Table des matières :

Table des figures :

INTRODUCTION GÉNÉRALE

Vu que l'informatique en Tunisie entre dans une phase d'accélération qui se manifeste par l'introduction rapide des applications informatiques et des NTIC généralement dans la plupart des aspects de la vie professionnelle et quotidienne, nos établissements se doient d'améliorer la qualité de leurs services et ce par le biais de l'informatisation et l'intégration de l'informatique dans toutes leurs activités.

Le présent travail s'inscrit dans le cadre d'un Projet de Fin d'Études conçu et réalisé au sein de la société « INFORM@TIX » qui est une entreprise basée à Tunis et ayant pour missions : La vente, l'installation, la maintenance et la mise en place de plusieurs types de systèmes, d'équipements et des solutions informatiques et réseaux ; Et qui pour pouvoir bien se positionner sur le marché et résister à la concurrence, elle a été obligée d'évoluer et de diversifier ses produits et ses solutions à plusieurs niveaux, entre autres : La conception et le développement des nouvelles solutions et applications informatiques, ...

Figure 0 : Logo de la société « INFORM@TIX ».

Compte tenu de l'absence d'un système informatisé de « Gestion de courriers (Bureau d'Ordre) » dans plusieurs sociétés en Tunisie, de la difficulté de la gestion manuelle de cette tâche et conscients de l'impact qu'aura son automatisation -pour moderniser les méthodes de travail et améliorer ses divers processus au sein des sociétés- sur la qualité des services de point de vue fiabilité, sécurité, exactitude, cohérence, flexibilité, rapidité, ... ; Il s'est avéré nécessaire, voire important, d'automatiser la gestion de courriers (Bureau d'Ordre) afin de remédier à touts les problèmes générés par la gestion manuelle dans les sociétés.

Notre projet à réaliser, comporte principalement des phases ou étapes cohérentes et claires :

- La première, est consacrée à « l'étude préalable et de quelques applications existantes » : Elle comporte une analyse de quelques cas et situations actuelles et d'exemples de systèmes existants au niveau de gestion de courriers (Bureau d'Ordre) en Tunisie, les critiques des existants qui vont permettre d'aboutir à une nouvelle approche dans la démarche du travail et à la proposition d'une solution pour un futur Système d'Information ; Car cette application de « Gestion de courriers (Bureau d'Ordre) d'une entreprise » est une nouvelle application, que la société « INFORM@TIX », compte concevoir et développer afin de la commercialiser comme l'un de ses nouveaux produits informatiques.

- La deuxième étape, consiste à dégager les fonctions que le nouveau système devrait fournir (C'est-à-dire « La valeur ajoutée ») et ce en mettant en évidence les besoins à l'origine de son développement qui va être la phase de « description des différents diagrammes (Ou « étude détaillée ») » : « La conception » (Elaboration des différents diagrammes).

- La dernière, sera destinée à « la réalisation » et à la mise en œuvre de la solution conçue.

CHAPITRE 1 :

ÉTAT DE L'ART ET ÉTUDE PRÉALABLE

Introduction

Dans le cadre de la collecte des informations nécessaires à la conception et à la réalisation de notre application et afin de mener à bien l'analyse de l'existant, nous avons opté pour l'étude et la compréhension des situations réelles, auxquelles on a essayé de répondre tout au long de notre démarche dans ce projet.

1.1 : État de l'art des applications de « Gestion de courries »

Auparavant, la gestion de courriers (Bureau d'Ordre) ainsi que celle des différentes interventions informatiques étaient manuelles. Compte tenu des insuffisances du système manuel ; Les défaillances persistent toujours par manque de suivi de différents types de courriers (Bureau d'Ordre) dans les entreprises et les établissements privés ainsi que ceux étatiques en Tunisie présente actuellement une grande importance, notamment avec les évolutions des technologies qui connaissent aujourd'hui une véritable révolution exponentielle ; D'où naisse l'idée de suivre la traçabilité d'un simple papier concernant un sujet bien déterminé, à une application pour tout suivre dans l'entreprise, … Pour remédier à cette situation au sein des sociétés, on a pris sur notre responsabilité la tâche qui consiste à réaliser une gestion de l'aspect organisationnel du service de Bureau d'Ordre, ayant pour but de faciliter la tâche de gestion et de suivi de courriers, …

En effet, un courrier doit être réceptionner, enregistrer, suivre son acheminement, … Il est ensuite affecté à un destinataire interne ou externe à la société.

Dans ce qui suit, nous présentons quelques exemples d'applications de gestion de courriers (Bureau d'Ordre) existantes sur le marché Tunisien, qui serviront comme étude de l'existant qui va précéder les différentes étapes jusqu'à la mise en place de notre nouveau système ;
Ce qui nous menera à une étude basée sur les activités liées aux champs d'études couverts par ce projet à partir desquels sera conçu le nouveau système et les directives à suivre pour la conception et le développement d'un système efficace et de qualité.

Cette partie reposera, donc, en premier lieu, sur la présentation et les critiques d'exemples de quelques applications existantes dont les objectifs se rapprochent des nôtres :

1.1.1 : « *MENARA-GBO : Gestion du courrier (Bureau d'Ordre)* », de la société « SIAT »

Figure 1.1 : Logo de l'application « MENARA-GBO », de la société « SIAT ».

La société « SIAT » est une société basée à Tunis et qui est au service des entreprises pour la mise en œuvre de solutions de dématérialisation des processus métier des entreprises avec une large couverture fonctionnelle ; Parmi ses logiciels, elle a « *MENARA-GBO* » qui est conçue pour les collectivités de toutes tailles qui souhaitent piloter la gestion de leurs courriers entrant et sortant en optimisant leurs traitements.

Les services de l'application « MENARA-GBO » :

- Enregistrement des courriers entrant et sortant.
- Suivie des courriers entrant et sortant.
- Richement paramétré.
- Diffusion des courriers.
- Recherches des courriers.

Pourquoi « MENARA-GBO » ?

- Interface intuitive.
- Suivi et contrôle des courriers.
- La réduction du volume de papiers utilisés.
- La diminution des traitements.

1.1.2 : « *Gestion de Bureau d'Ordre et de courriers (GBO)* », de la société « Xtensus »

Figure 1.2 : Logo de l'application « GBO », de la société « Xtensus ».

La société « Xtensus » est une société basée au Technopôle El Ghazala à l'Ariana, elle a une application de gestion de courriers appelée « *GBO* » (*Gestion de Bureau d'Ordre et de courriers*) ; Ce système permet non seulement la gestion des courriers arrivée et départ, mais aussi la gestion des informations relatives aux courriers.

Les principaux modules qui composent la solution « GBO » :

- *Gestion des courriers* : L'optimisation.
- *Suivi des documents* : La définition des différentes règles de circulation des documents et le suivi de leur circulation entre les différentes structures.
- *Administration* : La gestion des utilisateurs, leurs profils et leurs droits d'accès.

1.1.3 : « *MyGBO (Gestion Bureau d'Ordre)* », de la société « nTIS »

La société « nTIS » (*New Technologie et Innovation Services*) est une société basée à Tunis et qui a un logiciel de gestion de courriers du Bureau d'Ordre : « *MyGBO* ».

Les services de l'application « MyGBO » :

- *Gestion des utilisateurs* : Permet la gestion des profils des utilisateurs et des droits d'accès.
- Paramétrage des natures des courriers.
- Paramétrage de la structure d'entreprise.
- Gestion des courriers entrant et Sortant.
- Accès aux courriers à travers la recherche.

1.2 : Critiques des solutions existantes

Dans cette section, nous mettons l'accent sur la problématique de notre travail afin de dévoiler les raisons justifiants la mise en œuvre de notre nouvelle application informatique.

Tout nouveau projet procède toujours de l'existance d'un ou des problèmes auxquels on s'attend à donner une explication et en proposer une ou des solutions à y remedier.
La problématique est l'ensemble de questions précises et concises que l'on se pose au sujet à étudier.

En effet, jadis le traitement des données et des courriers au niveau des Bureaux d'Ordre des entreprises et des établissements se faisait manuellement (Enregistrer et suivre les couriers arrivée et départ dans des registres), ce qui était lent, ennuyeux, harassant et sujet à des nombreuses erreurs. Ses multiples problèmes du système manuel et surtout les évolutions technologiques et scientifiques ont conduit à l'émergence d'un système informatique ; De là, ce dernier s'est développé et a connu une grande expansion à tel enseigne qu'aucun domaine de la vie socio-économique ne peut s'en passer.

De ce fait, notre préoccupation majeure est d'analyser « La gestion de courriers (Bureau d'Ordre) d'une entreprise » ;

Après les études, parmi les problèmes qui ont été relevés, on cite :

- Minimiser l'utilisation des documents en papier et les risques de pertes ou d'usures des différents documents en papier (= *Risques de pertes des informations*) ;
- Le besoin d'eviter l'utilisation des cahiers-registres en papier pour le suivi de courriers ;
- Le besoin d'eviter, parfois, l'utilisation des fichiers « Microsoft Excel » pour les enregistrements des données et les redondance qu'ils peuvent engendrer ;
- Comment pourrions-nous éviter les problèmes liés à la gestion des courriers et leurs achiminements aux destinataires ? ;
- L'absence de moyens clairs de recherche et de consultation de différents types de courriers et de correspondances entrants et sortants des entreprises, notamment ceux anciens ;
- Une perte de temps dans le recherche des informations voulues ;

- Peut-on être en mesure d'établir des listes des courriers selon des différents critères, pour en avoir leur gestion (Par période, par source d'arrivée, par destinataire, ...) ;
- Et il y en a un besoin majeur qui est à l'origine de manque de nombre de personnel suffisant dans les entreprises et les établissements (Au niveau de leurs Bureaux d'Ordre) pour le traitement de différentes tâches se rapportant à la gestion des courriers et des correspondances, ce qui exige l'automatisation de ce travail ;
- ...

Ce sont des interrogations, parmi plusieurs, qui suscitent un intérêt certain, auxquelles il faudrait trouver une ou des solutions et des réponses.

Conclusion

Cette étude, diagnostic et compréhension de l'existant, va nous permettre d'élaborer la spécification des besoins de notre projet, qui fera l'objet du chapitre suivant.

CHAPITRE 2 :

ANALYSE ET SPÉCIFICATION DES BESOINS

Introduction

Tout nouveau projet à des raisons d'être pour en exister ; Et quand il s'agit d'un projet informatique, on a éventuellemnt des besoins fonctionnels et d'autres non-fonctionnels dont-il repose pour son développement et qui seront dévoilés par la suite.

2.1 : Solution proposée et objectifs

Dans le but de changer en mieux le Système d'Information actuel et d'acquiescer au même temps aux éventuelles attentes du futur système, en apportant certaines améliorations dans le déroulement de travail et le fonctionnement de service concerné par la gestion de courriers, afin de réduire certaines erreurs et difficultés ayant trait à cette gestion, c'est-à-dire d'essayer de suivre toute procédure à effectuer ;

Et vu ce qui précède, on doit concevoir et développer une application qui nous permet de prendre la solution aux problématiques et de remédier aux dysfonctionnements et des obsolescences les plus frappantes de certains systèmes actuels et aux problèmes qui ont été relevés et étudiés ; Donc il s'agit de proposer la réalisation d'une application pour le service de « Gestion de courriers (Bureau d'Ordre) d'une entreprise » avec une Base de Données ; Ce qui permettra d'alléger les charges des traitements manuels, le coût et d'assurer un accès facile et rapide aux différentes données, historiques, ...

Le choix de cette solution, est dans le but de pouvoir satisfaire les objectifs attendus de différents acteurs et services intéragissants dans la gestion de courriers (Bureau d'Ordre), qui permettra de :

- Gagner du temps pour mieux mâtriser la chaîne de traitement de courriers en évitant les problèmes liés à l'enregistrement, le suivi, la traçabilité et la gestion des courriers et des correspondances (*Arrivée et départ*) des entreprises : Courrier simple, fax, facture, note de service, demande, réclamation, ... ;
- Etablir des listes des courriers : *Impression de différents états* ;
- Obtenir une vision claire sur les différents courriers selon des critères : *Recherche* ;
- Avoir les possibilités d'ajouter, de rechercher, de consulter, de modifier et de supprimer facilement ;
- Gestion des droits d'accès à l'application : Pour l'enregistrement des courriers, la recherche, la consultation, la modification et la suppression ;
- ...

Les performances du système que nous proposons constitueraient une garantie pour la gestion des courriers (Bureau d'Ordre) d'une société, le choix porté sur ce sujet revêt d'une importance capitale au bon déroulement du travail au sein des entreprises et établissements, car le traitement et la diffusion de l'information constituent une force ou une faiblesse dans le rendement de toute entreprise.
« Le service de gestion de courriers » ne peut plus être un service secondaire dans l'entreprise : C'est le coeur des flux entrants et sortants ; De ce fait, ce service requiert l'attention des entreprises à la hauteur de sa valeur stratégique afin d'en assurer la traçabilité et le suivi de tout courrier.

Ce travail nous permettra d'apporter une certaine amélioration dans le fonctionnement de différents services d'une entreprise, dans le but de réduire certaines erreurs et difficultés ayant trait à la gestion et l'acheminement et le déroulement de travail, par la suite suivre toutes procédures effectuées, afin de pouvoir éditer automatiquement les résultats ; C'est-à-dire que sur le plan pratique, cette nouvelle application apportera des nouvelles évolutions technologiques sur les méthodes de gestion de courriers (Bureau d'Ordre) de la société en assurant une bonne gestion :
La rapidité, la performance et aussi bien la présence de l'information immédiatement tant au présent que dans le futur, ce qui permettra à la société de réaliser un pas de plus vers les nouvelles technologies.

2.2 : Les Acteurs

Nous commençons, tout d'abord, par définir ce qui est « *un Acteur* » :

« Un Acteur » représente l'abstraction d'un rôle joué par des entités externes (Utilisateur, dispositif matériel, ... ou autre système) qui interagissent directement avec le système étudié.

Nous allons, par la suite, énumérer les acteurs susceptibles d'interagir avec le système ;
« Les Acteurs » de notre application sont :

- « Le Responsable du Bureau d'Ordre » : C'est " *le responsable de courriers* ", qui peut :

  - Ajouter les courriers.
  - Rechercher les courriers.
  - Consulter les courriers.
  - Mettre à jour les courriers.
  - Imprimer la liste des courriers.
  - Supprimer les courriers départ.

- « L'Agent du Bureau d'Ordre » : Cet acteur peut :

  - Ajouter les courriers.
  - Rechercher les courriers.
  - Consulter les courriers.
  - Imprimer la liste des courriers.

- « L'Administrateur » : C'est " *le responsable de gestion de l'application et des utilisateurs* ", qui peut :

  - Ajouter les utilisateurs.
  - Modifier les utilisateurs.
  - Supprimer les utilisateurs.
  - Rechercher les courriers.
  - Consulter les courriers.

2.3 : Besoins fonctionnels

Les besoins fonctionnels représentent les réponses du système aux demandes formulées, auxquelles doit répondre notre application :

- *Enregistrement des différents types de courriers arrivée et départ*,
- *Gestion de courriers* : Ajout, recherche, modification (*Mise à jour*), suppression, lister et / ou imprimer,
- *Recherche de courriers selon critère* : Par type, par date, …,
- *Impression de différents états de suivi*,
- *Établir la liste de différents établissements / structures en relation avec l'entreprise (Concernant les courriers arrivée et départ)*,
- *Impression de différentes listes*,
- *Gestion des droits d'accès à l'application* : Pour l'enregistrement des courriers, la recherche, la consultation, la modification et la suppression,
- …

2.4 : Besoins non-fonctionnels

Les performances du système que nous proposons constitueraient une garantie pour la gestion de courriers (Bureau d'Ordre), nous mettons l'accent sur les besoins d'ordre non-fonctionnels qui spécifient les propriétés du système, telles que les contraintes d'environnement et d'implémentation, la performance, la maintenance, l'extensibilité et la flexibilité.
Certains besoins non-fonctionnels sont généraux et ne peuvent pas être rattachés à un cas d'utilisation particulier, à part les besoins fondamentaux ; Notre futur système doit répondre aux critères suivants, à savoir :

- *La performance* : Une application doit être, avant tout, performante c'est à-dire à diverses fonctionnalités, répond aux exigences des futurs utilisateurs d'une manière presque optimale, cohérente et avec le minimum de manipulations.
- *La fiabilité* : L'application doit fonctionner sans erreurs, ni bugs d'une façon cohérente entre ses différents modules.
- *La convivialité* : La future application doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples et adaptées aux besoins des utilisateurs.
- *L'ergonomie* : Qui a pour objectif d'améliorer l'Interaction Homme-Machine (*Interface Homme-Machine : I.H.M.*), la facilité d'utilisation, en mettant l'accent, par exemple, sur la conception des interfaces utilisateurs simples et compréhensibles afin qu'elles soient en adéquation avec les caractéristiques, perceptives et cognitives de leurs utilisateurs.

- Présentation des informations aux utilisateurs d'une façon simple, claire et compréhensive des différentes rubriques, des interfaces, des menus, ...
- *La sécurité* : Notre nouvelle application doit permettre un accès sécurisé aux données (Authentification et sécurité d'accès et droits attribués aux différents utilisateurs).
- *Gérer les accès* : Application Multi-Utilisateurs.
- L'application doit signaler les erreurs de manipulations ou les alertes par des messages d'erreur (Par exemple : Date erronée, manque de destinataire, ...).
- ...

2.5 : Les Diagrammes des Cas d'Utilisations

Dans cette partie, nous schématisons les Diagrammes des Cas d'Utilisations (En Anglais : « *Use Case Diagram* ») (« *Un cas d'utilisation » décrit une fonctionnalité du système, utilisée par un utilisateur / acteur et telle que se manifeste pour ce dernier*) qui représentent la structure des grandes fonctionnalités et besoins nécessaires et possibles que doit fournir le système à ses utilisateurs (Acteurs) en permettant d'identifier et de décrire les possibilités des interactions entre eux, le comportement et les fonctions du système du point de vue de l'utilisateur, c'est-à-dire la relation entre l'utilisateur et les objets que le système met en œuvre.

Les diagrammes des cas d'utilisation modélisent à " QUOI " sert le système.

- ***Diagramme des Cas d'Utilisations : « Responsable du Bureau d'Ordre » :***

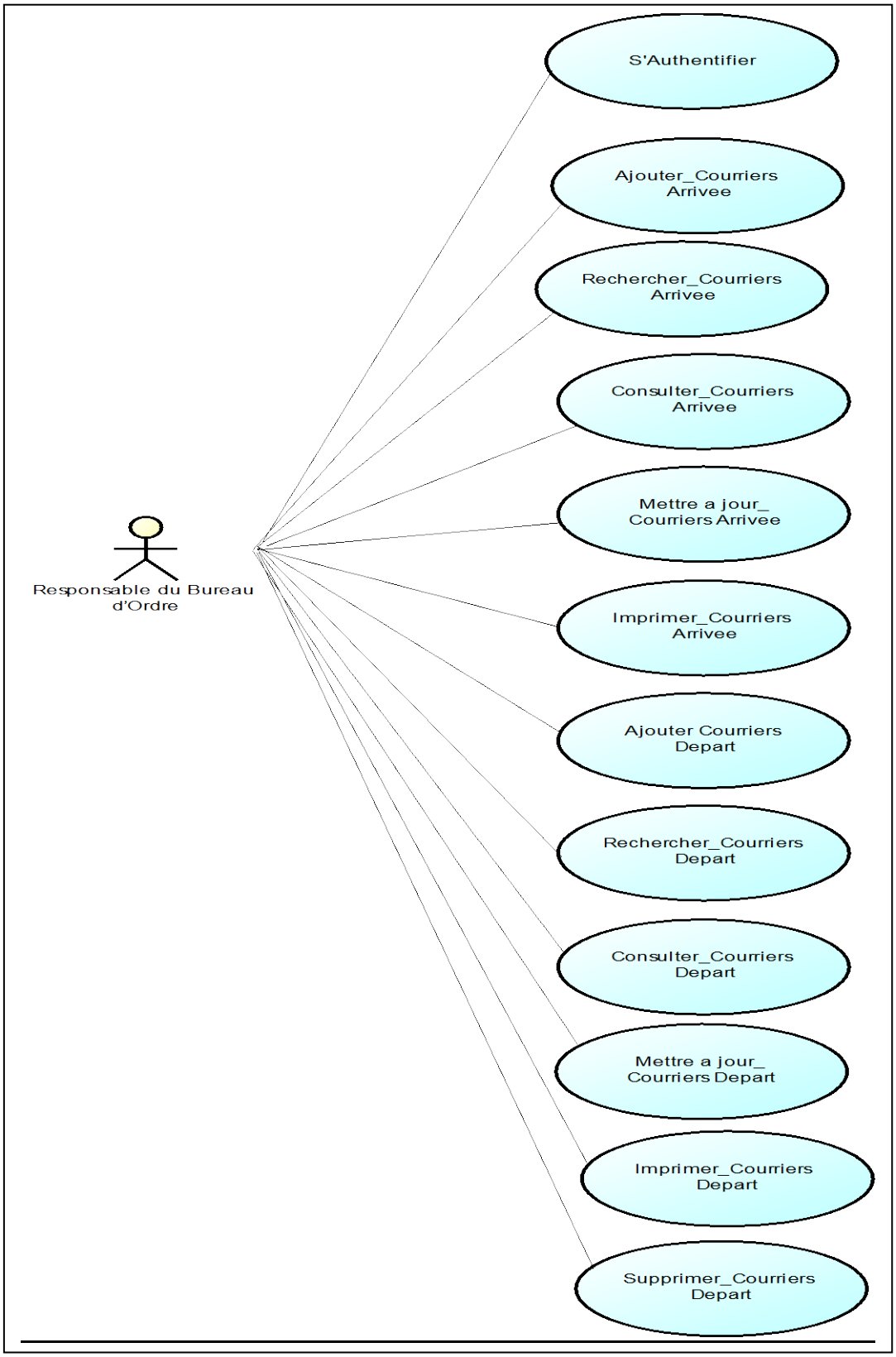

Figure 2.1 : Diagramme des Cas d'Utilisations : « Responsable du Bureau d'Ordre ».

- *Diagramme des Cas d'Utilisations : « Agent du Bureau d'Ordre » :*

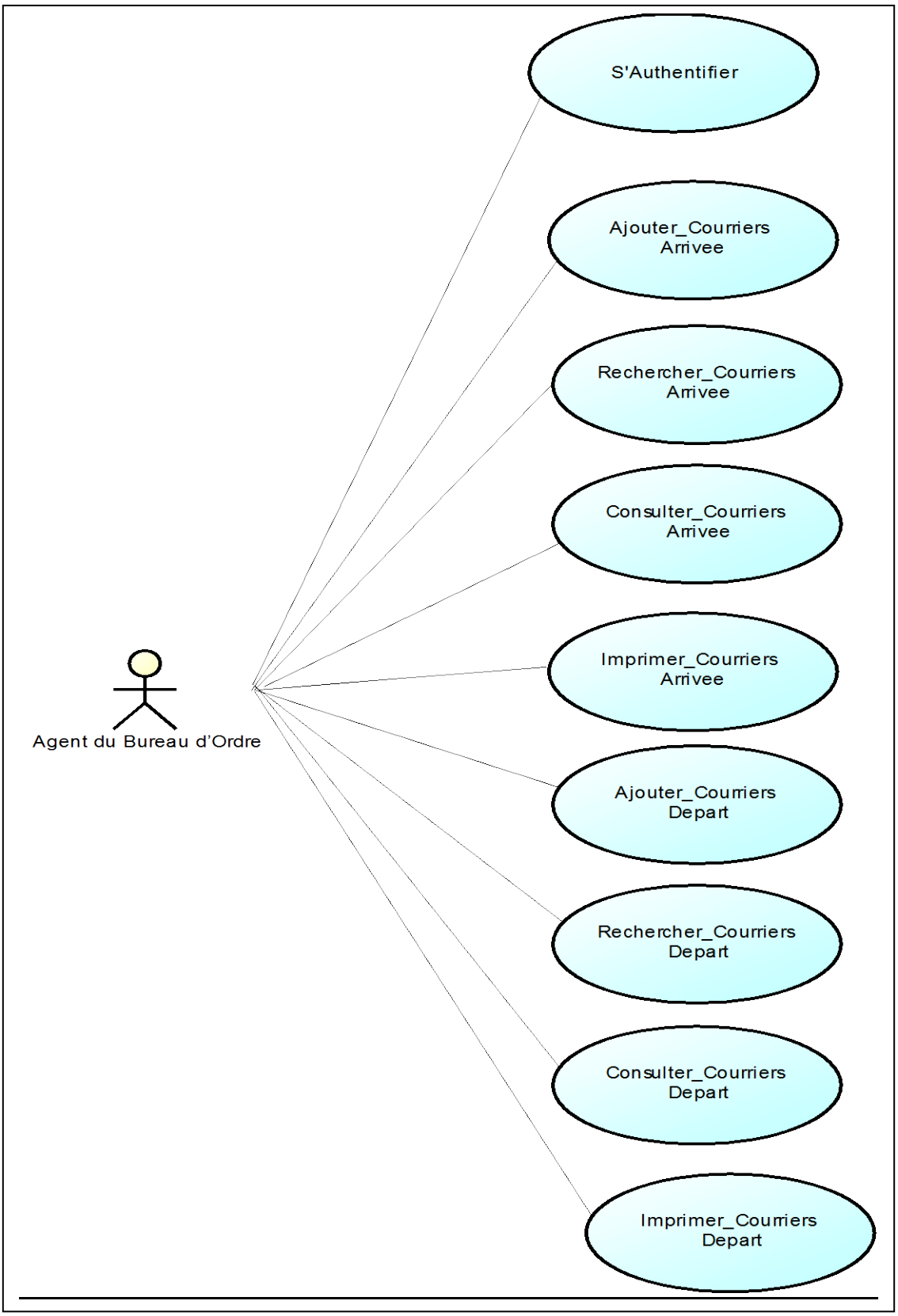

Figure 2.2 : Diagramme des Cas d'Utilisations : « Agent du Bureau d'Ordre ».

- *Diagramme des Cas d'Utilisations : « Administrateur » :*

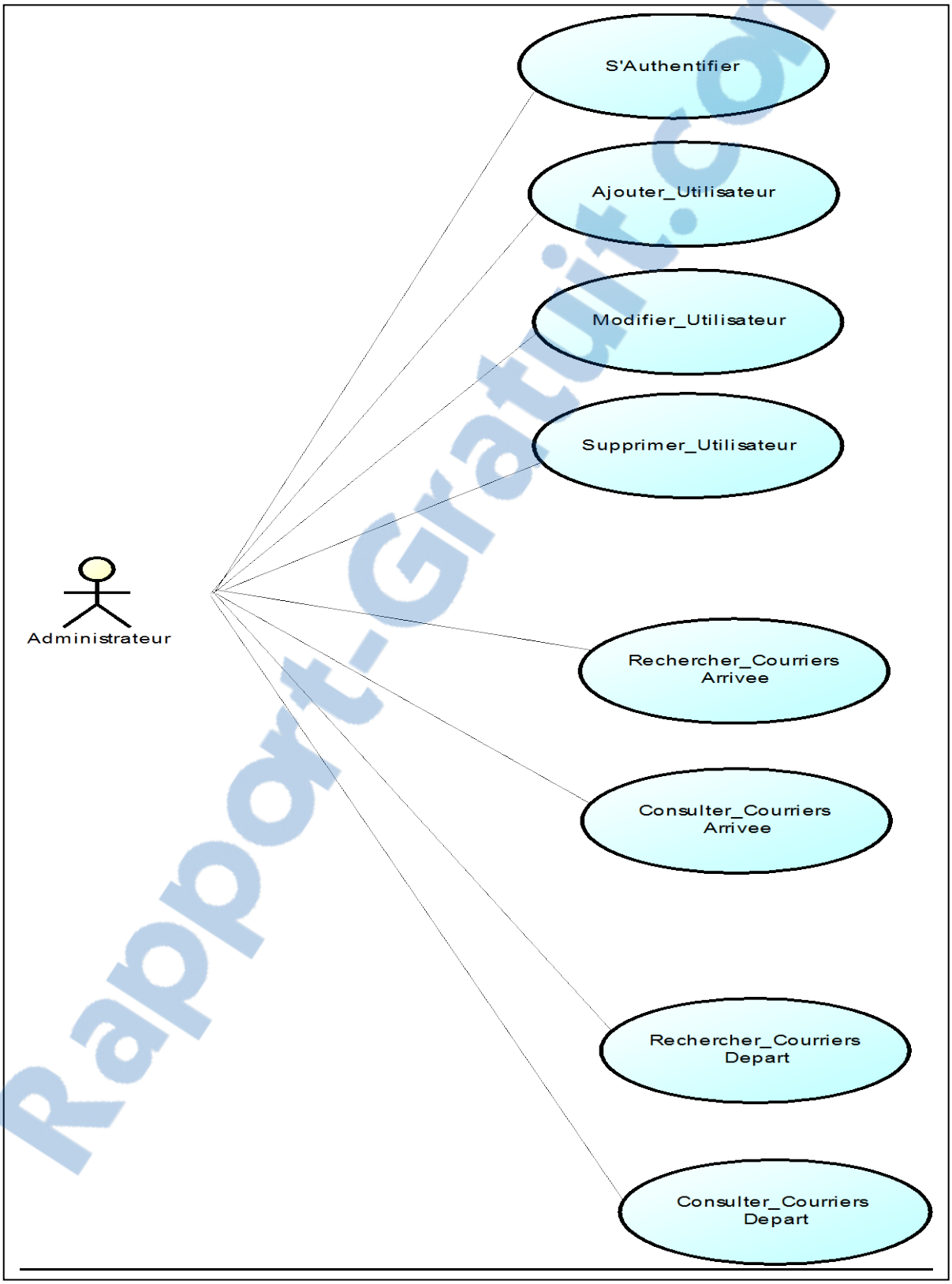

Figure 2.3 : Diagramme des Cas d'Utilisations : « Administrateur ».

2.6 : Description textuelle des Cas d'Utilisations

Les diagrammes des cas d'utilisations décrivent les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisations ; Bien que de nombreux diagrammes d'UML permettent de décrire un cas, il est recommandé de rédiger une « description textuelle », car c'est une forme souple qui convient dans beaucoup des situations pour décrire le déroulement des actions.

L'informaticien Suédois " Ivar JACOBSON " (Qui est principalement connu pour être l'un des concepteurs du langage de modélisation UML) définit en 2005, la description textuelle d'un cas d'utilisation, comme suit : « La description d'un cas d'utilisation définit ce qui survient dans le système quand ce cas est exécuté ; Il correspond à une séquence de transactions exécutées par le système, qui fournit un résultat à un acteur particulier ».

La description textuelle d'un cas d'utilisation permet, donc, de :

- Clarifier le déroulement de la fonctionnalité ;
- Décrire la chronologie des actions qui devront être réalisées ;
- Identifer les parties redondantes pour en déduire des cas d'utilisation plus précises qui seront utilisées, extension ou généralisation / spécialisation ;
- Indiquer d'éventuelles contraintes déjà connues et dont les développeurs vont devoir tenir compte lors de la réalisation / la programmation de l'application.

Les descriptions peuvent aider à améliorer certains « cas d'utilisations » déjà existants ;
Ou même, parfois, à découvrir d'autres « cas d'utilisations » que l'on pourrait ajouter, il s'agit, dans ce cas, d'une nouvelle itération sur les diagrammes des cas d'utilisations.

Nous allons, donc, décrire les scénarios qui explicitent la chronologie des actions qui seront réalisées lors de l'interaction entre les utilisateurs (Les acteurs) et le système lui-même.

2.6.1 : Cas d'Utilisations : « *S'Authentifier* »

La description textuelle des cas d'utilisations « ***S'Authentifier*** » :

1 - Acteur(s) : « Utilisateur » (*« Le Responsable du Bureau d'Ordre », « l'Agent du Bureau d'Ordre » ou même « l'Administrateur » de l'application*).

2 - Objectifs : Ce cas d'utilisation vise à décrire toutes les étapes relatives à l'authentification d'un « *Utilisateur* » à fin de s'identifier pour en accéder à l'application.

3 - Les Pré-Conditions :
- L'application doit être accessible.
- Le compte de « l'Utilisateur » en question, doit être présent dans la Base de Données parmi la liste des utilisateurs autorisés à utiliser l'application.

4 - Le(s) Post-Condition(s) :
- « L'Utilisateur » est authentifié, avec succès.
- La fenêtre d'accueil appropriée de l'application s'affiche.

5 - Le Scénario Nominal :

N_a) « L'Utilisateur » ouvre l'application.

N_b) Le système affiche la fenêtre d'authentification.

N_c) « L'Utilisateur » saisit son Login et son Mot De Passe dans leurs champs appropriés.

N_d) « L'Utilisateur » valide la fenêtre d'authentification, déjà, remplie.

N_e) Le système vérifie l'existence de ce Login et ce Mot De Passe dans la Base de Données parmi la liste des utilisateurs autorisés à utiliser l'application.

N_f) Le système affiche la fenêtre d'accueil de l'application qui est appropriée au profil de l'Utilisateur en question.

6 - Les Scénarios Alternatifs :

A_a) Lors de l'étape de vérification du Login et du Mot De Passe, le système trouve qu'un / des champ(s) (« Login » et / ou « Mot De Passe ») sont vides ou remplis avec des informations non-valides ou erronées : Cet enchainement démarre au point " N_e) du Scénario Nominal ".

A_b) Dans ce cas, le système va demander à cet utilisateur de re-saisir les données des champs de la fenêtre d'authentification : Le scénario reprend au point "N_b) du Scénario Nominal ".

2.6.2 : Cas d'Utilisations : « *Ajouter_Courrier* »

La description textuelle des cas d'utilisations « *Ajouter_Courrier* » :

1 - Acteur(s) : « Utilisateur » (*« Le Responsable du Bureau d'Ordre » ou « l'Agent du Bureau d'Ordre »*).

2 - Objectifs : Ce cas d'utilisation vise à décrire toutes les étapes relatives à l'ajout d'un courrier (« Arrivée » ou « Départ ») dans l'application, par un « *Utilisateur* » à fin de l'enregistrer dans la Base de Données de l'application.

3 - Les Pré-Conditions :

- L'application doit être, déjà, en marche (Ouverte).
- « L'Utilisateur » doit être, déjà, authentifié (C'est-à-dire qu'il a déjà accédé à l'application).

4 - Le(s) Post-Condition(s) :

- Courrier ajouté dans la Base de Données avec succès.

5 - Le Scénario Nominal :

N_a) « L'Utilisateur » demande l'accès à la fenêtre d'ajout d'un courrier.

N_b) Le système affiche l'interface demandée.

N_c) « L'Utilisateur » remplit les champs de la fenêtre concernant le nouveau courrier.

N_d) « L'Utilisateur » valide l'interface (Les champs) déjà remplit, en enregistrant.

N_e) Le système vérifie les champs remplis de la fenêtre.

N_f) Le système enregistre le courrier et l'ajoute dans la Base de Données.

6 - Les Scénarios Alternatifs :

A_a) Lors de l'étape de vérification des champs remplis de la fenêtre d'ajout d'un courrier, le système trouve qu'un / des champ(s) requis sont vides ou remplis avec des données non-valides ou erronées : Cet enchainement démarre au point " N_e) du Scénario Nominal ".

A_b) Le système notifie une / des erreur(s) dans les données saisies : Cet enchainement démarre au point " N_c) du Scénario Nominal ".

2.6.3 : Cas d'Utilisation : « *Ajouter_Utilisateur* »

La description textuelle du cas d'utilisation « *Ajouter_Utilisateur* » :

1 - Acteur : « L'Administrateur » de l'application.

2 - Objectifs : Ce cas d'utilisation vise à décrire toutes les étapes relatives à l'ajout d'un nouvel utilisateur de l'application, par « *L'Administrateur* » à fin de l'enregistrer dans la Base de Données de l'application.

3 - Les Pré-Conditions :

- L'application doit être, déjà, en marche (Ouverte).
- « L'Administrateur » doit être, déjà, authentifié (C'est-à-dire qu'il a déjà accédé à l'application).

4 - Le(s) Post-Condition(s) :

- Nouvel utilisateur ajouté dans la Base de Données avec succès.

5 - Le Scénario Nominal :

N_a) « L'Administrateur » demande l'accès à la fenêtre d'ajout d'un nouvel utilisateur.

N_b) Le système affiche l'interface demandée.

N_c) « L'Administrateur » remplit les champs (Login et Mot De Passe) de la fenêtre concernant le nouvel utilisateur.

N_d) « L'Administrateur » valide l'interface (Les champs) déjà remplit, en enregistrant.

N_e) Le système vérifie les champs remplis de la fenêtre.

N_f) Le système enregistre le nouvel utilisateur et l'ajoute dans la Base de Données.

6 - Les Scénarios Alternatifs :

A_a) Lors de l'étape de vérification des champs remplis de la fenêtre d'ajout d'un nouvel utilisateur, le système trouve qu'un / des champ(s) requis sont vides ou remplis avec des données non-valides ou erronées ou Login déjà existant : Cet enchainement démarre au point " N_e) du Scénario Nominal ".

A_b) Le système notifie une erreur dans les données saisies : Cet enchainement démarre au point " N_c) du Scénario Nominal ".

2.7 : Cycle de vie logiciel

« Le cycle de vie d'un logiciel » désigne toutes les étapes du développement d'un logiciel, dès sa conception jusqu'au sa disparition.

« Le modèle en spirale (*Spiral model*) » a été défini par l'ingénieur Américain "Barry BOEHM" en 1988 dans son article « *A Spiral Model of Software Development and Enhancement* », qui est un modèle de cycle de développement logiciel que pour l'implémentation de versions successives, le cycle recommence en proposant un produit de plus en plus complet et dur. Le cycle en spirale met cependant plus l'accent sur la gestion des risques ;
En effet, c'est un modèle permettant de définir une estimation de l'effort à fournir, c'est-à-dire les charges dans un développement logiciel et la durée que ce dernier prendra, en fonction des ressources allouées ; Le résultat de ce modèle n'est qu'une estimation.

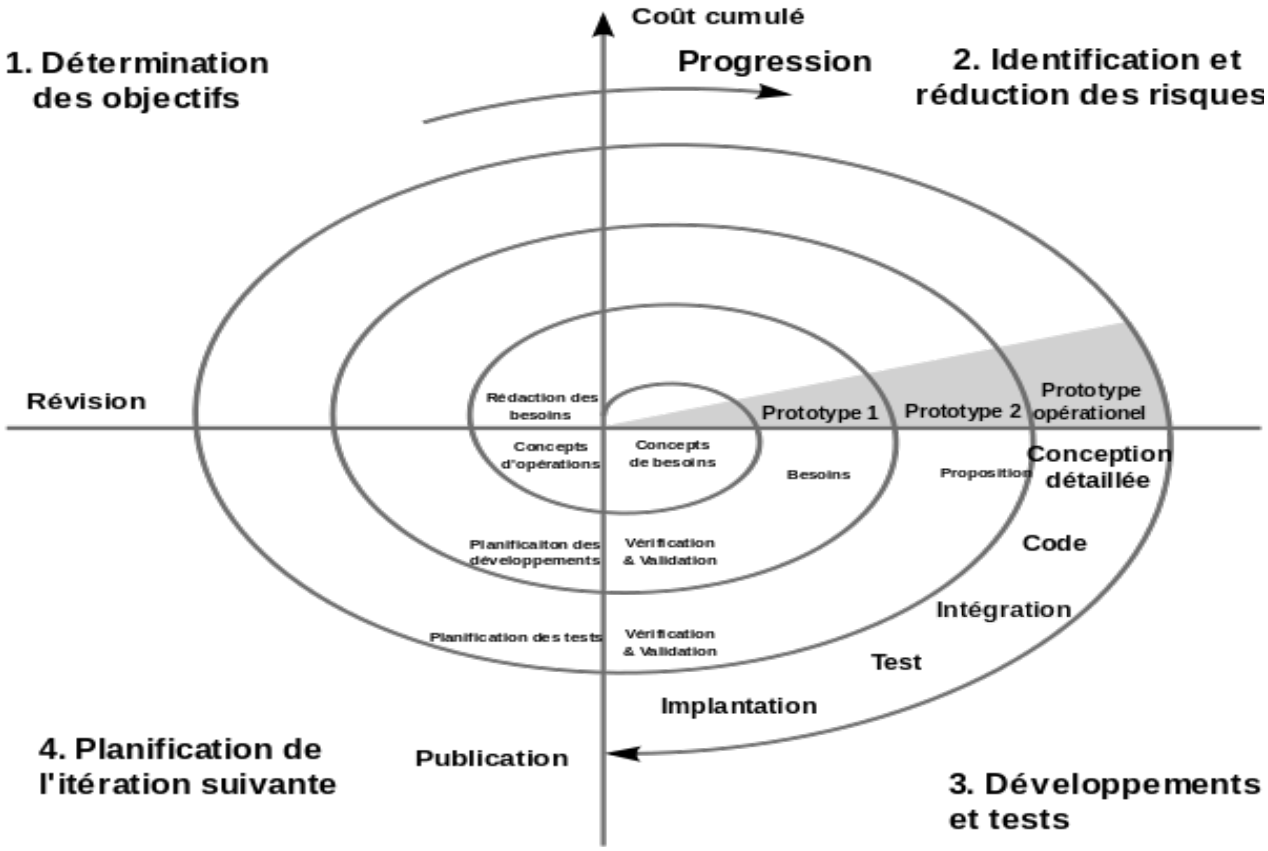

Figure 2.4 : Modèle en spirale.

On distingue quatre phases qui décrivent le déroulement du chaque cycle de vie de la spirale :

1. Détermination -À partir des résultats des cycles précédents, ou de l'analyse préliminaire des besoins- des objectifs du cycle, des alternatives pour les atteindres et des contraintes ;
2. Identification et réduction des risques : Analyse des risques, évaluation des alternatives ;
3. Développements et tests : Vérification de la solution retenue ;
4. Planification de l'itération suivante : Revue des résultats et vérification du cycle suivant.

" BOEHM " identifie des visions erronées de son modèle provenant de simplifications excessives ; Selon lui, les principales erreurs à éviter seraient les suivantes :

- Considérer la spirale comme une suite d'incrémentation en cascade,
- Tous les éléments du projet suivent une seule séquence en spirale,
- Tous les éléments du diagramme doivent être faits dans l'ordre indiqué sans possibilité de retour en arrière.

- ***Points forts du cycle en spirale :***

- Donne des indications sur les risques majeurs sans coût élevé.
- La conception ne doit pas forcément être terminée.
- Le développement se fait en interaction avec les clients (Utilisateurs du logiciel).

- ***Points faibles du cycle en spirale :***

- Ce modèle est complexe.
- La spirale peut être infinie.
- Il est difficile de définir les objectifs et les points de validation intermédiaires entre les différentes étapes.

2.8 : Chronogramme prévisionnel (Théorique) du projet

Le chronogramme servira à organiser le déroulement des phases de notre projet dans le temps, tout en indiquant les durées de chaque phase en précisant leurs enchaînements.

Voici donc, le « diagramme de Gantt » estimatif qui indique d'une façon théorique les durées prévisionnelles des différentes étapes, au départ de notre projet de « Conception et développement d'une application de gestion de courriers (Bureau d'Ordre) d'une entreprise » :

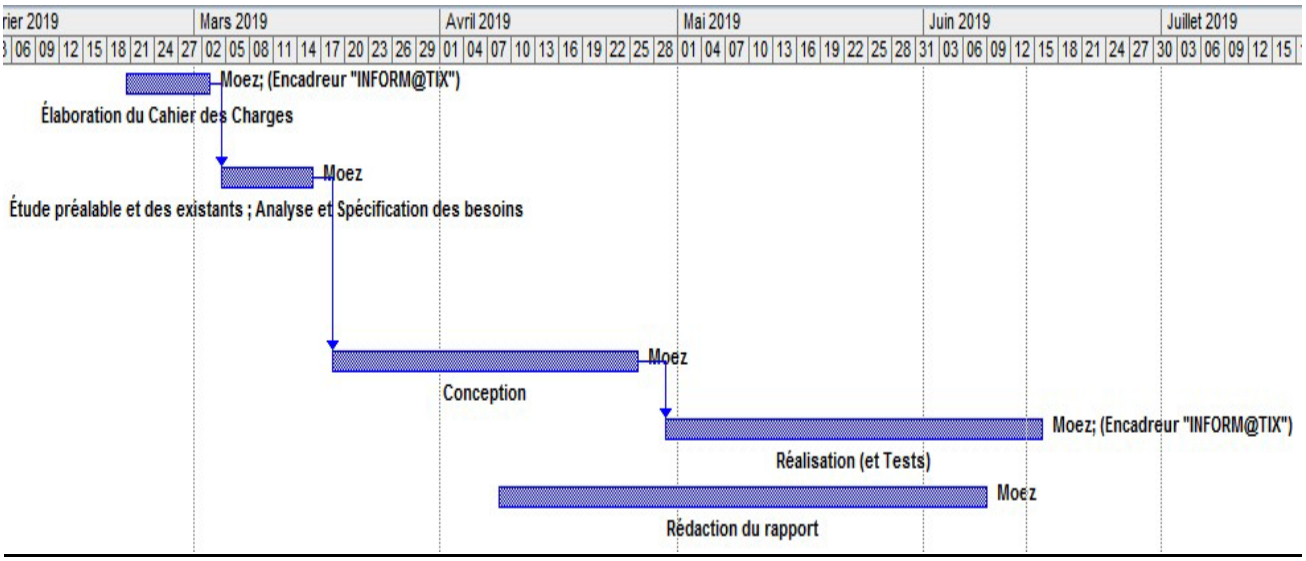

Figure 2.5 : Chronogramme prévisionnel (Théorique) du projet.

Conclusion

On va se baser sur ces besoins fonctionnels et ceux non-fonctionnels, déjà établis, pour pouvoir réaliser notre conception et les différents diagrammes de notre nouvelle application.

CHAPITRE 3 :

CONCEPTION

Introduction

Dans ce chapitre, nous présentons notre projet en faisant recours à une démarche sipmle utilisant des concepts claires considérés dans l'élaboration des différentes étapes de la gestion de la plupart des projets, que nous allons adpater pour notre projet de « Gestion de courriers (Bureau d'Ordre) d'une entreprise », et ceci est dans le but d'approfondir notre compréhension du sujet et avoir une idée plus claire sur les exigences de notre nouveau système et ses fonctions attendues.

À l'issue des étapes pécédentes « d'étude préalable et de l'existant » et « analyse et spécification des besoins », nous avons pu cerner les besoins et par conséquent les attentes de notre nouvelle application de « Gestion de courriers (Bureau d'Ordre) » et nous allons déduire le schéma conceptuel auquel nous nous alignerons pour la réalisation de notre nouvelle application ;
« *La conception* » de notre nouveau système fera l'objet de cette phase.

La conception est une étape très importante qui a pour objectif de faire l'étude des données et des traitements à effectuer. C'est en général, dans cette phase, que s'appliquent les techniques de modélisation, qui ont comme objectif de constituer une représentation claire et cohérente des données manipulées dans le Système d'Information.

Cette phase de conception décrira de manière précise et concise, en utilisant le formalisme UML, le fonctionnement du futur système, afin d'en faciliter la phase de réalisation.

3.1 : Modélisation des Besoins

3.1.1 : Description d'UML

UML « *Unified Modeling Language* » est un Langage Unifié pour la Modélisation objet ; UML est un Langage (Basée sur : Les notations, grammaire, sémantique) et pas une Méthode (Basée sur : Le recueil des besoins, analyse, conception, mise en oeuvre, validation, ...) ; Donc c'est un langage de modélisation graphique qui est à la base conçu pour fournir une méthode pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception Orientée Objet.

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel et / ou application Orienté Objet en offrant un standard de modélisation, pour représenter l'architecture logicielle.

UML est un langage visuel constitué d'un ensemble de schémas, appelés « *des diagrammes* », qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour concevoir et représenter le logiciel et / ou l'application à développer : Son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées, etc ... ;

Réaliser ces diagrammes revient donc à modéliser les besoins de l'application à développer, ces diagrammes modélisant autant de vues distinctes pour représenter des concepts particuliers du système d'information vont donner sur un tel système des vues partielles, analogues chacune à une photographie d'un statut, et dont la conjonction donnera une idée utilisable et réalisable en pratique.

3.1.2 : Modélisation Statique ou Diagrammes Structurels du système

La Modélisation Statique d'un système ou les Diagrammes Structurels (En Anglais : « *Structural Diagrams* ») -Ce que le système " EST "- : Consiste à décrire les composantes de ce système sans tenir compte de leurs évolutions dans le temps en se référant aux éléments d'un système et les relations entre elles ; Ainsi ces diagrammes sont utilisés pour modéliser " les choses " (C'est-à-dire les classes, les objets, les composants physiques, ...) qui composent un modèle, en outre, ils sont utilisés pour modéliser les relations et les dépendances entre les éléments.

3.1.2.a : Le Diagramme des Classes

Les diagrammes des classes permettent de spécifier " QUI " intervient à l'intérieur du système.

Nous essayons, donc, de définir des mécanismes d'extraction des données traitant de « *L'aspect statique* » des données (Les données et leurs structures), pour générer les structures et les objets des classes.

« Les données » représentent l'aspect statique du système d'information : " CE QUI EST ". Les données présentent, dans leur signification, une certaine stabilité et une invariance dans le temps. Cette signification (Sémantique) est essentiellement déterminée par le type d'activité.

Tout système Orienté Objet est organisé autour « Des Classes » ; Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets, et qui est un concept abstrait qui permet de représenter toutes les entités d'un système ; Une classe peut donc représenter tout élément devant être modélisé.

Une classe est donc, un ensemble de « *fonctions* » et de « *données* » (Attributs) qui sont liées ensemble par un champ sémantique, présentant la description formelle d'un ensemble d'objets.

- ***Le Diagramme des Classes du système :***

Dans cette étape, nous schématisons le Diagramme des Classes (En Anglais : « *Class Diagram* ») qui est utilisé pour représenter l'architecture conceptuelle des classes que le système utilise et les types des ensembles de leurs objets composants le système, ainsi que les différentes relations entre celles-ci ; *(« Une classe » est une description d'un ensemble d'objets partageant les mêmes attributs et opérations*).

Le diagramme des classes montre la structure interne du système. Il permet de fournir une représentation abstraite des objets qui interviennent dans le système qui s'interagissent.

Il s'agit, donc, d'une vue statique, car on ne tient pas compte du facteur temporel dans la structure du système. Le diagramme des classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Le diagramme des classes permet de modéliser les classes du système et leurs relations (*Associations*) indépendamment d'un langage de programmation particulier.

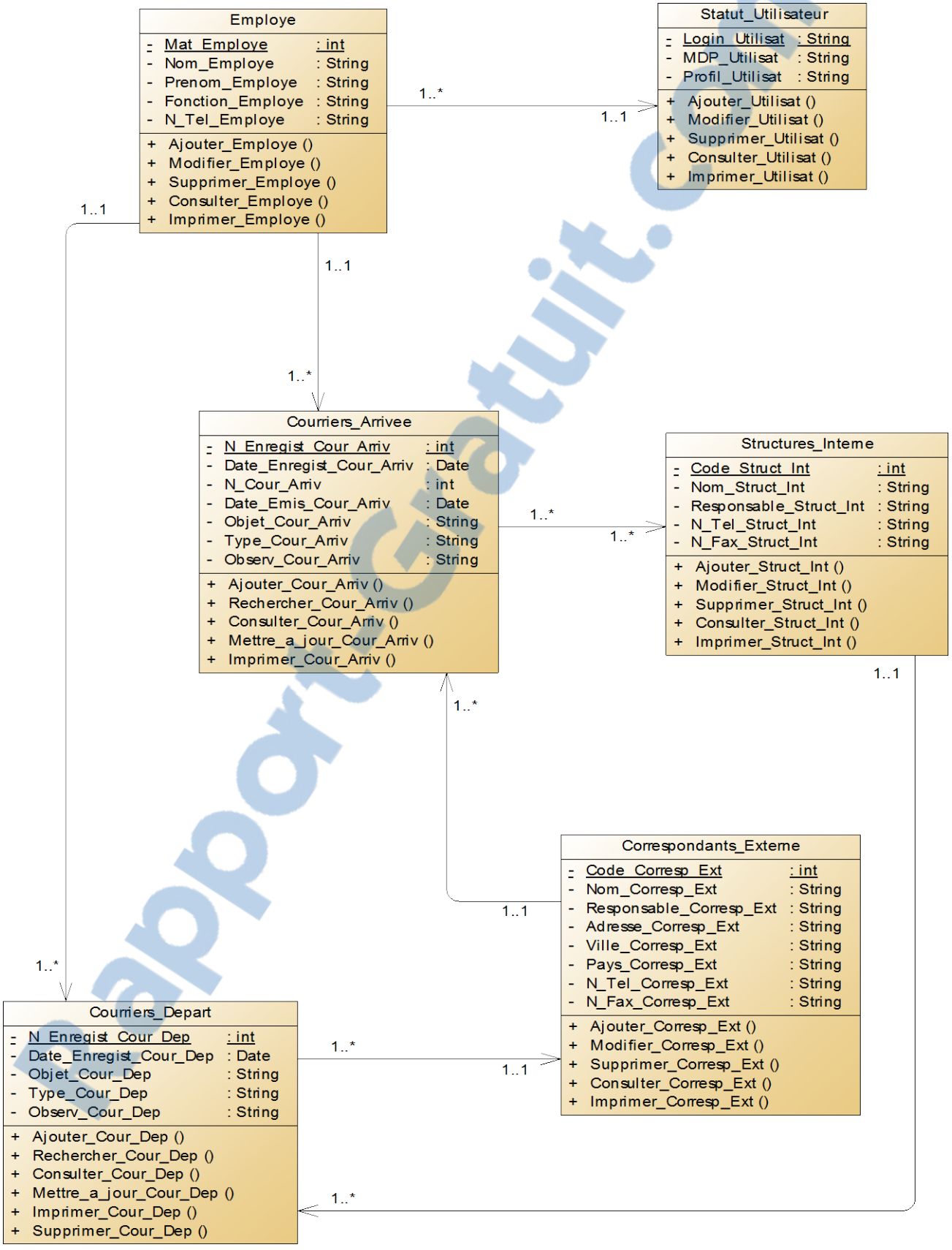

Figure 3.1 : Diagramme des Classes.

3.1.2.b : Le Diagramme de Composants

Le Diagramme de Composants (En Anglais : « *Component Diagram* ») décrit le système modélisé sous forme de composants réutilisables, en permettant de modéliser les relations entre les composants physiques du système et l'architecture interne de l'application, tout en mettant en évidence leurs relations de dépendance qui décrivent la façon selon laquelle les composants logicielles seront réalisées ; Donc ce diagramme décrit l'organisation du système du point de vue des éléments logiciels comme les modules (Fichiers sources, bibliothèques, exécutables, ...), des données (Fichiers, Bases de Données, ...) ou encore des éléments de configuration (Paramètres, scripts, fichiers de commandes, ...).

Ce diagramme permet de mettre en évidence les dépendances entre les composants du système de point de vue physique, tels qu'ils sont mis en œuvre (" QUI, UTILISE QUOI ? ").
Ainsi, le diagramme de composants va identifier, donc, les différents événements venant du monde externe et montre l'enchaînement dans le système que provoquent ces événements externes qui se produisent à un moment donné dans le temps.

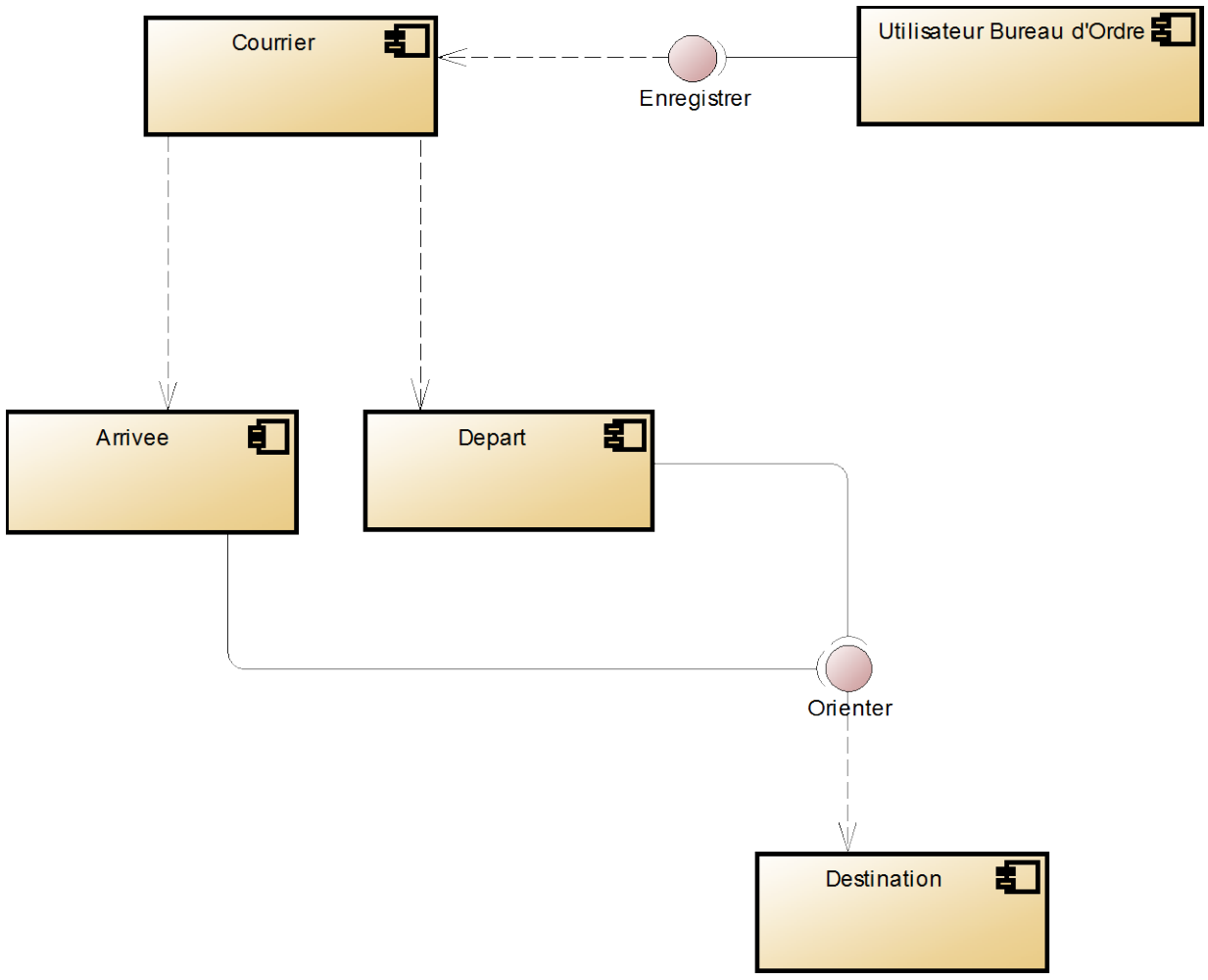

Figure 3.2 : Diagramme de Composants.

3.1.2.c : Le Diagramme de Paquetages

Dans cette partie, nous schématisons le Diagramme de Paquetages (En Anglais : « *Package Diagram* ») dans le quel nous allons mettre en évidence les services offerts par l'application qui décrivent les modules de notre application, et ce-ci en décomposant notre système en plusieurs parties (Appelées « *paquetage* ») ; « Un paquetage » est, donc, un regroupement ou un conteneur logique permettant de regrouper et d'organiser les différents éléments du système (Regroupement de classes, diagrammes, fonctions, ...).
Cela permet de clarifier le modèle en l'organisant par représentation des paquetages composant notre système, ainsi que les relations qui lient ces différents paquetages :

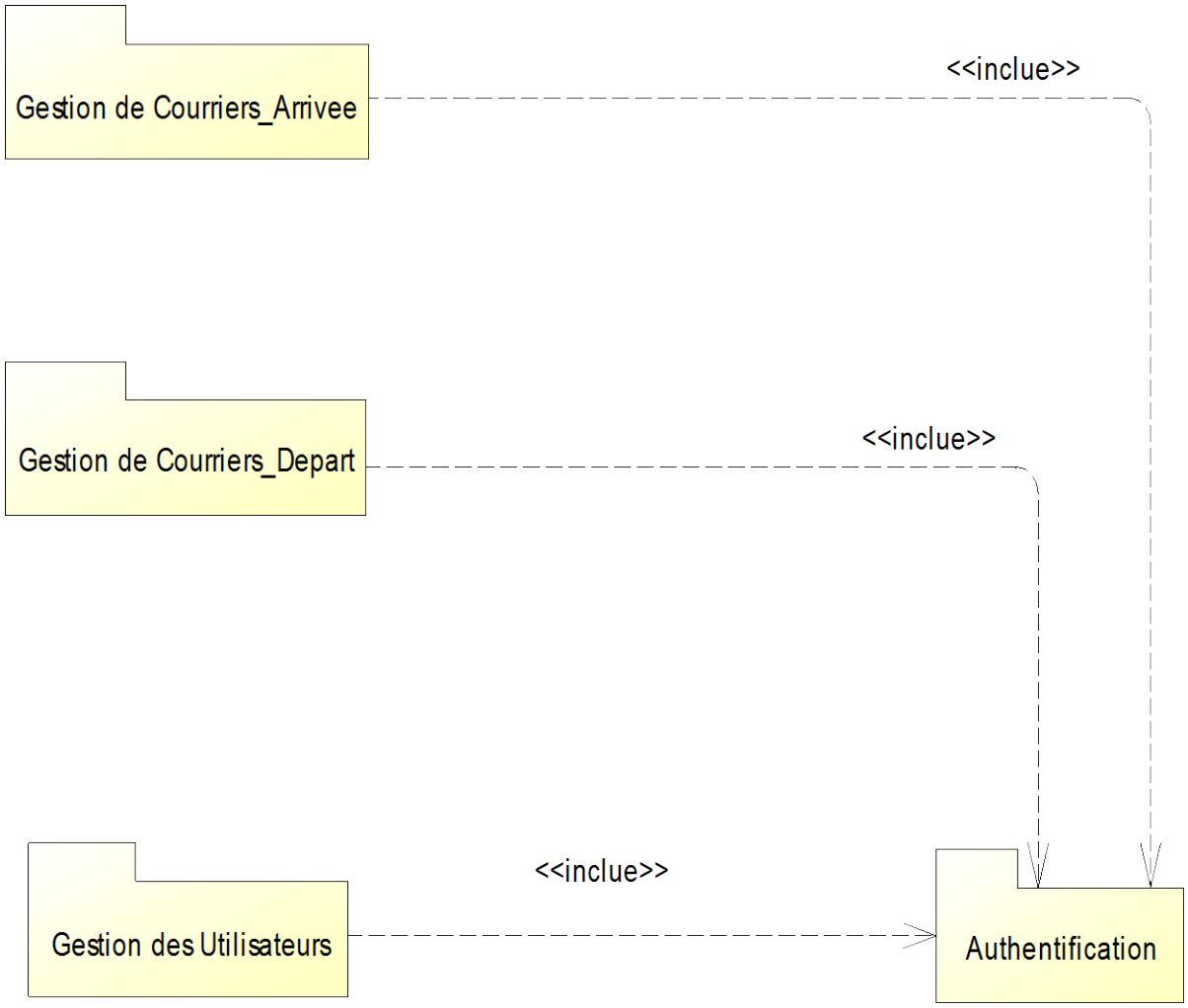

Figure 3.3 : Diagramme de Paquetages.

3.1.3 : Modélisation Dynamique ou Diagrammes Comportementaux du système

La Modélisation Dynamique d'un système ou les Diagrammes Comportementaux (En Anglais : « *Behavioral Diagrams* ») -COMMENT le système " ÉVOLU " ? et ce que le système " FAIT "- : Consiste à décrire le comportement de ce système lors de sa réaction à son environnement, afin de définir l'évolution des divers états des éléments et des informations à modéliser du modèle dans le temps, en distinguant les interactions entre les éléments du système.

Le modèle dynamique ou les diagrammes de comportements capturent, donc, les variétés de l'interaction et les instantanées états dans un modèle comme il l'exécute et l'utilise pour exprimer et modéliser le comportement du système au fil du temps, suivi de COMMENT le système va agir dans un environnement réaliste et en observant les effets d'une opération ou d'un événement, y compris ses résultats.

3.1.3.a : Les Diagrammes des Séquences

Nous essayons de définir des mécanismes d'extraction des données traitant de « *l'aspect dynamique* » (Le comportement des données) et décrivent de manière temporelle COMMENT les éléments du système (Les objets) interagissent entre eux et avec « les acteurs » à travers des messages échangés, pour générer celui des classes en s'appuyant sur un processus d'analyse des éléments nécessaires aux méthodes (Propriétés, objets et méthodes).

Les traitements représentent l'aspect « dynamique » ou « cinématique » (Mais par abus de langage, on parlera de « dynamique » plutôt que de « cinématique) du Système d'Information : " CE QUI SE FAIT ". Les traitements, et en particulier leur organisation, présentent une plus grande variabilité, en fonction essentiellement de l'évolution des besoins.

Dans cette phase, nous schématisons les Diagrammes des Séquences (En Anglais : « *Sequence Diagram* » ou « *Interaction Diagram* ») qui représentent les interactions entre les objets de point de vue temporel pour détailler un objet du monde réel sous forme d'un scénario d'un cas d'un diagramme de cas d'utilisations en y mettant l'accent sur la succession chronologique séquentielle du déroulement des opérations ou des traitements et les interactions réalisées entre les éléments du système et / ou ses acteurs, et l'ordre d'envoi des messages.

- ***Diagramme des Séquences des Cas d'Utilisations : « S'Authentifier » :***

(*« L'Utilisateur », dans cette séquence : Peut être un « Utilisateur de l'application » appartenant au service du Bureau d'Ordre (C'est-à-dire « le Responsable du Bureau d'Ordre » ou « l'Agent du Bureau d'Ordre »), ou même « l'Administrateur » de l'application*).

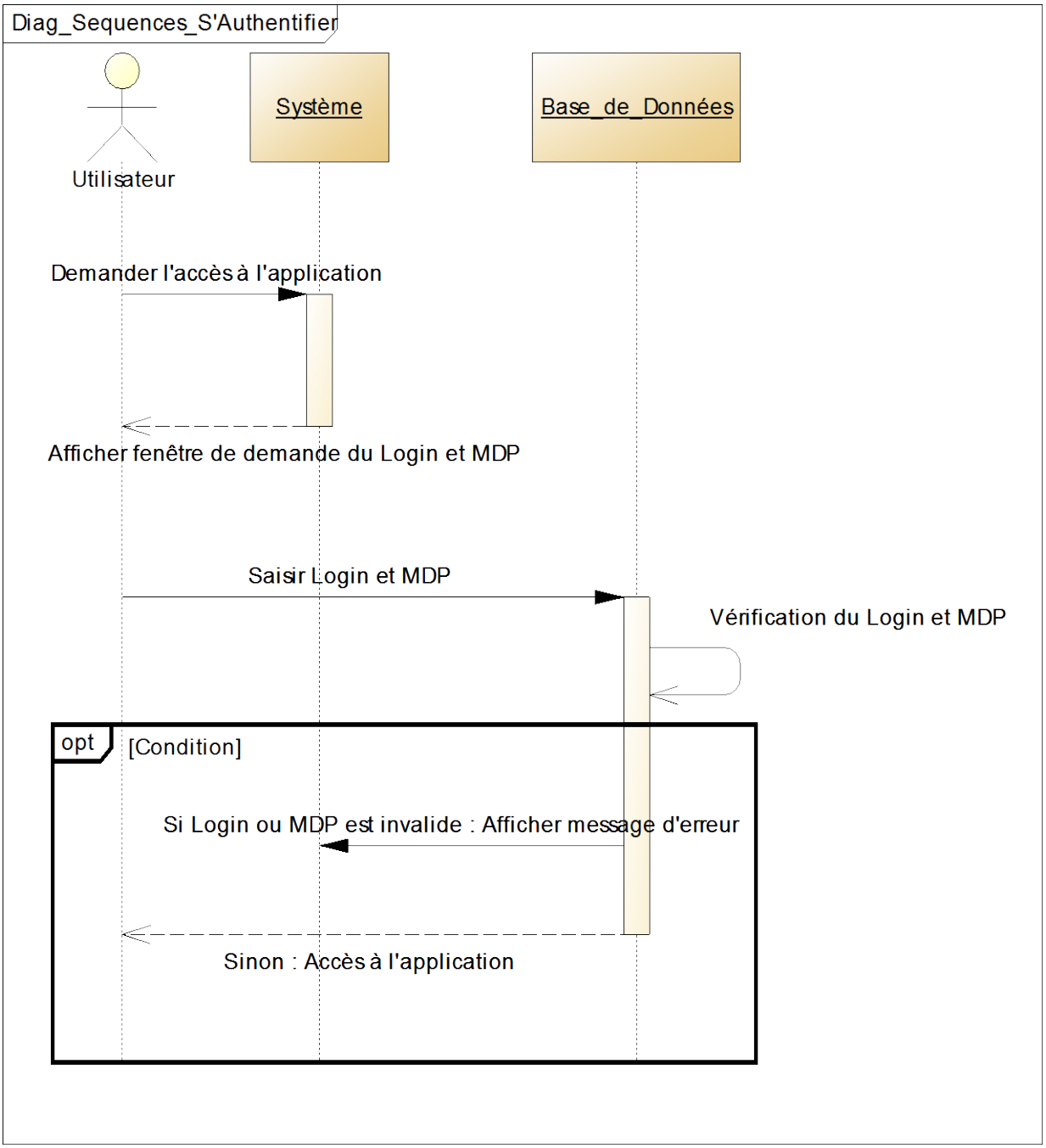

Figure 3.4 : Diagramme des Séquences des Cas d'Utilisations : « S'Authentifier ».

- ***Diagramme des Séquences des Cas d'Utilisations : « Ajouter_Courrier » :***

(*« Le Courrier » : Peut être un « Courrier Arrivée » à l'entreprise, ou même un « Courrier Départ » de l'entreprise ; « L'Utilisateur », dans cette séquence : Ne peut être qu'un « Utilisateur de l'application » appartenant au service du Bureau d'Ordre (C'est-à-dire « le Responsable du Bureau d'Ordre » ou « l'Agent du Bureau d'Ordre »)*).

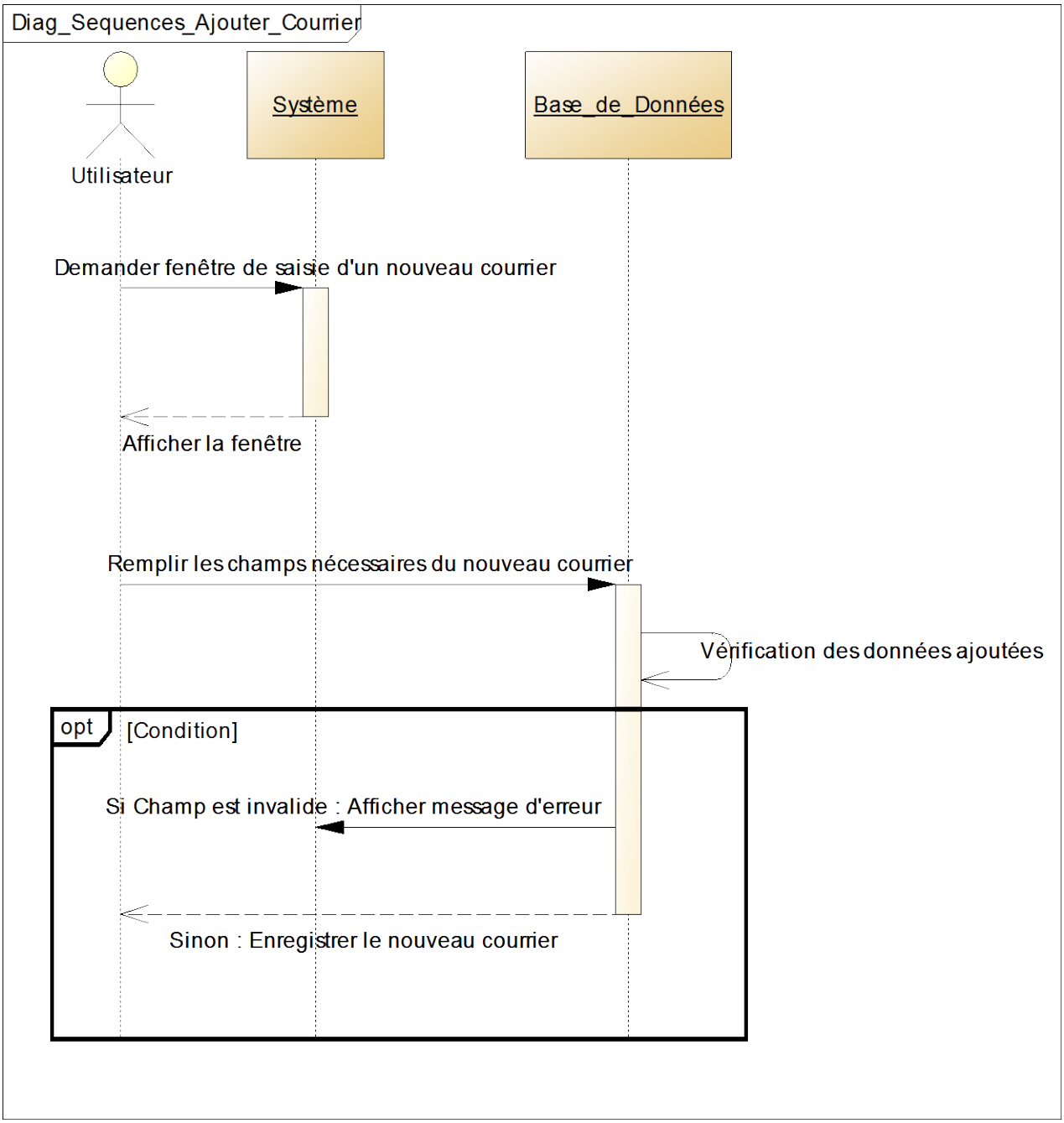

Figure 3.5 : Diagramme des Séquences des Cas d'Utilisations : « Ajouter_Courrier ».

- ***Diagramme des Séquences de Cas d'Utilisation : « Ajouter_Utilisateur » :***

(*« L'Utilisateur », dans cette séquence : Ne peut être que « l'Administrateur » de l'application lui-même*).

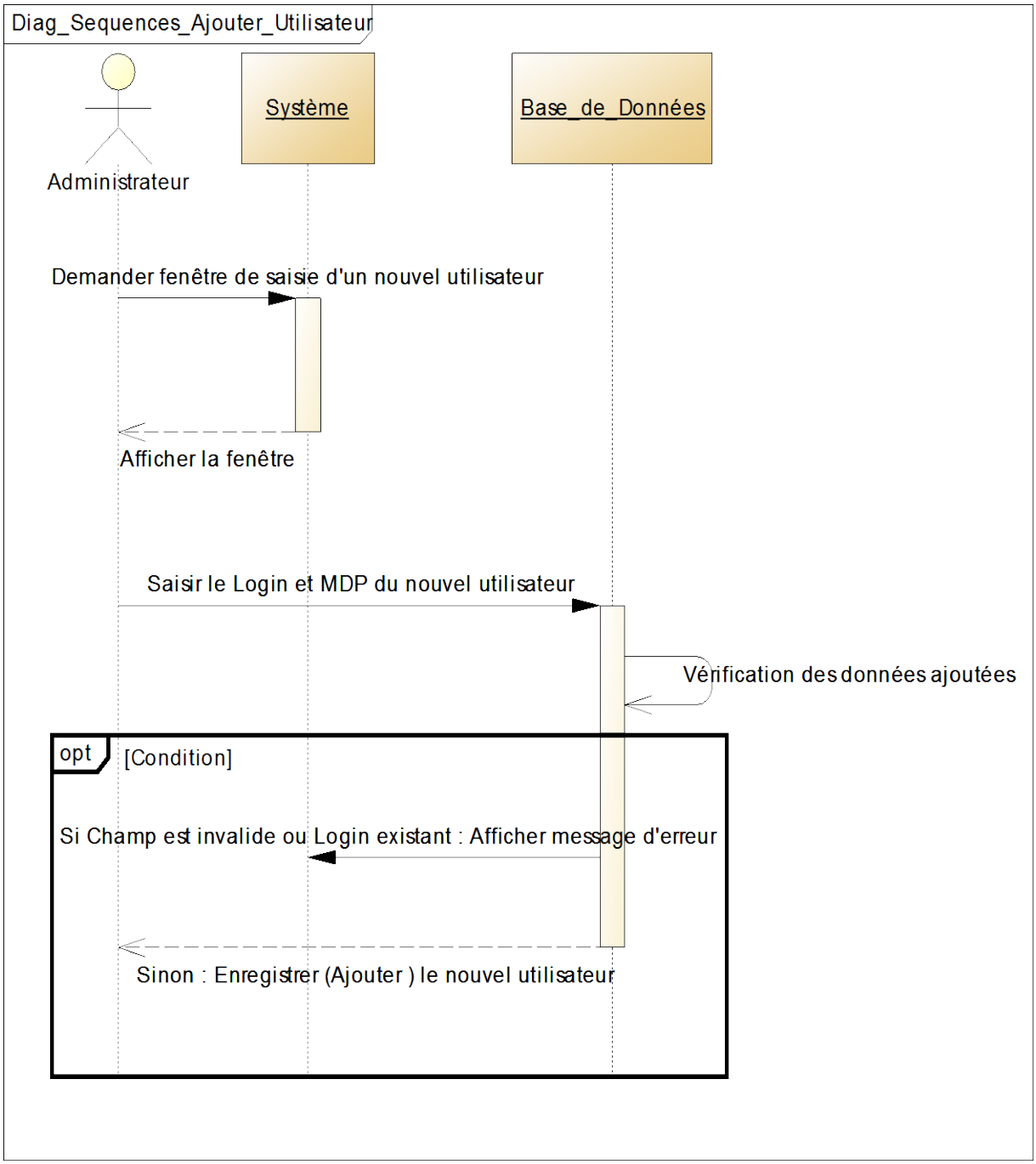

Figure 3.6 : Diagramme des Séquences de Cas d'Utilisation : « Ajouter_Utilisateur ».

3.1.3.b : Les Diagrammes d'États-transitions

Le diagramme d'états-transitions, (En Anglais : « *State-machine Diagram* ») -(*« Un état » est une abstraction des valeurs des attributs d'un objet, il correspond, donc, à une situation significative ou une condition dans laquelle est l'objet ; « Une transition » représente le passage instantané d'un objet d'un état vers un autre et qui se déclenche par un événement*)-, représente et décrit la façon dont évoluent les objets appartenant à une même classe donnée (Sous forme de graphes d'états), leurs états possibles, ainsi que le comportement d'une classe, ou du système ou même de ses composants en termes d'états ainsi que les transitions entre les états (Qui les relis par des arcs orientés), en se basant sur le principe d'automate pour décrire la dynamique des objets qui représente les changements des états d'un objet ou d'un composant en réponse aux interactions avec d'autres.

- ***Diagramme d'États-transitions : « S'Authentifier » :***

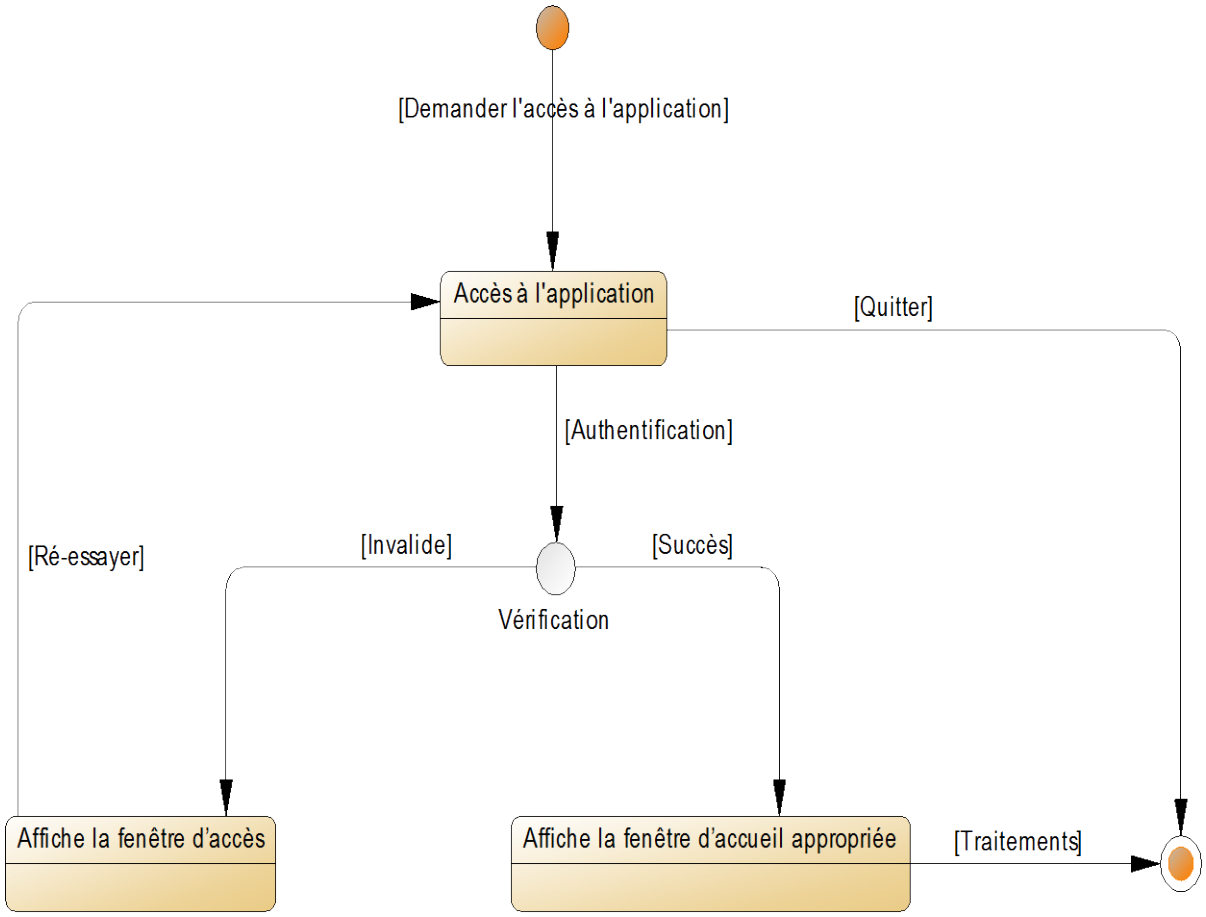

Figure 3.7 : Diagramme d'États-transitions : « S'Authentifier ».

- ***Diagramme d'États-transitions : « Ajouter_Courrier » :***

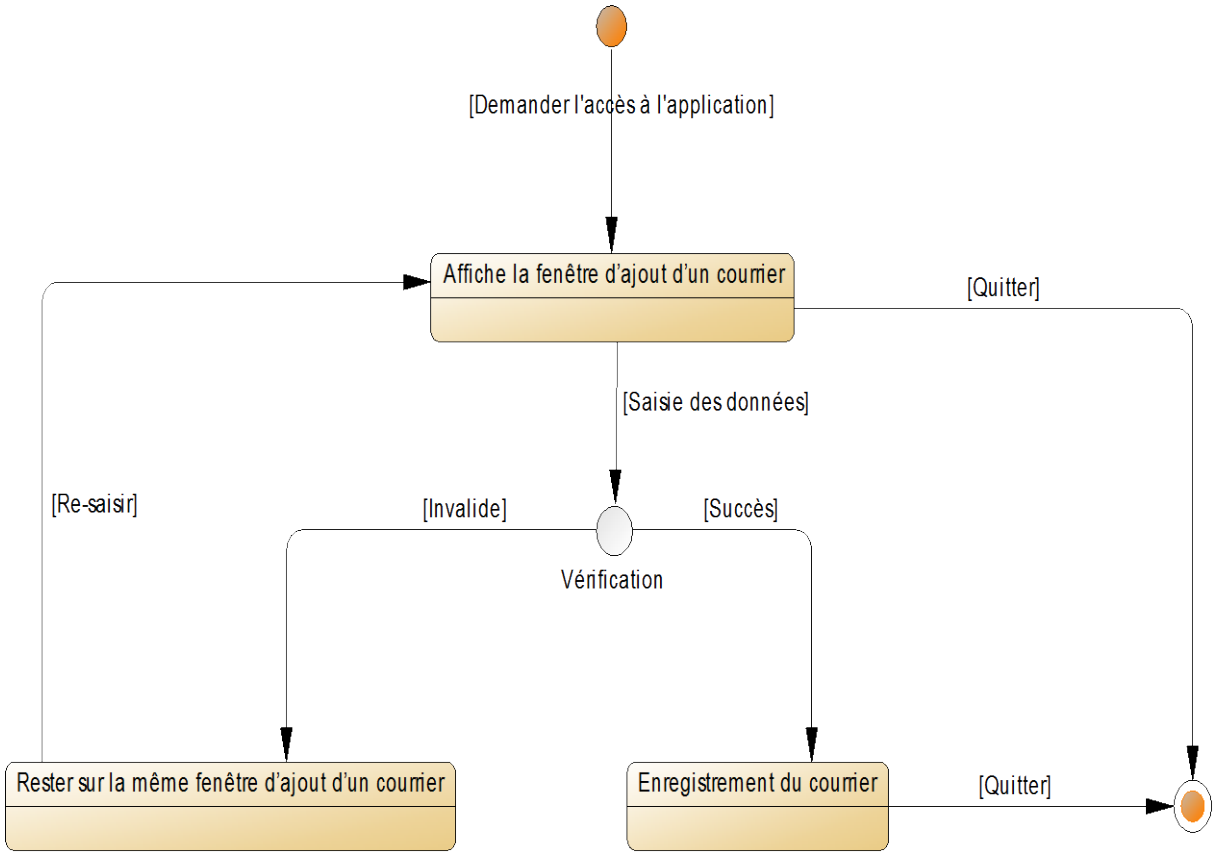

Figure 3.8 : Diagramme d'États-transitions : « Ajouter_Courrier ».

3.1.3.c : Les Diagrammes d'Activités

Le Diagramme d'Activités (En Anglais : « *Activity Diagram* ») (*« Une activité » représente une exécution d'un mécanisme ou un déroulelment d'étapes séquentielles*), est assez proche du « Diagramme d'états-transitions » mais permettant de modéliser et décrire les traitements et le comportement des opérations en termes d'actions, en représentant graphiquement sous forme de flux ou d'enchaînement d'activités le comportement d'une méthode, ou du système ou même de ses composants, ou le déroulement de réalisation d'un cas d'utilisation, pour permettre de répondre à la question : " QUI, FAIT QUOI ? " ; Le diagramme d'activités n'est autre que la représentation du processus telle qu'elle a été élaborée lors du travail qui a préparé la modélisation, il montre l'enchaînement des activités qui concourent au processus.

- ***Diagramme d'Activités : « Courriers_Arrivee » :***

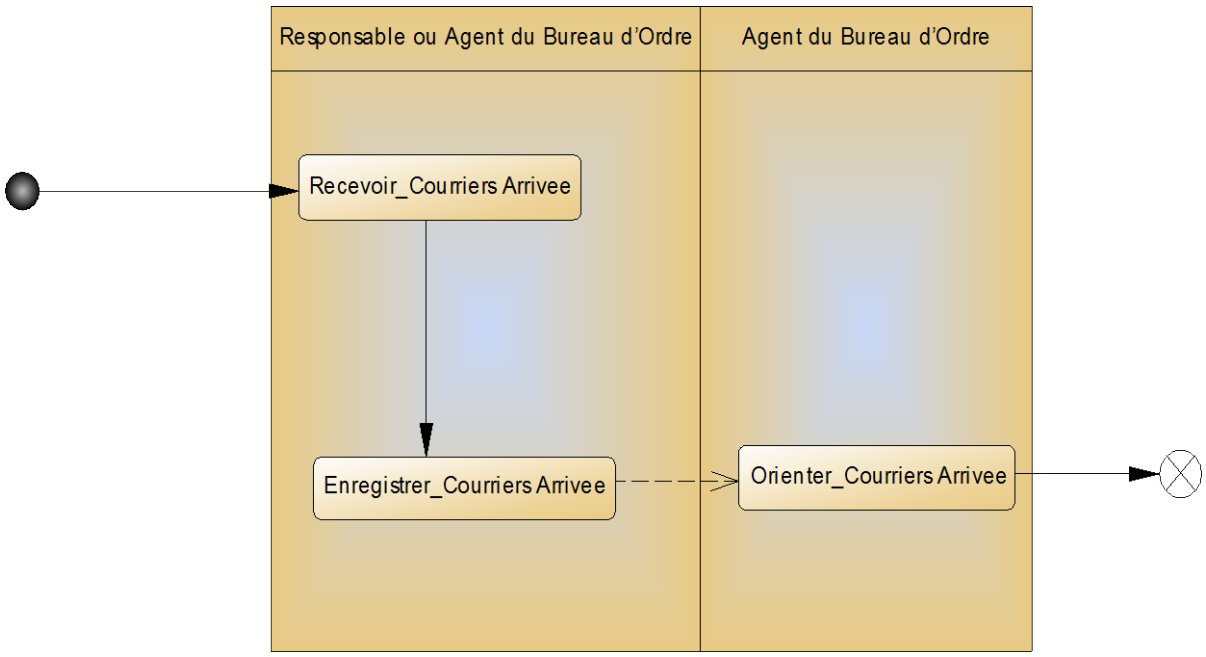

Figure 3.9 : Diagramme d'Activités : « Courriers_Arrivee ».

- ***Diagramme d'Activités : « Courriers_Depart » :***

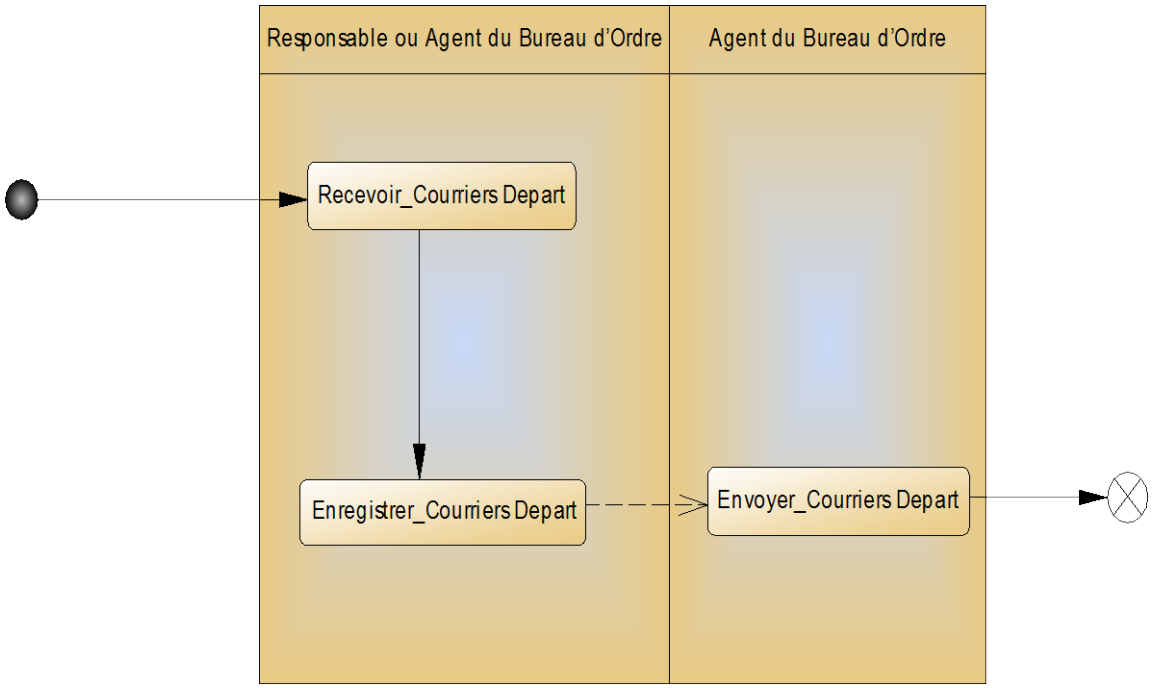

Figure 3.10 : Diagramme d'Activités : « Courriers_Depart ».

3.2 : Schéma Relationnel de la Base de Données

Le modèle relationnel représente la Base de Données comme un ensemble de tables, sans préjuger de la façon dont les informations sont stockées dans la machine. Les tables constituent donc la structure logique du modèle relationnel. Au niveau physique, le système est libre d'utiliser n'importe quelle technique de stockage (Fichiers séquentiels, indexage, adressage dispersé, séries de pointeurs, compression, ...) dès lors qu'il est possible de relier ces structures à des tables au niveau logique. Les tables ne représentent donc qu'une abstraction de l'enregistrement physique des données en mémoire.

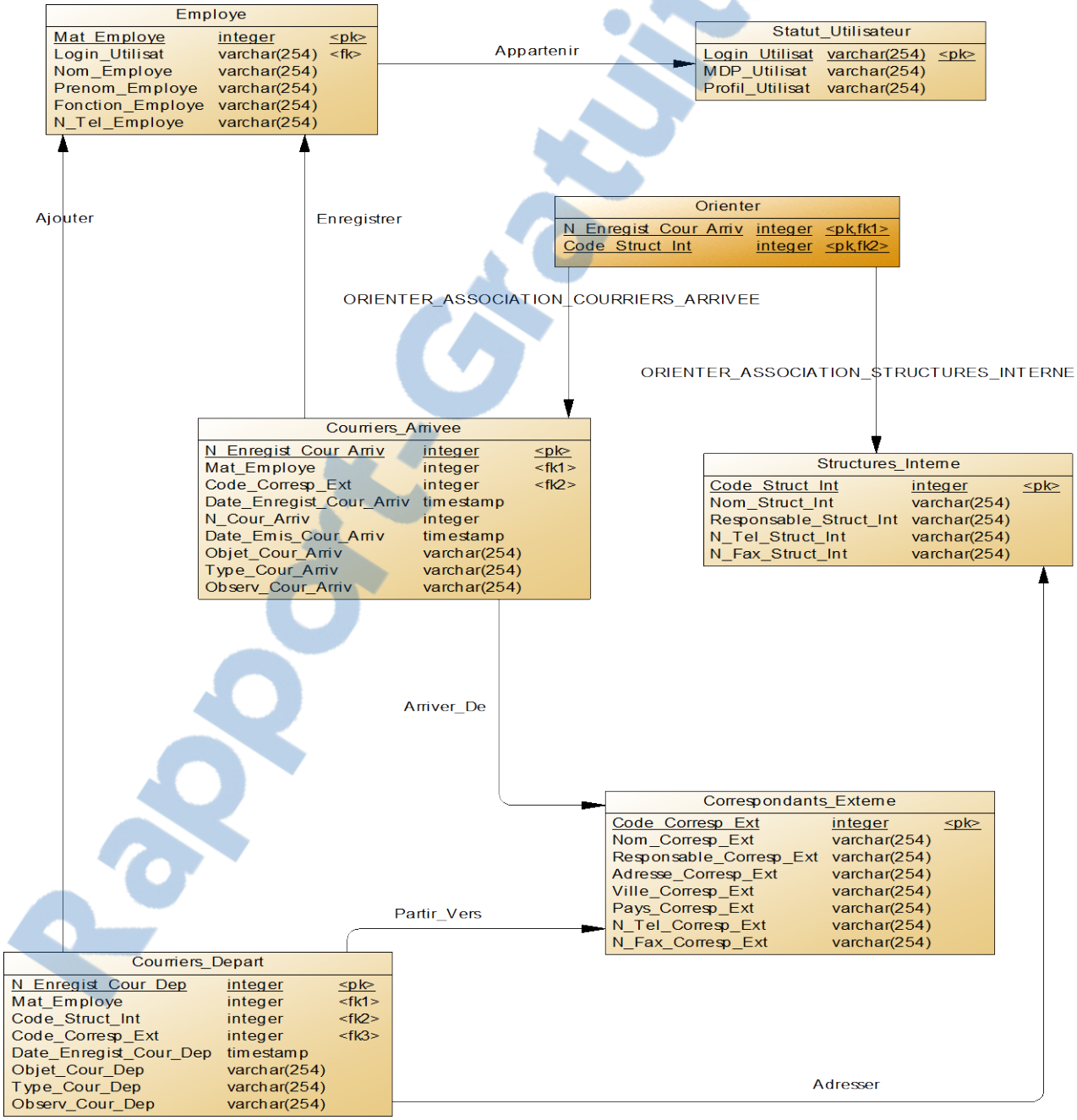

Figure 3.11 : Schéma Relationnel de la Base de Données.

Conclusion

Dans cette étape de conception, toutes les questions concernant la manière de réaliser le système à développer ont été élucidées. Le produit obtenu est un modèle graphique (*Ensemble de diagrammes*) prêt à être codé.

Après avoir achevé cette phase de « conception » des différents diagrammes de l'application et l'élaboration du schéma de la Base de Données de notre application, on va entamer, dans la phase suivante, la partie réalisation et dans laquelle on s'assurera que le système sera prêt pour être exploité par les utilisateurs finaux.

CHAPITRE 4 :

RÉALISATION

Introduction

À l'issue de l'étape pécédente de « la conception », nous avons pu schématiser les différents modèles conceptuels de diagrammes de notre nouvelle application de « gestion de courriers (Bureau d'Ordre) », auxquels nous nous alignerons pour la réalisation et le développement de notre nouvelle application, dont fera l'objet cette phase ; et ce-ci est dans le but de développer une application de «gestion de courriers (Bureau d'Ordre) » adaptable aux différentes conditions mentionnées dans les phases précédentes et d'être capable d'automatiser les tâches détaillées.

Cette phase de « réalisation » développera de manière précise et concise -En utilisant le langage de programmation « Delphi » en se connectant à une Base de Données-, le fonctionnement du système à réaliser, afin de concrétiser l'implémentation et l'exploitation future de la dite application.

4.1 : Le Diagramme de Déploiement

Le diagramme de déploiement (En Anglais : « *Deployment Diagram* ») se rapproche encore plus de la réalité physique, puisqu'il identifie et représente la structure physique du système informatique et la répartition des composants logiciels sur ce système et des ressources physiques (PC, Modem, Station de travail, Serveur, etc ...), leurs dispositions physiques (Connexions) dans un monde réel et les répartitions des programmes exécutables des composants sur ces différents éléments 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, montrant la configuration physique des différents matériels qui participent à l'exécution ou l'exploitation du système, ainsi que les artefacts qu'ils supportent.

Les éléments utilisés par un diagramme de déploiement sont principalement « *les nœuds* » (Représentés par « *des boîtes en trois dimensions* »), « *les composants* » (Qu'ils soient logiciels ou matériels), « *les associations* » (Représentés par « *les lignes entre les nœuds* ») et « *les artefacts* » logiciels qui sont déployés (Représentés par « *les petites formes à l'intérieur des boîtes* »).

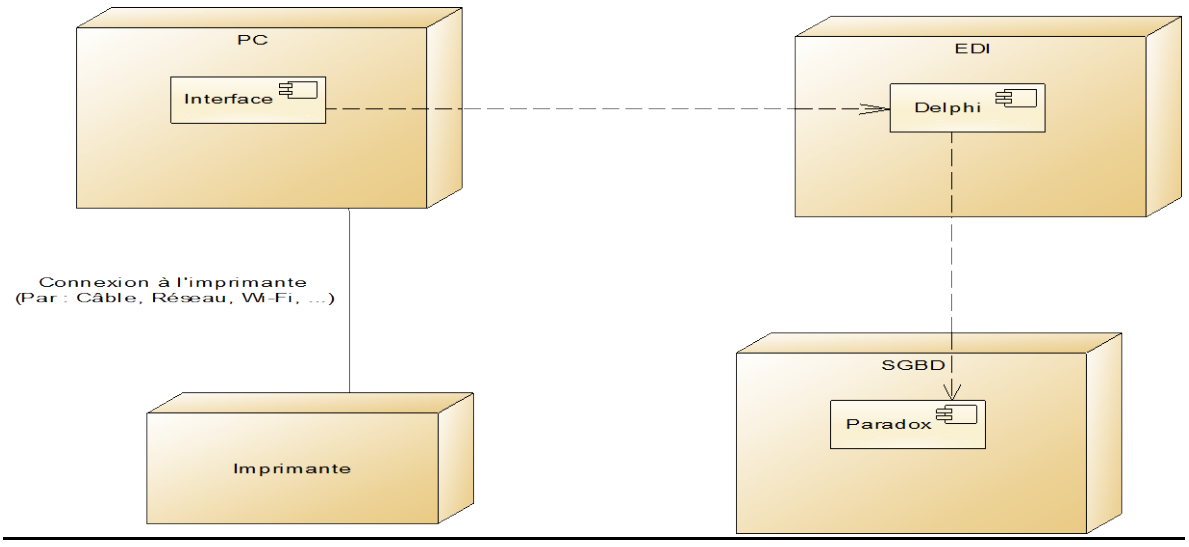

Figure 4.1 : Diagramme de Déploiement.

4.2 : Environnement de travail

Dans la phase de développement d'une application ou de la mise en œuvre d'un Système d'Information, la réalisation de notre système consiste à faire les choix de technologies d'environnements et d'organisation de composantes matérielles et logicielles les plus adaptées aux besoins réels et aux contraintes.
Ces choix sont ensuite relayés au sein de notre projet, guidant la réalisation et permettant l'implémentation d'une application performante et robuste.

De nos jours et avec l'évolution très rapide des technologies dans tous les domaines, plusieurs environnements matériels et logiciels ont existé, pour répondre aux attentes des utilisateurs et des clients également.

4.2.1 : Environnement matériel

L'environnement matériel de travail que nous avons utilisé pour l'élaboration de la conception et ensuite pour le développement de notre application est :

- « *Micro-Ordinateur* », avec les caractéristiques techniques suivantes :
  - *Micro-Processeur* : Intel (Dual-Core).
  - *Horloge* : 2,5 GHz.
  - *Mémoire (RAM)* : 4 Go.
  - *Disque Dur* : 500 Go.
- « *Imprimante - A4* ».

4.2.2 : Environnement logiciel

Notre application est conçue et developpée sous les environnements logiciels suivants :

- **« *Système d'Exploitation* » :** Cette application est conçue et développée sous l'environnement du Système d'Exploitation « **Microsoft Windows 7** ».

Figure 4.2 : Logo de « Microsoft Windows 7 ».

- **« *Atelier de Génie Logiciel (AGL)* » : « PowerAMC » :**

Figure 4.3 : Logo de « PowerAMC ».

« PowerAMC », (Anciennement crée par la société « SDP » et connu sous le nom de : « AMC*Designor »), -racheté par « Powersoft » qui lui-même a été racheté par « Sybase » en 1995 ; Que depuis 2010 « Sybase » elle même appartient à l'éditeur Allemand « SAP »-, est un logiciel de conception qui permet de modéliser les traitements informatiques et leurs Bases de Données associées.

« PowerAMC » permet de réaliser presque tous les types de modèles et diagrammes informatiques. « PowerAMC » permet de travailler et modéliser avec la méthode de conception « MERISE » et également « UML » ; Cela permet d'améliorer la modélisation, les processus, le coût et la production d'applications.

« PowerAMC » intègre en outre, des fonctions de génération de code pour plusieurs Bases de Données et divers langages de programmation.

- **« *Système de Gestion de Bases de Données (SGBD)* » : « Paradox » :**

Figure 4.4 : Logo de « Paradox ».

« Paradox » est un Système de Gestion de Base de Données-Relationnelles « *SGBD-R* », actuellement édité par la société Canadienne « Corel ».

« Paradox » fut créé par la société « Ansa-Software » sous MS-DOS auparavant, racheté en 1987 par la société Américaine « Borland » qui l'a fait évoluer sous « Microsoft Windows » avant de céder la licence de développement et d'exploitation à la société Canadienne « Corel » (Qui a créé une version de « Paradox » pour « Linux »).

« Paradox » est un logiciel utilisant des fichiers dans deux formats : Le format « dBase (.dbf) » et le format « Paradox (.db) ». Il est compatible avec les requêtes « SQL » et dispose d'une interface graphique pour saisir les requêtes.

Il permet aussi de configurer, avec des assistants ou librement, des formulaires de saisie incorporant des tables, des états imprimables, des pages HTML liées aux données d'une Base, des macros, d'incorporer des fiches créées sous « Delphi », ...

Comme beaucoup de Système de Gestion de Base de Données-Relationnelles, ses données peuvent être utilisées dans des programmes écrits dans divers langages de programmation ; Les langages couramment utilisés avec « Paradox » sont : Son langage natif « ObjectPAL » (Object Pascal, sous Windows) et les langages qui disposent de modules d'accès aux données pour les fichiers « .db », Telsque : Delphi, Visual Basic, C++,... ou plus généralement tout langage sous Microsoft Windows pouvant utiliser l'accès « ODBC (*Open DataBase Connectivity*) ».

- **« *Environnement de Développement Intégré (EDI)* » : « Delphi (Sous « Microsoft Windows ») » :**

Figure 4.5 : Logo de « Delphi ».

« Delphi » est à la fois un langage de programmation Orienté Objet et un Environnement de Développement Intégré « *EDI* » (En Anglais : *IDE : Integrated Development Environment*) pour ce langage.

L'EDI « Delphi », est un EDI fonctionnant sous Windows, créé en 1995 par la société Américaine « Borland » ; À l'époque, créer des programmes graphiques sous Windows se faisait en grande majorité en utilisant soit la chaîne de compilation « Visual C++ », soit le « Visual Basic » : Le premier outil étant excessivement complexe et le second assez peu structuré, « Delphi » apparut alors comme une alternative viable pour beaucoup de développeurs qui souhaitaient créer des programmes pour Windows.

Depuis 2011, la société Américaine « Embarcadero Technologies » (Celle qui a acheté en 2008, la division d'outils de développement de logiciels de la société « Borland Software Corporation ») a introduit, dans « Delphi », le framework « Firemonkey » en complément de la Visual Component Library (*VCL*) (Toujours orientée Windows) qui a permis progressivement de compiler les mêmes programmes sur d'autres plate-formes.

Actuellement, « Delphi » permet de générer des exécutables sous Windows, MacOS, iOS, Android et Linux, depuis des programmes écrits en « Object Pascal (*ObjectPAL*) sous Windows ».

4.3 : Création de la Base de Données

C'est le passage de la description Conceptuelle à l'implémentation Physique de la Base de Données (Modèle Physique des Données « *MPD* ») dans un SGBD-R (En utilisant le SGBD-R « Paradox », pour la Base de Donnée de notre application), afin de pouvoir en effectué les différentes opérations de modification, d'ajout et de suppression.

Le Modèle Physique des Données « *MPD* » de la Base de Données de notre application de gestion de courriers (Bureau d'Ordre), est :

- Employe (#Mat_Employe, Nom_Employe, Prenom_Employe, Fonction_Employe, N_Tel_Employe, #Login_Utilisat).
- Statut_Utilisateur (#Login_Utilisat, MDP_Utilisat, Profil_Utilisat).
- Courriers_Arrivee (#N_Enregist_Cour_Arriv, Date_Enregist_Cour_Arriv, N_Cour_Arriv, Date_Emis_Cour_Arriv, Objet_Cour_Arriv, Type_Cour_Arriv, Observ_Cour_Arriv, #Mat_Employe, #Code_Corresp_Ext).
- Courriers_Depart (#N_Enregist_Cour_Dep, Date_Enregist_Cour_Dep, Objet_Cour_Dep, Type_Cour_Dep, Observ_Cour_Dep, #Mat_Employe, #Code_Struct_Int, #Code_Corresp_Ext).
- Structures_Interne (#Code_Struct_Int, Nom_Struct_Int, Responsable_Struct_Int, N_Tel_Struct_Int, N_Fax_Struct_Int).
- Correspondants_Externe (#Code_Corresp_Ext, Nom_Corresp_Ext, Responsable_Corresp_Ext, Adresse_Corresp_Ext, Ville_Corresp_Ext, Pays_Corresp_Ext, N_Tel_Corresp_Ext, N_Fax_Corresp_Ext).
- Orienter (#N_Enregist_Cour_Arriv, #Code_Struct_Int).

4.4 : Chronogramme réel du projet

Avec l'avancement du déroulement de travail, dans le temps, il s'avère que quelques phases nécessitent réellement des modifications de leurs durées, d'où il se doit obligatoirement modifier le chronogramme prévisionnel (Théorique) du départ de notre projet en l'adaptant aux exigences du plan réel (Pratique) et par la suite naisse un nouveau « diagramme de Gantt » qui représentera le «chronogramme réel » pour notre projet, en justifiant les décalages vis-à-vis le planning théorique :

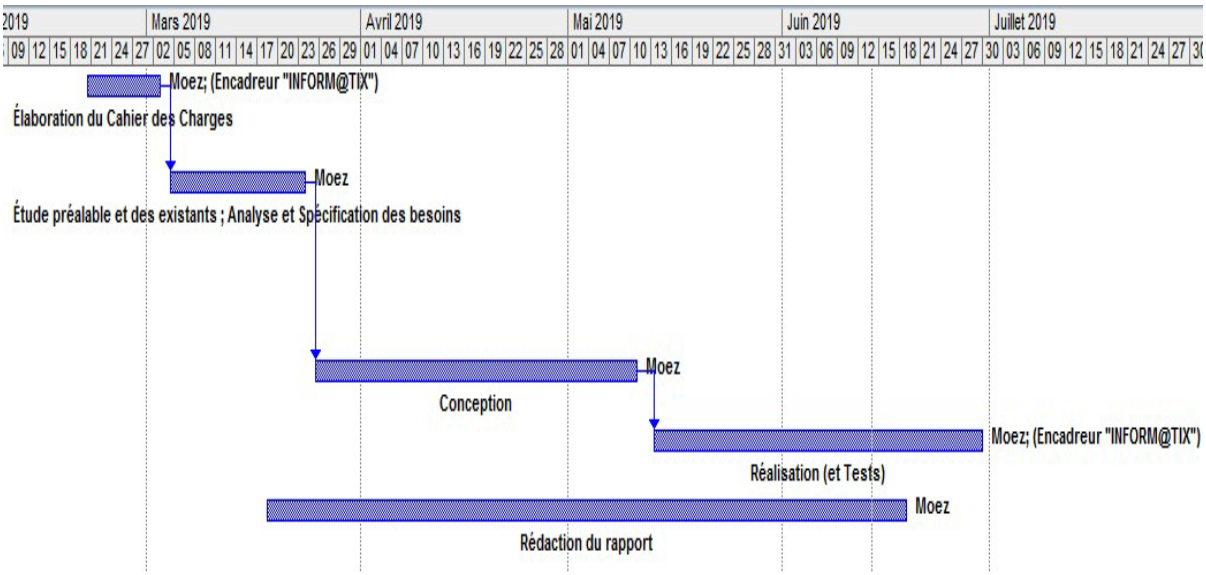

Figure 4.6 : Chronogramme réel du projet.

Conclusion

À la fin de cette étape de réalisation, nous avons obtenu une application de gestion de courriers en une première *Version Beta*, exploitable, répondant à nos attentes et aux objectifs définis au départ de notre projet.

Le produit obtenu est une première *Version Beta* qui va s'adapter et peut être modifié selon les recommandations et les exigences des clients finaux.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Ce présent travail ou projet ayant pour objectif l'automatisation de la « Gestion de courriers (Bureau d'Ordre) d'une entreprise », avec plus de fiabilité et d'efficacité.

En effet, l'application ainsi réalisée, répond aux spécifications énoncées au début du lancement de notre projet ; Cette application a permis :

- La création des différents Diagrammes relatifs à la Conception de l'application,
- La création d'une Base de Données,
- La gestion des courriers, des utilisateurs, ...
- La possibilité d'édition de quelques états de consultation,
- ...

Ce projet m'a donné une occasion très enrechissante pour mettre l'accent sur des différentes connaissances scientifiques acquises durant mes études, en m'offrant la chance pour bien m'en familiariser et avoir un complément de savoir pratique considérable avec la méthologie de conception UML et les outils informatiques utlisés pour le développement.

D'un point de vue humain, mon P.F.E. m'a également permis d'avoir un aperçu sur une autre vue du monde réel de travail car j'étais face à un engagement, puisque j'avais un cahier des charges au début et des délais à respecter au cours de la réalisation du projet.

Ce Projet de Fin d'Études, m'a été très bénéfique, en effet, il m'a permis de confronter de plus près les problèmes partiques et techniques, de concevoir une application informatique et de la développer ce qui m'a donné la chance d'apprfondir mes connaissances des différentes étapes de suivi et de réalisation d'un projet informatique en exerçant de près le métier de « concepteur » et celui de « développeur » informatique.

Les perspesctives de modifications, d'évolutions et d'extension restent cependant ouvertes pour poursuivre ce projet, et à l'enrechir par l'intégration d'autres modules complémentaires selon l'évolution des besoins futurs du Système d'Informations et des différents clients / utilisateurs finaux, notamment penser au fait qu'elle soit une solution sous autre architecture (Architecture « Web », possibilité de consulter les courriers sur une application mobile « Smartphone », ... par exemples).

Finalement, j'espère avoir bien présenter un travail qui facilitera les tâches de « Gestion de courriers (Bureau d'Ordre) » au sein d'une entreprise.

BIBLIOGRAPHIE

- ✓ [B_1] : Christian SOUTOU « *UML 2 Pour les Bases de Données* » (Éditions EYROLLES, 2002).
- ✓ [B_2] : Joseph GABAY et David GABAY « *UML 2 Analyse et Conception* » (Éditions DUNOD, 2008).
- ✓ [B_3] : Pascal ROQUES « *UML 2 Par la pratique* » (Éditions EYROLLES, 2008).
- ✓ [B_4] : Benoît CHARROUX, Aomar OSMANI et Yann THIERRY MIEG « *UML 2 Partique de la Modélisation* » (Éditions PEARSON Education France, 2009).
- ✓ [B_5] : Olivier DAHAN et Paul TOTH « *Delphi 7 Studio* » (Éditions EYROLLES, 2002).
- ✓ [B_6] : Steve TEIXEIRA et Xavier PACHECO « *Borland Delphi 6* » (Éditions CampusPress, 2002).
- ✓ [B_7] : M^{me} Houda REKAYA HOUISSA « *Cours du Module : " Bases de Données "* » (À l'U.V.T., Année Universitaire : 2017 / 2018).
- ✓ [B_8] : M^{me} Ahlem BEN YOUNÈS « *Cours du Module : " Ingénierie des Systèmes d'Information "* » (À l'U.V.T., Année Universitaire : 2018 / 2019).

WEBOGRAPHIE

- ✓ [W_1] : *www.cours-gratuit.com* : C'est un portail pédagogique prposant des solutions de formations continues gratuites.
- ✓ [W_2] : *https://coursinformatiquepdf.com* : C'est un site Web de cours et tutoriels informatique en pdf, à télécharger gratuitement.
- ✓ [W_3] : *www.developpez.com* : C'est un club sous forme de site Web des développeurs et professionnels en informatique du monde entier.
- ✓ [W_4] : *www.openclassrooms.com* : C'est une plate-forme Européenne d'éducation en ligne.
- ✓ [W_5] : *www.wikipedia.org* : C'est la plus célèbre encyclopédie collective en ligne, universelle, multilingue, qui vise à offrir un contenu dans tous les domaines librement réutilisable.
- ✓ [W_6] : *www.youtube.com* : C'est un site Web international d'hébergement de vidéos, sur lequel les utilisateurs peuvent envoyer, regarder, commenter, évaluer et partager des vidéos.

مُلَخَّص

يندرج هذا العمل في إطار مشروع نهاية الدّراسات للحصول على شهادة "الماجستير المهني في التّقنيات الحديثة للإتّصالات و الشّبكات" في "جامعة تونس الإفتراضيّة" ؛ هذا المشروع هو عبارة عن تصوّر و برمجة "تطبيقة إعلاميّة للتّصرّف في مراسلات مكتب الضّبط بشركة"، و ذلك بإعتماد : تطبيقة التّصميم و هندسة البرمجيّات "PowerAMC"، لغة التّصميم و هندسة البرمجيّات "UML"، نظام التّصرّف في قواعد البيانات "Paradox" و لغة البرمجة "Delphi".

كلمات مفاتيح : مكتب الضّبط، المراسلات الواردة، المراسلات الصّادرة، UML، Delphi.

Résumé

Le présent travail s'inscrit dans le cadre du Projet de Fin d'Études pour l'obtention du Diplôme de « Mastère Professionnel en Nouvelles Technologies des Télécommunications et Réseaux (N2TR) » à « l'Université Virtuelle de Tunis (UVT) » et qui consiste à concevoir et développer une application de «Gestion de courriers (Bureau d'Ordre) d'une entreprise », en utilisant : L'AGL « PowerAMC », « UML » comme langage de modélisation pour la conception, le SGBD « Paradox » et l'Environnement de Développement Intégré « Delphi ».

Mots clés : Bureau d'Ordre, Courrier Arrivée, Courrier Départ, UML, Delphi.

Abstract

The present work is part of the graduation project of studies in order to obtain the Diploma « Professional Master in New Technologies of Telecommunications and Networks (N2TN) » at the «Virtual University of Tunis (VUT) » and which consists in designing and developing a computer software of « Mail Management (Order Office) of a company », by using : The Software Design « PowerAMC », with « UML » for modelization, the « Paradox » DBMS and the « Delphi » Integrated Development Environment.

Key words : Order Office, Arrival Mail, Departure Mail, UML, Delphi.