<b>IV</b>
<b>LISTE DES FIGURES .....</b>	<b>V</b>
<b>LISTE DES TABLEAUX.....</b>	<b>VII</b>
<b>LISTE DES ABREVIATIONS.....</b>	<b>VIII</b>
<b>SOMMAIRE.....</b>	<b>XI</b>
<b>INTRODUCTION GENERALE .....</b>	<b>1</b>
<b>PARTIE I : PRESENTATION .....</b>	<b>2</b>
<i>Chapitre 1 :           Présentation de l'école nationale d'informatique.....</i>	<i>3</i>
1.1.           Informations d'ordre général .....	3
1.2.           Missions et historique .....	3
1.3.           Organigramme institutionnel de l'ENI .....	5
1.4.           Domaines de spécialisation.....	7
1.5.           Architecture des formations pédagogiques .....	8
1.6.           RELATIONS DE L'ENI AVEC LES ENTREPRISES ET LES	
ORGANISMES.....	11
1.7.           PARTENARIAT AU NIVEAU INTERNATIONAL .....	12
1.8.           DEBOUCHES PROFESSIONNELS DES DIPLÔMES .....	14
1.9.           RESSOURCES HUMAINES .....	16
<i>Chapitre 2 :           Présentation de NETAPSYS .....</i>	<i>18</i>
2.1.           Présentation du Groupe Netapsys .....	18
2.2.           Netapsys Madagascar.....	19
2.3.           Organigramme .....	21
2.4.           Les valeurs du Groupe .....	21
2.5.           Technologies et outils maîtrisés .....	22
2.6.           Industrialisation des savoir-faire.....	22
2.7.           Partage et diffusion des connaissances .....	23
2.8.           Outils de pilotage projets .....	23
2.9.           Références Clients.....	24

<i>Chapitre 3 :</i>	<i>Description du projet.....</i>	<i>27</i>
3.1.	Formulation.....	27
3.2.	Objectif et besoin de l'utilisateur.....	27
3.3.	Moyens nécessaires pour la réalisation du projet.....	28
3.4.	Résultats attendus.....	28
PARTIE II : ANALYSE ET CONCEPTION.....		29
<i>Chapitre 4 :</i>	<i>Analyse préalable .....</i>	<i>30</i>
4.1.	Analyse de l'existant.....	30
1.1.1.	Organisation actuelle.....	30
4.1.1.	Inventaire des moyens matériels et logiciels.....	30
4.2.	Critique de l'existant.....	31
4.3.	Conception avant-projet.....	31
4.3.1.	Proposition des solutions .....	31
4.3.2.	Outils utilisés.....	32
4.3.2.1	UML.....	32
a.	Présentation d'UML.....	32
4.3.2.2	La méthode AGILE notamment SCRUM .....	33
a.	Définition .....	33
b.	Principes .....	34
c.	Vocabulaires de SCRUM.....	34
4.3.2.3	Le système d'exploitation Windows .....	35
4.3.2.4	Le langage JAVA .....	35
4.3.2.5	Le SGBD SQLite.....	36
4.3.2.6	L'ORM Activeandroid .....	36
4.3.2.7	L'IDE Android Studio .....	37
4.3.2.8	GIT.....	39
4.3.2.9	Web service REST Client.....	39
<i>Chapitre 5 :</i>	<i>Analyse conceptuelle .....</i>	<i>41</i>
5.1.	Spécification des besoins .....	41
5.1.1.	Règles de fonctionnalité.....	41
5.1.2.	Dictionnaire des données .....	41
5.1.3.	Diagramme des cas d'utilisations .....	43
5.1.3.1	Formalisme .....	44
5.1.3.2	Relations entre cas d'utilisation.....	45
a.	Relation d'inclusion.....	45
b.	Relation d'extension.....	45
c.	Relation de généralisation .....	45

5.1.4.	Cas d'utilisation de l'application .....	46
5.1.5.	Priorisation des cas d'utilisations.....	47
5.2.	Modélisation du domaine.....	48
5.2.1.	Diagramme des séquences systèmes.....	48
<i>Chapitre 6 :</i>	<i>Conception détaillée .....</i>	<i>51</i>
6.1.	Architecture du système MVC.....	51
6.2.	Diagramme de séquence de conception pour chaque cas d'utilisation	52
6.3.	Diagramme de classe de conception pour chaque cas d'utilisation .....	57
6.3.1.	Formalisme.....	57
6.3.1.1	Les classes .....	57
a.	Définition .....	57
6.3.1.2	Les associations.....	58
a.	Multiplicité des associations.....	58
b.	Agrégation .....	59
c.	Composition .....	59
d.	Généralisation .....	60
6.4.	Diagramme de classe de conception globale .....	63
6.5.	Diagramme de déploiement .....	63
<b>PARTIE III : REALISATION .....</b>		<b>65</b>
<i>Chapitre 7 :</i>	<i>Mise en place de l'environnement de développement .....</i>	<i>66</i>
7.1.	Installation des outils .....	66
7.1.1.	Java Development Kit (JDK) et Java Runtime Environment (JRE)	66
7.1.2.	Android Studio.....	67
7.1.3.	Le SDK d'Android.....	69
7.2.	Architecture et système d'exécution d'Android .....	70
<i>Chapitre 8 :</i>	<i>Développement de l'application.....</i>	<i>71</i>
8.1.	Création de la base de données .....	71
8.2.	Codage de l'application .....	75
8.3.	Présentation de l'application .....	80
<b>CONCLUSION .....</b>		<b>X</b>
<b>REFERENCE BIBLIOGRAPHIQUES.....</b>		<b>XI</b>
<b>REFERENCE WEBOGRAPHIQUES.....</b>		<b>XI</b>
<b>GLOSSAIRE .....</b>		<b>XII</b>



<b>TABLE DES MATIERES.....</b>	<b>XIV</b>
<b>RESUME .....</b>	<b>XVIII</b>
<b>ABSTRACT.....</b>	<b>XVIII</b>

# RESUME

Le stage effectué au sein de NETAPSYS MADAGASCAR a porté sur la conception et la réalisation d'une application de prise de commande mobile pour une société client de Netapsys. L'objectif principal était de leur fournir une application mobile Android qui permettra de prendre des commandes auprès des points de vente.

Grâce à cette application, le processus de vente du client sera amélioré et leurs employé pourront alors finir leur travail en même temps vu que les commerciaux utilisant l'application envoie en temps réel les commandes via un web service.

Pour atteindre ces objectifs, on a développé l'application avec le langage de programmation JAVA avec l'IDE Android studio utilisant une base de donnée SQLite gérée par un ORM Activeandroid avec un web service REST, le langage de modélisation UML et la méthode SCRUM.

**Mots clés :** , Android, commande, commerciaux, points de vente, SCRUM, web service

---

# ABSTRACT

The internship at the "NETAPSYS MADAGASCAR" focused on conception and release a mobile application of taking a command for a Netapsys's customer. The main objective was to give them Android application with possibility of taking command to a grocery store.

With this application, the sell process will be optimized and their employee could end their day together because the commercial who used the application send in real time the commands by the service web.

To achieve these objectives, we used JAVA programming language with Android Studio IDE using SQLite Databases managed with Activeandroid ORM with REST web server, UML modeling language et SCRUM method.

**Keywords:** Android, command, commercial, grocery store, SCRUM, web server