## INTRODUCTION GENERALE

### CHOIX ET INTERET DU SUJET

L’avènement de nouvelles technologies de l’information et de la communication (NTIC) font de l’informatique la révolution la plus importante et innovante qui marque l’univers télématique moderne. Le fait de chercher à échanger le plus grand nombre d'informations dans un délai plus court et parfois à des distances plus grandes constitue un grand défi que les inventeurs et les chercheurs se sont décidés de relever en faisant progresser des techniques de la communication et de l'information : de l'invention de l'écriture à la création des réseaux organisés de télécommunication ;

En effet, l’assention des applications informatique propose des solutions imposantes, incontournables et précisent aux différent problèmes de gestion dans les domaines professionnels que pour les applications personnelles, pour ceux le souci qui nous anime à aborder ce sujet se justifie par le fait que nous constatons dans la plus part des installations aéroportuaires gérer par la régie des voies aériennes l’absence d’un logiciel adéquat des gestion des informations du trafic aérien vus la sensibilités que ces dernières présente pour la RVA Katanga dans le domaines de traçabilités de déplacement des appareils ainsi que des passages sur le ciel congolais tous en facilitant un suivit de facturation de redevances que tous client utilisant que tous client devait payé après avoir utilisé les l’espaces aériens ainsi que tous les services qui vont de pair avec.

### ETAT DE LA QUESTION

L’état de la question est définie comme un produit documentaire établissant le bilan critique des travaux effectués sur un sujet donné pendant une période déterminée et pouvant se présenter sous forme écrite ou orale. Il définit le sujet dans le temps et dans l'espace, en précise les acteurs et les différents aspects (politiques, économiques, juridiques, etc.), les sources et ressources d'information. Il s'appuie sur une importante bibliographie de la littérature du domaine.[[1]](#footnote-1)

Le progrès scientifique est cumulatif n’est pas l’œuvre d’une personne mais de plusieurs chercheurs qui révisent, critiquent, actualisent et élargissent, afin de mener l’originalité de l’œuvre scientifique en évidence une synthèse de critiques des écrits existant nous permet de tracer une ligne de démarcation entre les différent travaux afin de nous situer par rapport à eux.

Des travaux consultés, nous citons :

* CHIKURU MUGOSHO ALAIN « conception et réalisation d’une application de gestion des passagers sur tous les vols nationaux et internationaux ».

Il analyse les problèmes liés aux informations des passagers sur le vol et conclus en proposant une application web(site) affichant toutes les information de ces derniers en le classant dans une base des données comme de carte d’identité avec image du passager mais en ne spécifiant pas la destination de ce dernier et ou les passagers devais se faire capturé pour que ses informations se retrouve à la régies des voies aériennes du Katanga.

* KABUNDA KASANSA PREMIS « conception d’une application de vente de billet dans une compagnies aviation cas de korongo Airlines ».

Le chercheur s’est investi dans la recherche des informations liée à la vente des billets sans faire état de manifeste il s’est beaucoup plus intéressé a l’analyse du système d’information de korongo sans tenir compte de relation existant entre la compagnie et la régies des voies aériennes qui sont appelé à être toujours en étroite collaboration sur le plan de services réciproque que ces entreprises. Il a conclu pour dire que le problème de la gestion des ventes des billets serait résolu en concevant une application de gestion programmé sous les scriptsVB.NET et une base de données Access fonctionnant en en mode monoposte.

En ce qui concerne notre travail, nous avons choisi de faire comme examen de notre recherche un sujet qui va aborder sur « la conception et la réalisation d’une application de gestion intégrée du trafic Aériens » cas de l’aéroport International de Lubumbashi

En effet notre démarcation par rapport a nos prédécesseurs ci –haut se situe au niveau ou nous parlons de la conception et réalisation d’une application de gestion intégrée capable de fournir toutes les informations des gestion des services dans un aéroport de type international, entre autre les informations du client a l’administration, les calculs de redevances sur bases des informations fournies par les compagnies clientes , ainsi que la facturation, en même temps faire les statistiques des vols, des passagers par compagnie, des mouvements par avion ainsi que du fret

Cette application sera déployée en réseau sous une architecture client-serveur via un serveur Dns et sous une architecture n/3 sous un intranet qui devrait interconnecter toute les compagnies clientes en vue de partager toutes les informations y afférant.

### PROBLEMES DE GESTION (PROBLEMATIQUES)

Dans une recherche active, le chercheur ne choisit pas les problèmes à résoudre l'histoire les lui impose mais il crée sa problématique, c'est-à-dire que pour résoudre un problème donné il choisit un certain nombre de critères et élabore à partir de ceux-ci son système de recherche [[2]](#footnote-2)

L’aéroport international de Lubumbashi est un aéroport le plus fréquenté de la république démocratique du Congo vu que son trafic aérien croit chaque année.

De ce fait le trafic aérien est depuis longtemps un secteur globalement en forte croissance et qui nécessite une bonne gestion des informations vu que ce dernier occupe une place stratégique dans croissance comique d’un pays. Face à un trafic qui double tous les dix ans, les services chargés du contrôle et de la régulation de la circulation aérienne ainsi que de la gestion des passagers sont dû trouver des solutions pour faire face à cette augmentation en ce qui concerne le stockage des informations ainsi que leurs restitutions quand on en a besoin et au moment voulut.

On constate toutefois depuis quelques années des difficultés grandissantes à accroître de façon significative la capacité des systèmes de gestion du trafic aérien actuellement à l’aéroport international de Lubumbashi qui est incapable de fournir les information de manière précise sur les passagers et la compagnie en détail, vu que ces information se traite sur un fichier incapable de remplir le norme d’une application de gestion du trafic aériens. l’accroissement exponentiel des retards quand a la recherche de mouvement des avions ainsi que des passagers pris à bord fait défaut du faits que le manifeste de passager ne sont pas bien conserver ou n’arrive pas dans les mains des gestionnaires du trafic aériens vu que le système d’information actuelle soit incapable d’enregistrer ces manifeste du point de vu leur rubrique.

Vus le manque d’archivage de donnée dont le volume croit chaque jour ainsi que le formulaire de trafic qui a ce jours se font concevez dans des malles métallique entreposé à la merci des intempéries et qui d’une recherche d’un mouvement ancien un casse-tête a l’agent commis.

Partant aussi du manque d’une communication direct entre l’aéroport et ses client qui sont les compagnie d’aviation via une application déployer en réseaux pouvant facilité une collaboration direct entre les deux entité précité et diminuer sensiblement le travail de comptage de passage qui se manuellement par pointage causant un retard sur le décollage des avions.

Pour arriver à pallier à ces différents problèmes et à automatiser le suivi des passagers, nous préconisons la conception d’une application de gestion intégrée du trafic aérien. Pour y parvenir nous nous posons les questions suivantes :

* Comment pouvons-nous arriver à modéliser le suivi des passagers sur tous les vols au niveau de la RVA/Katanga ?

### AMELIORATIONS ATTENDUES (HYPOTHESES)

Une hypothèse est définie comme une proposition ou une explication que l’on se contente d’énoncé sans prendre position sur son caractère véridique c'est-à-dire sans l’affirmer ou la nier[[3]](#footnote-3)

Etant donné que la problématique a déjà été posée, il est important d'y assimiler quelques réponses provisoires, les quelles réponses pourront être confirmées ou infirmées après l’analyse et la conception du nouveau système d’information.

Ainsi, vu notre sujet, nous pouvons envisager de donner les proposions et possibilité suivantes :

* Utiliser le langage UML avec la méthodologie UP pour bien modéliser le suivi des passagers sur tous les vols.

### DELIMITATIONS DU SUJET

Tout travail scientifique pour être précis doit être limité. En ce qui concerne notre objet de recherche, nous allons aborder des questions lies à la conception d’une application de gestion intégrée du trafic aérien

Ainsi nous allons délimiter spécialement notre recherche a la régie des voies aériennes de Lubumbashi e précisément à l’aéroport international de la luano a la division commercial au bureau de vérification du trafic aérien.

### METHODES ET TECHNIQUES

### METHODES

La méthode scientifique désigne l’ensemble des canons guidant ou devant guidé le processus de production des connaissances scientifique qu’il s’agisse d’observation, d’expériences, de raisonnement ou de calcul théorique.[[4]](#footnote-4)

Dans le cadre de notre recherche nous allons utiliser le langage de model unifier(UML) associer à la démarche UP ( processus unifier)

1. **LE LANGAGE UML**

Le langage uml est un langage de model unifier UML hérite principalement des méthodes objets de Booch(Booch), OMT (Rumbaugh) et OOSE (Jacobson)[[5]](#footnote-5)

Ce langage intègre également d'autres approches, comme les machines à états de Harel. Ayant comme But initial de définir un processus/méthode de développement complet (de l'analyse à l'implémentation) orienté objet.

* Problème
* Pas de notation, langage pour écrire les modèles ou les artefacts définis par ce processus devenu le but final d'UML
* UML n'est donc pas une méthode ou un processus
* UML propose un ensemble de notations pour que chacun ait à sa disposition les éléments nécessaires à la conception d'une application Normalisé par l'OMG (Object Management Group ) qui a comme caractéristique :
* Notation standard pour la modélisation

D’applications à base d'objets (et de composants depuis la version 2)

* Mais utilisable dans de nombreux autres contextes de conception ou spécification
* Langage utilisant une notation graphique

1. **LA METOHDE UP**

Définie comme processus unifier **UP (Unified Process)** est une méthode générique de développement de logiciel[[6]](#footnote-6).

Générique signifie qu'il est nécessaire d'adapter UP au contexte du projet, de l'équipe, du domaine et/ou de l'organisation (*exemple: R.UP ou X.UP*). C'est, entre parenthèses, plus ou moins vrai pour toute méthode, qu'elle se définisse elle-même comme générique ou pas. Mener pour transformer les besoins d’un utilisateur en système logiciel.

Caractéristiques essentielles du processus unifié :

* Le processus unifié est à base de composants,
* Le processus unifié utilise le langage UML (ensemble d'outils et de diagramme),
* Le processus unifié est piloté par les cas d’utilisation,
* Centré sur l’architecture,
* Itératif et incrémental.

### TECHNIQUES

C’est ainsi, nous avons recouru à deux types des techniques pour la réalisation de ce présent travail à savoir : la technique documentaire et la technique d’interview.

### L’interview

Elle consiste à un processus de communication verbale entre l’interviewé et l’interviewer, processus au cours duquel le premier tente d’obtenir du second des informations dont il a besoin pour l’élaboration d’un travail scientifique Cette technique nous permettra d’entrer en contact avec les responsables des différents services à la Direction des Recettes du Katanga pour avoir des informations à notre préoccupation.

### La documentation

Celle-ci permet à consulter quelques documents entre autres nous citons : les ouvrages, revues, articles, notes de cours ayant trait au sujet étude, autres archives relatifs au sujet sous examen.

### La technique d’observation

Elle nous a permis de descendre dans les installations aéroportuaire de la régies des voies aériennes de Lubumbashi en vue d’y effectuer une observation directe de ce qui se passa sur terrain et comprendre ce qui se fait et comment cela se fait.

### SUBDIVISION DU TRAVAIL

Notre travail pote sur « la conception et la réalisation d’une application intégrée des gestions du trafic aériens. «  Cas de la luano ».

Pour avoir une idée générale de notre travail nous nous donnons un aperçu de ce que nous allons faire en trois chapitres or mis l’introduction et la conclusion

* Le chapitre premier sera intitulé « cadrage théorique » dans ce chapitre il sera question d’aborder sur toutes les théories nécessaire de l’analyse, de la conception ainsi que du trafic aériens que nous allons utiliser du début jusqu’ à la fin de notre travail scientifique
* Le deuxième chapitre se nomme « description du milieu d’étude » ce chapitre portera sur l’étude de l’environnement dans lequel nous allons implémenter notre application en étudiant délicatement le système existant pour en déduire les failles
* Le troisième chapitre portera sur « conception du système informatique » qui va nous conduire à modéliser le système d’information future à mettre en place
* Le quatrième chapitre sera « implémentation et déploiement » ce chapitre fera l’objet de la matérialisation de notre projet ou il sera question de faire le choix des différentes technologies logiciel et matérielles à utiliser pour le déploiement de note application et à présenter quelque capture de l’interface de l’application.

## CHAPITRE I : CADRAGE THEORIQUE

## **I.1. BALISAGE CONCEPTUEL**

## **I.1.1 CONCEPTION**

## **I.1.1.1. Présentation du langage de modélisation UML**

UML (Unifiedprocess) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue[[7]](#footnote-7).

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, de communiquer les divers aspects d'un système d'information (aux graphiques sont bien sûr associés des textes qui expliquent leur contenu). UML est donc un métalangage car il fournit les éléments permettant de construire le modèle qui, lui, sera le langage du projet.

UML 2 s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ces types de diagrammes sont répartis en deux grands groupes

* **SIX DIAGRAMMES STRUCTURELS :**
* **Diagramme de classe** : Les diagrammes de classes expriment de manière générale la structure statique d’un système, en termes de classes et de relations entre ces classes.[[8]](#footnote-8)
* **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[[9]](#footnote-9) objets. On peut donc dire :
* un objet est une instance de classe,
* un lien est une instance de relation

Le diagramme de classe est un modèle permettant de décrire de manière abstraite et générale les liens entre objets.

* **Diagramme d'objets**: il montre les instances des éléments structurels et leurs liens à l'exécution.
* **Diagramme de paquetage** : ces diagramme qui regroupe des éléments de la modélisation appelés aussi membre, portant sur un sous ensemble du système. Le découpage en paquetage doit traduire du découpage logique du système à construire qui correspond a des espace de nommage homogène.
* **Diagramme de structure composite** : il montre l'organisation interne d'un élément statique complexe.
* **Diagramme de composants**: il montre des structures complexes, avec leurs interfaces fournies et requises.[[10]](#footnote-10)
* **Diagramme de déploiement** : le diagramme de déploiement permet de représenter l’architecture physique supportant l’exploitation du système. Cette architecture comprend de nœuds correspondant aux supports physiques ainsi que la répartition des artéfacts logiciels sur ces nœuds [[11]](#footnote-11)
* **SEPT DIAGRAMMES COMPORTEMENTAUX :**
* **Diagramme de cas d’utilisation** : u diagramme de ca d’utilisation permet de décrire l’interaction entre les acteurs et le système. La description de l’interaction est réalisée suivant le point de vue de l’utilisateur. La présentation met en jeu trois concepts : l’acteur, le système et l’interaction entre l’acteur et le cas d’utilisation[[12]](#footnote-12)
* **Diagramme de vue d'ensemble des interactions**: il fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.[[13]](#footnote-13)
* **Diagramme de séquence** : est un diagramme qui a pour mission de représenter l’interaction entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser pas cas d’utilisation en considérant les différents scenarios associés un diagramme de séquence se représente dans un grand rectangle avec indication du nom du diagramme en haut[[14]](#footnote-14)
* **Diagramme de temps** : il fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps.[[15]](#footnote-15)
* **Diagramme d’activité** : est un diagramme qui représente un certain nombre des points communs avec le diagramme d’état puisqu’il concerne le comportement interne des opérations ou des cas d’utilisation. Ce pendant le comportement visé ici s’applique au flot de contrôle et flot des données propres a un ensemble d’activité et plus relativement a une seule classe[[16]](#footnote-16)
* **Diagramme d'états** : il montre les différents états et transitions possibles des objets d'une classe.[[17]](#footnote-17)

## **I.2. LA DEMARCHE UP (Unified Process)**

Processus unifié (PU ou UP en anglais pour **UnifiedProcess**) est une méthode de développement pour les logiciels orientés objets. C’est une méthode générique, itérative et incrémentale, contrairement à la méthode séquentielle [Merise](http://fr.wikipedia.org/wiki/Merise_(informatique)) (ou [SADT](http://fr.wikipedia.org/wiki/Analyse_fonctionnelle_descendante)).

PU vient compléter la systémique des modèles [UML](http://fr.wikipedia.org/wiki/Unified_Modeling_Language). Elle est le résultat final d’une évolution de l’approche d’Ericsson qui est au fondement d’une des premières méthodes de développement pour applications orientées objets, la méthode ObjectoryProcess (1987). ObjectoryProcess (version 1 à 3.8 en 1995) a elle-même servi de base à la société Rational pour la création de Rational ObjectoryProcess (1997) (version 4.1), parente direct de RUP en 1998. PU : Processus unifié, désigne les préceptes généraux de la méthode.

## LES ABREVIATIONS UTILISER

* UP : UnifiedProcess, la dénomination anglaise.
* USDP : *Unified Software DevelopmentProcess, autre dénomination courante.*
* [*RUP*](http://en.wikipedia.org/wiki/RUP) : Rational UnifiedProcess, Instanciation par Rational Software (IBM) des préceptes UP.
* EUP : Enterprise UnifiedProcess, Instanciation intégrant les phases de post-implantation et décrivant le cycle de vie du logiciel.
* XUP : ExtremeUnifiedProcess, Instanciation hybride intégrant UP avec *[ExtremeProgramming](http://fr.wikipedia.org/wiki/Extreme_Programming" \o "Extreme Programming)*.
* AUP : Agile UnifiedProcess, partie des préceptes UP permettant l’agilité du développement, instanciation partielle de la méthode mettant l’accent sur l’optimisation et l’efficacité sur le terrain plus que sur le modèle théorique.
* [2TUP](http://fr.wikipedia.org/wiki/2TUP) : TwoTracksUnifiedProcess, Instanciation de UP proposé par [Valtech](http://fr.wikipedia.org/wiki/Valtech" \o "Valtech) prenant en compte les aléas et contraintes liées aux changements perpétuels et rapides des SI des entreprises.
* AM : Agile Methods, méthode Agile

Ainsi, une réalisation conforme à PU, pour transformer les besoins des utilisateurs en logiciel, doit nécessairement présenter les caractéristiques suivantes :

* PU est à base de composants
* PU utilise le langage UML
* PU est piloté par les [cas d’utilisation](http://fr.wikipedia.org/wiki/Cas_d%27utilisation)
* PU est centré sur l’architecture
* PU est itératif et incrémental

## FONCTIONNEMENT DE LA MATHODE UP

Le pilotage par les [diagrammes de cas d'utilisation](http://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation) (*DCU*) revêt une signification concrète : des DCU, on doit tirer les modèles d’analyse, de conception, de déploiement. Ce sont eux qui sont implantés et ce sont les cas d’utilisation prévus qui vont présider à l’élaboration des lignes de tests : les cas d’utilisation doivent finalement être permis par le nouveau logiciel. Les DCU sont les modèles garantissant la cohérence du processus du développement. Ce sont eux qui unifient. Enfin les DCU sont, de par leur nature, intrinsèquement liés à l’architecture du système.

Celle-ci est conçue dès le départ de façon très pragmatique : elle est adaptée au contexte de travail, aux besoins de l’utilisateur, aux possibilités de réutilisation (*re-use*) de bibliothèques ou de « briques » préexistantes.

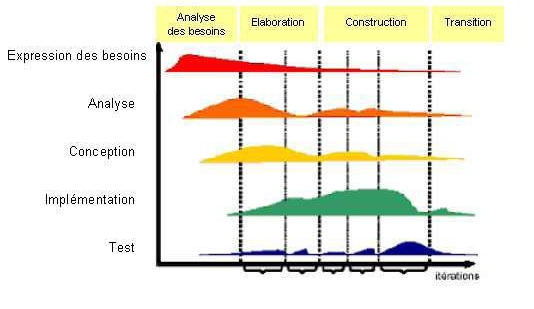
L’élaboration de l’architecture est d’abord grossière et indépendante des cas d’utilisation (on veillera cependant à ce qu’elle n’empêche pas leur réalisation) puis, un sous-ensemble des fonctions essentielles est identifié et l’architecture est reprise et détaillée suivant cet ensemble. De la spécification à la précision des cas, l’architecture évolue, incluant finalement de nouveaux cas, ainsi de suite, jusqu’à ce que l’architecture ait atteint un niveau de développement suffisamment élevé et stable pour donner lieu au développement d’un prototype qui sera présenté au client achevant ainsi une itération.

Une itération est la succession des étapes d’une activité. Un incrément est une avancée dans les stades de développement. À chaque itération on retrouve les phases de spécification des besoins, de conception, jusqu’au prototypage exécutable. Une nouvelle itération, par exemple après démonstration du prototype aux utilisateurs, reprendra la spécification en la précisant ou la corrigeant, puis reprenant l’élaboration, etc.

Les incréments sont définis par le projet, et chaque incrément va ajouter de nouvelles fonctionnalités. Les incréments peuvent suivre les différents cas d’utilisation par exemple. La difficulté résidera dans le fait de combiner finalement les sous-projets ou incréments ensemble et de respecter leurs interdépendances et la cohérence générale du produit envisagé. C’est donc également un développement sous forme de composants qui est proposé. Il utilisera au mieux les apports des technologies objets.

PU intègre les deux visées dans le but de minimiser les risques de contre-sens par rapport aux besoins ainsi que le risque de développements infinis, indéfinis, mal définis ou inachevés : Ici l’utilisateur peut corriger lui-même, sur les prototypes, la tournure que prend le développement. Dès le début, des résultats tangibles sont obtenus même s’ils ne sont que prototypiques. Certaines implémentations de PU considèrent d’ailleurs les prototypes comme des versions à part entière du système final. Les fonctions subalternes peuvent très bien, dans une telle optique, être abandonnées en cours de route pour des questions de coûts ou de délais par exemple. Enfin, si les besoins utilisateurs changent en cours de développement, cette évolution peut être, dans une certaine mesure, intégrée. Ce n’est pas le cas dans le cadre d’un développement séquentiel.

PU prévoit globalement un cycle de vie où les itérations (spécifications + analyse + conception + implémentation + tests)



sont regroupées en catégories d’activités. Ces activités sont soit initiales (création), soit intermédiaires (élaboration, construction) soit finales (transition vers l’utilisateur ou mise en production). Ces catégories d’activité découpent la réalisation du produit comme une succession temporelle (séquences) mais comprennent toutes les tâches structurantes du projet (raffinage successifs, itérations) et proposent une organisation matricielle du volume d’activité total fourni : il est évident qu’en phase de création, une plus grande partie du temps sera consacrée à l’analyse des besoins qu’aux tests ; inversement, en phase de transition, les tests sont encore en cours alors que l’analyse des besoins et son raffinage sont théoriquement bouclés depuis la phase de construction.

EUP (Entreprise PU) ajoute des catégories d’activités décrivant la vie du logiciel en production jusqu’à son retrait de la production.

## UP COMME METHODE AGILE

Les méthodes dites [agiles](http://fr.wikipedia.org/wiki/Methode_agile) (AM) décrivent des processus de développement d’application basés sur la modélisation, la conception et la documentation réalisés de façon efficace. Les pratiques de modélisation agiles sont en fait des améliorations (Best practices) censées pouvoir être appliquées à des méthodes déjà existantes, déjà instanciées. Ainsi AUP (agile modelingunifiedprocess) doit être considéré comme un sous-ensemble de Rational UP. Or l’emboîtement des pratiques agiles avec les concepts UP n’est pas évident :[[18]](#footnote-18)

Souvent l’adoption de PU par une organisation découle en fait de la volonté de discipliner les pratiques de développement à l’utilisation d’outils particuliers et au suivi de règles de conduites établies, homogènes. Ils constituent eux-mêmes des traitements dans les directions informatiques des entreprises et font l’objet d’ingénierie (BPR : [Business ProcessReengineering](http://fr.wikipedia.org/wiki/R%C3%A9ing%C3%A9nierie_des_processus_d%27affaires)). Le développement agile, au contraire, préconise le choix des outils les plus simples et l’utilisation en douceur ou « sur le mode de la boîte à outils » des modèles du langage ou des phases de la méthode. Il y a donc paradoxe à vouloir rigidifier en les codifiant des pratiques par nature destinées à la souplesse.

Pour autant, la plupart des concepts agiles sont implantés, décrits, dans PU sous la forme de mécanismes de développement :

La participation active telle que promue par AM est facilitée par le développement itératif et incrémental. L’utilisateur peut théoriquement intervenir au bon moment et annihiler toute erreur d’interprétation.

La modélisation en parallèle est préconisée par PU comme par AM : en effet, si la « sérialisation » telle qu’elle peut être induite par le découpage en activité organise la matrice des tâches, il n’en reste pas moins qu’à chaque itération les différents types de modèles peuvent être effectués en même temps.

AM préconise une formalisation des zones de contacts entre le projet et l’existant ou système en place. RUP prévoit l’étude de « classes frontières » servant d’interface avec le SI existant tel qu’il demeurera après l’implantation du projet.

La modélisation dans chaque incrément est conçue par PU comme étant le résultat de raffinages successifs.

Favoriser la réutilisation de codes, de classes : Plus encore que PU ce sont les langages objets et la conception en classe qui permettent cela. Par contre, selon Scott Ambler [[19]](#footnote-19)certains fondements de PU ne peuvent coexister avec les préceptes d’agilité : La prééminence et le rôle moteur des DCU doit être abandonné car s’ils permettent de documenter correctement les comportements du logiciel ils ne pourront pas servir à piloter quelque autre activité du projet que ce soit : les contraintes, les interfaces utilisateurs et leur cinématique, les [règles métier](http://fr.wikipedia.org/wiki/R%C3%A8gles_m%C3%A9tier) que devront respecter le logiciel ne peuvent être déduites des DCU. PU ajoute d’ailleurs un ensemble de « spécifications supplémentaires » pour pallier ce manque. Ainsi, toujours selon Ambler, si l’analyse des besoins peut conduire le projet, les cas d’utilisations ne le peuvent pas et ne constituent qu’une rhétorique marketing propre aux instanciations telles que RUP ou EUP.

En second lieu, la méthode de développement en incrément et itération, si elle est assimilée par le chef de projet, ne l’est pas forcément des développeurs et encore moins des utilisateurs qui peuvent y être associés. Ces concepts ne sont pas simples à appréhender et à implanter dans une gestion de projet. Cela nécessite une démarche active de la part de celui qui décide de mener le projet selon ces préceptes de façon que la démarche soit effectivement itérative et incrémentale. Autant de temps consacré à la « métaphysique » du projet et de sa gestion qui vont à l’encontre de l’optimisation agile.

* **Classe** : Description d’un ensemble d’objets partageant la même sémantique, ainsi que les mêmes attributs, opérations et relations[[20]](#footnote-20).
* **Opération**: La définition d’une classe est complétée par l’ensemble des opérations qu’elle peut exécuter[[21]](#footnote-21).

Une opération est une fonctionnalité assurée par la classe.

Le niveau de détail à retenir pour décrire les opérations est fonction du niveau d’avancement de l’étude

* **Association** : Une association représente une relation structurelle entre classes d’objets. La plupart des[[22]](#footnote-22)

Associations sont binaires, c’est à dire qu’elles connectent deux classes. On représente une association en traçant une ligne entre les classes associées.

## **I.3. THEORIE SUR L’APPLICATION**

* **Sql** : structued query langage (langage structuré des requête)
* **Html**: hyper texmark uplangage
* **Css:** cascading steel sheet
* **Serveur apache**: est un serveur web
* **http**: hypertext transfert protocol
* **dns**: domaine Name service
* **note pad+++** : éditeur de texte tout ce qui est relatif a l’aviation civil et cela englobe le transport des passages ainsi des marchandises par avion d’un aéroport a vers un aéroport[[23]](#footnote-23).

1. **Concepts**

* **IFR** : (Instrument Flight Rouler)
* **AD** (adulte) : désigne les passagers adultes
* **CH** (enfants) : désigne les passagers enfants
* **INF(**Infant) : désigne les bébés
* **Logiciel** : ensemble des programme, procédé et règles éventuellement la documentation relatif au fonctionnement de trainement de données[[24]](#footnote-24)
* **Gestion**: action de gérer
* **Application**: tout programme informatique capable de répondre aux problèmes de gestion
* **Agl** : atelier de génie logiciel désigne toute plate forme au quel on fait recourt pour analyse ou implémenter un système d’information
* **Informatisé**: traitement organisé par les méthodes et les moyens de l’informatique.
* **Système d’information :** Un système d’information (SI) qui est un moyen humain et technique nécessaire aux stockages et au traitement des informations d’une organisation
* **objet :** est une représentation artificielle de ce que l’on pense avoir compris d’une réalité.
* **Programme :** logiciel applicatif, définie comme une suite hiérarchisé d’instruction dont le but et résoudre de problèmes d’une manière automatique
* **Application :** est un logiciel conçu pour aider utilisateur a exécuter une tache spécifique, telle que les traitements de texte, comptabilité ou la gestion des stocks.
* **Banque des données :** ensemble des données relatif a un domaine défini des connaissances et organisée pour être offert aux consultations d’utilisateurs.
* **base des données :** est un ensemble d’informations organisées stocker sur un support amovible et capable d’être manipulées par un SGBD
* **SGBD :** système de gestion des bases des données
* **SGBDR :** ensemble des fonctions permettant de définir la structure des données, de formuler des requêtes, de modifier et des stocker les informations  « système de gestion de bases des données relationnelles
* **Table :** est une entité contenant des informations hiérarchisées dépendant d’un identifiant (clé primaire)

## **I.4. TRAFIC AERIEN**

Le trafic aérien désigne

* **FL** : fytlevel (niveau de vol)
* **PMAD** : poids maximum autorisé au décollage
* **MTOW** : poids de l’avion l’atterrissage
* **STA**: stationnement
* **TD**: heure du décollage
* **TA**: heure d’atterrissage
* **Balisage** : système d’éclairage destiné a éclairé la piste pour donner une bonne visibilité de la piste
* **Fret**: marchandise pris a bord d’un avion destiné a l’importation ou a l’exportation
* **VFR** : (Visual Flight Roul)
* **Mouvement** : ensembles des informations sur un avion quittant ou provenant d’un autre aéroport[[25]](#footnote-25)
* **Pax**: passagers a bord d’un avion
* **Formulaires de trafic**: un document sur quel son repris toutes les informations sur un avion en détails servant de pro formant.
* **Handling**: l’escalier sur le quelle le passagers montent pour accéder dans l’avion ou descendre de l’avion
* **Tarmac**: lieu sur le quel les avions embarquent ou débarquent leurs passagers
* **Passagers en transit**: c’est l’ensemble des passagers qui restent dans l’avion destinés à débarquer a une autre destination
* **Route**: chemin aérien que doit emprunter tous les avions vers une destination précise

1. **Redevance aéronautique**

La redevance aéronautique est celle qui est purement liée a la facturation des avions après avoir utilise les services de la régie de des voies aériennes

La redevance aéronautique est constitué de :

* La redevance d’atterrissage : La redevance d’atterrissage correspond a l’usage des infrastructures et équipement aéroportuaires nécessaire a l’atterrissage ou au décollage et a la circulation au sol d’un aéronef[[26]](#footnote-26)

La redevance est due pour tout aéronef qui effectue un atterrissage et est calculé d’après le PMAD de l’aéronef

* La redevance d’atterrissage international

La redevance balisage lumineux : Cette redevance s4applique aux aérodromes équipés d’une installation des balisages lumineux

* La redevance stationnement : La redevance stationnement avion est due par tout aéronef stationnant sur les aires de trafic (poste stationnement sur le tarmac) ou dans les hagards de maintenance (mis a la disposition par la régie des voies aériennes) en zone technique ou entretient.
* Les redevances passagères : La redevance passagère est due pour l’utilisation des installations aménagées pour l’embarquement et l’accueil des passagers et du public

La redevance est due pour chaque vol au départ et calculer sur base du nombre des passagers embarquant à bord de l’aéronef[[27]](#footnote-27)

* La redevance fret : La redevance fret est due pour l’utilisation des installations aménagées pour l’embarquement ou débarquement du fret[[28]](#footnote-28)

La redevance est due pour chaque vol au départ et a l’arrivée est calculée sur la base du fret embarqué au départ exprimé en kilogramme

* La redevance sécurité : La redevance sécurité vise de manière générale à contribuer à améliorer les conditions de sécurité sur les plates formes[[29]](#footnote-29)

Les sommes perçues aident de financer les investissements en matière de sécurité

La redevance s’applique aux aéronefs ayant 20 places assises et plus, sur les vols domestiques au départ

* La redevance sureté : La redevance sureté vise de manière générale à contribuer ou à améliorer les conditions de sureté sur les plates-formes

Les sommes perçues aident à financer les investissements en matière de sureté

* **La redevance route** : La redevance de route est prévue pour l’usage des aides et services en route fournis par la régie des voies aériennes en espace supérieur (>FL245) et en espace inferieur à (<FL245) quelque soit le point d’entré ou le point de sortie dans l’espace aérien congolais[[30]](#footnote-30)

Les vols en provenance ou à destination des aéroports non gérer par la régie des voies aériennes ne sont pas ne sont pas exonérés du paiement de la redevance route

* **La redevance anti incendie** : Cette redevance est instituée aux aéroports ou ces services sont rendus pour la sécurité des aéronefs[[31]](#footnote-31)

L’assistance anti-incendie est requise sur la demande express de la compagnie aérienne lors du ravitaillement en carburant contre un payement de 50USD, sur tout trafic confondu

## **CONCLUSION**

Nous voici arrivé a la fin de notre premier chapitre qui porté sur le cadrage théorique, dans ce chapitre il a été question de faire un cadrage sur tous les concepts liés au développement de notre projet jusqu’ a sa matérialisation, pour bien définir ces différents concepts nous l’avons repartie en cinq sous bien conformément a notre sujet, entre autre : le balisage conceptuel, la conception, la théorie sur l’application, le trafic aérien.

## CHAPITRE II DESCRITION DU MILIEU D’ETUDE

## II.1. APERCU HISTORIQUE

## II.1.1. BREVE RADIOSCOPIE DE LA REGIE DES VOIES AERIENNE DE LUBUMBASHI(RVA)

La régie des voies aérienne est une entreprise publique à caractère technique et commerciale placée sous tutelle du ministère des transports et communications et elle fut créée par l’ordonnance loi NO 72 – 002 du 21 février 1972. Elle est régie par la loi NO 72 – 200 du 05 mai 1978 telle que modifiée en ce jour.

Elle dirigé par un commandant (yankee delta) secondé par un vice commandant

## II.2. L’AEROPORT INTERNATIONAL DE LUBUMBASHI/LUANO

## II.2.1. HISTORIQUE DE L’AEROPORT INTERNATIONAL DE LA LUANO

Les travaux de construction de l’aéroport de la luano ont commencé au début de l’année 1954 le 02/11/1959, le piste, le tarmac, furent officiellement inaugurés et immédiatement livrés a l’exploitation. Mais le bâtiment de l’aérogare n’a été officiellement inauguré que le 02/09/1958 par le gouverneur de la province du Katanga Monsieur Jean PEANCLINK

Les raisons fondamentales qui ont justifie la construction de cet aéroport international de la luano sont les suivantes :

* L’agrandissement rapide de la ville et la peur que le premier aéroport TSHOMBE ne soit englouti dans la ville.
* La longueur du premier aéroport (1800m de piste) ainsi que l’état général des infrastructures (piste, bâtiments) ne permettaient plus sa bonne exploitation par rapport au nouveau qui répondent à toutes les exigences avec ses 3200m de piste sur 50m et 90cm d’épaisseur. Cette piste est capable de supportés des avions de plus de 300 tonnes tels que la DC 10.
* Des nouvelles exigences économiques d’exploitation du transport aérien.

Voila autant des raisons qui ont milité à la construction de cet aéroport. De l’importance de son trafic et sa vocation à accueillir les vols internationaux en partance ou en provenance de l’étranger, il est classé dans la catégorie « 1 » parce que c’est un aéroport douanier d’entrée et sortie, aussi un point stratégique d’importance militaire et économique. Vu son importance sur le plan national, luano est donc un point sensible, en particulier pour la ville de Lubumbashi et en général la province qui donc bénéficier de la protection dévolue aux installations classifiées de l’Etat.

## II.2.2. SITUATION GEOGRAPHIQUE

La Régie des Voies Aérienne/Katanga à son siège de travail dans les installations de l’aéroport international de Lubumbashi. Celle-ci est situé au nord-est de la ville à 8km en vol d’oiseau et a plus au moins 13km de la route pour quelqu’un qui désirerait s’y rendre à partir du centre de la ville, il doit emprunter la route qui mène vers Likasi et prendre la droite après le tunnel. Il se trouve dans le quartier luano de la commune annexe qui est une localité au nord-est de la ville de Lubumbashi.

#### 

## II.2.3. PRESENTATION DE L’ORGANIGRAMME GENERAL DE LA RVA

## II.2.3.1. ORGANIGRAMME ET STRUCTURE GENERALE DE L’AEROPORTINTERNATIONAL DE LUBUMBASHI

*Figure1 : organigramme General de la régie des voies Aérienne[[32]](#footnote-32)*

COMMANDANT D’AEROPORT YD

COMMANDANT D’AEROPORT ADJOINT YA

SECRETARIAT

Staff Coordination

Division CA

Division

Administra & Finance

Division RADIO

Division

Moyens Généraux

Division Surfal

Division Commerciale

Bureau Statistique

Bureau Telecom-Aéronautique

Bureau Navigation Aérienne

Bureau TWR

|  |
| --- |
| SVC C.A |

|  |
| --- |
| SVC TELECOM |

|  |
| --- |
| SVC MTO |

|  |
| --- |
| SVC ANTI - INCENDIE |

|  |
| --- |
| SVC TELECOM |

|  |
| --- |
| SVC ENERGIE |

|  |
| --- |
| SVC NAVDS |

|  |
| --- |
| SVC PERSONNEL |

|  |
| --- |
| SVC COMPTABILITE |

|  |
| --- |
| SVC SURETE - AERO |

|  |
| --- |
| SVC MEDICAL |

|  |
| --- |
| SVC FACILITATION |

|  |
| --- |
| SVC HAROI |

|  |
| --- |
| SVC ENTRETIEN |

|  |
| --- |
| SVC FACTURATION SVC CONTROLE |

|  |
| --- |
| SVC RECOUVREMENT SVC IDEF |

|  |
| --- |
| SVC VTA |

## II.2.3.2. ORGANIGRAMME DE LA DIVISION COMMERCIALE (DICOM)

*Figure2 : Organigramme de la Division Commercial[[33]](#footnote-33)*

Division commerciale

Secrétariat

Service VTA

Service Parking

Service IDEF

Service Contrôle Recettes

R

Service Recouvrement

Service Facturation

Bureau Facturation

Aéronautique

Bureau Facturation

Extra-Aéronautique

Bureau Recouvrement

Aéronautique

Bureau Recouvrement

Aéronautique

Perception

Cash

Contrôle

Bureau Report

Eng

Bureau Handling

Bureau Trafic

Bureau Parking

Bureau Access

IDEF Fret

IDEF Passagers

## II.3. ANALYSE DU METIER

L'analyse du métier est la prise de connaissance du domaine d'application et du diagnostic des points forts et des points faibles permettant une approche du problème. Il faut d'abord récolter les informations ensuite assurer la présentation c'est-à-dire inventorier ce qui existe et ainsi tirer les conséquences nécessaires au changement ou au maintien de cet existant.

Ceci dit, voici comment se déroule le processus qui fait l'objet de notre étude

## II.3.1. DESCRIPTION TEXTUEL DU PROCESSUS METIER

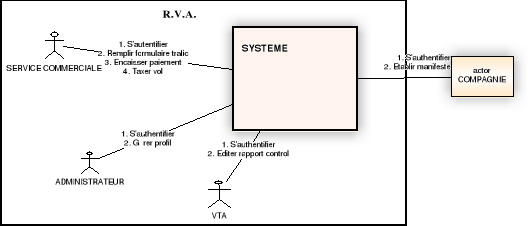
Le processus de notre domaine se déroule de la manière suivante :

A La genèse d’une nouvelle compagnie d’aviation la rva enregistre son contrat avant qu’un avion puis voler dans le cial congolais, Le jour où une compagnie a un vol ; deux heures avant, elle dépose au bureau de navigation (BNA) de la RVA différents documents en rapport avec le vol, dont le formulaire de trafic reprit sous deux exemplaires un au départ et l’autre à l’arrivée sur base des quels sera facturé le mouvement, le plan de vol, etc.

Pour ce qui est de notre domaine d'étude, les documents concernés sont le manifeste passager, le mouvement, et le contrat de la compagnie.

* **Le manifeste passager** ce document est élaboré par la compagnie et contient les informations suivantes :
* La date du vol
* Le lieu d'embarquement
* Le lieu de débarquement
* L'identifiant de l'avion
* Le trajet complet
* Le numéro du vol
* Le numéro, nom et prénoms du passager
* Le numéro de fauteuil du passager
* Son genre : celui-ci peut être soit masculin, féminin, bébé ou enfant
* Le nombre de bagage du passager
* Le poids des bagages du passager
* Le numéro du ticket du passager
* **Le formulaire de trafic :** ce document est élaborés par la compagnie et déposé a la facturation au sein de la division commerciale pour paiement des redevances aéronautiques, ce document contient les informations suivantes :
* Le nom de l'exploitant (la compagnie)
* Le nom de l'aéroport de départ
* Le nom et le code de la ville où se trouve l'aéroport de destination
* Le code de la ville où se trouve l'aéroport de départ
* La date de départ
* La nature du vol (régulier ou non régulier)
* Le type de l'aéronef
* L'immatriculation de l'aéronef
* L'heure de décollage
* Piste en service
* Le régime de vol ;
* AD (adulte) : désigne les passagers adultes
* CH (enfants) : désigne les passagers enfants
* INF(Infant) : désigne les bébés
* Les frets sont mentionnés en Kg
* Le balisage
* **Licence de la compagnie :** ce document est déposé par la compagnie reprend les informations suivantes :
* Le code de la compagnie
* Le siège social
* L’adresse
* Nombre d’avion
* Le numéro téléphonique
* Type de compagnie

## II.3.2. DIAGRAMME DE CONTEXTE



*Figure3 : Diagramme de contexte métier*

## II.4. IDENTIFICATION DES ACTEURS DU METIER.

* La vérification des trafics aériens (VTA) : ce service se charge du control des passagers figurants sur le manifeste et ceux qui monte dans l'appareil de qui doit effectuer le mouvement. Ce control se passe au pied de l'avion avant le décollage.
* Le bureau de navigation (BNA) :c'est le service qui porte le nom de tout le bureau et qui se charge de la réception des tous les documents ayants trait au vol dont le plan de vol et le relever du mouvement aérien après leur élaboration par les compagnies d'aviation. C'est ce service qui communique avec la tour de contrôle et le bureau de statistique pour leur renseigner sur les différents mouvements de vol.
* Le service commercial : ce service se charge du calcul de différentes redevances à payer par les compagnies en tenant compte des informations se trouvant sur le manifeste ainsi que sur d'autres documents qui concernent le vol. C'est ce service qui se charge aussi de l'élaboration du formulaire de trafic. Voici la liste de quelques redevances.

## II.5. DIAGRAMME D’ACTIVITE

 *Figure4 : diagramme d’activité Métier*

## II.6. CAPTURE DES BESOIN FONCTIONNEL

Cette phase va nous permettre d'identifier les différents cas d'utilisations de notre métier. Par le biais des cas d'utilisation, nous serons en contact permanent avec les acteurs du système en vue de définir les limites de celui-ci, et ainsi éviter de trop s'éloigner des besoins réels de l'utilisateur final.

## II.6.1. IDENTIFICATION DES CAS D’UTILISATION

|  |  |  |
| --- | --- | --- |
| **NOM CAS D’UTILISATION** | **INTENTION** | **ACTION** |
| **Etablir Manifeste** | Donner les informations précisent sur les passagers a embarqué dans l’avion | Enregistrer les informations suivantes : nom, post nom, prénom, sexe, adresse, Age destination, bagage |
| **Payer redevance** | S’acquitter des taxes | Verser argent |
| **Etablir formulaire de trafic** | Renseigner toutes les redevances | Donner les informations en détail sur le vol et les différents services évidents |
| **Etablir facture** | Donner le net a payé par la compagnie | Enregistrer le montant par mouvement |
| **Contrôler passagers** | Vérifier tous les passagers embarquant dans l’avion | Comparer les informations sur les formulaires et l’effectif embarquant |

Les différents cas d’utilisations retenus pour notre métier sont les suivant

## II.6.2. DIAGRAMME DE CAS D'UTILISATION DU METIER

C'est un diagramme qui représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UL, celui où s'assure la relation entre l'utilisateur et les objets avec lesquelles le système interagit[[34]](#footnote-34)



*Figure 5: diagramme des cas d’utilisation Métier*

## II.6.3.1. ETUDE DE RELATION ENTRE CAS D’UTILISATION



*Figure 6 : diagramme des cas d’utilisation métier avec relation*

## II.6.3.2. CLASSEMENT DE CAS D’UTILISATION EN ITERATION

|  |  |  |  |
| --- | --- | --- | --- |
| **NOM CAS D'UTILISATION** | **PRIORITE** | **RISQUE** | **ITERATION** |
| Etablir Manifeste | Haute | Haut | 1 |
| Payer redevance | Moyenne | Moyen | 4 |
| Etablir Facture | Haute | Haut | 3 |
| Etablir formulaire de trafic | Haute | Haut | 2 |
| Contrôler passagers | Moyenne | Moyen | 6 |

## II.6.3.3. REGROUPEMENT DES CAS D’UTILISATION EN PAQUETS

Pour améliorer notre modèle d’analyse, nous allons organiser les cas d'utilisation et les regrouper en ensembles fonctionnels cohérents. Pour ce faire, nous utilisons le concept général d'UML, le package.

Ce diagramme donne une vue d'ensemble du système structuré en paquetage. Chaque paquetage représente un ensemble homogène d'éléments du système



*Figure7 : Diagramme de paquetage métier*

## II.6.4. DESCRIPTION TEXTUELLE DES CAS D’UTILISATIONS

1. Nom cas d’utilisation **: Etablir manifeste**
   * **Acteur** : Compagnie
   * **But** : donner les informations sur tous les passagers du vol
   * **Résumé** : donner les nombres et les noms de tous les passagers qui doivent embarquer dans l’aéronef
   * **Pré condition** : avoir été enregistré a la régie des voies aériennes
   * **Scenario Nominale** :
2. Lancer le formulaire manifeste
3. Renseigner toutes les informations sur les passagers
4. Saisir les informations sur l’aéronef
5. Saisir les informations sur les vols
6. Imprimer les manifestes
   * **Scenario Alternatif** :
7. Afficher les champs mal renseigné en rouge(2)
8. Annuler manifester si pas des passagers
9. Afficher les champs non renseigné en rouge
   * **Post Condition** : valider manifeste et imprimer Manifeste



*Figure 8: diagramme de séquence Etablir Manifeste*

1. **Nom cas D’utilisation** : **Etablir Formulaire de trafic**
   * + **Acteur** : bureau de navigation aérienne
     + **But** : mettre au point un document qui retrace les différentes taxes que la compagnie doit payer avant d’effectuer un mouvement
     + **Résumé** : compléter les informations sur les différentes redevances qui feront l’objet de la facture que doit payer la compagnie
     + **Pré condition**:
2. Etre en procession du manifeste
3. Mouvement validé
4. Plan de vol établi
   * + **Scenario nominale :**
5. Saisir toutes les redevances en rapport avec le mouvement
6. Saisir Nom de l’exploitant
7. Valider formulaire
   * + **Scenario alternatif :**

* Afficher les champs mal renseigné en rouge
* Afficher un message d’erreur si la compagnie ne pas reconnue
* Vérifier compagnie
* Vérifier l’aéronef affecté au vol si elle existe dans la base de données si non afficher erreur
* Fermer formulaire
  + - **Post condition** : formulaire de trafic établi et formulaire enregistré dans la base de données.



*Figure 9:diagramme de séquences Etablir formulaire de trafic*

1. Nom cas d’utilisation **: Etablir facture**
   * + - **Acteur** : facturation
       - **But** : facturer compagnie
       - **Résumé** : le facturier Matérialisé les taxes à payer par la compagnie
       - **Pré condition** : formulaire de trafic
       - **Scenario Nominale** :
2. Compléter redevance
3. Renseigner toutes les informations sur les redevances
4. Saisir les informations sur l’aéronef
5. Saisir les informations sur l’aéronef
6. Saisir les informations sur les vols
7. Valider la facture
   * **Scenario Alternatif** :
8. Afficher les champs mal renseigné en rouge Erreur(2)
9. Annuler facture
10. Modifier facture
11. Imprimé facture

* **Post Condition** : facture



*Figure 10 : diagramme de séquence établir Facture*

1. Nom cas d’utilisation **: Payer redevance**
   * + - **Acteur** : Compagnie
       - **But** : s’acquitter des redevances enfin d’avoir l’autorisation d’effectuer un mouvement
       - **Résumé** : après avoir établit la facture il revient à l’exploitant aérien de s’acquitter de sa facture
       - **Pré condition** : facture
       - **Scenario Nominale** :
       - **Scenario Alternatif** :
2. Imprimer facture
3. Valider
   * **Post Condition** : facture payé



*Figure 11: diagramme des séquences payer Redevance*

1. Nom cas d’utilisation **: contrôlé passagers**
   * + - **Acteur** : Compagnie
       - **But** : donner les informations sur tous les passagers du vol
       - **Résumé** : donner les nombres et les noms de tous les passagers qui doivent embarquer dans l’aéronef
       - **Pré condition** : avoir été enregistré à la régie des voies aériennes
       - **Scenario Nominale** :
2. Lancer le formulaire manifeste
3. Renseigner toutes les informations sur les passagers
4. Saisir les informations sur l’aéronef
5. Saisir les informations sur les vols
6. Imprimer les manifestes
   * + - **Scenario Alternatif** :
         1. Afficher les champs mal renseigné en rouge(2)
         2. Annuler manifester si pas des passagers
         3. Afficher les champs non renseigné en rouge
       - **Post Condition** : valider manifeste et imprimer Manifeste



*Figure 12 : diagrammes des séquences contrôlées passagers*

## II.5. LE MODELE DU DOMAINE

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation[[35]](#footnote-35).

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 pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation. A ce niveau, il nous est impératif de procéder à l'identification des concepts du domaine ; plutôt que d'aller en bésogne et tomber dans le trou du problème à résoudre



*Figure12 :modèle du domaine métier*

## II.6. CRITIQUES DE L’EXISTANT

Nous voici arrivés à la fin de notre analyse métier qui a pour objectif de décomposer le système en différent module question de bien le comprendre et en dégagé ses point fort et ses point faibles ;

1. Point fort

La régies des voies aériennes dispose des agents bien qualifier et compétent qui savent bien faire leurs travail en ce qui concerne la gestion du trafic aérien bien qu’ils manquent certains outils des précisions ils arrivent a bien s’en sortir dans l’exécution de leurs tâches quotidiennes

1. Point négatif

La régie des voies aériennes est une institution qui gère plusieurs compagnies d’aviation a caractère national et international Apres notre analyse nous avons découvert les faillesci-après :

* + Les compagnies déposes des fois les formulaires de trafic sans manifestes
  + Manque de communication directe entre RVA et compagnie en ce qui concerne la gestion du manifeste
  + Lenteur dans la recherche des mouvements antérieurs vus que le volume des données a traité manuellement est grande
  + Perte des documents par manque d’endroit des stockages
  + L’incapacité de modifier un manifeste déjà préétablit en cas d’ajout ou manque des passagers lors de l’embarquement
  + L’incapacité d’identifier les passagers en cas d’accident aérien
  + Manque de partage des données a des différent poste de travail par manque d’une application client-serveur pouvant relier les différent service travaillant en synergie avec la Rva

## II.7. PROPOSITION DES SOLUTIONS

Suite aux multiples faiblesses du système soulignées dans le point précédent, nous proposons de mettre en place une application informatique avec laquelle on pourra faire un suivi automatique du trafic Aérien. Celle-ci permettra de gérer toutes les opérations du manifeste jusqu’au décollage de l’aéronef via un intranet qui reliera toutes les divisions commerciales de la RVA ainsi que les compagnies Aériennes.

D'où son apport se fera sentir dans :

-la centralisation des données,

- Taxation Automatique à partir du formulaire de trafic

- Impression et stockages des factures par compagnie,

- la statistique des compagnies, aéronefs, et utilisateurs

- La création des comptes utilisateurs avec contrôle d’accès aux données,

- La réduction de temps d'accès aux données,

-La facilitation à l'accès aux données,

- L'automatisation de certaines tâches,

- L'amélioration de la sécurité des données,

Avant de passer à l'implémentation de l'application, nous devons d'abord faire l'analyse et la conception, puis passer au choix des outils qu'on utilisera pour la mise en œuvre de celle-ci. C'est ce qui sera le sujet dans les chapitres qui suivent.

## **CHAPITRE III : ANALYSE ET CONCEPTION DU SYSTEME INFORMATIQUE**

## III.1. INTODUCTION

Le [Système d’information](http://fr.wikipedia.org/wiki/Syst%C3%A8me_d%E2%80%99information) est un élément essentiel de l’entreprise de nos jours. Les entreprises sont prêtes à investir de fortes sommes afin d’obtenir un système d’information performant et adapté à leur activité. C’est également un enjeu majeur pour la compétitivité de l’entreprise.[[36]](#footnote-36)

C’est pendant L'analyse nous donne les activités du micro processus (niveau d'abstraction constant). A chaque niveau d'abstraction, un micro processus régit la construction des modèles. Pour la formalisation de besoins des utilisateurs, nous utiliserons un sous-ensemble du langage de modélisation UML qui nous sera également nécessaire et suffisant pour notre travail.

Dans un premier temps, les besoins des utilisateurs seront modélisés au moyen des cas d'utilisation UML, et seront représentés de façon concrète au moyen d'Interface Homme Machine (IHM) destinée à faire réagir les futurs utilisateurs.

Chaque cas d'utilisation est décrit textuellement de façon détaillée, mais donne également lieu à un diagramme de séquence simple représentant graphiquement la chronologie des interactions entre les acteurs et le système. Dans le cadre du scenario nominal, ces diagrammes sont appelés : « diagramme de séquence système ».

Enfin, nous allons élaborer un modèle du domaine ainsi que le diagramme de classes participantes (qui décrivent cas d'utilisations par cas d'utilisation à travers les trois principales classes, à savoir : Dialogue, Contrôle et Entité pour aboutir au code

## III.2. EXPRESSIONS DE BESOINS FONCTIONNELS

Notre application de gestion du trafic aérien sera en mesure de couvrir les fonctionnalités suivantes :

1. **Besoins fonctionnels**

* Gestion des utilisateurs par privilèges
* La gestion des manifestes passagers
* La gestion de la facturation
* La gestion de la statistique de mouvements
* Enregistrement des aéronefs par compagnies
* Taxation de redevance aéronautique
* Consultation du mouvement et facture mensuelle

1. **Besoins non fonctionnels**

* Sécurité d’accès aux données
* Ergonomie
* La licence

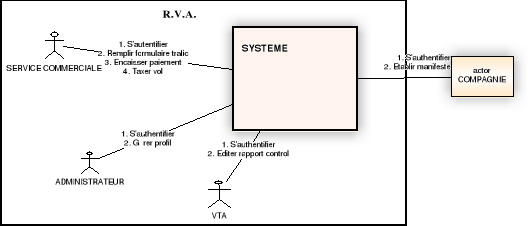
## III.3. FORMALISATION DES BESOINS

## III.3.1. IDENTIFICATION DES ACTEURS DU SYSTÈME INFORMATIQUE

La question majeure à ce niveau est de savoir *qui devient les acteurs identifiés au niveau du métier par rapport au Système Informatique.*Nous partons avec le principe que tout travailleur d'interface du système métier devient l'acteur du cas d'utilisation au Système Informatique.

Les acteurs retenus pour notre système informatique sont :

* **La compagnie** : une compagnie peut être vus comme un exploitant aérien pouvant s’enregistrer pour avoir accès aux différente fonctions du système informatique qui sont : établir manifeste sur base du quel il va payer ses redevance avant d’embarquer les passagers.
* **La facturation** : est un acteur qui doit s’identifier pour établir le formulaire de trafic, sur base du quel il doit établir la facture en fonction des différent services que la compagnie a recouru pour effectuer un mouvement
* **La caisse**: est un acteur qui pour rôle d’encaisser le montant et valider la facture
* **Vta** : est un acteur qui a pour rôle : contrôler les passagers avant l’embarquement, et enregistrer le formulaire de trafic mais qui doit évidement s’authentifier.
* **L’administrateur du système** : cette acteur doit s’identifier pour créer, modifier, supprimer, contrôler les comptes des utilisateurs selon leurs droit d’accès et leurs priorité dans le système informatique.



***Figure 13 : Diagramme de contexte du système informatique***

## III.3.2. IDENTIFICATION DES MESSAGES DU SYSTEME

Par définition un message désigne la communication unidirectionnelle entre les objets qui transportent de l'information avec l'intention de déclencher une activité chez le récepteur[[37]](#footnote-37).

**Les messages qui seront émis par notre système sont :**

* + L’erreur si le mot de passe est incorrect
  + Les factures par compagnies
  + Le nombre des utilisateurs du système selon les privilèges
  + Le nombre des compagnies Aérienne affiliés
  + Les nombres d’aéroport de la Rva
  + Les nombres des types d’avion opérant à Luano
  + Créer, modifier, et supprimer un compte utilisateur
  + Créer, modifier, supprimer une facture
  + Créer, modifier, supprimer un formulaire
  + Créer, modifier, supprimer un manifeste passagers
  + Créer, modifier, supprimer un mouvement
  + Impression de la facture

***Les messages qui seront reçus par notre système sont :***

* + Ajout, modification et suppression d’une compagnie
  + Ajout, modification et suppression d’un Aéroport
  + S’inscrire
  + Payer
  + Ajout manifeste
  + La recherche des données

## III.3.3. IDENTIFICATION DES CAS D’UTILISATIONS DU SYSTEME

Pour constituer les cas d'utilisations, nous allons considérer l'intention fonctionnelle de chaque acteur par rapport au système dans le cadre de l'émission ou de la réception de massage. Ainsi lorsque nous regroupons ces intentions fonctionnelles en unité cohérentes, nous obtenons les cas d'utilisations suivants :

|  |  |  |  |
| --- | --- | --- | --- |
| **Cas d'utilisation** | **Acteurs principaux Secondaires** | **Messages émis/reçu par les acteurs** | **itérations** |
| **S’inscrire** | **compagnie** | **Emet** : Donner la licence d’exploitation | 8 |
| Administrateur | **Reçoit** : identité de la compagnie |
| **Etablir Manifestes** | **Compagnie** | **Emet** : créer/modifier/supprimer/imprimer/enregistrer les Manifestes | 1 |
| Vta | **Reçoit** : Manifeste établit |
| **Payer Redevances** | **Compagnie** | **Emet** : verser Argent | 10 |
| **Caissier** | **Reçoit** : Compter argent remettre différence, convertir au taux du jour |
| **Facturation** | **Emet :**création du formulaire de trafic, |
| **Etablir Formulaire de trafic** | **Facturation** | **Emet :**création du formulaire de trafic, modification, annulation, Ajout de redevances Aéronautique, Enregistrement | 2 |
| Compagnie | **Reçoit :**Formulaire de trafic. |
| **Etablir Facture** | **Facturation** | **Emet :**Crée Facture pour la compagnie, | 3 |
| **compagnie** | Ajout, modification, des diffèrent taxes, total à payer en dollars et la conversion en franc congolais au taux du jour. |
|  |  |
| **Encaisser payement** | **caisse** | **Emet**: Total a payé | 9 |
| **Compagnie** | **Reçoit :**facture Payée |
| **Contrôler Passagers** | **Vta** | **Emet**: les formulaires de trafic Contrôler Passagers avant embarquement dans l’aéronef | 4 |
| **Facturation** | **Reçoit :** effectif embarqué dans l’aéronef Comparaison des chiffres du manifeste |
| **Etablir Rapport** | **VTA** | **Emet : créer formulaire après contrôle** Modifier, Ajouter, supprimer redevances | 5 |
| **Facturation** | **Reçoit**: Formulaire Recalculer Redevances |
| **Gérer Profil Utilisateur** | **Administrateur** | **Emet** : contrôle d’accès au système Par Privilèges Créer, Ajouter, Modifier, supprimer les comptes utilisateur | 6 |
| **S’authentifier** | **utilisateurs** | **Saisir login et mot de passe** | 7 |

## III.3.5. DIAGRAMME DES CAS D’UTILISATION SYSTEME

****

*Figure14 : cas d’utilisation système informatique*

## III.3.6. DESCRIPTION TEXTUELLE DE CAS D’UTILISATION PAR ITERATION

1. **Nom du cas d’utilisation : « Etablir Manifeste »**

**- but**: Créer, Modifier, Supprimé un Manifeste

- **Objectif** : Enregistrer les informations sur les identités des passagers a embarqués dans un mouvement

-**Acteur Principale** : Compagnie

**-Acteur Secondaire** :Vta

-**Pré condition** : Avoir une licence d’exploitation, avoir au moins un Aéronef, Un siège social

-**Scenario nominal** :

1. Ouvrir menu principal
2. lancer le formulaire Manifeste
3. Renseigner tous les champs
4. valider le formulaire

-**Scenario Alternatif** :

* Afficher le message d’erreur s’il y a les champs mal renseignés en rouges en relançant le formulaire

**-Post-condition :** Manifeste Etablit



*Figure15: diagramme de séquence établir manifeste*

**2. Diagramme de cas d’utilisation : « établir Formulaire de trafic »**

**- but**: Créer, Modifier, Supprimé le Formulaire de trafic

- **Objectif** : Enregistrer les informations sur les différentes Taxes

-**Acteur Principale** : Facturation

**-Acteur Secondaire** : Compagnie

-**Pré condition** : réception d’un Manifeste pour un Vols

-**Scenario nominal** :

1. Ouvrir écran menu principal
2. lancer le formulaire Manifeste
3. Renseigner tous les champs
4. valider

-**Scenario Alternatif** :

* Afficher le message d’erreur s’il y a les champs mal renseignés en rouges en relançant le formulaire

**-Post-condition :** Formulaire Etablit



*Figure16: diagramme de séquence établir Formulaire de trafic*

**3. Diagramme de cas D’utilisation : « Etablir Facture »**

**- but**: Créer une Facture

- **Objectif** : taxé les Redevances

-**Acteur Principale** : Facturation

**-Acteur Secondaire** : Compagnie

-**Pré condition** : Avoir un formulaire de Trafic Préétablit

-**Scenario nominal** :

1. Ouvrir écran menu principal
2. lancer le formulaire facture
3. Renseigner tous les champs
4. valider

-**Scenario Alternatif** :

Afficher le message d’erreur s’il y a les champs mal renseignés en rouges en relançant le formulaire



*Figure17: diagramme de séquence établir facture*

**4. Nom du cas d’utilisation : inscrire Compagnie**

**- but :** la compagnie doit s’inscrire a la régie des voies aériennes pour voler sur les cieux congolais

- **Résumé** : donner les informations sur la compagnie et se mettre en règle avec la régie des voies aériennes avant d’effectuer tous mouvement

-**Acteur Principale** : administrateur

**-Acteur Secondaire** : Compagnie

-**Pré condition** : Avoir une licence d’exploitation

-**Scenario nominal** :

1. Ouvrir écran menu principal
2. Ecran ajout compagnie
3. Renseigner tous les champs
4. valider

-**Scenario Alternatif** :

* Afficher le message d’erreur s’il y a les champs mal renseignés en rouges en relançant le formulaire



*Figure17: diagramme de séquence inscrire compagnie*

1. Nom cas d’utilisation **: valider paiement**

**- but :** valider la facture

- **Résumé** : donner les informations sur le montant a été encaisser avant d’effectuer le mouvement

-**Acteur Principale** : Caisse

**-Acteur Secondaire** : Compagnie

-**Pré condition** : Avoir une facture

-**Scenario nominal** :

1. Ouvrir écran menu principal
2. Ecran consultation
3. imprimer

-**Scenario Alternatif** :

- **Post condition :** facture payée



*Figure18: diagramme de séquence payer facture*

**6. Nom du cas d’utilisation : s’authentifier**

**- but :** accéder au système

- **Résumé** saisir son login et son mot de passe

-**Acteur Principale** : utilisateur

**-Acteur Secondaire** :

-**Pré condition** : Avoir un compte

-**Scenario nominal** :

1. écran authentification
2. Saisir login et mot de passe
3. valider

-**Scenario Alternatif** :

* Afficher le message d’erreur s’il y a les champs mal renseignés en rouges en relançant le formulaire
* Login et mot de passe incorrect
  + - **Post condition** : ouverture menu principal



*Figure19: diagramme de séquence s’authentifier*

**7. Nom du cas d’utilisation : contrôlé passagers**

**- but :** vérifier les nombres de passagers embarquant physiquement dans l’aéronef et passagers sur formulaire de trafic

- **Résumé** : compté passagers

-**Acteur Principale** : Vta

**-Acteur Secondaire** : compagnie

-**Pré condition** : vol

-**Scenario nominal** :

1. écran menu principal
2. Ecran mouvement
3. imprimer

-**Scenario Alternatif** :

* + - **Post condition** : valider facture

**8. Diagramme de cas d’utilisation : établir rapport**

**- but**: Créer, Modifier, Supprimé le Formulaire de trafic

- **Objectif** : Enregistrer les informations sur les différentes Taxes

-**Acteur Principale** : Vta

**-Acteur Secondaire** : Compagnie

-**Pré condition** : réception d’un Manifeste pour un Vols

-**Scenario nominal** :

1. Ouvrir écran menu principal

1. lancer le formulaire Manifeste
2. Renseigner tous les champs
3. valider

-**Scenario Alternatif** :

* Afficher le message d’erreur s’il y a les champs mal renseignés en rouges en relançant le formulaire

**-Post-condition :** rapport établit



*Figure20: diagramme de séquence contrôler passagers*

**9. Diagramme de cas d’utilisation : payer redevance**

**- but**: s’acquitter de la facture

- **Objectif** : payé montant sur la facture

-**Acteur Principale** : compagnie

**-Acteur Secondaire** : caisse

-**Pré condition** : facture

-**Scenario nominal** :

-**Scenario Alternatif** :

- **post condition :** facture payé



*Figure21: diagramme de séquence Payer Redevance*

**10. Nom du cas d’utilisation : gérer utilisateur**

**- but :** ajout, suppression, modification de compte utilisateur

- **Résumé** : mettre compte utilisateur à jour

-**Acteur Principale** : utilisateur

**-Acteur Secondaire** :

-**Pré condition** : Avoir un compte

-**Scenario nominal** :

1. ouvrir écran administration

2. gérer compte utilisateur

3. ajout, suppression, modification

4. valider

-**Scenario Alternatif** :

* + - **Post condition** : profil mis a jour



*Figure22: diagramme de séquence gérer utilisateur*

## III.4. ANALYSE DU SYSTEME

## III.4.1.MODEL DU DOMAINE SYSTEME



*Figure23: Model du domaine système*

## III.4.2.DIAGRAMME DE CLASSE PARTICIPANTE

1. Diagramme des classes participantes DCU Etablir Formulaire Manifeste



*Figure24: diagramme de classe participante établir manifeste*

1. Diagramme des classes participantes DCU Etablir Formulaire de trafic



*Figure25: diagramme de classe participante formulaire trafic*

1. Diagramme des classes participantes DCU Etablir Facture



*Figure26: diagramme de classe participante établir facture*

1. Diagramme des classes participantes DCU Etablir Rapport



*Figure27: diagramme de classe participante établit rapport*

1. Diagramme de classes Participantes DCU s’authentifier



*Figure28: diagramme de classe participante s’authentifier*

1. Diagrammes des classes participantes DCU encaisser paiement



*Figure29: diagramme de classe participante encaisser paiement*

1. Diagramme des classes participantes Contrôler passagers



*Figure30: diagramme de classe participante contrôler passagers*

1. Diagramme des classes participantes DCU « payer Facture »



*Figure31: diagramme de classe participante payer facture*

1. Diagramme de classes Participante DCU  «  Enregistrer Compagnie »



*Figure32: diagramme de classe participante enregistrer compagnie*

1. Diagramme des classes participantes DCU « Gérer Utilisateurs »



*Figure33: diagramme de classe participante gérer utilisateurs*

## III.5. CONCEPTION DU SYSEME INFORMATIQUE

## III.5.1.DIAGRAMME DE COMMUNICATION

1. Diagramme de communication établir manifeste



*Figure34: diagramme de communication établir manifeste*

1. Diagramme de communication établir formulaire de trafic



*Figure34: diagramme de communication établir formulaire*

1. Diagramme de communication établir facture



*Figure36: diagramme de communication établir facture*

1. Diagramme de communication s’authentifier



*Figure36: diagramme de communication s’authentifier*

1. Diagramme de communication Gérer Utilisateur



*Figure38: diagramme de communication gérer utilisateurs*

1. Diagramme de communication établir rapport



*Figure39: diagramme de communication établir rapport*

1. Diagramme de communication contrôlé Passagers



*Figure40: diagramme de communication contrôlé passagers*

1. Diagramme de communication DCU inscrire compagnie



*Figure 41: diagramme de communication inscrire compagnie*

## III.5.2. DIAGRAMME DE CLASSE DE CONCEPTION



*Figure42: diagramme de classe conception système*

## III.5.2.1. PASSAGE AU MLD



*Figure43: diagramme de modèle logique de donnée*

## CONCLUSION

Nous voici arrivé au terme de notre troisième chapitre qui est intitulé analyse et conception du système informatique. Dans ce dernier il a été question de procéder au dégagement des grandes fonctionnalités du système informatique a crée en s’appuyant sur l’expression de besoins fonctionnels, formalisme de besoins fonctionnels et la phase d’analyse pour chuter a la dernière phase qui est la conception du système.

## CHAPITRE IV : IMPLEMENTATION

## IV.1. CONCEPTION DETAILLEE

Qui vient juste après une activité qui s'inscrit dans l'organisation définie par la conception préliminaire. Nous allons construire des classes, les vues IHM, les interfaces et les tables qui vont donner une image « prête à coder ». Le modèle logique y est également important à ce niveau.

Pour ce faire, nous allons maintenant incorporer dans nos diagrammes *le choix d'architecture et les choix technologiques* qui vont modifier les classes de conceptionpréliminaire vues au chapitre précédent, les préciser, ajouter de nouvelles classes plus techniques, etc.

N'oublions pas que l'objectif principal du modèle de conception détaillée est de pouvoir être traduit directement en ***code* ;**ainsi l'implémentation des*3 types de classes d'analyse* sera réalisée comme suit :

Le dialogue est réalisé par les pages *HTML5* et du**CSS3 et JavaScript**pour le rendu : l'idée est également d'insérer des instructions de script dans des pages HTML5 (le langage peut être PHP). Les pages HTML5 peuvent contenir des *WebForms*,qui génèrent des vues en HTML5 et qui permettent également de placer des contraintes (*composant validators*) sur les champs de saisie utilisateur. Ces contraintes sont vérifiées côte serveur par défaut, mais un module JavaScript permet de pré valider les toutes côtes client sauf désactivation du code JavaScript Par un utilisateur Malveillant avec le navigateur évoluer comme Google chrome et Firefox37.

Le « contrôle » est implémenté soit par des classes associées à PHP (classes ***CodeBehind***), soit par des classes supplémentaires déployées dans l'application web. Les entités seront implémentées par les classes PHP ensemble avec des objets de Mapping/relationnels constituant une passerelle vers le SGBDR choisit.

## IV.2. CONCEPTION DE LA PERSISTANCE

**a. La dérivation du « Modèle Métier » en Modèle logique des données.**



La modélisation Relationnelle est la concrétisation d'une modélisation de classes. Le modèle des classes est un outil pour la conception et le modèle relationnel est un outil pour la réalisation du système.

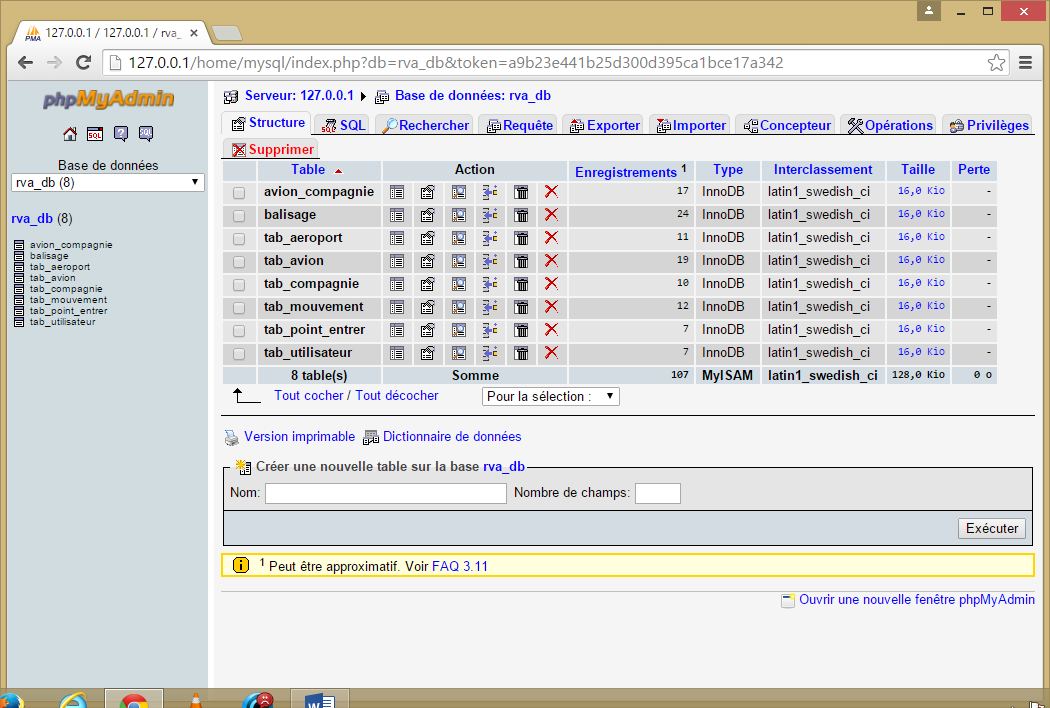
Le modèle « métier » définit dans le premier chapitre incorporait ce que les méthodes antérieurs nommaient `'Modèle Conceptuel de Données `' « MCD ». Les possibilités d'expressions d'un modèle UML obligent à compléter les règes de passage du modèle objet vers le modèle logique de données (MLD).

La conception du MLD prend entrée :

· Le modèle sémantique (essentiellement les objets du métier)

· Dans le modèle pragmatique (les objets de nature organisationnelle ou administrative) ;

***Structure de la base de données***



1. Architecture de l’application

La technologie orientée objet requiert une architecture, laquelle organise les interactions entre les objets. Souvent ces objets sont regroupés en classes, les classes en domaines et ces domaines en couches qui permettent de représenter l'architecture de l'application.

Ainsi construire une architecture en couche est un critère de qualité dans le cadre d'un développement Objet. Il ne nous reste qu'à déterminer le nombre des couches et leur contenu.

### A) L’Architecture client Serveur

L'environnement **client-serveur** désigne un mode de communication à travers un [réseau](http://fr.wikipedia.org/wiki/R%C3%A9seau_informatique) entre plusieurs [programmes](http://fr.wikipedia.org/wiki/Programme_(informatique)) ou [logiciels](http://fr.wikipedia.org/wiki/Logiciel) : l'un, qualifié de [client](http://fr.wikipedia.org/wiki/Client_(informatique)), envoie des requêtes ; l'autre ou les autres, qualifiés de [serveurs](http://fr.wikipedia.org/wiki/Serveur_informatique), attendent les requêtes des clients et y répondent. Par extension, le client désigne également l'[ordinateur](http://fr.wikipedia.org/wiki/Ordinateur) sur lequel est exécuté le logiciel client, et le serveur, l'ordinateur sur lequel est exécuté le logiciel serveur.[[38]](#footnote-38)

### B ) Architecture trois tiers

**architecture 3-tiers**, **architecture à trois niveaux** ou **architecture à trois couches** est l'application du modèle plus général qu'est le multi-tier. L'architecture logique du système est divisée en trois niveaux ou couches[[39]](#footnote-39) :

* couche présentation ;
* couche métier ;
* couche accès aux données.

Son nom provient de l'[anglais](http://fr.wikipedia.org/wiki/Anglais) *tier* signifiant étage ou niveau. Il s'agit d'un modèle logique d'architecture applicative qui vise à modéliser une[application](http://fr.wikipedia.org/wiki/Application_(informatique)) comme un empilement de trois couches [logicielles](http://fr.wikipedia.org/wiki/Logiciel) (étages, niveaux, tiers ou strates) dont le rôle est clairement défini :

* **la présentation** : Elle correspond à la partie de l'[application](http://fr.wikipedia.org/wiki/Application_(informatique)) visible et interactive avec les utilisateurs. On parle d'[interface homme machine](http://fr.wikipedia.org/wiki/Interface_homme_machine). En informatique, elle peut être réalisée par une application graphique ou textuelle ([WPF](http://fr.wikipedia.org/wiki/Windows_Presentation_Foundation)). Elle peut aussi être représentée en [HTML](http://fr.wikipedia.org/wiki/Hypertext_Markup_Language) pour être exploitée par un [navigateur web](http://fr.wikipedia.org/wiki/Navigateur_web) ou en [WML](http://fr.wikipedia.org/wiki/Wireless_Markup_Language) pour être utilisée par un téléphone portable. le **traitement** métier des données : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;
  1. la couche Métier**:**

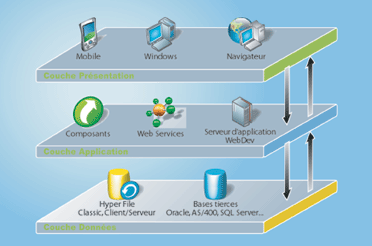
Elle correspond à la partie fonctionnelle de l'[application](http://fr.wikipedia.org/wiki/Application_(informatique)), celle qui implémente la « logique », et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation.

Les différentes règles de gestion et de contrôle du système sont mises en œuvre dans cette couche.

La couche métier offre des services applicatifs et métier à la couche présentation. Pour fournir ces services, elle s'appuie, le cas échéant, sur les données du système, accessibles au travers des services de la couche inférieure. En retour, elle renvoie à la couche présentation les résultats qu'elle a calculés[[40]](#footnote-40).

### La couche d’accès aux données [persistantes](http://fr.wikipedia.org/wiki/Persistance_(informatique)) :

Correspond aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.



*Figure 44:* architecture trois tiers

## IV.4.DIAGRAMME DE COMPOSANT

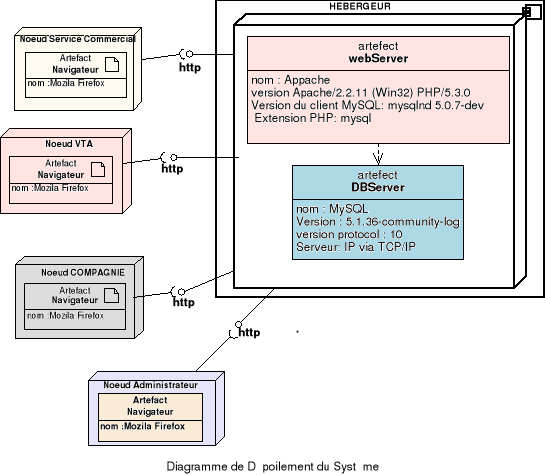
Les diagrammes des composants tombent sous la catégorie des diagrammes d'implémentation, un genre de diagramme qui modélise l'implémentation et le déploiement du système.



*Figure45: diagramme de composant*

## IV.5.DIAGRAMME DE DEPLOIEMENT

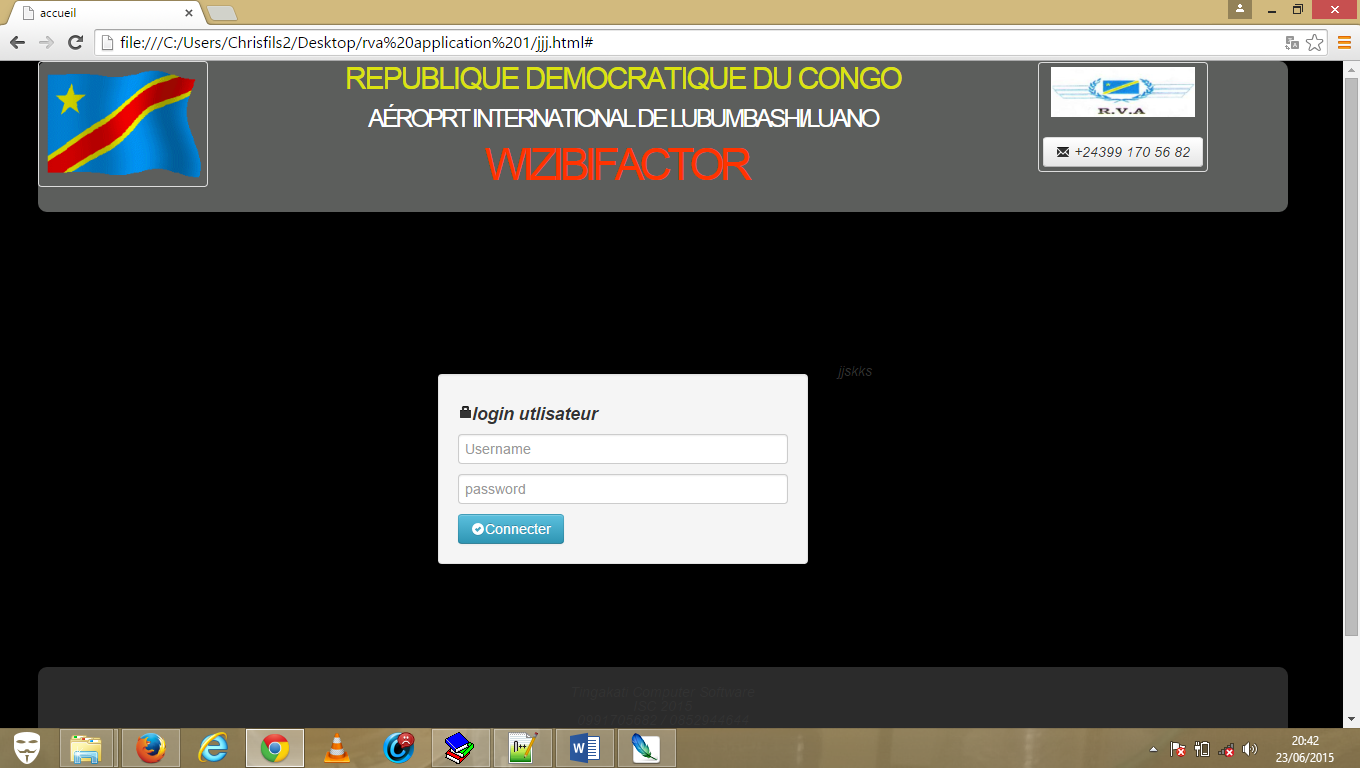
Le diagramme de déploiement modélise les composants matériels utilises pour implémenter un système et l'association entre ces composants. Des composants peuvent apparaître également sur un diagramme de déploiement pour montrer le lieu géographique leur déploiement



*Figure46: diagramme de déploiement*

### IV.6. PRESENTATION DE L’APPLICATION

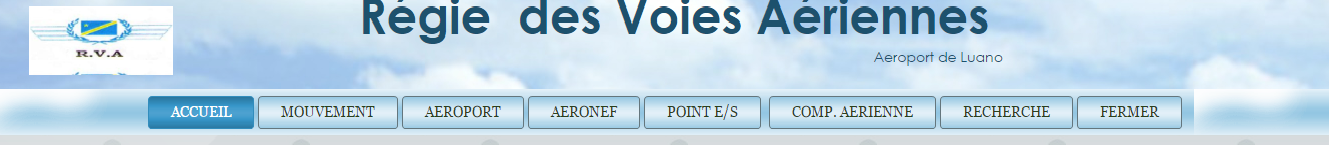
### IV.6.1. interface d’authentification



### IV.6.2. interface page d’accueil



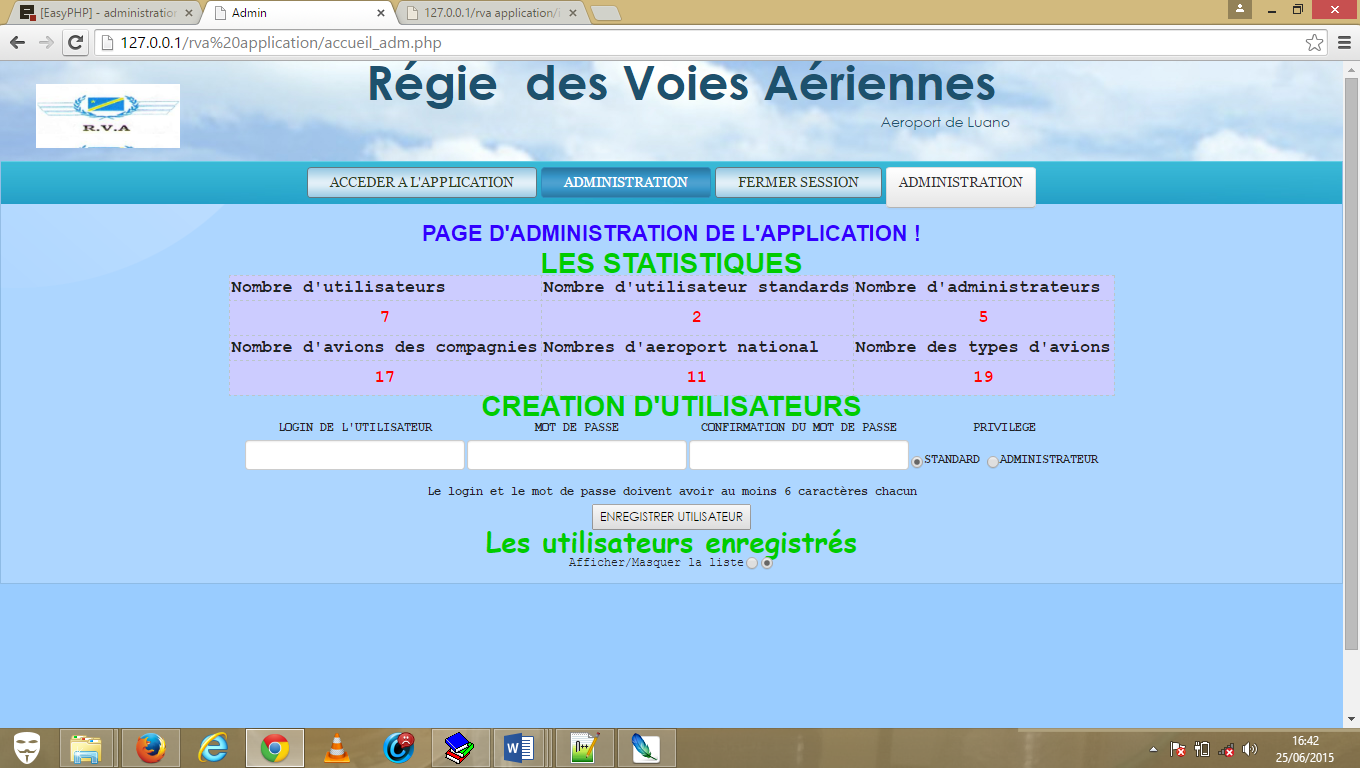
### IV.6.3. interface menu principal



### IV.6.4. interface formulaire de trafic



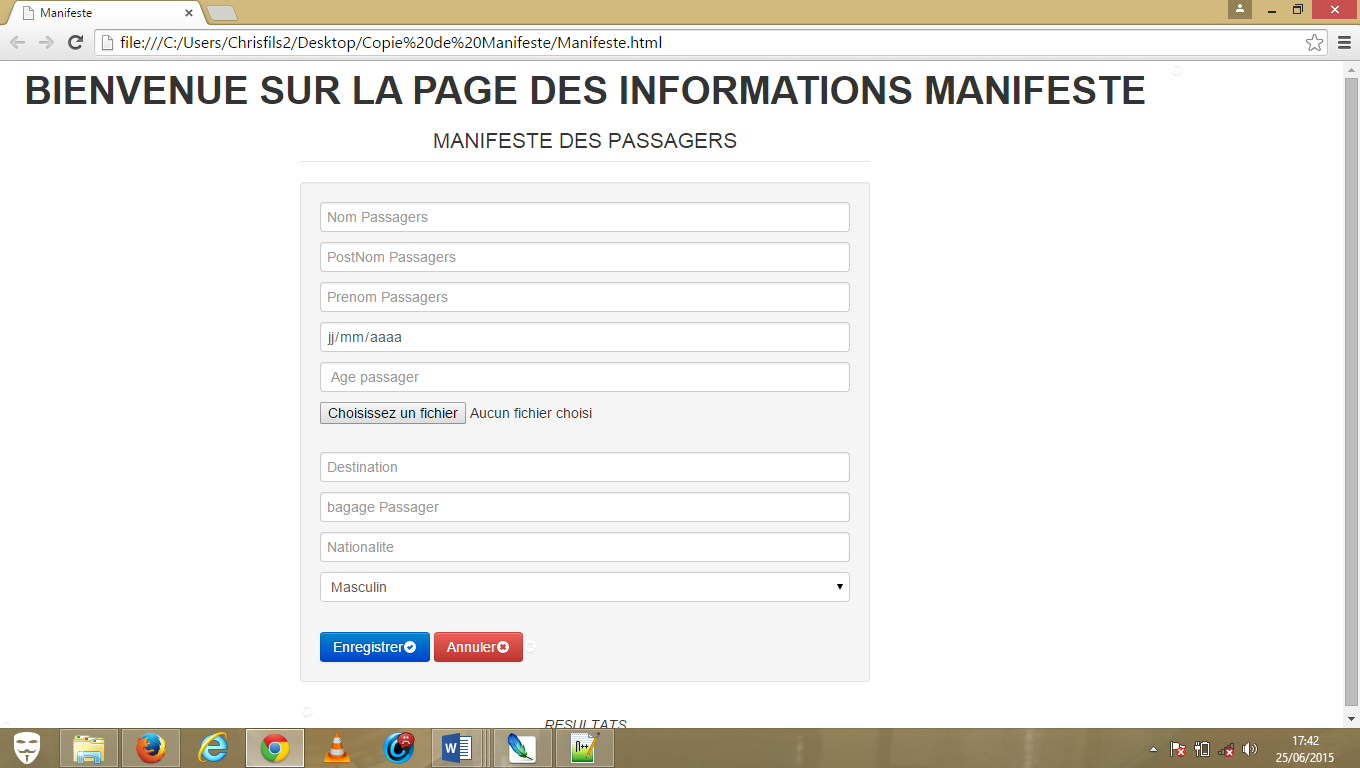
### IV.6.5. interface page d’administration



### IV.6.6. interface page mouvement



### IV.6.7. interface page manifeste



## CONCLUSION GENERALE

Nous voici arrivé à terme notre sujet de recherche qui porter sur la conception et réalisation d’une application de gestion du trafic aérien cas de la régie de voies aériennes Lubumbashi.

La réalisation d’une application de gestion du trafic aérien a un atout important pour les entreprise aéroportuaires voulant améliorer leurs gestions avec efficacité(les échanges des informations appropriées, améliorer les processus et adaptation aux nouvelles technologies d’informations et de la communication)

C’est dans cette vision que s’inscrit notre projet de fin d’étude dans laquelle nous nous sommes confié la charge de concevoir et de réaliser une application de gestion du trafic aérien, cette application ayant pour objectif non seulement de gérer les trafics aériens (gestion passagers, manifeste, facturation, statistiques) mais aussi une gestion centraliser des utilisations ainsi que les exploitants aériens et leurs aéronefs.

Pour répondre aux besoins des utilisateurs ; nous avons recourus au langage UML (Unified modeling laguage) qui nous a permis de représenté graphiquement les différents diagrammes et la méthode UP (unified process) qui a son tour nous a amené a découpé notre projet en différente phase entre autre phase d’inception, d’élaboration, de construction et de transition. Le processus UP étant itératif et incrémental basé sur le risques et piloté par cas d’utilisation fait en sorte que nous puissions découpés notre projet en quatre chapitre pour bien analysé notre problème de gestion.

Donc le premier chapitre de notre travail nous eut à définir les concepts et les considérations théoriques, dans le second chapitre nous sommes passés à l’analyse de l’existant et le troisième chapitre nous l’avons consacré l’analyse et la conception détaillée de l’application et le quatrième chapitre enfin nous a conduit à l’implémentation d’un processus du domaine

Ce projet a fait l’objet d’une expérience intéressante qui nous a permis d’améliorer nos connaissances et nos compétences dans le domaine de la programmation, nous avons appris à manipuler les langages PHP4, HTML5, CSS3, JQUERY, BOOTSTRAP, AJAX.

En effet, ce travail étant une œuvre humaine ne pas un modèle unique et parfait, c’est reste ouvert à toutes les critiques et nous sommes prêt à recevoir toutes les remarques et suggestions tendant à améliorer cette étude étant donné que tout projet informatique a toujours était l’œuvre d’une équipe

### BIBLIOGRAPHIE

* 1. **Livres**
  + PASCAL ROQUES Les cahiers du Programmeur ED .EYROLLES 2002,
  + PASCAL ROQUES UML2 par la Pratique 6°ed EYROLLES,
  + JOSEPH GABAY et DAVID GABAY Uml2 Analyse et Conception ed. EYROLLES 2008,
  1. **Notes des cours**

-PATRCK KASONGA cours de conception des Systèmes Informatiques 2, Isc 2015,

- ASSAUT BIATSHINI Conception des Systèmes Informatiques 1, isc 2014,

-JEAN ENGELS PHP5 et exercices corrigés Paris 2010,

- DI GALLO Frédéric méthodologies des systèmes d’information UML 2010,

- ASSAUT BIATSHINI réseau et télématique 1 isc 2014,

- RUPHIN NYAMI Atelier de génie logiciel isc 2015,

* 1. **Web**
  + http/www.Openclassroom.com
  + http/www.Developper.com
  + http://www.agilemodeling.com

Table des matières

[INTRODUCTION GENERALE 1](#_Toc423107554)

[1. CHOIX ET INTERET DU SUJET 1](#_Toc423107555)

[2. ETAT DE LA QUESTION 1](#_Toc423107556)

[3. PROBLEMES DE GESTION (PROBLEMATIQUES) 2](#_Toc423107557)

[4. AMELIORATIONS ATTENDUES (HYPOTHESES) 3](#_Toc423107558)

[5. DELIMITATIONS DU SUJET 4](#_Toc423107559)

[6. METHODES ET TECHNIQUES 4](#_Toc423107560)

[6.1. METHODES 4](#_Toc423107561)

[6.2. TECHNIQUES 5](#_Toc423107562)

[6.2.1. L’interview 5](#_Toc423107563)

[6.2.2. La documentation 6](#_Toc423107564)

[6.2.3. La technique d’observation 6](#_Toc423107565)

[SUBDIVISION DU TRAVAIL 6](#_Toc423107566)

[CHAPITRE I : CADRAGE THEORIQUE 7](#_Toc423107567)

[*I.1. BALISAGE CONCEPTUEL* 7](#_Toc423107568)

[*I.1.1 CONCEPTION* 7](#_Toc423107569)

[*I.1.1.1. Présentation du langage de modélisation UML* 7](#_Toc423107570)

[*I.2. LA DEMARCHE UP (Unified Process)* 9](#_Toc423107571)

[LES ABREVIATIONS UTILISER 9](#_Toc423107572)

[FONCTIONNEMENT DE LA MATHODE UP 10](#_Toc423107573)

[UP COMME METHODE AGILE 12](#_Toc423107574)

[*I.3. THEORIE SUR L’APPLICATION* 14](#_Toc423107575)

[*I.4. TRAFIC AERIEN* 15](#_Toc423107576)

[*CONCLUSION* 17](#_Toc423107577)

[CHAPITRE II DESCRITION DU MILIEU D’ETUDE 18](#_Toc423107578)

[II.1. APERCU HISTORIQUE 18](#_Toc423107579)

[II.1.1. BREVE RADIOSCOPIE DE LA REGIE DES VOIES AERIENNE DE LUBUMBASHI(RVA) 18](#_Toc423107580)

[II.2. L’AEROPORT INTERNATIONAL DE LUBUMBASHI/LUANO 18](#_Toc423107581)

[II.2.1. HISTORIQUE DE L’AEROPORT INTERNATIONAL DE LA LUANO 18](#_Toc423107582)

[II.2.2. SITUATION GEOGRAPHIQUE 19](#_Toc423107583)

[II.2.3. PRESENTATION DE L’ORGANIGRAMME GENERAL DE LA RVA 20](#_Toc423107584)

[II.2.3.1. ORGANIGRAMME ET STRUCTURE GENERALE DE L’AEROPORTINTERNATIONAL DE LUBUMBASHI 20](#_Toc423107585)

[II.2.3.2. ORGANIGRAMME DE LA DIVISION COMMERCIALE (DICOM) 21](#_Toc423107586)

[II.3. ANALYSE DU METIER 22](#_Toc423107587)

[II.3.1. DESCRIPTION TEXTUEL DU PROCESSUS METIER 22](#_Toc423107588)

[II.3.2. DIAGRAMME DE CONTEXTE 23](#_Toc423107589)

[II.4. IDENTIFICATION DES ACTEURS DU METIER. 24](#_Toc423107590)

[II.5. DIAGRAMME D’ACTIVITE 25](#_Toc423107591)

[II.6. CAPTURE DES BESOIN FONCTIONNEL 25](#_Toc423107592)

[II.6.1. IDENTIFICATION DES CAS D’UTILISATION 26](#_Toc423107593)

[II.6.2. DIAGRAMME DE CAS D'UTILISATION DU METIER 27](#_Toc423107594)

[II.6.3.1. ETUDE DE RELATION ENTRE CAS D’UTILISATION 28](#_Toc423107595)

[II.6.3.2. CLASSEMENT DE CAS D’UTILISATION EN ITERATION 28](#_Toc423107596)

[II.6.3.3. REGROUPEMENT DES CAS D’UTILISATION EN PAQUETS 28](#_Toc423107597)

[II.6.4. DESCRIPTION TEXTUELLE DES CAS D’UTILISATIONS 29](#_Toc423107598)

[II.5. LE MODELE DU DOMAINE 34](#_Toc423107599)

[II.6. CRITIQUES DE L’EXISTANT 36](#_Toc423107600)

[II.7. PROPOSITION DES SOLUTIONS 36](#_Toc423107601)

[CHAPITRE III : ANALYSE ET CONCEPTION DU SYSTEME INFORMATIQUE 38](#_Toc423107602)

[III.1. INTODUCTION 38](#_Toc423107603)

[III.2. EXPRESSIONS DE BESOINS FONCTIONNELS 38](#_Toc423107604)

[III.3. FORMALISATION DES BESOINS 39](#_Toc423107605)

[III.3.1. IDENTIFICATION DES ACTEURS DU SYSTÈME INFORMATIQUE 39](#_Toc423107606)

[III.3.2. IDENTIFICATION DES MESSAGES DU SYSTEME 40](#_Toc423107607)

[III.3.3. IDENTIFICATION DES CAS D’UTILISATIONS DU SYSTEME 41](#_Toc423107608)

[III.3.5. DIAGRAMME DES CAS D’UTILISATION SYSTEME 43](#_Toc423107609)

[III.3.6. DESCRIPTION TEXTUELLE DE CAS D’UTILISATION PAR ITERATION 43](#_Toc423107610)

[III.4. ANALYSE DU SYSTEME 56](#_Toc423107611)

[III.4.1.MODEL DU DOMAINE SYSTEME 56](#_Toc423107612)

[III.4.2.DIAGRAMME DE CLASSE PARTICIPANTE 57](#_Toc423107613)

[III.5. CONCEPTION DU SYSEME INFORMATIQUE 62](#_Toc423107614)

[III.5.1.DIAGRAMME DE COMMUNICATION 62](#_Toc423107615)

[III.5.2. DIAGRAMME DE CLASSE DE CONCEPTION 66](#_Toc423107616)

[III.5.2.1. PASSAGE AU MLD 67](#_Toc423107617)

[CONCLUSION 67](#_Toc423107618)

[CHAPITRE IV : IMPLEMENTATION 68](#_Toc423107619)

[IV.1. CONCEPTION DETAILLEE 68](#_Toc423107620)

[IV.2. CONCEPTION DE LA PERSISTANCE 69](#_Toc423107621)

[L’Architecture client Serveur 70](#_Toc423107622)

[Architecture trois tiers 71](#_Toc423107623)

[ la couche Métier 71](#_Toc423107624)

[La couche d’accès aux données persistantes : 72](#_Toc423107625)

[IV.4.DIAGRAMME DE COMPOSANT 72](#_Toc423107626)

[IV.5.DIAGRAMME DE DEPLOIEMENT 73](#_Toc423107627)

[IV.6. PRESENTATION DE L’APPLICATION 74](#_Toc423107628)

[IV.6.1. interface d’authentification 74](#_Toc423107629)

[IV.6.2. interface page d’accueil 75](#_Toc423107630)

[IV.6.3. interface menu principal 75](#_Toc423107631)

[IV.6.4. interface formulaire de trafic 75](#_Toc423107632)

[IV.6.5. interface page d’administration 76](#_Toc423107633)

[IV.6.6. interface page mouvement 76](#_Toc423107634)

[IV.6.7. interface page manifeste 77](#_Toc423107635)

[CONCLUSION GENERALE 78](#_Toc423107636)

[BIBLIOGRAPHIE 79](#_Toc423107637)

1. 1sensagent [↑](#footnote-ref-1)
2. Dumazedier, Ripert, Loisir et cult. 1966, p.30 [↑](#footnote-ref-2)
3. Flux ahom de Wikipédia [↑](#footnote-ref-3)
4. René Descartes [↑](#footnote-ref-4)
5. Pascal roques (cahier du programmeur) Edit Eroylles 2002 p3 [↑](#footnote-ref-5)
6. Di gallo Fréderic cous de conception de système d’information p.15 [↑](#footnote-ref-6)
7. Pascal Roques les cahiers du programmeur ed.Eyrolles P.4 [↑](#footnote-ref-7)
8. Idem [↑](#footnote-ref-8)
9. Idem [↑](#footnote-ref-9)
10. Pascal Roque UML 2 ed : Erroylles 2006 P 89 [↑](#footnote-ref-10)
11. Pascal Roque UML 2 ed : Erroylles 2006 P 86 [↑](#footnote-ref-11)
12. DI GALLO Frédéric méthodologies des systèmes d’information UML [↑](#footnote-ref-12)
13. Pascal Roque UML 2 ed : Erroylles 2006 P 87 [↑](#footnote-ref-13)
14. [↑](#footnote-ref-14)
15. Pascal Roque UML 2 ed : Erroylles 2006 P 88 [↑](#footnote-ref-15)
16. Pascal Roque UML 2 ed : Erroylles 2006 P 89 [↑](#footnote-ref-16)
17. Pascal Roque UML 2 ed : Erroylles 2006 P 90 [↑](#footnote-ref-17)
18. G Gabail UML 2 ;opsite [↑](#footnote-ref-18)
19. <http://www.agilemodeling.com/> , [↑](#footnote-ref-19)
20. Cours de UML2 Univesité de dophine [↑](#footnote-ref-20)
21. UML2 Michael Blaha 2005 [↑](#footnote-ref-21)
22. Pascal Roques UML par la Pratique P/323 [↑](#footnote-ref-22)
23. Dictionnaire de l’internaute [↑](#footnote-ref-23)
24. wikipedia [↑](#footnote-ref-24)
25. Cours de navigation Aerienne [↑](#footnote-ref-25)
26. Division Commercial RVA 2015 [↑](#footnote-ref-26)
27. Division Commercial RVA 2015 [↑](#footnote-ref-27)
28. Division Commercial RVA 2015 [↑](#footnote-ref-28)
29. Division Commerciale RVA 2015 [↑](#footnote-ref-29)
30. Division commerciale RVA 2015 [↑](#footnote-ref-30)
31. [↑](#footnote-ref-31)
32. Source bureau du personnel régie de voies aériennes Lubumbashi [↑](#footnote-ref-32)
33. Source division commerciale de la régie des voies aériennes Lubumbashi [↑](#footnote-ref-33)
34. Pascal Roques edit Eyrolles UML par pratique [↑](#footnote-ref-34)
35. Pascal Roque Uml 2 de l’apprentissage a la pratique. [↑](#footnote-ref-35)
36. www. Wikipedia.Org/evaluation d’un système d’information [↑](#footnote-ref-36)
37. 36 dictionnaires et Recueil [↑](#footnote-ref-37)
38. [↑](#footnote-ref-38)
39. http://fr.wikipedia.org/wiki/Architecture\_trois\_tiers [↑](#footnote-ref-39)
40. Assaut Biatshini Cours de reseau et telematique 1 [↑](#footnote-ref-40)