**UNIVERSITE DE FIANARANTSOA**

**ECOLE NATIONALE D’INFORMATIQUE**

Mémoire de fin d’études pour l’obtention du diplôme de Master Professionnel en Informatique

Mention : Informatique

Parcours : Informatique Générale

**Conception et Réalisation d’une application pour la gestion de Pharmacies**

Présenté le 02 décembre 2022

Par Monsieur ANDRIANAMBININA Zo

Membres du Jury :

|  |  |  |
| --- | --- | --- |
| Président | : |  |
| Examinateur | : |  |
| Rapporteurs | : | Monsieur RATIARSON Venot, Maître de Conférences  Monsieur ANDRIAMPARANY Solofo Nirina, Directeur de société Isalosy |

Année Universitaire 2020-2021

# Curriculum Vitae

**Etat Civil**

Nom : ANDRIANAMBININA

Prénoms : Zo

Date et lieu de naissance : 04 Juillet 1997 à Toamasina I

Adresse : Lot 312AE/3602 Ankofafa Fianarantsoa

Téléphones : 034 20 935 90 – 033 43 356 73

E-Mail : [zo.andrianambinina1@gmail.com](mailto:zo.andrianambinina1@gmail.com)

**Formations et diplômes**

|  |  |
| --- | --- |
| **2021-2022**: | Deuxième année de formation en Master professionnel en Informatique à l’Ecole Nationale d’Informatique, Université de Fianarantsoa. Parcours : Informatique Générale. |
| **2020-2021** : | Première année de formation en Master professionnel en Informatique à l’Ecole Nationale d’Informatique, Université de Fianarantsoa. Parcours : Informatique Générale. |
| **2018-2019**: | Troisième année de formation en Licence professionnelle en Informatique à l’Ecole Nationale d’Informatique, Université de Fianarantsoa. Parcours : Informatique Générale.  Diplôme obtenu : Licence Professionnelle***.*** |
| **2017-2018** : | Deuxième année de formation en Licence professionnelle en Informatique à l’Ecole Nationale d’Informatique, Université de Fianarantsoa. Parcours : Informatique Générale. |
| **2016-2017** : | Première année de formation en Licence professionnelle en Informatique à l’Ecole Nationale d’Informatique, Université de Fianarantsoa. Parcours : Informatique Générale. |
| **2014-2015** : | Terminale au Lycée Privée Les Semeur Ankofafa Fianarantsoa.  Diplôme obtenu : Baccalauréat, Série D, Mention :Assez Bien***.*** |

**Stages et expériences professionnelles**

**Année 2022 :**

* Stage pratique de 5 mois au sein de la Société ISALOSY Antananarivo Madagascar.

|  |  |  |
| --- | --- | --- |
| **Thème** | : | Conception et réalisation d’une Application de Gestion de Pharmacie |
| **Méthode de gestion de projet** | : | SCRUM |
| **Méthode de conception** | : | 2TUP |
| **Outil de conception** | : | Visual Paradigm |
| **Langage de programmation** | : | C # |
| **Frameworks** | : | ASP Net Core, React Js |
| **Système de gestion de base de données** | : | Ms SQL |

* Conception et mise en place d’un Data Warehouse avec l’outil Pentao Data Integrator.

**Année 2021 :**

* Mise en place d’une application permettant la résolution d’un problème de transport (Recherche opérationnelle) en **Javascript**;
* Mise en place d’une application pour la classification d’objets en utilisant la technologie **Tensorflow** et le langage de programmation **Python** (Intelligence artificielle) ;

**Année 2019 :**

* Stage pratique de 3 mois au sein de Lycée Privée Les Semeurs.

|  |  |  |
| --- | --- | --- |
| **Thème** | : | Conception et réalisation d’un application gestion scolarité |
| **Méthode de conception** | : | 2TUP |
| **Langage de modélisation** | : | UML |
| **Outil de modélisation** | : | Visual Paradigm |
| **Langage de programmation** | : | PHP en utilisant le Framework Laravel |
| **Système de Gestion de Base de données** | : | MySQL |

* Création d’une application mobile pour la géolocalisation des agences bancaires en utilisant les technologies **Ionic** et **Cordova**;
* Gestion Bancaire (Projet ASP.NET) ;

**Année 2018 :**

* Stage pratique de 3 mois au sein Atelier Vatolahy Fianarantsoa.

|  |  |  |
| --- | --- | --- |
| **Thème** | : | Conception et réalisation d’une application pour la gestion de stock |
| **Méthode de conception** | : | Merise |
| **Outil de modélisation** | : | WinDesign |
| **Technologie de développement** | : | Php |
| **Systèmes de Gestion de Base de données** | : | MySQL |

* Gestion des heures Complémentaires (Projet PHP) ;
* Réalisation de projets réseaux :Surveillance de machine distant avec Mib Browser et CACTI

**Année 2017 :**

* Gestion de vente des matériels informatiques (Projet Qt Creator) ;
* Résolution d’une équation seconde degré et Résolution de trois équations trois inconnues (Projet JAVASCRIPT).
* Vente en ligne (Création d’un Site Web) ;

**Compétences en Informatique**

|  |  |  |
| --- | --- | --- |
| **Systèmes d’exploitation** | **:** | Windows, Linux |
| **Méthodes de conception** | **:** | Merise, 2TUP |
| **Langage de modélisation** | **:** | UML |
| **Langages de programmation** | **:** | C, C++, Visual Basic, C#, Java, Python, Shell |
| **Technologies web** | **:** | HTML, CSS, Javascript, ASP, ASP.NET, JSP, AJAX, PHP |
| **Technologies mobiles** | **:** | Ionic, Cordova, React Native, |
| **Frameworks** | **:** | Laravel, Symfony, ASP Net, JQuery, Bootstrap, React js, Angular, Django, Spring MVC, Express, Node js |
| **Service Web** | **:** | REST |
| **Outil Bureautique** | **:** | Microsoft Word |

**Compétences linguistiques**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Aptitude à comprendre à l’audition** | **Aptitude à lire** | **Aptitude à écrire et à rédiger** | **Aptitude à parler et à communiquer oralement** |
| Français | B | B | AB | P |
| Anglais | AB | AB | AB | P |

**Grille d’évaluation** : B(Bien), AB (Assez Bien), P (Passable).

**Divers et activités extracurriculaires**

**Activité sportive pratiquée** : Football.

**Passe-temps** : Lire, Regarder des films, Ecouter de la musique.

# Sommaire général

[Curriculum Vitae I](#_Toc118338334)

[Sommaire général V](#_Toc118338335)

[Remerciements VII](#_Toc118338336)

[Liste des figures VIII](#_Toc118338337)

[Liste des tableaux XI](#_Toc118338338)

[Liste des abréviations ou nomenclature XIII](#_Toc118338339)

[Introduction générale 1](#_Toc118338340)

[PARTIE I : PRESENTATIONS 2](#_Toc118338341)

[Chapitre 1. Présentation de l’Ecole Nationale d’Informatique (ENI) 3](#_Toc118338342)

[1.1. Informations d’ordre général 3](#_Toc118338343)

[1.2. Missions et historique 3](#_Toc118338344)

[1.3. Organigramme institutionnel de l’ENI 4](#_Toc118338345)

[1.4. Domaines de spécialisation 6](#_Toc118338346)

[1.5. Architecture des formations pédagogiques 6](#_Toc118338347)

[1.6. Relations de l’ENI avec les entreprises et les organismes 9](#_Toc118338348)

[1.7. Partenariat au niveau international 10](#_Toc118338349)

[1.8. Débouchés professionnels des diplômés 11](#_Toc118338350)

[1.9. Ressources humaines 12](#_Toc118338351)

[Chapitre 2. Présentation de la société Inovatic 14](#_Toc118338352)

[2.1 Création et description 14](#_Toc118338353)

[2.2 Localisation 14](#_Toc118338354)

[2.3 Spécialisations 14](#_Toc118338355)

[2.4 Contacts 15](#_Toc118338356)

[Chapitre 3. Description du projet 16](#_Toc118338357)

[3.1 Formulation 16](#_Toc118338358)

[3.2 Objectif et besoins de l’utilisateur 16](#_Toc118338359)

[3.3 Moyens nécessaires à la réalisation du projet 16](#_Toc118338360)

[3.4 Résultats attendus 17](#_Toc118338361)

[PARTIE II : ANALYSE ET CONCEPTION 18](#_Toc118338362)

[Chapitre 4. Analyse préalable 19](#_Toc118338363)

[4.1 Analyse de l’existant 19](#_Toc118338364)

[4.2 Critique de l’existant 20](#_Toc118338365)

[4.3 Proposition de solution 21](#_Toc118338366)

[4.4 Conception avant-projet 21](#_Toc118338367)

[Chapitre 5. Analyse conceptuelle 39](#_Toc118338368)

[5.1 Dictionnaire des données 39](#_Toc118338369)

[5.2 Règles de gestion 40](#_Toc118338370)

[5.3 Règles de calcul 40](#_Toc118338371)

[5.4 Représentation et spécification des besoins 40](#_Toc118338372)

[5.5 Spécification des besoins techniques 51](#_Toc118338373)

[5.6 Diagramme des cas d’utilisation global du système 54](#_Toc118338374)

[5.7 Modélisation du domaine 55](#_Toc118338375)

[Chapitre 6. Conception détaillée 69](#_Toc118338376)

[6.1 Architecture de l’application 69](#_Toc118338377)

[6.2 Diagramme de séquence de conception de chaque cas d’utilisation 70](#_Toc118338378)

[6.3 Diagramme des classes de conception de chaque cas d’utilisation. 86](#_Toc118338379)

[6.4 Diagramme des classes global 88](#_Toc118338380)

[6.5 Diagramme des paquetages 89](#_Toc118338381)

[6.6 Diagramme de déploiement 90](#_Toc118338382)

[PARTIE III : REALISATION 91](#_Toc118338383)

[Chapitre 7. Mise en place de l’environnement de développement 92](#_Toc118338384)

[7.1 Installation et configuration des outils 92](#_Toc118338385)

[7.2 Architecture de l’application 95](#_Toc118338386)

[Chapitre 8. Développement de l’application 97](#_Toc118338387)

[8.1 Création de la base de données 97](#_Toc118338388)

[8.2 Codage de l’application 97](#_Toc118338389)

[8.3 Présentation de l’application 99](#_Toc118338390)

[Conclusion 104](#_Toc118338391)

[Références bibliographiques VI](#_Toc118338392)

[Références webographiques VIII](#_Toc118338393)

[Glossaire XI](#_Toc118338394)

[Annexes XV](#_Toc118338395)

[Table des matières XXI](#_Toc118338396)

[Résumé XXV](#_Toc118338397)

[Abstract XXV](#_Toc118338398)

# Remerciements

En préambule à ce mémoire, je tiens tout d’abord à rendre grâce au Seigneur Dieu tout Puissant et Miséricordieux, de m’avoir toujours donné la force, la santé et le courage, qui m’ont permis de franchir toutes les épreuves que j’ai connues durant mon cursus universitaire et surtout à l’aboutissement de ce modeste travail.

Je remercie en particulier, Monsieur HAJALALAINA Aimé Richard, Professeur, Président de l’ Université de Fianarantsoa, de m’avoir encore offert l’opportunité d’étudier dans son grand monde universitaire.

De même pour Monsieur MAHATODY Thomas, Professeur et Directeur de l’Ecole Nationale d’Informatique, je le remercie pour toutes ses sacrifices tant en termes d’enseignement qu’il dispense que sur sa bonté de diriger convenablement l’ENI.

Je remercie également Monsieur ANDRIAMPARANY Solofo Nirina, Directeur général de la Société Isalosys Madagascar, d’avoir proprement approuvé mon intégration au sein de son organisme, et qui par lui, j’ai pu découvrir davantage les activités exercées dans la société et surtout la réalité du monde professionnel. Il a aussi mon encadreur professionnel, pour son dévouement à me guider, orienter et motiver face à toutes les épreuves que j’ai connues tout au long de ce stage et qui sans lui, je n’ai pas pu aboutir à ce présent stade.

Un grand merci à Monsieur VENOT, Maître de Conférences, mon encadreur pédagogique, pour sa précieuse collaboration tant sur les relectures intégrales de ce présent travail et sa correction que sur les bons conseils et techniques qu’il a personnellement inoculés.

J’adresse mes sincères remerciements à Monsieur X, Titre, Fonction et Monsieur Y, Titre, Fonction, et qui sont respectivement le Président du Jury et l’examinateur de ce présent travail. Leurs remarques et suggestions ont été des vrais atouts pour l’amélioration de ce mémoire.

L’enseignement de qualité dispensé par l’équipe pédagogique de l’Ecole Nationale d’Informatique a également su nourrir mes réflexions et a représenté une profonde satisfaction intellectuelle, merci donc aux Professeurs de l’ENI ainsi qu’aux restes de ses pédagogues.

Enfin, je remercie ma famille et mes ami(e)s pour leurs prières et soutiens tant au niveau moral que financier et qui, par leurs encouragements, j’ai pu surmonter tous les obstacles.

A tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

# Liste des figures

[Figure 1. Organigramme de l'ENI. 5](#_Toc30668084)

[Figure 2. Organigramme de la société Inovatic. 15](#_Toc30668085)

[Figure 3. Diagramme des flux de données. 21](#_Toc30668086)

[Figure 4. Vue d'ensemble de Scrum. [10.a] 27](#_Toc30668087)

[Figure 5. Processus de développement en forme de Y. [12.a] 31](#_Toc30668088)

[Figure 6. Arborescence des diagrammes UML. [13.a] 33](#_Toc30668089)

[Figure 7. Diagramme de contexte dynamique. 59](#_Toc30668090)

[Figure 8. Formalisme graphique d'un diagramme de cas d'utilisation. 62](#_Toc30668091)

[Figure 9. Diagramme des cas d'utilisation fonctionnels du système. 63](#_Toc30668092)

[Figure 10. Diagramme des cas d'utilisation techniques du projet. 74](#_Toc30668093)

[Figure 11. Diagramme des cas d'utilisation global du système. 77](#_Toc30668094)

[Figure 12. Diagramme de séquence système du cas d'utilisation « S'authentifier ». 78](#_Toc30668095)

[Figure 13. Diagramme de séquence système du cas d'utilisation « Se déconnecter ». 78](#_Toc30668096)

[Figure 14. Diagramme de séquence système du cas d'utilisation « Ajouter client ou prospect ». 79](#_Toc30668097)

[Figure 15. Diagramme de séquence système du cas d'utilisation « Modifier client ou prospect ». 80](#_Toc30668098)

[Figure 16. Diagramme de séquence système du cas d'utilisation « Supprimer client ou prospect ». 80](#_Toc30668099)

[Figure 17. Diagramme de séquence système du cas d'utilisation « Ajouter contact ». 81](#_Toc30668100)

[Figure 18. Diagramme de séquence système du cas d'utilisation « Modifier contact ». 82](#_Toc30668101)

[Figure 19. Diagramme de séquence système du cas d'utilisation « Supprimer contact ». 83](#_Toc30668102)

[Figure 20. Diagramme de séquence système du cas d'utilisation « Envoyer document commercial ». 83](#_Toc30668103)

[Figure 21. Diagramme de séquence système du cas d'utilisation « Ajouter événement ». 84](#_Toc30668104)

[Figure 22. Diagramme de séquence système du cas d'utilisation « Modifier événement ». 85](#_Toc30668105)

[Figure 23. Diagramme de séquence système du cas d'utilisation « Supprimer événement ». 86](#_Toc30668106)

[Figure 24. Diagramme de séquence système du cas d'utilisation « Voir force de vente ». 86](#_Toc30668107)

[Figure 25. Diagramme de séquence système du cas d'utilisation «Voir historique des actions avec les clients». 87](#_Toc30668108)

[Figure 26. Diagramme de séquence système du cas d'utilisation « Se dialoguer entre personnel ». 87](#_Toc30668109)

[Figure 27. Diagramme de séquence système du cas d'utilisation « Contacter prospect ou client ». 88](#_Toc30668110)

[Figure 28. Diagramme de séquence système du cas d'utilisation « Voir statistiques ». 88](#_Toc30668111)

[Figure 29. Diagramme de séquence système du cas d'utilisation « Voir performance des commerciaux ». 89](#_Toc30668112)

[Figure 30. Diagramme de séquence système du cas d'utilisation « Réinitialiser mot de passe ». 89](#_Toc30668113)

[Figure 31. Modèle de domaine du projet. 90](#_Toc30668114)

[Figure 32. Architecture MVC. [46.a] 91](#_Toc30668115)

[Figure 33. Diagramme de séquence de conception « S'authentifier ». 92](#_Toc30668116)

[Figure 34. Diagramme de séquence de conception « Ajouter client ou prospect ». 93](#_Toc30668117)

[Figure 35. Diagramme de séquence de conception « Modifier client ou prospect ». 94](#_Toc30668118)

[Figure 36. Diagramme de séquence de conception « Supprimer client ou prospect ». 95](#_Toc30668119)

[Figure 37. Diagramme de séquence de conception « Contacter client ou prospect ». 96](#_Toc30668120)

[Figure 38. Diagramme de séquence de conception « Ajouter événement ». 97](#_Toc30668121)

[Figure 39. Diagramme de séquence de conception « Modifier événement ». 98](#_Toc30668122)

[Figure 40.. Diagramme de séquence de conception « Annuler événement ». 99](#_Toc30668123)

[Figure 41. Diagramme de séquence de conception « Envoyer document commercial ». 100](#_Toc30668124)

[Figure 42. Diagramme de séquence de conception « Voir force de vente ». 100](#_Toc30668125)

[Figure 43. Diagramme de séquence de conception « Voir historique des actions avec les clients ». 101](#_Toc30668126)

[Figure 44. Diagramme de séquence de conception « Voir statistiques ». 101](#_Toc30668127)

[Figure 45. Diagramme de séquence de conception « Se dialoguer entre personnel ». 102](#_Toc30668128)

[Figure 46. Diagramme de séquence de conception « Voir performance des commerciaux ». 102](#_Toc30668129)

[Figure 47. Diagramme de séquence de conception « Ajouter contact ». 103](#_Toc30668130)

[Figure 48. Diagramme de séquence de conception « Modifier contact ». 104](#_Toc30668131)

[Figure 49. Diagramme de séquence de conception « Supprimer contact ». 105](#_Toc30668132)

[Figure 50. Diagramme de séquence de conception « Ajouter personnel ». 105](#_Toc30668133)

[Figure 51. Diagramme de séquence de conception « Modifier personnel ». 106](#_Toc30668134)

[Figure 52. Diagramme de séquence de conception « Supprimer personnel ». 107](#_Toc30668135)

[Figure 53. Diagramme de séquence de conception « Se déconnecter ». 107](#_Toc30668136)

[Figure 54. Diagramme de séquence de conception « Réinitialiser mot de passe ». 108](#_Toc30668137)

[Figure 55. Diagramme des classes de conception « Gérer prospect ou client ». 108](#_Toc30668138)

[Figure 56. Diagramme des classes de conception « Gérer événement ». 109](#_Toc30668139)

[Figure 57. Diagramme des classes de conception « Envoyer document commercial ». 109](#_Toc30668140)

[Figure 58. Diagramme des classes de conception « Voir force de vente, Voir statistiques, Voir performance des commerciaux ». 110](#_Toc30668141)

[Figure 59. Diagramme des classes de conception « Gérer personnel, Réinitialiser mot de passe, S'authentifier, Se déconnecter ». 110](#_Toc30668142)

[Figure 60. Diagramme des classes global. 111](#_Toc30668143)

[Figure 61. Diagramme des paquetages. 112](#_Toc30668144)

[Figure 62. Diagramme de déploiement. 112](#_Toc30668145)

[Figure 63. Page d'accueil de WampServer. 114](#_Toc30668146)

[Figure 64. Message de bienvenue de CodeIgniter. 115](#_Toc30668147)

[Figure 65. Postman testant un API REST. 116](#_Toc30668148)

[Figure 66. Capture d'écran de Sublime Text. 117](#_Toc30668149)

[Figure 67. Architecture de l'application. 117](file:///C:\Memoire%20M2\Memoire%20M2.docx#_Toc30668150)

[Figure 68. Algorithme de création d'un client de type société. 120](#_Toc30668151)

[Figure 69. Algorithme de suppression d'un événement. 120](#_Toc30668152)

[Figure 70. Algorithme de récupération d'un ou de plusieurs événements. 120](#_Toc30668153)

[Figure 71. Algorithme JavaScript d'envoi d'une requête AJAX. 121](#_Toc30668154)

[Figure 72. Page d'authentification de l'application. 121](#_Toc30668155)

[Figure 73. Tableau de bord de l'application. 122](#_Toc30668156)

[Figure 74. Page d'agenda du CRM. 122](#_Toc30668157)

[Figure 75. Création de devis. 123](#_Toc30668158)

[Figure 76. Mail reçu par le client et le devis correspondant. 124](#_Toc30668159)

[Figure 77. SMS envoyé au client à chaque envoi de devis. 124](#_Toc30668160)

[Figure 78. Facture créée à partir d'un devis validé. 125](#_Toc30668161)

[Figure 79. Devis utilisé dans l'ancien système. XVII](#_Toc30668162)

[Figure 80. Facture dans l'ancien système. XVIII](#_Toc30668163)

[Figure 81. Fichier de prospection dans l'ancien système. XIX](#_Toc30668164)

# Liste des tableaux

[Tableau 1. Organisation du système de formation pédagogique de l'Ecole. 6](#_Toc30665093)

[Tableau 2. Architecture des études correspondant au système LMD 7](#_Toc30665094)

[Tableau 3. Liste des formations existantes à l’Ecole Nationale d’Informatique. 8](#_Toc30665095)

[Tableau 4. Débouchés professionnels éventuels des diplômés. 12](#_Toc30665096)

[Tableau 5. Caractéristiques de l'ordinateur utilisé. 17](#_Toc30665097)

[Tableau 6. Moyens matériels de la société Inovatic. 22](#_Toc30665098)

[Tableau 7. Etude comparative des solutions préconisées. 23](#_Toc30665099)

[Tableau 8. Etude comparative des scénarios possibles pour la solution retenue. 24](#_Toc30665100)

[Tableau 9. Etude comparative entre les méthodes agiles SCRUM et XP. 26](#_Toc30665101)

[Tableau 10. Etude comparative entre 2TUP et RUP. 30](#_Toc30665102)

[Tableau 11. Etude comparative entre ArgoUML et Visual Paradigm. 34](#_Toc30665103)

[Tableau 12. Etude comparative entre PHP, JSP et ASP. 36](#_Toc30665104)

[Tableau 13. Etude comparative entre MySQL, PostgreSQL, Oracle et SQL Server. 39](#_Toc30665105)

[Tableau 14. Etude comparative entre Symfony, CodeIgniter et Laravel. 41](#_Toc30665106)

[Tableau 15. Etude comparative entre JQuery, Angular et React. 42](#_Toc30665107)

[Tableau 16. Etude comparative entre Bootstrap et Materialize. 43](#_Toc30665108)

[Tableau 17. Etude comparative entre les IDE PHP Storm et Sublime Text. 48](#_Toc30665109)

[Tableau 18. Etude comparative entre Postman, Dell Boumi et IBM API Management. 50](#_Toc30665110)

[Tableau 19. Etude comparative entre Filezilla, FireFTP et SmartFTP. 51](#_Toc30665111)

[Tableau 20. Dictionnaire des données. 53](#_Toc30665112)

[Tableau 21. Identification des acteurs avec leurs descriptions respectives. 57](#_Toc30665113)

[Tableau 22. Identification des messages. 57](#_Toc30665114)

[Tableau 23. Capture des besoins fonctionnels. 60](#_Toc30665115)

[Tableau 24. Description textuelle du cas d'utilisation « Ajouter client ou prospect ». 64](#_Toc30665116)

[Tableau 25. Description textuelle « Modifier prospect ou client ». 65](#_Toc30665117)

[Tableau 26. Description textuelle « Supprimer prospect ou client ». 65](#_Toc30665118)

[Tableau 27. Description textuelle « Contacter prospect ou client ». 66](#_Toc30665119)

[Tableau 28. Description textuelle « Ajouter événement ». 66](#_Toc30665120)

[Tableau 29. Description textuelle du cas d'utilisation « Modifier événement ». 67](#_Toc30665121)

[Tableau 30. Description textuelle du cas d'utilisation « Annuler événement ». 67](#_Toc30665122)

[Tableau 31. Description textuelle « Envoyer document commercial ». 68](#_Toc30665123)

[Tableau 32. Description textuelle du cas d'utilisation « Voir force de vente ». 68](#_Toc30665124)

[Tableau 33. Description textuelle du cas d'utilisation « Voir historique des actions avec les clients ». 69](#_Toc30665125)

[Tableau 34. Description textuelle du cas d'utilisation « Voir statistiques ». 69](#_Toc30665126)

[Tableau 35. Description textuelle du cas d'utilisation « Dialoguer avec d’autres personnels ». 70](#_Toc30665127)

[Tableau 36. Description textuelle du cas d'utilisation « Voir performance des commerciaux ». 70](#_Toc30665128)

[Tableau 37. Description textuelle du cas d'utilisation « Ajouter contact ». 71](#_Toc30665129)

[Tableau 38. Description textuelle du cas d'utilisation « Modifier contact ». 71](#_Toc30665130)

[Tableau 39. Description textuelle du cas d'utilisation « Supprimer contact ». 72](#_Toc30665131)

[Tableau 40. Description textuelle du cas d'utilisation « Ajouter personnel ». 72](#_Toc30665132)

[Tableau 41. Description textuelle du cas d'utilisation « Modifier personnel ». 73](#_Toc30665133)

[Tableau 42. Description textuelle du cas d'utilisation « Supprimer personnel ». 73](#_Toc30665134)

[Tableau 43. Description textuelle du cas d'utilisation « S'authentifier ». 75](#_Toc30665135)

[Tableau 44. Description textuelle du cas d'utilisation « Se déconnecter ». 75](#_Toc30665136)

[Tableau 45. Description textuelle du cas d'utilisation Réinitialiser mot de passe. 76](#_Toc30665137)

[Tableau 46. Description des besoins fonctionnels. XVI](#_Toc30665138)

[Tableau 47. Chronogramme des travaux. XX](#_Toc30665139)

# Liste des abréviations ou nomenclature

|  |  |
| --- | --- |
| 2TUP | **2** Tracks **U**nified **P**rocess |
| A | **A**lphabétique |
| AB | **A**ssez **B**ien |
| AJAX | **A**synchronous **J**avascript **A**nd **X**ml |
| AN | **A**lpha**n**umérique |
| ANSI | **A**merican **N**ational **S**tandards **I**nstitute |
| API | **A**pplication **P**rogramming **I**nterface |
| ASP | **A**ctive **S**erver **P**ages |
| AUF | **A**gence **U**niversitaire de la **F**rancophonie |
| B | **B**ien |
| BD | **B**ase de **D**onnées |
| BNGRC | **B**ureau **N**ational de **G**estion des **R**isques et des **C**atastrophes |
| CARI | **C**olloque **A**fricain sur la **R**echerche en **I**nformatique |
| CCNA | **C**IS**C**O **N**etworking **A**cademy |
| CHU | **C**entre **H**ospitalier **U**niversitaire |
| CITEF | **C**onférence **I**nternationale des **F**ormations d’**I**ngénieurs et **T**echniciens d’**E**xpression **F**rançaise |
| CLI | **C**ommand **L**ine **I**nterface |
| CNH | **C**ommission **N**ationale d’**H**abilitation |
| CNRE | **C**entre **N**ational de **R**echerche sur l’**E**nvironnement |
| CNRIT | **C**entre **N**ational de **R**echerche **I**ndustrielle et **T**echnologique |
| COBOL | **CO**mmon **B**usiness **O**riented **L**anguage |
| COFAV | **C**orridor **F**orestier **A**mbositra-**V**ondrozo |
| CPU | **C**entral **P**rocessing **U**nit |
| CRM | **C**ustomer **R**elationship **M**anagement |
| CRUD | **C**reate **R**ead **U**pdate **D**elete |
| CUR | **C**entre **U**niversitaire **R**égionale |
| D | **D**ate |
| DBA | **D**ata**B**ase **A**dministrator |
| DEA | **D**iplôme d’**E**tudes **A**pprofondies |
| DFD | **D**iagramme des **F**lux de **D**onnées |
| DLC | **D**evelopment **L**earning **C**enter |
| DOM | **D**ocument **O**bject **M**odel |
| EMIT | **E**cole de **M**anagement et d’**I**nnovation **T**echnologique |
| ENI | **E**cole **N**ationale d’**I**nformatique |
| ERD | **E**ntity **R**elation **D**iagram |
| ESIROI-ESTIM | **E**cole **S**upérieure d’**I**ngénieurs **R**éunion-**O**céan **I**ndien – **E**cole **S**upérieure des **T**echnologies d’**I**nformatique et du **M**anagement |
| ESPA | **E**cole **S**upérieure **P**olytechnique d’**A**ntananarivo |
| FID | **F**onds d’**I**ntervention pour le **D**éveloppement |
| FPPSM | **F**orêts, **P**arcs, **P**auvreté dans le **S**ud de **M**adagascar |
| FSF | **F**ree **S**oftware **F**oundation |
| FTM | **F**oibe Tao-**T**saritanin’i **M**adagasikara |
| FTP | **F**ile **T**ransfer **P**rotocol |
| GNU | **G**nu’s **N**ot **U**nix |
| GPL | **G**eneral **P**ublic **L**icense |
| GRC | **G**estion de la **R**elation **C**lient |
| HT | **H**ors **T**axe |
| HTML | **H**yper**T**ext **M**arkup **L**anguage |
| HTTP | **H**yper**T**ext **T**ransfer **P**rotocol |
| IBM | **I**nternational **B**usiness **M**achines |
| IDE | **I**ntegrated **D**evelopment **E**nvironment |
| IIS | **I**nternet **I**nformation **S**ervices |
| INPG | **I**nstitut **N**ational **P**olytechnique de **G**renoble |
| INSTAT | **I**nstitut **N**ational de la **S**tatistique |
| IRD | **I**nstitut de **R**echerche pour le **D**éveloppement |
| IREMIA | **I**nstitut de **R**echerche en **M**athématiques et **I**nformatique **A**ppliquées |
| JIRAMA | **J**iro sy **R**ano **M**alagasy |
| JSP | **J**ava **S**erver **P**ages |
| JVM | **J**ava **V**irtual **M**achine |
| LDAP | **L**ightweight **D**irectory **A**ccess **P**rotocol |
| LMD | **L**icence-**M**aster-**D**octorat |
| MAEP | **M**inistère de l’**A**griculture, de l’**E**levage et de la **P**êche |
| MEF | **M**inistère de l’**E**conomie et des **F**inances |
| MEM | **M**odélisation pour l’**E**nvironnement à **M**adagascar |
| MENSupRES | **M**inistère de l’**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique |
| MFB | **M**inistère des **F**inances et du **B**udget |
| MIC | **M**adagascar **I**nformatique **C**orporation |
| MID | **M**inistère de l’**I**ntérieur et de la **D**écentralisation |
| MIT | **M**assachusetts **I**nstitute **T**echnology |
| MOF | **M**eta-**O**bject-**F**acility |
| MVC | **M**odèle-**V**ue-**C**ontrôleur |
| MySQL | **M**y **S**tructured **Q**uery **L**anguage |
| N | **N**umérique |
| NB | **N**ota **B**ene |
| NoSQL | **N**ot **o**nly **SQL** – **N**ot **o**nly **S**tructured **Q**uery **L**anguage |
| NS | **N**iveau **S**colaire |
| OMG | **O**bject **M**anagement **G**roup |
| OO | **O**rienté **O**bjet |
| ORM | **O**bject **R**elational **M**apping |
| OTME | **O**ffice des **T**ransmissions **M**ilitaires de l’**E**tat |
| P | **P**assable |
| PDF | **P**ortable **D**ocument **F**ormat |
| PEAR | **P**HP **E**xtension and **A**pplication **R**epository |
| PHP | **P**ersonal **H**ome **P**age – **P**HP **H**ypertext **P**reprocessor |
| PME | **P**etite et **M**oyenne **E**ntreprise |
| PRESUP | **P**rogramme de **R**enforcement de l’**E**nseignement **S**upérieur |
| RAM | **R**andom **A**ccess **M**emory |
| REST | **Re**spresentational **S**tate **T**ransfer |
| ROI | **R**eturn **O**n **I**nvestment |
| RRR | **R**abais, **R**emise et **R**istourne |
| RUP | **R**ational **U**nified **P**rocess |
| SaaS | **S**oftware **a**s **a** **S**ervice |
| SGBD | **S**ystème de **G**estion de **B**ase de **D**onnées |
| SGBDR | **S**ystème de **G**estion de **B**ase de **D**onnées **R**elationnelles |
| SMMC | **S**ociété de **M**anutention des **M**archandises **C**onventionnelles |
| SMS | **S**hort **M**essage **S**ystem – **S**ystème de **M**essage **S**uccinct |
| SOA | **S**ervices **O**riented **A**rchitecture |
| SOAP | **S**imple **O**bject **A**ccess **P**rotocol |
| SPARC | **S**calable **P**rocessor **ARC**hitecture |
| SPAT | **S**ociété du **P**ort à gestion **A**utonome de **T**oamasina |
| SQL | **S**tructured **Q**uery **L**anguage |
| SSH | **S**ecure **S**hell |
| SSH | **S**ecure **Sh**ell |
| SSL | **S**ecure **S**ocket **L**ayer |
| SSL | **S**ecure **S**ocket **L**ayer |
| TB | **T**rès **B**ien |
| TIC | **T**echnologie de l’**I**nformation et de la **C**ommunication |
| TPE | **T**rès **P**etite **E**ntreprise |
| TTC | **T**oute **T**axe **C**omprise |
| TVA | **T**axe sur la **V**aleur **A**joutée |
| UML | **U**nified **M**odeling **L**anguage |
| UML | **U**nified **M**odeling **L**anguage |
| UPST | **U**niversité **P**aul **S**abatier de **T**oulouse |
| URI | **U**niform **R**esource **I**dentifier |
| URL | **U**niform **R**esource **L**ocator |
| USAID | **U**nited **S**tates **A**gency for **I**nternational **D**evelopment |
| VB | **V**isual **B**asic |
| WWW | **W**orld **W**ide **W**eb |
| XML | e**X**tensible **M**arkup **L**anguage |
| XP | e**X**treme **P**rogramming |

# Introduction générale

Actuellement, l’informatique constitue un domaine pratiquement incontournable dans la vie quotidienne et surtout dans le monde professionnel. Elle est très utile pour les entreprises en termes de communication, sauvegarde et gestion des informations pour l’automatisation des opérations à effectuer dans les différents domaines du travail.

Dans le cadre de nos études, la formation en troisième année (L3) est complétée par un stage dans une entreprise pour l’obtention du diplôme de licence professionnelle en informatique. Cela a été l’opportunité pour moi de mettre en pratique et d’approfondir mes connaissances dans l’Atelier Vatolahy à Ambatolahimandravina Fianarantsoa. L’Atelier Vatolahy qui est une société individuelle, oeuvrant dans la confection de produits en béton de granulats préfabriqués et briques compressées stabilisées au ciment /Chaux. La société cherche toujours à se rapprocher aux clients afin de répondre effectivement à leurs besoins. C’est précisément pour cette raison qu’a été mise en place une application web pour la « gestion de stock des produits en bétons de granulats préfabriqués ».

La contribution principale de ce présent travail porte sur la mise en place d’une application web pour la gestion de pharmacies selon les besoins des clients.

Pour réaliser ce projet, Nous allons utiliser un langage de modélisation, une méthode de modélisation, une architecture, un langage de programmation, des Frameworks pour le Back-end et le Front-end, un système de versionning et un SGBD.

Ce présent mémoire s’articulera donc autour de trois grandes parties principales : d’abord, nous allons passer par la présentation de l’Ecole Nationale d’Informatique (ENI) suivi de celle de Société ISALOSY ainsi que la description du projet. Par la suite, nous allons se concentrer sur l’analyse préalable et l’analyse conceptuelle. Et enfin, nous allons expliquer le déroulement du projet, en visant la réalisation du projet.

# PARTIE I : PRESENTATIONS

## Présentation de l’Ecole Nationale d’Informatique (ENI)

### Informations d’ordre général

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

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

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

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

### Missions et historique

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### Organigramme institutionnel de l’ENI

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

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

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

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

Trois départements de formation caractérisent l’organigramme :

* Le département de formation théorique à l’intérieur de l’Ecole ;
* Le département de formation pratique pour la coordination et la supervision des stages en entreprise et des voyages d’études ;
* Le département de formation doctorale pour l’organisation de la formation de 3ème cycle.

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

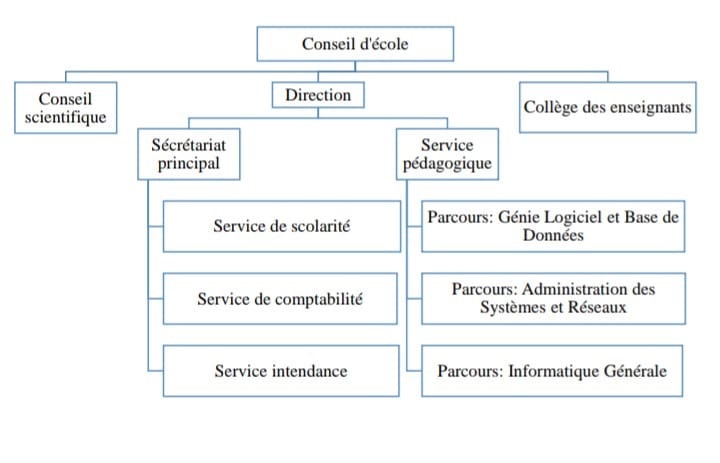


Figure 1. Organigramme de l'ENI.

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

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

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

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

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

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

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

### Domaines de spécialisation

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

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

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

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

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

|  |  |
| --- | --- |
| Formation théorique | Formation pratique |
| Enseignement théorique  Travaux dirigés  Travaux pratiques | Etude de cas  Travaux de réalisation  Projets / Projets tutorés  Voyage d’études  Stages |

### Architecture des formations pédagogiques

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

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

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

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

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

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

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

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

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

* D = Doctorat (Bac +8)

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

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

Tableau 2. Architecture des études correspondant au système LMD

|  |  |  |  |
| --- | --- | --- | --- |
| +8 | Doctorat | Master Recherche  Master Professionnel  Licence Professionnelle | **Marché de travail** |
| +7 |
| +6 |
| +5 | Master 2 |
| +4 | Master 1 |
| +3 | Licence 3 |
| +2 | Licence 2 |
| +1 | Licence 1 |  |

La licence peut avoir une vocation générale ou professionnelle.

Le master peut avoir une vocation professionnelle ou de recherche.

Le tableau 3 montre la liste des formations existantes à l’Ecole Nationale d’Informatique.

Tableau 3. Liste des formations existantes à l’Ecole Nationale d’Informatique.

|  |  |  |
| --- | --- | --- |
|  | FORMATION EN : | |
|  | LICENCE PROFESSIONNELLE | MASTER |
| Condition d’admission | Par voie de concours Formation Professionnelle : 100 candidats  Formation hybride : 150 candidats |  |
| Condition d’accès | Bac de série C, D ou Technique | Être titulaire de licence professionnelle |
| Durée de formation | 3 années | 2 années |
| Diplôme délivré | Diplôme de Licence Professionnelle en Informatique | Diplôme de Master Professionnel  Diplôme de Master Recherche |

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

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

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

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

Depuis l’année universitaire 2010-2011, l’ENI s’est mise à organiser des formations hybrides en informatique dans les différentes régions (Fianarantsoa, Toliara) en raison de l’insuffisance de la capacité d’accueil des infrastructures logistiques. En effet, le système de formation hybride semi - présentielle utilise la visioconférence pour la formation à distance.

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

### Relations de l’ENI avec les entreprises et les organismes

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

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

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

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

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

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

* L’informatique de gestion d’entreprise
* Les technologies de l’information et de la communication (TIC)
* La sécurité informatique des réseaux
* L’administration des réseaux et des systèmes
* Les services bancaires et financiers, notamment le *MobileBanking*
* Les télécommunications et la téléphonie mobile
* Les *BigData*
* Le commerce, la vente et l’achat, le Marketing
* L’ingénierie informatique appliquée
* L’écologie et le développement durable

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

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

### Partenariat au niveau international

Entre 1996 et 1999, l’ENI avait bénéficié de l’assistance technique et financière de la Mission Française de Coopération et d’action culturelle dans le cadre du Programme de Renforcement de l’Enseignement Supérieur (PRESUP) consacré à l’Ecole a notamment porté sur :

* Une dotation en logiciels, micro-ordinateurs, équipements de laboratoire de maintenance et de matériels didactiques ;
* La réactualisation des programmes de formation assortie du renouvellement du fonds de la bibliothèque ;
* L’appui à la formation des formateurs ;
* L’affectation à l’Ecole d’Assistants techniques français.

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

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

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

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

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

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

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

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

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

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

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

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

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

### Débouchés professionnels des diplômés

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

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

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

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

L’Ecole bénéficie aujourd’hui de 36 années d’expériences pédagogiques et de reconnaissance auprès des sociétés, des entreprises et des organismes. C’est une Ecole Supérieure de référence en matière informatique.

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

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

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

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

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

Le tableau 4 présente les débouchés professionnels éventuels des diplômés de l’ENI.

Tableau 4. Débouchés professionnels éventuels des diplômés.

|  |  |
| --- | --- |
| LICENCE | * Analyste * Programmeur * Administrateur de site web/de portail web * Assistant Informatique et internet * Chef de projet web ou multimédia * Développeur Informatique ou multimédia * Intégrateur web ou web designer * Hot liner/Hébergeur Internet * Agent de référencement * Technicien/Supérieur de help desk sur Informatique * Responsable de sécurité web * Administrateur de réseau * Administrateur de cybercafé |
| MASTER | * Administrateur de cybercafé * Administrateur de réseau et système * Architecture de système d’information * Développeur d’application /web /java/Python /IOS /Android * Ingénieur réseau * Webmaster /web designer * Concepteur Réalisateur d’applications * Directeur du système de formation * Directeur de projet informatique * Chef de projet informatique * Responsable de sécurité informatique * Consultant fonctionnel ou freelance |

### Ressources humaines

Pour mener aux différentes activités ainsi qu’aux organisations éducatives de l’Ecole, multiples sont les personnes physiques jouant chacune leurs propres rôles pour faire fonctionner l’Etablissement dont :

* Directeur de l’Ecole : Professeur RAMAMONJISOA Andriantiana Bertin Olivier ;
* Responsable de Mention « Informatique » : Monsieur RABETAFIKA Louis Haja ;
* Responsable de Parcours « Génie Logiciel et Base de Données » : Monsieur RALAIVAO Jean Christian ;
* Responsable de Parcours « Administration des Systèmes et Réseaux » : Monsieur SIAKA ;
* Responsable de Parcours « Informatique Générale » : Monsieur GESAZAFY Gilante ;
* Nombre d’Enseignants permanents : 13 dont deux (02) Professeurs Titulaires, six (06) Maîtres de Conférences et cinq (05) Assistants d’Enseignement Supérieur et de Recherche ;
* Nombre d’Enseignants vacataires : 10 ;
* Personnel Administratif : 23 ;

## Présentation de la société Inovatic

### Création et description

Inovatic est une entreprise spécialisée dans quelques domaines de l’informatique tels que la création de site internet ou d’application (lourde, web ou mobile), maintenance informatique, conception graphique, etc. Elle a été créée en 2017 par deux jeunes entrepreneurs et depuis, la forme juridique de la société est une Société à Responsabilité Limitée. Vu le nombre actuel de personnel que la société emploie, Inovatic est désormais considérée comme un PME ou Petites et Moyennes entreprises.

### Localisation

Le siège central de la société se trouve à Toamasina, plus précisément à la Pépinière d’entreprises, 02 Rue Richelieu (Près de l’hôpital Kely). Mais uniquement pour son développement, les gérants ont décidé en 2018 de créer un nouveau local dans la ville capitale, plus précisément à 67 Ha, près de la pharmacie Mamy. La société dispose donc de deux locaux.

### Spécialisations

Parmi les spécialisations de la société, on distingue :

* Création de site internet ;
* Conception et réalisation d’application (lourde, web ou mobile) ;
* Maintenance informatique ;
* Conception graphique ;
* Animation publicitaire ;
* Assistance informatique ;
* etc.

Pour parvenir à l’accomplissement de ces services, la société dispose d’une équipe jeune et dynamique ayant chacune sa propre spécialité.

La figure 2 représente l’organigramme de la société Inovatic.

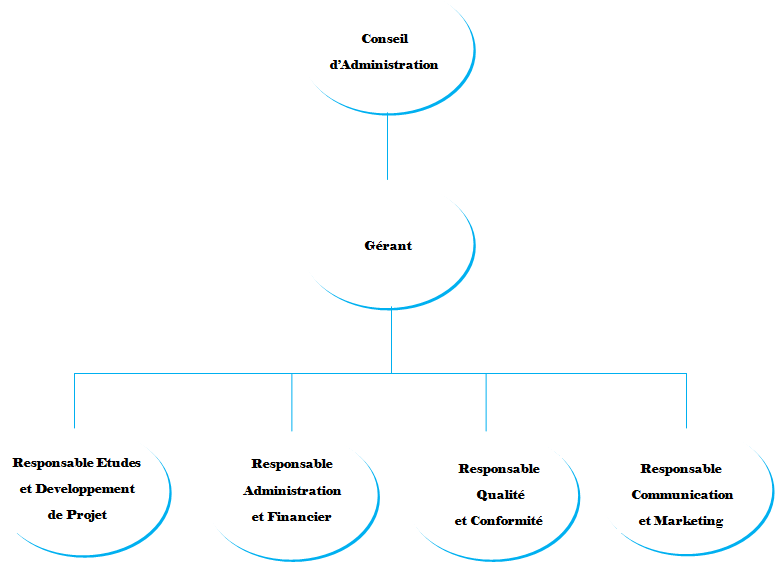


Figure 2. Organigramme de la société Inovatic.

### Contacts

Multiples sont les moyens permettant de contacter la société, parmi ces moyens, on distingue :

* Téléphones : 034 74 299 98 – 034 04 235 01
* Mail : [info@inovatic.mg](mailto:info@inovatic.mg)
* Site web: www.inovatic.mg

## Description du projet

### Formulation

Ce projet intitulé « conception et réalisation d’une application web pour la gestion de pharmacie » consiste essentiellement à concevoir initialement le logiciel « HJGEST ». Certes, cette conception initiale s’est réalisée au sein de la pharmacie. Force est de marteler que ce logiciel est une application web, déjà utilisée dans cette entreprise, cependant sa fiabilité n’est pas encore si authentique. C’est pour cette raison que des améliorations sont nécessairement requises afin de répondre aux besoins des utilisateurs, à ne dire que les nouvelles fonctionnalités de l’application, la robustesse, etc.

### Objectif et besoins de l’utilisateur

#### Objectif

L’objectif principal et suprême dans ce projet est de concevoir et réaliser une application pour la gestion de pharmacie.

L’objectif de ce projet est la mise en place d’une application web pour la gestion de pharmacie qui pourra être fonctionnelle, utilisable, fiable, robuste, évolutive, apte à répondre aux besoins des utilisateurs.

#### Besoins de l’utilisateur

L’application doit satisfaire les besoins fonctionnels en matière de gestion de stock comme :

* Gestion des médicaments ;
* Gestion des approvisionnements des médicaments ;
* Mise à jour de stock des médicaments ;
* Gestion des fournisseurs ;
* Gestion des clients ;
* Gestion des commandes ;
* Générer de bon de commande et de bon de livraison ;
* Archivage des livraisons effectuées.

### Moyens nécessaires à la réalisation du projet

#### Moyens humains

Tenant compte de tous les moyens et informations nécessaires à la réalisation de ce projet, toutes ces personnes ont chacun apporté leur moyen :

* Chef de Projet : Encadreur professionnel, fournisseur de toutes les informations, mais aussi, participant pour la validation du projet.
* Testeur : Encadreur professionnel, participant pour le test de l’application
* Un concepteur-développeur : le stagiaire qui a assuré la conception et le développement de l’application.

#### Moyens matériels et logiciels

Le tableau 5 présente la liste des matériels et logiciels nécessaires à la réalisation du projet.

Tableau 5. Caractéristiques de l'ordinateur utilisé.

|  |  |  |
| --- | --- | --- |
| **Désignation** | **Caractéristiques/Descriptions** | **Commentaires** |
| -Un Ordinateur Portable  -Un Ordinateur Bureau | **-TOSHIBA CORE i5 :**   * **RAM : 4G** * **Disque Dur : 128GB**   **-HP CORE i3 :**   * **RAM :4G** * **Disque Dur : 320GB** | **Pour la réalisation du projet.** |
| Visual Paradigm | **Libre** | **Outil de modélisation.** |
| -Microsoft Visual Studio  Community 2020 Preview  - Microsoft Visual Code | **Version 22.0.40629.00 Update 3** | **Pour le développement de l’application.** |
| SQL Server 2019 | **SQL Server Data Tools 19.0.41012.0** | **Système de gestion de base de données.** |
| Postman | **Libre** | **Testeur API** |
| Chrome | **Navigateur web** | **Pour lancer l’application** |

### Résultats attendus

Ce projet devra généralement aboutir à une application web facile à manipuler et à utiliser. Elle doit être fiable, valide et robuste, évolutive, paramétrable et apte à répondre aux besoins de l’entreprise. En outre, il faut surtout que l’application ait la capacité de faire la mise à jour de stock des médicaments lors de la production et après livraison, d’une part, et d’autre part et d’élaborer un bon de commandes.

Ce sont :

* Gérer approvisionnement ;
* Gérer médicament ;
* Gérer Commande ;
* Gérer vente et livraison
* Elaborer facture (bon de commande et bon de livraison).

# PARTIE II : ANALYSE ET CONCEPTION

## Analyse préalable

Avant tous travaux de développement informatique, il est convenable d’étudier le système actuel pour établir une cohérence aux choix stratégiques à mettre en place. Dans cette partie, l’analyse de l’existant sera effectuée en premier lieu, ensuite les critiques de l’existant, et enfin la conception avant-projet pour savoir les solutions à la réalisation du projet.

### Analyse de l’existant

Comme pour tout développement voué à réussir, la base de son élaboration est une phase d’analyse complète de l’existant, en plongeant totalement dans le domaine du projet.

#### Organisation actuelle

De toute façon, la Pharmacie rencontre des problèmes à propos de traitements manuels des données (enregistrement de commandes, la gestion de clients et fournisseurs, etc.). De plus, la société utilise l’application Microsoft Excel mais avec difficultés parce que cette application est difficile à manier pratiquement. Pharmacie a une application « HJGEST », spécialisée pour la gestion de stock ayant la capacité de gérer les produits(médicaments), les commandes, les approvisionnements, les clients, les fournisseurs. Pourtant des améliorations sont impérativement nécessaires pour avoir l’excellence.

#### Inventaire des moyens matériels et logiciels

##### Moyens matériels

Au niveau conceptuel, l’inventaire des moyens matériels consiste à énumérer les différents matériels existants et utilisés au sein d’une structure ou bien même domaine que touche le système d’information étudié. L’importance de cet inventaire réside sur le fait qu’il permet au préalable d’aider à l’option des outils techniques qui serviront à automatiser le système d’information.

Le tableau 6 représente l’inventaire des moyens matériels utilisés au sein de la structure touchée par le système.

Tableau 6. Moyens matériels de la pharmacie.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ordinateur** | **Marque** | **Processeur** | **Mémoire** | **Système** | **Nombre** |
| Laptop | HP,  DELL, | CORE i3 | 4Go | Windows10,  Windows 7 | 2 |
| Desktop | DELL, ASUS | CORE 2 DUO, CORE i3, CORE | 2Go | Windows7, | 2 |

##### Moyens logiciels

Le tableau 7 illustre les logiciels existants dans la pharmacie.

**Tableau 7:Liste des logiciels existants**

|  |  |
| --- | --- |
| Systèmes d’exploitation : | * Windows 7 Professionnel 64 bits, * Windows 10 Edition Familiale Basique 64 bits, |
| Logiciels bureautiques : | * Microsoft office 2016 * Foxit Reader, * Adobe Reader. |
| Navigateurs web : | * Mozilla Firefox, * Google chrome, * Internet Explorer, * Opera Mini. |
| Antivirus : | * SMADAV, * Panda Protection, * Avast, * 360 Security, * Windows Defender. |
| Outil pour gérer la base des données | * Wamp Server |
| Autres | * Pot Player, * VLC. |

### Critique de l’existant

Les études précédentes sur l’organisation actuelle ont amené à constater quelques détails :

* **Points forts :**
* Le logiciel HJGEST est déjà sur place, une application web spécialisée pour la gestion de stock ;
* L’enregistrement dans un cahier des informations reçues (documents) permet de consulter et de vérifier l’existence des informations en cas de coupure l’électricité ;
* Le travail manuel est plus pratique.
* **Points faibles :**
* HJGEST n’est pas encore fiable, moindre fonctionnalité, et pourtant MySQL perd son efficacité sur de très grandes quantités de données ;
* Il y a difficulté de l’accès à l’information car le travail manuel prend beaucoup de temps ;
* La disposition de la liste et les détails des informations sur des papiers risquent d’être abimés ou perdus.
* Le manque de personnel pose un problème sur la répartition des tâches.

### Proposition de solution

L’analyse préalable nous permet de définir les solutions possibles pour améliorer l’organisation et le traitement des informations dans la Pharmacies.

On a deux (02) solutions à proposer :

**Solution 1 :** Recherche et achat d’un logiciel (Gestion de stock, gestion de vente, etc).

**Solution 2 :** Création d’une application spécialisée basée sur la conception initiale du logiciel HJGEST et avec amélioration, ainsi que l’utilisation d’autres SGBD comme ORACLE, SQL Server, …

Le tableau 8 présente la comparaison entre ces deux solutions.

**Tableau 8:Comparaison des solutions**

|  |  |  |
| --- | --- | --- |
| Solutions | Avantages | Inconvénients |
| Solution 1 | - Pas de temps à perdre  - Beaucoup de choix | - Prix cher  - Imperfectionnent |
| Solution 2 | - Fiabilité et Robustesse  - Ergonomie  - Plus de fonctionnalités | - Perte de temps  - Imperfectionnent |

### Conception avant-projet

Suite à la formulation des besoins, de l’analyse de l’existant et de la préconisation de solutions, nous procédons maintenant à la conception avant-projet. Cette étape consiste à analyser parmi un ensemble de solutions potentielles, celle qui est la plus adéquate au présent projet et selon ses aspects fondamentaux. A chaque analyse, nous ferons une brève étude comparative des outils candidats et adaptés au projet, ensuite nous préciserons l’outil retenu tout en justifiant le choix, ladite analyse se terminera par la présentation de l’outil choisi.

#### Méthode de gestion de projet

Sans organisation, ni méthode, la gestion de projet n’est rien. Afin de réussir un projet en respectant le délai, le budget et les ressources, l’équipe doit être organisée et efficace. Les méthodes de gestion de projet aident à accomplir chaque étape du projet, de la planification à la mise en œuvre, dans un souci d’efficacité et de rentabilité. Choisir une méthode pour conduire un projet permet à tous les acteurs de travailler efficacement ensemble, en suivant des règles clairement définies.

En informatique, quand il s’agit de gestion de projet, on entend souvent parler des méthodes classiques ou traditionnelles et des méthodes agiles. Mais, quelles grandes différences existent entre ces deux approches et lesquelles sont celles qui conviennent au mieux au présent projet ?

##### Etude comparative

[5.a] Les méthodes classiques sont les plus utilisées en gestion de projet. Ces méthodes sont aussi appelées « en cascade » car chaque étape doit être terminée pour passer à la suivante. En appliquant cette méthodologie, l’équipe du projet suit le cahier des charges à la lettre et travaille sur la totalité du projet jusqu’à sa livraison. Il n’y a pas d’interaction avec le client qui recevra son projet une fois que celui-ci soit terminé. Tout doit être prévu. L’équipe s’engage sur un planning précis et définit l’ensemble des tâches à réaliser. L’inconvénient majeur c’est qu’il n’y a pas de place pour les changements imprévus, il est donc préférable de tout réussir du premier coup. Au final, le client est souvent déçu car ses attentes ont évolué ou le contexte a changé. Des imprévus sont survenus, et des ajustements doivent être effectués, ce qui implique des retards et des coûts supplémentaires. Pour que ces méthodes fonctionnent correctement, une bonne communication entre les membres de l’équipe est essentielle.

[5.a, 6.a] Plus efficaces et moins rigides que les méthodes traditionnelles, les méthodes agiles placent les besoins du client au centre des priorités du projet. Elles offrent une plus grande flexibilité et une meilleure visibilité dans la gestion du projet, ce qui permet à l’équipe d’être plus réactive aux attentes du client. Le projet est ainsi découpé en mini-projets, chacun nécessitant la validation du client pour passer au suivant. Le dialogue avec le client est privilégié, les retours et les ajustements sont possibles. On prend davantage en considération l’évolution des besoins du client. Les méthodes agiles font de plus en plus d’adeptes.

Selon cette brève étude, vu que contrairement aux méthodes classiques ou traditionnelles, les méthodes agiles incluent le client dès le début à la fin du projet. En plus, elles considèrent que les besoins ne peuvent être figés et proposent au contraire de s’adapter aux changements de ces derniers. C’est pourquoi, nous les avons optées.

En parlant des méthodes agiles, il existe différents exemples potentiels à opter pour bien conduire le présent projet. Parmi ces exemples se trouvent SCRUM et XP (eXtreme Programming). Pour l’option de l’approche qui est en adéquation au projet, il est indispensable d’effectuer une étude comparative. Ainsi, le tableau 9 représente l’étude comparative entre SCRUM et XP.

Tableau 9. Etude comparative entre les méthodes agiles SCRUM et XP.

|  |  |  |
| --- | --- | --- |
|  | Avantages | Inconvénients |
| SCRUM | * Sprint durant de 2 à 4 semaines ; * Ne prescrit aucune pratique d’ingénierie ; * Pas besoin de priorisation de fonctionnalité ; | * N’autorise pas les changements dans les sprints une fois la réunion de planification de ces derniers terminé ; * Cadre non entièrement décrit ; |
| XP | * Accepte les modifications dans les itérations ; * Disposition de fonctionnalités prêtes à appliquer ; | * Itération durant seulement d’une à deux semaines ; * Très stricte sur l’exigence des pratiques d’ingénierie ; * Priorisation des fonctionnalités ; |

##### Choix et justification

Après ces analyses, nous avons pu constater que SCRUM met l’accent sur l’auto-organisation tandis que XP lui, il met l’accent sur des fortes contraintes sur les pratiques d’ingénierie. C’est pourquoi il est beaucoup plus préférable d’opter pour SCRUM plutôt que XP seulement dans le but d’inclure instantanément le propriétaire du projet qui n’a éventuellement les connaissances techniques revendiquées par la majorité de l’équipe.

##### Présentation de SCRUM

Scrum est un cadre de développement dans lequel des équipes plurifonctionnelles réalisent des produits de manière itérative et incrémentale. Scrum structure le développement en cycles de travail appelés Sprints. Ces itérations ne durent jamais plus de quatre semaines (deux semaines la plupart du temps), et s’enchaînent l’une après l’autre sans interruption. Les Sprints sont d’une durée limitée, ou « timeboxés » -ils se terminent à une date spécifique, que le travail soit terminé ou non, et ne sont jamais prolongés. Généralement les équipes Scrum choisissent une durée de Sprint et la maintiennent durant le projet, jusqu’à ce qu’elles puissent encore augmenter leurs productivités et utiliser alors un cycle plus court. Au début de chaque sprint, une équipe plurifonctionnelle sélectionne des éléments (exigences du client) dans une liste priorisée. L’équipe s’accorde collectivement sur une cible constituée de ce qu’elle pense pouvoir livrer à la fin du sprint, de manière tangible et réellement « terminée ». Aucun nouvel élément n’est ajouté durant le sprint. Scrum accepte le changement pour le Sprint suivant, mais la durée fixe d’un Sprint en cours est faite pour se focaliser sur un objectif relativement stable, clair et limité. Chaque jour, l’équipe se réunit brièvement afin de contrôler sa progression et ajuster les prochaines étapes nécessaires à la finalisation du travail restant. A la fin de chaque Sprint, une revue est organisée avec les parties prenantes durant laquelle l’équipe montre ce qu’elle a réalisé. Le feedback obtenu peut être pris en compte sur le Sprint suivant. Scrum insiste sur la nécessité de livrer un produit opérationnel à la fin de chaque Sprint, et réellement « terminé ». Dans le cas de logiciels, cela signifie un système intégré, entièrement testé, documenté pour ses utilisateurs et potentiellement déployable. Les rôles, les artefacts et les évènements clés, constituent la vue ensemble de Scrum représentée par la figure 3.

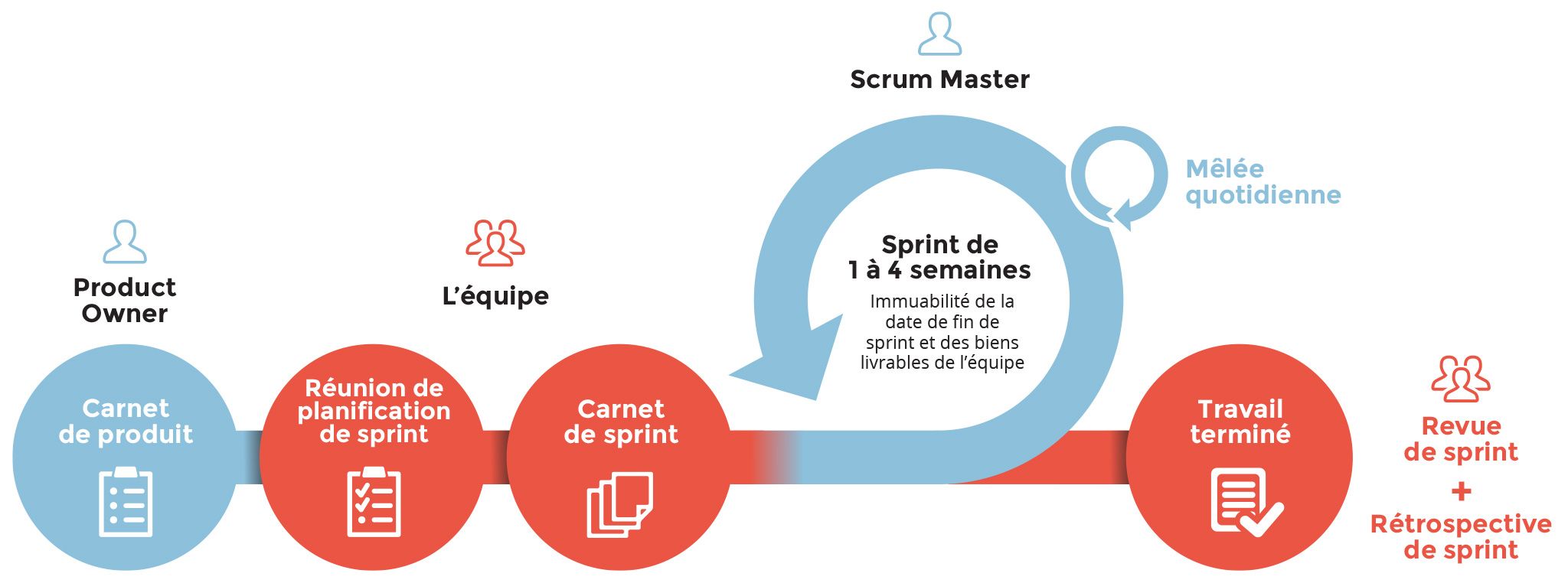


Figure 3. Vue d'ensemble de Scrum. [10.a]

L’un des thèmes majeurs de Scrum c’est « inspecter et adapter ». Etant donné que le développement implique inévitablement l’apprentissage, de l’innovation, et des surprises, Scrum souligne la nécessité de développer par étapes courtes, d’inspecter à la fois l’efficacité des pratiques mises en œuvre et le produit qui en résulte, et adapter ainsi les objectifs du produit et les activités du processus.

**Les rôles dans Scrum**

Dans Scrum, il y a trois rôles : le Product Owner, l’Equipe, et le Scrum Master. Ensemble ils forment l’Equipe Scrum.

**Le Product Owner** est chargé de maximiser le retour sur investissement (ROI) en identifiant les fonctionnalités du produit, en les transférant dans une liste priorisée, en choisissant celles qui doivent être en tête de liste pour le prochain Sprint, et en repriorisant et en affinant continuellement cette liste. Lorsqu’il s’agit d’un produit commercial, le Product Owner est responsable des pertes et profits. Dans le cas d’une application interne, le Product Owner n’est pas responsable du ROI au sens commercial du terme (génération de revenus), mais il est tout de même en charge de maximiser le ROI en choisissant-à chaque Sprint-les éléments ayant à la fois la plus forte valeur métier et le moindre coût. En pratique, le mot « valeur » reste un terme relativement vague, et la priorisation peut être influencée par le désir de satisfaire des utilisateurs clés, par la nécessité de s’aligner avec des objectifs stratégiques, de prendre en compte des risques, de s’améliorer, ou par d’autres facteurs. Dans certains cas, le Product Owner et le client sont la même personne ; c’est souvent le cas pour les applications internes. Dans d’autres cas, le client peut représenter des millions de personnes avec une grande variété de besoins, auquel cas le rôle du Product Owner est similaire à celui du Responsable Produit ou du Responsable Marketing chez les éditeurs. Cependant, le Product Owner est quelque peu différent du Responsable Produit traditionnel avec l’ensemble des parties prenantes et passe en revue les résultats de chaque Sprint, au lieu de déléguer les décisions liées au développement à un chef de projet. Il est important de noter que dans Scrum, il y a une et une seule personne possédant la fonction et l’autorité du Product Owner. Cette personne est responsable de la valeur du travail réalisé ; même si cette personne ne doit pas travailler seule.

**L’équipe (également appelée Equipe de développement)** construit le produit qui est défini par le Product Owner : une application ou un site web par exemple. Dans Scrum, l’Equipe est « plurifonctionnelle » : elle inclut toute l’expertise nécessaire pour fournir une version du produit potentiellement livrable à chaque Sprint. Elle est également « auto organisée » (autogérée), avec un grand degré d’autonomie et de responsabilité. L’équipe décide des éléments à implémenter dans un Sprint (éléments issus de la liste proposée par le Product Owner), et des moyens les plus adaptés pour réaliser cet objectif.

Chaque personne dans l’équipe est simplement qualifiée de membre. Il faut noter qu’il n’y a pas de titre de spécialiste au sein d’un groupe qui adopte Scrum ; pas d’analyste fonctionnel, pas de DBA, pas d’architecte, pas de team leader, pas de concepteur IHM, pas de programmeur. Tous les membres travaillent ensemble durant chaque Sprint, par tous les moyens appropriés et nécessaires à l’atteinte de l’objectif qu’ils ont eux-mêmes déterminé.

**Le Scrum Master** aide le groupe à apprendre et à appliquer Scrum afin que la valeur métier se matérialise. Le Scrum Master fait tout ce qui est en son pouvoir pour aider l’équipe, le Product Owner et l’organisation à réussir le projet. Le Scrum Master n’est pas le manager de l’équipe, ni même un chef de projet, un chef d’équipe ou un représentant de l’équipe. Son rôle est plutôt de servir l’équipe ; il aide à supprimer les obstacles, protège l’équipe des interférences extérieurs, et facilite l’adoption par l’équipe des pratiques modernes du développement. Il enseigne, il coache et il guide le Product Owner, l’équipe ainsi que le reste de l’organisation dans une utilisation fine de Scrum. Le Scrum Master est un formateur et un coach. Il s’assure que chacun (y compris le Product Owner, ainsi que le top management) comprenne les principes et les pratiques de Scrum, et il accompagne l’organisation dans le changement souvent difficile mais nécessaire au succès du développement agile. Etant donné que Scrum met en lumière les obstacles et les menaces qui pèsent sur l’efficacité de l’équipe et du Product Owner, il est important d’avoir un Scrum Master engagé et travaillant énergétiquement à la résolution des problèmes auxquels seront confrontés aussi bien l’équipe que le Product Owner. Il y a idéalement un Scrum Master dédié et à plein temps, bien qu’une petite équipe puisse voir l’un de ses membres jouer ce rôle (on réduit alors sa charge sur les travaux courants). D’excellents Scrum Master peuvent provenir de n’importe quelle formation ou discipline : ingénierie, conception, test, responsable produit, chef de projet ou responsable qualité.

Le Scrum Master et le Product Owner ne peuvent pas être la même personne tant leurs objectifs sont différents, et la combinaison des deux rôles conduits souvent à la confusion et à des conflits. Cette combinaison pousse souvent le Product Owner à faire du micro management, ce qui est à l’opposé des équipes autoorganisées telles que Scrum le définit. A la différence d’un manager classique, le Scrum Master ne dit pas à chacun ce qu’il doit faire et n’assigne pas de tâches. Il facilite le processus, en soutenant l’équipe dans son organisation et sa gestion. Si le Scrum Master était précédemment en position de chef de projet au sein de l’équipe, il devrait significativement d’état d’esprit et de style d’interaction, afin que l’équipe puisse pleinement réussir avec Scrum.

#### Méthode de conception

[5] Une méthode de conception définit une démarche reproductible pour obtenir des résultats fiables. Tous les domaines de la connaissance utilisent des méthodes plus ou moins sophistiquées et plus ou moins formalisées. Une méthode d’élaboration de logiciel décrit comment modéliser et construire des systèmes logiciels de manière fiable et reproductible.

De manière générale, les méthodes permettent de construire des modèles à partir d’éléments de modélisation qui constituent des concepts fondamentaux pour la représentation de systèmes ou de phénomènes. Elles définissent également une représentation souvent graphique qui permet d’une part de manipuler aisément les modèles, et d’autre part de communiquer et d’échanger l’information entre les différents intervenants. Une bonne représentation recherche l’équilibre entre la densité d’information et la lisibilité.

##### Etude comparative

La mise en place d’un CRM requiert obligatoirement de conception. A ce stade, 2TUP (2Tracks Unified Process) et RUP (Rational Unified Process) sont des méthodes permettant chacune de concevoir une telle application. Quelles différences existe-t-il entre ces méthodes candidates et quelle est celle qui est la plus adaptée au présent projet ?

Le tableau 10 représente l’étude comparative entre 2TUP et RUP.

Tableau 10. Etude comparative entre 2TUP et RUP.

|  |  |  |
| --- | --- | --- |
|  | Points forts | Points faibles |
| 2TUP | * Fait une large place à la technologie et à la gestion de risque ; * Définit les profils des intervenants, les livrables, les plannings et les prototypes | * Plutôt superficiel sur les phases situées en amont et en aval du développement : capture des besoins, support, maintenance, etc ; |
| RUP | * Spécifie le dialogue entre les différents intervenants du projet. | * Coûteux à personnaliser ; * Très axé processus, au détriment du développement : peu de place pour le code et la technologie. |

##### Choix et justification

Selon la précédente étude correspondante à la comparaison de 2TUP et RUP, nous avons constaté que 2TUP définit les profils des intervenants, les livrables, les plannings et les prototypes. En plus, contrairement à RUP, cette méthode n’est pas coûteuse à personnaliser. C’est pourquoi nous l’avons optée comme méthode de conception pour concevoir le projet.

##### Présentation de la méthode de conception 2TUP

2TUP ou 2Tracks Unified Process est un processus unifié c’est-à-dire construit sur UML, itératif, centré sur l’architecture et conduit par les cas d’utilisation. Le principe fondateur du 2TUP est que toute évolution imposée à un logiciel peut se décomposer et se traiter parallèlement, suivant un axe fonction et un axe technique. Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste à étudier les acteurs qui vont interagir avec le système à construire, les messages qu’échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l’entourent et sont reliés à lui, sur l’axe qui relie un acteur au système on met les messages qu’échangent les deux avec le sens. Le processus s’articule ensuite autour de trois phases essentielles qui sont la branche fonctionnelle, la branche technique et la phase de réalisation.

A l’issu des évolutions du modèle fonctionnel et de l’architecture technique, la réalisation du logiciel consiste à fusionner les résultats de ces deux branches de processus. Cette fusion conduit ainsi à l’obtention d’un processus de développement en forme de Y qui est illustré par la figure 5.

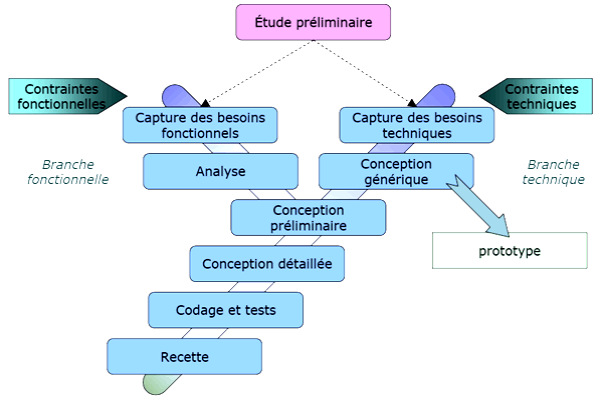


Figure 4. Processus de développement en forme de Y.

La branche gauche ou branche fonctionnelle comporte la capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs. De son côté, la maîtrise d’œuvre consolide les spécifications et en vérifie la cohérence et l’exhaustivité de l’analyse, qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier. Les résultats de l’analyse ne dépendent d’aucune technologie particulière.

La branche droite ou branche technique comporte la capture des besoins techniques, qui recense toutes les contraintes et les choix dimensionnant la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte de contraintes d’intégration avec l’existant conditionnent généralement des prérequis d’architecture technique. Elle comporte aussi la conception générique, qui définit ensuite les composants nécessaires à la construction de l’architecture technique. Cette conception est la moins dépendante possible des aspects fonctionnels. Elle a pour objectif d’uniformiser et de réutiliser les mêmes mécanismes pour tout un système.

L’architecture technique construit le squelette du système informatique et écarte la plupart des risques de niveau technique. L’importance de sa réussite est telle qu’il est conseillé de réaliser un prototype pour assurer sa validité.

Enfin, la branche du milieu comporte :

* La conception préliminaire, qui représente une étape délicate, car elle intègre le modèle d’analyse dans l’architecture technique de manière à tracer la cartographie des composants du système à développer ;
* La conception détaillée, qui étudie ensuite comment réaliser chaque composant ;
* L’étape de codage, qui produit ces composants et teste au fur et à mesure les unités de code réalisées ;
* L’étape de recette, qui consiste enfin à valider les fonctions du système développé.

#### Présentation de la notation UML

UML est le résultat d’un long processus initialisé depuis maintenant deux ans par trois des méthodologistes les plus réputés : Grady Booch, Ivar Jacobson et Jim Rumbaugh. UML est à la fois la synthèse et le successeur naturel de leurs différents travaux. Par sa standardisation par l’OMG, qui devrait aboutir dans le courant de l’année 1997, et de par son adoption par les plus grands acteurs du monde de l’informatique : IBM, Microsoft, Oracle, Hewlett Packard, pour ne citer que les plus grands, UML est appelé à devenir à très court terme le standard pour la modélisation des applications informatiques de demain.

Le langage de modélisation UML respecte un certain nombre de règles sur les concepts manipulés (classes, attributs, opérations, paquetages …) ainsi que sur la syntaxe d’écriture et le formalisme de représentation graphique. L’ensemble de ces règles constitue en soi un langage de modélisation qui a fait l’objet d’un méta-modèle UML. L’intérêt de disposer d’un méta-modèle UML permet de bien maîtriser la structure UML et de faciliter son évolution. Cette approche a été généralisée par l’OMG en normalisant la représentation des méta-modèles par la définition en 1997 d’un méta-modèle défini dans le MOF. Le MOF représente ainsi un super langage de définition des méta-modèles.

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

* Six diagrammes structurels :
* Diagramme de classes qui montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc.
* Diagramme d’objets qui montre les instances des éléments structurels et leurs liens à l’exécution.
* Diagramme de structure composite qui montre l’organisation interne d’un élément statique complexe.
* Diagramme de composants qui montre des structures complexes, avec leurs interfaces fournies et requises.
* Diagramme de déploiement qui montre le déploiement physique des « artefacts » sur les ressources matérielles.
* Diagramme de paquetages qui montre l’ensemble des packages et leurs relations.
* Sept diagrammes comportementaux :
* Diagramme de cas d’utilisation qui montre les interactions fonctionnelles entre les acteurs et le système à l’étude.
* Diagramme de vue d’ensemble des interactions qui fusionne les diagrammes d’activité et de séquence pour combiner des fragments d’interaction avec des décisions et des flots.
* Diagramme de séquence qui montre la séquence verticale des messages passés entre objets au sein d’une interaction.
* Diagramme de communication qui montre la communication entre objets dans le plan au sein d’une interaction ;
* Diagramme de temps qui fusionne les diagrammes d’états et de séquence pour montrer l’évolution de l’état d’un objet au cours du temps.
* Diagramme d’activité qui montre l’enchaînement des actions et décisions au sein d’une activité.
* Diagramme d’états qui montre les différents états et transitions possibles des objets d’une classe.

La figure 6 représente l’arborescence des diagrammes UML.



Figure 5. Arborescence des diagrammes UML. [13.a]

#### Outil de conception

Un outil de conception est un logiciel informatique permettant d’automatiser des conceptions logicielles. Construire des modèles est l’un des principaux rôles des outils de conception et ces constructions permettront de représenter les problèmes dans le monde réel en un modèle compréhensible par une majorité d’experts d’un domaine bien déterminé.

##### Etude comparative

Multiples sont les outils de modélisation permettant de construire nos modèles de conception tels que ArgoUML, Visual Paradigm, Entreprise Architect, Modelio, UML Designer. Parmi ces outils, on entend souvent parler dans les grandes communautés la puissance des deux logiciels de conception ArgoUML et Visual Paradigm. Mais, qu’est-ce qui fait leurs puissances donc et comment se différencient-t-ils ?

Le tableau 11 représente l’étude comparative entre ArgoUML et Visual Paradigm.

Tableau 11. Etude comparative entre ArgoUML et Visual Paradigm. [10, 14. a]

|  |  |  |
| --- | --- | --- |
|  | Avantages | Inconvénients |
| ArgoUML | * Facilité d’utilisation ; * Logiciel libre ; * Facilité de l’analyse de son fonctionnement de base ; | * Ne prend en charge que quelques diagrammes UML ; * Génération de code uniquement en Java ; * N’est pas fait pour les professionnels. * N’utilise pas la notation UML pour la visibilité des attributs et des méthodes ; * Moins complet que d’autres logiciels de conception. |
| Visual Paradigm | * Génération de code informatique à partir de diagramme et vis-versa ; * Prise en charge de tous les diagrammes UML ; * Prise en charge de la modélisation des processus métiers ; * Dispose de différentes fonctions facilitant la conception des diagrammes aux débutants ; * Réalisation de mapping objet relationnel à partir de diagramme de classes. | * Beaucoup trop cher ; * Requiert un minimum d’expérience pour une meilleure pratique ; * Pas Open Source ; |

##### Choix et justification

Au regard de ces données, et selon des amples analyses, ArgoUML ne prend en charge que seulement quelques diagrammes UML contrairement à Visual Paradigm qui a la capacité de produire la totalité des diagrammes UML. En plus, l’expérience du concepteur du système en utilisant Visual Paradigm est un atout considérable pour pouvoir accélérer le processus de création des modèles de conception. Ce sont les raisons pour lesquelles nous avons choisi Visual Paradigm comme outil de conception.

##### Présentation de Visual Paradigm [15.a]

Visual Paradigm est un outil de conception et de gestion puissant, multiplateforme et pourtant facile à utiliser pour les systèmes informatiques. Visual Paradigm fournit aux développeurs de logiciel la plateforme de développement de pointe pour créer des applications de qualité plus rapidement, mieux et moins cher. Il facilite une excellente interopérabilité avec d’autres outils CASE et la plupart des principaux IDE, ce qui excelle l’ensemble des processus de développement Model-Code-Deploy dans cette solution à guichet unique. Avec Visual Paradigm, il est possible et facile de dessiner toutes sortes de diagrammes UML 2.x, des Diagrammes de Relation d’Entité (ERD), des diagrammes d’ORM et les diagrammes qui servent à modéliser les processus métiers tels que les diagrammes de flux de données, les diagrammes de processus métiers, les diagrammes de chaîne de processus de conduite d’évènements et les diagrammes de carte processus.

#### Langage de programmation

* **Back end**

Pour le côté back-end, nous avons utilisé comme langage, C# (C Sharp) est un langage de programmation aux nombreux paradigmes mais principalement considéré comme oriente objet. Le langage a été développé par Microsoft en 2002. Il est fortement inspiré de langages comme java ou C++.

* **Avantages offerts par C# :**

C# offre les fonctionnalités suivantes :

* Gestion automatique de la mémoire grâce à un ramasse-miette
* Introspection pour manipuler dynamiquement les objets
* Gestion des erreurs grâce aux exceptions
* Support de fonctionnalités intéressantes telles que les indexeurs, les délégués et les événements
* Surcharge des opérateurs
* Génération de fichier XML pour la documentation des codes

C# possède une forte similarité avec Java. Cependant, il dispose également des fonctionnalités importantes qui lui sont propres notamment : les passages par références, les delegates, les événements, les indexeurs, les objets de types valeur, les surcharges des opérateurs, les pointeurs. C# propose également un langage permettant de faire des requêtes sur des données en faisant abstraction de leur types appelé LINQ. Ce langage permet d’utiliser facilement un jeu d’instructions supplémentaires afin de filtrer des données, faire des sélections. Il existe plusieurs domaines d’applications pour LINQ :

* Linq To Entities ou Linq To SQL qui utilisent ces extensions du langage sur les bases de données
* Linq To XML qui utilise ces extensions de langage pour travailler avec les fichiers XML
* Linq To Object qui permet de travailler avec des collections d’objets en mémoire.

* **Entity Framework**

C’est un ORM initialement créé par Microsoft et maintenant disponible en open source. C’est un projet mature permettant de se connecter à diverses bases de données . Le rôle d’Entity Framework consiste entre autres à :

* Modéliser ses données et générer la base de données correspondante
* Générer un modèle à partir d’une base de données existante
* Gérer tous les accès à la base de données (lecture, écriture, suppression, …)
* **Front-end**

Il existe plusieurs frameworks pour développer le front-end de l’application. Le tableau 14 montre la comparaison de quelques-uns.

**Tableau 12:Comparaison de Framework Front-end**

|  |  |  |
| --- | --- | --- |
|  | Avantages | Inconvénients |
| Angular | - Un framework complet avec une architecture élégante  - Une documentation très étoffée qui permet aux développeurs de trouver toutes les informations nécessaires rapidement  - Une grande communauté, qui permet toujours de pouvoir poser des questions en cas de blocage  - Des améliorations constantes et régulières  - Modèle MVVM (Modèle-View- View Model), qui permet aux développeurs de travailler séparément sur la même section d’application en utilisant le même ensemble de données  - Un moteur DI (Injection de dépendances)  - Les avantages de TypeScript : Vérification statique et opérationnelle de la saisie, fonction de saisie très performante | - Complexité de la syntaxe  - Spécificités du TypeScript, qui a une courbe d’apprentissage difficile |
| React js | - React est centré sur Javascript en encapsulant le HTML dans JS  - Réduction des nombres d’opérations sur le DOM, optimisation et accélération du processus de mises à jour  -Plus facile à apprendre  -Une véritable de boite à outils qui offre aux développeurs un large éventail d’options Flexibilité et réactivité  -React nécessite une bonne compréhension des base du HTMK et ne nécessite pas de syntaxe complexe tel qu’apprendre TypeScript  -100% open source, nombreuses amélioration et contribution régulières de la communauté des développeurs  -La bibliothèque est adaptée au SEO et elle se concentre sur la vitesse de rendu | -Très peu de documentation officielle  - Pas de moteur pour injection de dépendances  - Malgré la légèreté de la syntaxe, React requiert une connaissance approfondie pour intégrer l’interface |

D’après la comparaison, Nous avons choisi comme framework, React pour le Front-end puisque il est plus rapide à apprendre et à démarrer. Des plus React permet d’utilisateurs des bibliothèque tierces pendant le processus de développement. Il propose également un processus de liaison de donnes bidirectionnel.

#### Système de Gestion de Base de Données (SGBD)

En informatique un système de gestion de base de données (SGBD) est un logiciel destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

Un serveur informatique est un dispositif informatique matériel et logiciel qui offre des services à un ou plusieurs clients.

Un client est un programme installé sur l’ordianteur qui envoie des demandes à un serveur pour effectuer une requête.

Le tableau 11 montre la comparaison entre MySQL, SQL et Oracle.

**Tableau 13:Comparaison entre MySQL, Oracle et MS SQL Server**

|  |  |  |
| --- | --- | --- |
| **SGBD** | **Avantages** | **Inconvénients** |
| MySQL | - Dispose d’une interface graphique pour saisir les requêtes ;  - OpenSource, bien que les critères de licence soient de plus en plus difficiles à supporter ;  - Facilité de déploiement et de prise en main. | - Ne convient pas pour les BD de grande taille ;  - Assez peu de richesse fonctionnelle ;  - Manque de robustesse avec de fortes volumétries ;  - Pas d'héritage de table ;  - Pas de vue matérialisée. |
| SQL Server | - Administration aisée ;  - Fonction d'audit évolué ;  - Langage T-SQL très convivial,intégration de CLR;  - Niveau de SQL très près de la norme SQL ;  - Compression des données et des sauvegardes ;  - Une des bases plus performantes sous Windows. | - Les versions Express sont limitées en termes de fonctionnalités.  - Pas de prise en charge du LDA ;  - Pas d'intégration Java, orientation C# |
| **Oracle** | - Richesse fonctionnelle ;  - Fonction d'audit évolué ;  - Gestion centralisée de plusieurs instances ;  - Pérennité de l'éditeur | - Administration complexe ;  - Prix élevé ;  -Fort demandeur de ressource |

* **Choix retenu**

**SQL Server** est un système de gestion de base de données (SGBD) en langage SQL incorporant entre autres un SGBDR (SGBD relationnel ») développé et commercialisé par la société Microsoft. Il fonctionne sous les OS Windows et Linux (depuis mars 2016), mais il est possible de le lancer sur Mac OS via Docker, car il en existe une version en téléchargement sur le site de Microsoft.

Dans ce projet, on a utilisé le SGBD SQL Server en raison de :

* La richesse en fonctionnalité et la sécurité ;
* L’utilisation de la version gratuite permis pour les entreprises.

#### IDE

Visual Studio est adopté en tant qu’environnement de développement intégré aussi bien pour le front end que pour le back end. Durant ce projet, nous avons utilisé la version 2022 de l’application.

#### Service web

Un service web est un ensemble de fonctions d’application associées pouvant être appelées par programme via internet. Les entreprises peuvent associer différents services web de manière dynamique afin d’exécuter des transactions complexes demandant un minimum de programmation. Les services web permettent aux acheteurs et aux fournisseurs du monde entier de se trouver, d’entrer en contact de manière dynamique, et d’exécuter des transactions en temps réel, le tout avec très peu d’interaction humaine.

Les services web sont des applications modulaires autonomes et autodescriptives qui peuvent être publiées, localisées et appelées à travers internet.

Les services web sont autonomes : côté client, aucun logiciel supplémentaire n’est requis. Un simple langage de programmation prenant en charge XML et les clients http suffit pour commencer. Côté serveur, un serveur web et un moteur de servlets sont nécessaires. Le client et le serveur peuvent être mis en œuvre dans différents environnements. Le service web peut activer une application existante sans qu’il soit nécessaire d’écrire la moindre ligne de code.

Les services web contiennent leurs propres descriptions. Le client et le serveur doivent simplement reconnaître le format et le contenu des messages de requête et de réponse. La définition du format de message est transmise avec le message. Aucun référentiel de métadonnées externe n’est nécessaire, ni aucun outil de génération de code.

Les services web sont modulaires. Les services web simples peuvent être associés pour former des services web plus complexes, soit en utilisant des techniques d’enchaînement d’opérations, soit en faisant appel à des services web de niveau inférieur à partir d’une implémentation de service web.

Les services web sont non tributaires de plateforme : les services web sont basés sur un ensemble concis de normes XML ouvertes, conçus pour favoriser l’interopérabilité entre un service web et des clients sur de nombreuses plateformes informatiques utilisant différents langages de programmation.

##### Etude comparative

En parlant de service web, REST et SOAP sont deux éléments souvent comparés l’un à l’autre comme étant des services web dans la conception des applications client-serveur. Qu’est-ce qui les différencie ?

La différence majeure entre ces deux éléments est le degré de liaison entre le client et le serveur. Un client développé avec le protocole SOAP ressemble à un logiciel d’ordinateur, car il est étroitement lié au serveur. Si une modification est effectuée d’un côté ou de l’autre, l’ensemble peut ne plus fonctionner. Il faut effectuer des mises à jour du client s’il y a des changements sur le serveur et vice-versa.

Un client de type REST sait utiliser un protocole et des méthodes standardisées. Son application doit rentrer dans ce modèle. On ne crée pas de méthodes supplémentaires, on utilise les méthodes standardisées que l’on développe pour le type de media dont on a besoin. Il y a en conséquence beaucoup moins de couplage entre le client et le serveur : un client peut utiliser un service de type REST sans aucune connaissance de l’API. A l’inverse, un client SOAP doit tout savoir des éléments qu’il va utiliser pendant son interaction avec le serveur, sinon cela ne fonctionnera pas.

Par ailleurs, REST possède des caractéristiques spécifiques qui le différencient de SOAP. REST est indépendant d’un protocole. On peut utiliser aussi bien le protocole HTTP que FTP, tant qu’il s’agit d’un protocole possédant un schéma standard pour une URI. REST est aussi standard que les éléments utilisables pour développer des applications. Si on utilise le protocole HTTP, alors certaines parties comme l’authentification ou encore la sécurité seront standardisées, car elles le sont avec HTTP.

En plus, REST fournit une alternative plus légère. Au lieu d’utiliser XML pour effectuer une requête, REST s’appuie souvent sur une URL simple. Dans certaines situations, on doit fournir des informations supplémentaires de manière particulière, mais la plupart des services web ²²utilisant REST reposent exclusivement sur l’obtention des informations nécessaires à l’aide de l’approche URL.

##### Choix et justification

Selon ces critères de comparaison, nous avons choisi à utiliser REST ou Representational State Transfer comme service web car REST semble beaucoup plus facile à utiliser avec la technologie PHP plutôt que SOAP.

##### Présentation du service web REST [21]

Qu’est-ce que REST ? Il s’agit d’un modèle d’architecture proche des origines du web. REST est le tenant de la simplicité utile face à l’usine tentaculaire des web service et des normes WS-\* associées. Au premier abord, il est vrai que REST est un exemple rutilant de simplicité, en expliquant que l’infrastructure web existante est un socle parfaitement suffisant pour faire(presque) tout ce qui est nécessaire à une SOA.

REST signifie Respresentational State Transfert : c’est un concept formalisé en 2000 par Roy Fielding. Ce style d’architecture est censé représenter l’essence d’une architecture web bien fondée et efficace : un réseau de ressources (vue comme une machine à états virtuelle) au travers desquelles l’utilisateur avance dans son application en utilisant des liens (transition d’état) qui lui donnent accès à d’autres ressources (état de l’application) qui lui sont transmises et « rendues » pour son utilisation. Il s’agit du style d’architecture aujourd’hui communément utilisé dans une infrastructure web.

Ce style prône la standardisation des services pour garantir l’interopérabilité, une approche assez sensée en soi. Pour REST, le monde est un ensemble de ressources, accessibles par quatre services uniquement, représentés par les méthodes http : GET, PUT, POST, DELETE. Tout est ensuite affaire d’identification de ces ressources, via le concept d’URI. Depuis le milieu des années 1990, nous avons vu éclore des solutions d’envergure basées sur cette architecture, et en premier lieu le web lui-même (au sens de réseau de sites). L’idée est de reconnaître dans cette technologie un support pour implémenter une SOA.

#### Outil de test d’API

Par définition, un outil de test d’API est une application permettant de tester le bon fonctionnement des interfaces de programmation ou bien même API. Comme principal rôle, les outils de test d’API aident les développeurs à mettre en place des APIs fonctionnels et indépendamment de la conception au préalable d’IHM.

##### Etude comparative

Dans le monde du développement informatique, il existe plusieurs outils permettant de d’envoyer des requêtes HTTP et parmi ces derniers se trouvent les plus connus qui sont Postman, Dell Boumi ainsi que IBM API Management. Mais, lequel parmi ces trois outils est le mieux placé pour le présent projet.

Le tableau 18 représente l’étude comparative entre les outils de test d’API Postman, Dell Boumi et IBM API Management.

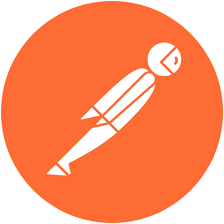
Tableau 14. Etude comparative entre Postman, Dell Boumi et IBM API Management.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Postman | Dell Boumi | IBM API Management |
| Prix à partir de | | 8,00 USD/mois | 550,00 USD/mois | 650,00 USD/mois |
| Version d’essai gratuite | | Oui | Oui | Non |
| Version gratuite | | Oui | Non | Non |
| Plateforme | | Windows, Mac | Windows | Windows |
| Facilité d’utilisation | | 4.5/5 | 4.4/5 | 4.7/5 |
| Service client | | 4.3/5 | 4.2/5 | 4.7/5 |
| Rapport qualité-prix | | 4.8/5 | 4.2/5 | 4/5 |
| Fonctionnalités | Conception d’API | Oui | Non | Non |
| Versionning | Oui | Non | Non |
| Gestion de test | Oui | Non | Non |
| Gestion de cycle de vie des API | Oui | Non | Non |
| Tableau de bord | Oui | Non | Non |
| Portails de développeurs | Non | Oui | Oui |

##### Choix et justification

Selon les informations représentées dans le précédent tableau, nous avons constaté que parmi les trois outils préconisés, Postman est beaucoup plus moins coûteux et en plus, il dispose d’une version d’essai gratuite et d’une version non payante. En termes de fonctionnalités, malgré que le portail de développeur soit absent, Postman conçoit des API, gère les tests et les cycle de vie de ces API, c’est pourquoi on s’est focalisé à l’utiliser.

##### Présentation de l’outil de test d’API Postman

Postman est une plateforme de collaboration pour le développement d’API. Les fonctionnalités de Postman simplifient chaque étape de la création d’une API et simplifient la collaboration afin de permettre de créer de meilleurs API plus rapidement. Postman s’intègre aux cas d’utilisation spécifiques, que ça soit dans le développement, les tests et l’assurance qualité ou dans la gestion de produit. Parmi ces cas d’utilisation, Postman peut s’avérer être utile dans le développement API-First qui consiste à libérer des services fiables en construisant des API avant de déployer du code. Dans le développement d’application, cet outil élimine les dépendances et réduit les délais de production en faisant en sorte que les équipes Front-end et Back-end travaillent en parallèle. En plus, il automatise les tests manuels et met rapidement les consommateurs au courant de ce que l’API peut faire et de son fonctionnement.

#### Système de Versionning

Ce système permet de mutualiser un développement. Un groupe de développeurs autour d'un même développement se sert de l'outil pour stocker toute évolution du code source. Le système gère les mises à jour des sources pour chaque développeur, conserve une trace de chaque changement. C’est pour cette raison qu’on nous a imposé d’utiliser GIT. Tout d’abord, Git est un système libre. Il ne peut donc pas y avoir de limitations contractuelles sur son utilisation. Quel que soit le nombre de collaborateurs, de projets ou de mises à jour, Git sera toujours gratuit.

Il possède également l'avantage d'un système de gestion de versions déconnecté. C’est-à-dire qu’il est possible de travailler en mode hors connexion avec son dépôt tout en gardant des fonctionnalités avancées de collaboration lorsqu’on est connecté.

## Analyse conceptuelle

### Dictionnaire des données

Le dictionnaire des données est le résultat de la phase de collecte des données. C’est l'ensemble des données correspondant à la description de toutes les entités du modèle. Chaque donnée représentera une rubrique d'information homogène pour chaque entité du système d'information.

Pour chaque donnée, il indique :

* **La rubrique** : Il s’agit d’un libellé désignant une donnée.
* **La description** : Il s’agit d’une mention décrivant ce à quoi la donnée correspond
* **Le type de données** :
* **A ou** **Alphabétique** : lorsque la donnée est uniquement de caractères alphabétiques.
* **N ou Numérique :** lorsque la donnée est composée uniquement des nombres (entiers ou réels).
* **AN ou Alphanumérique :** lorsque la donnée peut être à la fois de caractères alphabétiques et numériques.
* **Date :** lorsque la donnée est une date.
* **Booléen :** Vrai ou faux.
* **La taille :** elle s’exprime en nombre de caractères ou de chiffres.
* Et parfois des **remarques** ou **observations** complémentaires.

Le tableau 15 représente le dictionnaire des données du projet.

Tableau 15. Dictionnaire des données.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom de rubrique** | **Description** | **Type** | **Taille** | **Example** |
| AdresseCli | Adresse du client | AN | 30 |  |
| DateAchat | DateAchat | Date | 10 | jj-mm-aaaa |
| DateEntre | Date entrée médicament | Date | 10 | jj-mm-aaaa |
| DatePer | Date Péremption | Date | 10 | jj-mm-aaaa |
| DateProd | Date Production | Date | 10 | jj-mm-aaaa |
| DateVente | DateVente |  |  | jj-mm-aaaa |
| Client | Nom client | AN | 25 |  |
| CodeFrs | Code fournisseurs | AN | 10 | Fr001 |
| NomFa | Nom famille médicament | AN | 25 |  |
| NomMed | Nom médicament | AN | 25 |  |
| numClient | Numéro téléphone client | AN | 13 | xxx xx xxx xx |
| numFrs | Numéro téléphone fournisseur | 13 | 13 | xxx xx xxx xx |
| PrixAchat | Prix Achat | D | 7 |  |
| PrixVente | Prix de vente | D | 7 |  |
| Stock | Stock médicament | N | 6 |  |
| Type | Type action vente | AN | 10 |  |

### Règles de gestion

Les règles de gestion sont des normes léguées par les acteurs du système actuel qui sont aussi les futurs usagers de l’application une fois mise en œuvre. Elles sont aussi les logiques qui mettent en relation la plupart des données recensées dans le précédent dictionnaire. En plus des règles de gestion, nous donnerons aussi les règles de calcul qui font la logique entre certaines données présentes dans le dictionnaire.

Les règles de gestion du système sont :

|  |  |  |
| --- | --- | --- |
| * RG1 | : | Un client peut effectuer une commande ou plusieurs ; |
| * RG2 | : | Une commande appartient à un seul client ; |
| * RG3 | : | Une commande contient au moins un produit ; |

### Règles de calcul

Ce sont :

RC1 : Le stock de médicament augmente après l’approvisionnement (achat médicament) (Somme du stock existant et de la quantité approvisionnée).

RC2 : Le stock de médicament diminue après la vente (Somme du stock existant et de la quantité vente).

RC3 : Prix de vente égale prix achat fois pourcentage (Ex : Prix Achat \* 3.5%).

RC4 : Prix total est égal de la quantité commandée et du prix unitaire.

### Représentation et spécification des besoins

Le processus unifié est un processus piloté par des cas d’utilisations et orienté vers la diminution de risques. Le but principal d’un système d’information est de satisfaire les besoins du client et les cas d’utilisations permettent d’illustrer ces besoins.

#### Diagramme de cas d’utilisation

#### Recueil des besoins (pré-étude)

* **Identification des acteurs**
* Un acteur représente un rôle joué par une personne ou une chose qui interagit avec le système.
* Une même personne physique peut jouer le rôle de plusieurs acteurs (client, fournisseur). Plusieurs personnes peuvent jouer le même rôle et agir comme un seul et même acteur (tous les clients).
* La détermination des acteurs permet de déterminer les limites du système.
* Il existe 4 grandes catégories d’acteurs : principaux, secondaires, le matériel externe, les autres systèmes.
* Une fois identifiés, les acteurs doivent être décrits de manière claire et concise avec notamment le détail de leurs responsabilités. [9]

La figure 6 montre le formalisme de représentation d’un acteur.

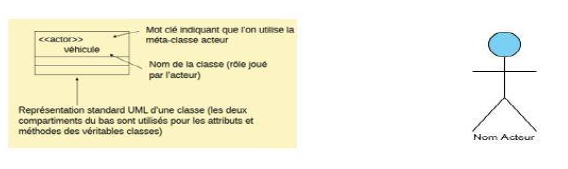


Figure 6:Formalisme d'un acteur

Le stick man est en général utilisé pour les acteurs humains et la méta-classe pour les dispositifs externes.

Pour ce projet, l’acteur participe au fonctionnement de l’application pour la gestion de Pharmacies. Il est dénommé Responsable ou Administrateur.

Responsable : gère les mouvements des stocks, c’est-à-dire, l’enregistrement de commande ainsi que la livraison et l’approvisionnement de médicament. Il gère aussi le fournisseur et famille de médicament.

* **Identification des messages**

Un message représente la spécification d’une communication unidirectionnelle entre objets qui transporte de l’information avec l’intention de déclencher une activité chez le récepteur.

Nous allons identifier les principaux messages échangés entre le système et les acteurs. C’est à dire la spécification d’une communication entre le système et les acteurs.

Le tableau 16 montre les messages entre l’acteur et le système.

Tableau 16:Message entre l’acteur et le système

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Acteur** | **Acteur** | **Système** | **Système** | **Acteur** |
| R  E  S  P  O  N  S  A  B  L  E  O  U  A  D  M  I  N  I  S  T  R  A  T  E  U  R | Demande de s’authentifier | | Résultat de l’authentification (Page de l’authentification) | |
| Demande de consulter une liste | | Résultat de la liste sous forme d’un tableau | |
| Demande de saisie | | Résultat avec ouverture de formulaire de saisie | |
| Demande d’enregistrer ou d’ajouter | | Message de confirmation | |
| Confirmation d’une action | | Enregistrement avec notification | |
| Demande d’une modification | | Affichage de formulaire contenant les données à modifier | |
| Demande de suppression | | Message de confirmation de la suppression | |
| Demande de se déconnecter | | Message de confirmation | |
| Confirmation de se déconnecter | | Résultat de la déconnexion | |

#### Capture des besoins fonctionnels

* 1. **Besoins fonctionnels**

Les besoins fonctionnels sont illustrés dans le tableau 17.

Tableau 17:Besoins fonctionnels

|  |  |
| --- | --- |
| **Besoin Fonctionnel** | **Description, Permet à l’utilisateur de :** |
| Gérer fournisseur | - Consulter la liste des fournisseurs ;  - Ajouter, modifier et supprimer fournisseurs. |
| Gérer client | - Consulter la liste des clients ;  - Ajouter, modifier et supprimer clients. |
| Gérer famille | - Consulter la liste des fournisseurs ;  - Ajouter, modifier et supprimer fournisseurs. |
| Gérer médicament | - Consulter la liste des médicaments existants ;  - Ajouter ou créer, modifier et supprimer produits. |
| Faire approvisionnement | - Consulter le stock des médicaments ;  - Achat des médicaments. |
| Gérer commande | - Consulter la liste des commandes effectuées ;  - Enregistrer, modifier et supprimer une commande ;  - Générer un bon de commande. |
| Gérer la livraison | - Consulter la liste des livraisons effectuées  - Générer un bon de livraison |

* **Cas d’utilisation**

Un cas d’utilisation doit produire un résultat observable pour un ou plusieurs acteurs ou parties prenantes du système. Il doit définir les exigences fonctionnelles du système, d’où chaque cas d’utilisation correspond à une fonction métier du système.

La figure 7 présente le formalisme d’un cas d’utilisation.

Formalisme :

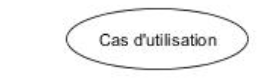
******

Figure 7:Formalisme d'un cas d'utilisation

* **La relation d’inclusion, formalisée par le mot-clé « include »**

Une instance du cas d’utilisation source comprend également le comportement décrit par le cas d’utilisation destination.

La figure 8 montre le formalisme de la relation d’inclusion.

Formalisme :

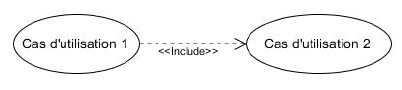
****

Figure 8:Formalisme de la relation d'inclusion

Signification : Cas d’utilisation 2 est une partie obligatoire de Cas d’utilisation 1 (Cas 1 inclut Cas 2).

* **La relation d’extension, formalisée par le mot-clé « extend »**

Le cas d’utilisation source ajoute son comportement au cas d’utilisation destination. L’extension peut être soumise à condition. Cette extension permet de modéliser des variantes de comportement d’un cas d’utilisation ou des parties de cas d’utilisation facultatives.

La figure 9 affiche le formalisme de la relation d’extension.

Formalisme :

****

Figure 9:Formalisme de la relation d'extension

Signification : Cas d’utilisation 2 est une partie optionnelle de cas d’utilisation 1(cas 2 est une extension du cas 1).

* **La relation de généralisation et spécialisation**

La figure 10 présente le formalisme de la relation de généralisation et spécialisation.

Formalisme :

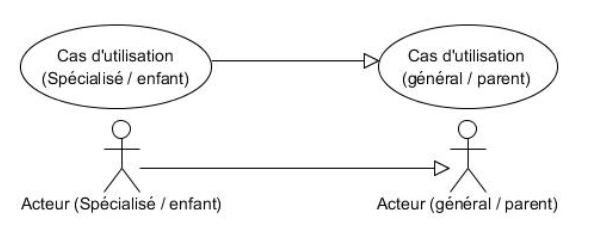


Figure 10:Formalisme de la relation de généralisation et spécialisation

Signification : L’élément (acteur ou cas d’utilisation) spécialisé (ou enfant) hérite de toutes les caractéristiques de l’élément général (ou parent).

D’où le formalisme de diagramme de cas d’utilisations, présenté dans la figure 11.

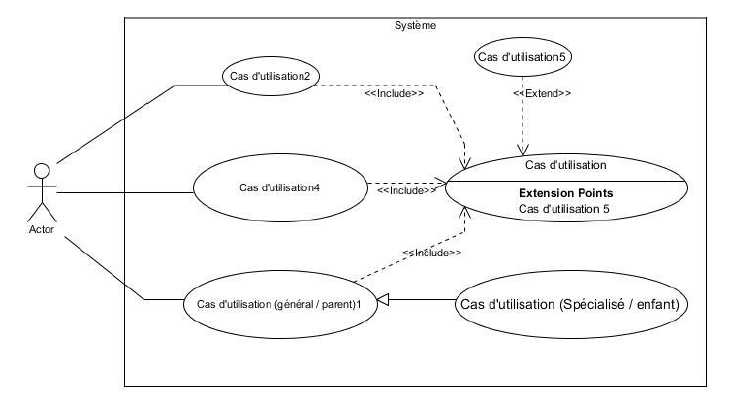


Figure 11:Formalisme de diagramme de cas d'utilisations

* **Diagramme de cas d’utilisation du système**

La figure 12 représente le digramme de cas d’utilisation du système.

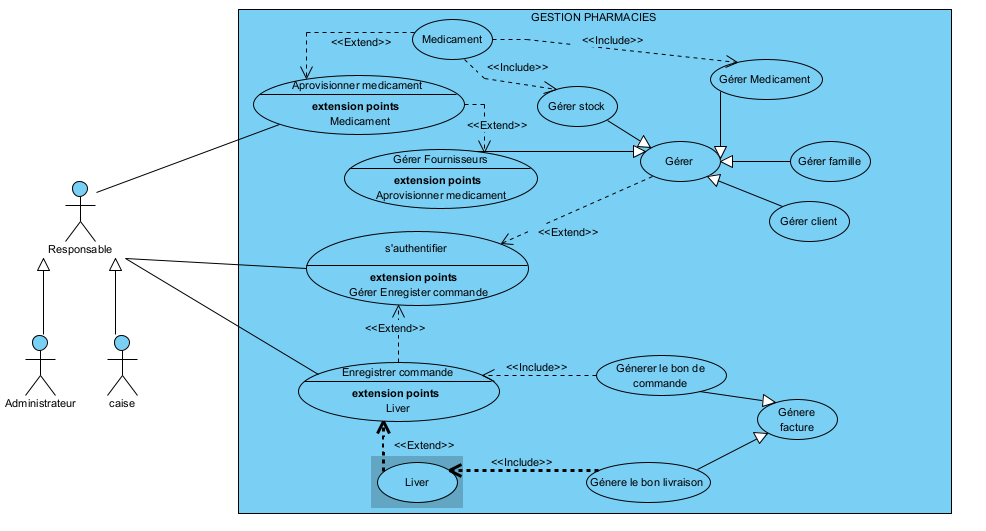


Figure 12:Diagramme de cas d'utilisation

* 1. **Les besoins non fonctionnels**

Les besoins non fonctionnels sont les suivants :

* **La sécurité et la confidentialité :** L’accès à l’application doit être sécurisé ;
* **La maintenance et la réutilisabilité :** Le code source de l’application doit être assez compréhensible pour faciliter sa maintenance ;
* **L’ergonomie et la convivialité :** L’application doit pouvoir présenter des interfaces bien lisibles pour faciliter l’utilisation de l’application ;
* **La robustesse et fiabilité :** la capacité du logiciel à fonctionner, même en présence d’événements exceptionnels tels que la saisie d’informations erronées par l’utilisateur. La gestion des erreurs qui doivent être signalées par des messages clair pour bien guider l’utilisateur de l’application.

#### Priorisation des cas d’utilisation

La priorisation des cas d’utilisation permet de décrire chronologiquement les opérations de l’utilisateur et les classés selon leur ordre d’importance.

Le tableau 18 montre la priorisation de cas d’utilisations.

Tableau 18:Priorisation de cas d'utilisations

|  |  |
| --- | --- |
| **Priorité** | **Cas d’utilisation** |
| Haute | S’authentifier |
| Haute | Gérer |
| Moyenne | Approvisionner |
| Moyenne | Enregistrer commande |
| Moyenne | Médicament |
| Bas | Livrer |
| Bas | Générer facture |

#### Description textuelle pour chaque cas d’utilisation

* + - 1. **Cas d’utilisation « S’authentifier »**

Le tableau 16 illustre la description textuelle pour « S’authentifier ».

Tableau 19:Description textuelle pour « S'authentifier »

|  |  |
| --- | --- |
| **Titre** | S’authentifier |
| **Objectif** | Permettre à l’utilisateur d’entrer au menu principal. |
| **Acteur** | Responsable |
| **Précondition** | L’utilisateur est à la page d’authentification |
| **Post condition** | L’utilisateur est à la page d’accueil |
| **Scenario nominal** | 1. L’utilisateur entre le mot de passe et le nom ;  2. Le système vérifie l’identification ;  3. Le système affiche le menu principal. |
| **Scenario alternatif** | **A1. Le système n’arrive pas à identifier l’utilisateur**  A1 démarre à l’étape 2 du scenario nominal ;  3. Le système affiche un message d’erreur. |

* + - 1. **Cas d’utilisation « Gérer »**

Pour les cas d’utilisation « Gérer » (Gérer client, Gérer fournisseur, Gérer médicament), les traitements sont identiques.

Le tableau 20 démontre la description textuelle pour le cas d’utilisation « Gérer ».

Tableau 20:Description textuelle pour « Gérer »

|  |  |
| --- | --- |
| **Titre** | Gérer |
| **Objectif** | Permettre à l’utilisateur de faire : l’ajout, la modification et la suppression d’un enregistrement. |
| **Acteur** | Responsable |
| **Scenario nominal** | 1. L’utilisateur demande à système d’afficher les enregistrements ;  2. Le système affiche les enregistrements demandés ;  3. L’utilisateur demande à système le formulaire d’ajout ;  4. Le système affiche le formulaire d’ajout ;  5. L’utilisateur remplit le formulaire ;  6. Le système vérifie, l’enregistre et rafraichit la liste. |
|  | * **A1. Modification d’un enregistrement**   A1 démarre à l’étape 2 du scenario nominal ;  3. L’utilisateur demande à système de modifier un  enregistrement ;  4. Le système affiche le formulaire rempli ;  5. L’utilisateur saisie les modifications nécessaires ;  6. Le scenario reprend à l’étape 6 du scenario nominal.   * **A2. Suppression d’un enregistrement**   A2 démarre à l’étape 2 du scenario nominal ;  3. L’utilisateur demande de supprimer un enregistrement ;  4. Le système demande de confirmation ;  5. L’utilisateur confirme la suppression ;  6. Le cas d’utilisation se termine en fin normal. |
| **Exception** | Le système renvoie un message d’erreur si la saisie par l’utilisateur est incorrecte. |

* + - 1. **Cas d’utilisation « Approvisionner médicament »**

Le tableau 20 illustre la description textuelle pour le cas d’utilisation « Approvisionner médicament (Achat) ».

Tableau 21:Description textuelle pour « Approvisionner médicament »

|  |  |
| --- | --- |
| **Titre** | Approvisionner médicament |
| **Objectif** | Permettre à l’utilisateur d’enregistrer l’achat de médicament et mettre à jour le stock. |
| **Acteur** | Responsable |
| **Précondition** | L’utilisateur est authentifié ;  L’utilisateur a déjà gérer le médicament et le fournisseur. |
| **Post condition** | La base de données se met à jour et la liste se rafraichît. |
| **Scenario nominal** | 1. L’utilisateur demande d’approvisionner ;  2. Le système demande de choisir le fournisseur ;  3. L’utilisateur saisit le code du fournisseur ;  4. Le système vérifie le code du fournisseur ;  5. Le système affiche des champs à remplir concernant le médicament ;  6. L’utilisateur remplit les autres champs obligatoires ;  7. Le système vérifie les données récemment saisit par le responsable ;  8. Le système enregistre et affiche la liste contenant les médicaments récemment achetés. |
| **Scenario alternatif** | **A1. Le fournisseur est introuvable**  A1 démarre à l’étape 4 du scenario nominal ;  5. Le système affiche un message d’erreur ;  6. Le scenario nominal reprend à l’étape 3.  **A2. Information de médicament incorrecte**  A2 démarre à l’étape 7 du scenario nominal ;  8. Le système affiche un message d’erreur ;  9. Le scenario nominal reprend à l’étape 6.  **A3. Ajout d’une autre médicament**  A3 démarre à l’étape 8 du scenario nominal ;  9. L’utilisateur demande d’ajouter une autre matière première ;  10. Le scenario nominal reprend à l’étape 5. |

* + - 1. **Cas d’utilisation « Stock »**

Le tableau 22 présente la description textuelle pour le cas d’utilisation « Stock »

Tableau 22:Description textuelle pour « Médicament »

|  |  |
| --- | --- |
| **Titre** | Stock |
| **Objectif** | Permettre à l’utilisateur de mettre à jour à la fois le stock du médicament |
| **Acteur** | L’utilisateur est authentifié ;  L’utilisateur a déjà gérer le médicament |
| **Post condition** | Le stock du médicament se met à jour. |
| **Scenario nominal** | 1. L’utilisateur demande le détail de stock ;  2. Le système affiche toute la liste de médicament avec stock existant (Quantité médicament) ;  3. L’utilisateur demande d’ajouter un nouveau stock ;  4.Le système affiche le formulaire à remplir ;  5. L’utilisateur remplit le formulaire d’ajouter un nouveau stock ;  6. Le système vérifie la saisie ;  7. Le système enregistre le nouveau stock et affiche la liste. |
| **Scenario alternatif** | **A1. Le stock de la matière première est insuffisant**  A1 démarre à l’étape 2 du scenario nominal ;  7. Le système affiche un message de demande d’approvisionnement. |

* + - 1. **Cas d’utilisation « Enregistrer commande »**

Le tableau 23 montre la description textuelle pour le cas d’utilisation « enregistrer commande ».

Tableau 23:Description textuelle pour « enregistrer commande »

|  |  |
| --- | --- |
| **Titre** | Enregistrer commande |
| **Objectif** | Permettre à l’utilisateur d’archiver la commande. |
| **Acteur** | Responsable |
| **Précondition** | L’utilisateur est authentifié |
| **Post condition** | La base de données se met à jour et la liste de commande se rafraîchit. |
| **Scenario nominal** | 1. L’utilisateur demande à système d’enregistrer une commande ;  2. Le système demande à l’utilisateur de choisir ou entrer le client ;  3. L’utilisateur choisir client ;  4. Le système vérifie le client et envoie le formulaire concernant le médicament ;  5. L’utilisateur remplit les champs obligatoires ;  6. Le système vérifie la saisie et enregistre. |
| **Scenario alternatif** | **A1. Le client est introuvable**  A1 démarre à l’étape 4 du scenario nominal ;  5. Le système renvoie un message de suggestion d’ajouter le client. |
| **Exception** | Le système renvoie un message d’erreur si l’utilisateur a saisie incorrect. |

* + - 1. **Cas d’utilisation « Livrer »**

Le tableau 24 montre la description textuelle pour le cas d’utilisation « Livrer ».

Tableau 24:Description textuelle pour le cas d'utilisation livrer

|  |  |
| --- | --- |
| **Titre** | Livrer |
| **Objectif** | Permettre à l’utilisateur d’effectuer une livraison. |
| **Acteur** | Responsable |
| **Précondition** | L’utilisateur est à la liste de la commande |
| **Post condition** | La base de données se met à jour. |
| **Scenario nominal** | 1. L’utilisateur demande à système d’effectuer une livraison ;  2. Le système affiche un formulaire à remplir ;  3. L’utilisateur remplit les champs obligatoires ;  4. Le système vérifie et enregistre les données saisies. |
| **Exception** | Le système renvoie un message d’erreur si la saisie contient d’erreur. |

* + - 1. **Cas d’utilisation « Générer facture »**

Le tableau 22 illustre la description textuelle pour le cas « Générer facture ».

Tableau 25:Description textuelle pour le cas d'utilisation « Générer facture »

|  |  |
| --- | --- |
| **Titre** | Elaborer facture |
| **Objectif** | Permettre à l’utilisateur de générer une facture. |
| **Acteur** | Responsable |
| **Précondition** | L’utilisateur est authentifié ; |
| **Post condition** | Facture (bon de commande ou bon de livraison ) |
| **Scenario nominal** | 1. L’utilisateur demande d’afficher un détail ;  2. Le système affiche le résultat ;  3. L’utilisateur demande de générer une facture. |
| **Scenario alternatif** | **A1. Bon de commande**  A1 démarre à l’étape 3 du scenario nominal ;  4. Le système vérifie et génère un bon de commande.  **A2. Bon de livraison**  A2. Démarre à l’étape 3 du scenario nominal :  5. Le système vérifie et génère un bon de livraison. |
| **Exception** | Le système renvoi un message d’erreur. |

#### Diagramme des séquences système pour chaque cas d’utilisation

Le diagramme de séquence fait parties des diagrammes comportementaux (dynamique) et plus précisément des diagrammes d’interactions. Il permet de représenter des échanges entre les d objets et acteurs du système en fonction du temps.

Les éléments du diagramme de séquence sont :

* **Les lignes de vie :** ce sont des traits pointillés à la verticale de l’objet qui peut être considéré comme un axe temporel (le temps s’écoule du haut vers le bas)**.**
* **Les messages :** Les messages sont des communications d’un objet vers un autre objet représentés sous forme de flèches.
* **Les occurrences d'exécution :** représente la période d’exécution d’une opération. Ils sont représentés par un rectangle vertical.
* **Les fragments combinés** : une ou plusieurs séquences de traitement enfermé dans un cadre et exécuté dans des circonstances spécifiques désignés. (ALT, OPT, LOOP…).

La figure 13 représente le formalisme d’un diagramme de séquence.

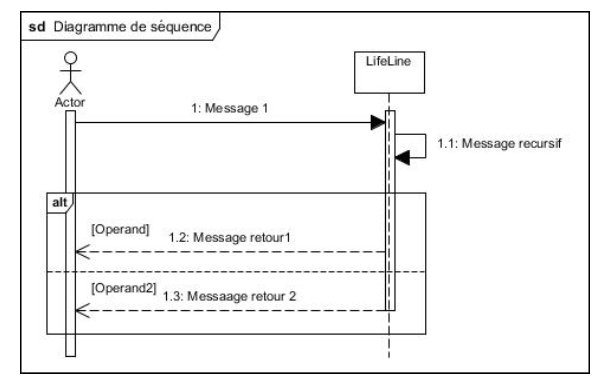


Figure 13:Formalisme diagramme de séquence système

* + - 1. **Diagramme de séquence système « S’authentifier »**

La figure 14 montre le diagramme de séquence système « s’authentifier ».

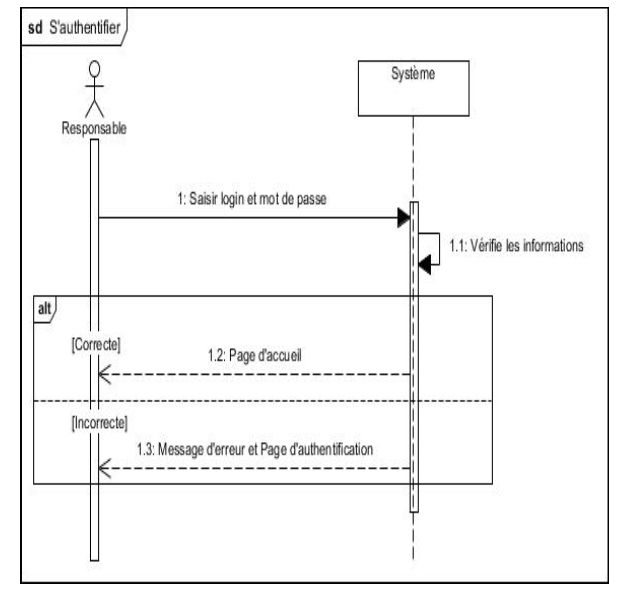
****

Figure 14:Diagramme de séquence système « S’authentifier »

* + - 1. **Diagramme de séquence système « Gérer »**

La figure 15 affiche le diagramme de séquence système « Gérer ».

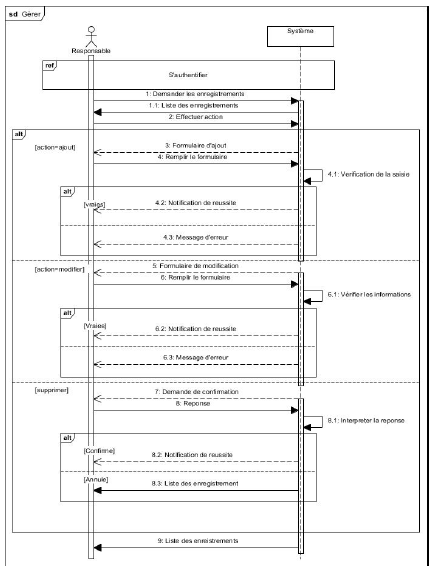
****

Figure 15:Diagramme de séquence système « Gérer »

### Description textuelle pour chaque cas d’utilisation

Les besoins techniques ou besoins non fonctionnels sont des contraintes techniques qui caractérisent le système et dont le traitement est généralement centré sur un exploitant. Ces besoins peuvent concerner aussi les différentes contraintes d’implémentation.

Dans notre cas, le système devra :

* Inclure la fonctionnalité d’authentification pour gérer l’accès aux différentes fonctionnalités fournies ;
* Permettre aux utilisateurs de se déconnecter pour quitter le système tout en éliminant les traces de connexion sur un ordinateur quelconque ;
* Permettre à un utilisateur de réinitialiser son mot de passe de connexion.

La figure 10 représente le diagramme des cas d’utilisation techniques du système.



Figure 16. Diagramme des cas d'utilisation techniques du projet.

Le tableau 43 représente la description textuelle du cas d’utilisation « S’authentifier ».

Tableau 26. Description textuelle du cas d'utilisation « S'authentifier ».

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | *Nom* | : | S’authentifier. | | *Objectifs* | : | Sécuriser l’application et contrôler l’accès aux différentes fonctionnalités. | | *Acteurs* | : | Responsable Commercial, Gérant, Administrateur. | | *Précondition* | : | Emplacement dans la page d’authentification. | | |
| *Scénarios nominaux* | *Scénarios alternatifs* |
| 1. Les acteurs concernés entrent les informations d’authentification (Login et Mot de passe). 2. Les acteurs concernés demandent de s’authentifier en cliquant sur le bouton « S’authentifier ». 3. Le système vérifie les informations saisies. | 1. Les informations d’authentification requises sont incomplètes. 2. Les informations saisies sont erronées. 3. Le système avertit les acteurs concernés sur l’erreur commise. |
| |  |  |  | | --- | --- | --- | | *Postcondition* | : | Enregistrement de quelques informations du Personnel (Code Personnel, nom, téléphone, mail) pendant sa session. | | |

Le tableau 44 représente la description textuelle du cas d’utilisation « Se déconnecter ».

Tableau 27. Description textuelle du cas d'utilisation « Se déconnecter ».

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | *Nom* | : | Se déconnecter. | | *Objectif* | : | Quitter l’application. | | *Acteurs* | : | Responsable commercial, Gérant, Administrateur. | | *Précondition* | : | Accès à l’application proprement dite. | | |
| *Scénarios nominaux* | *Scénario alternatif* |
| 1. Les acteurs concernés demandent de se déconnecter en cliquant sur l’icône de déconnexion. 2. Le système les redirige vers la page d’authentification. |  |
| |  |  |  | | --- | --- | --- | | *Postcondition* | : | Suppression de toutes les informations enregistrées pendant l’authentification. | | |

Le tableau 45 représente la description textuelle du cas d’utilisation « Réinitialiser mot de passe ».

Tableau 28. Description textuelle du cas d'utilisation Réinitialiser mot de passe.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | *Nom* | : | Réinitialiser mot de passe. | | *Objectif* | : | Permettre aux utilisateurs d’obtenir un nouveau mot de passe au cas où celui-ci a été oublié. | | *Acteurs* | : | Responsable commercial, Gérant, Administrateur. | | *Précondition* | : | Emplacement dans la page d’authentification. | | |
| *Scénarios nominaux* | *Scénarios alternatifs* |
| 1. Les acteurs concernés cliquent sur le lien mot de passe oublié. 2. Le système les redirige vers la page dédiée à la réinitialisation de mot de passe. 3. Le système leur avertit qu’un code de réinitialisation de 6 chiffres a été envoyé à leurs numéros de téléphone.   Les acteurs concernés entrent le code de 6 chiffres et valide la réinitialisation. | 1. Echec lors de l’envoi du Sms aux acteurs concernés. 2. Les acteurs concernés entrent un code incorrect ou erroné.   Le système leur avertit les erreurs commises. |
| |  |  |  | | --- | --- | --- | | *Postcondition* | : | Permission aux utilisateurs d’accéder aux fonctionnalités. | | |

### Diagramme des cas d’utilisation global du système

Le diagramme des cas d’utilisation global d’un système est un diagramme qui regroupe en plus des cas d’utilisation fonctionnels de ce même système les cas d’utilisation techniques tout en prenant en compte les relations qui font leurs liaisons.

La figure 11 représente le diagramme des cas d’utilisation global du système.

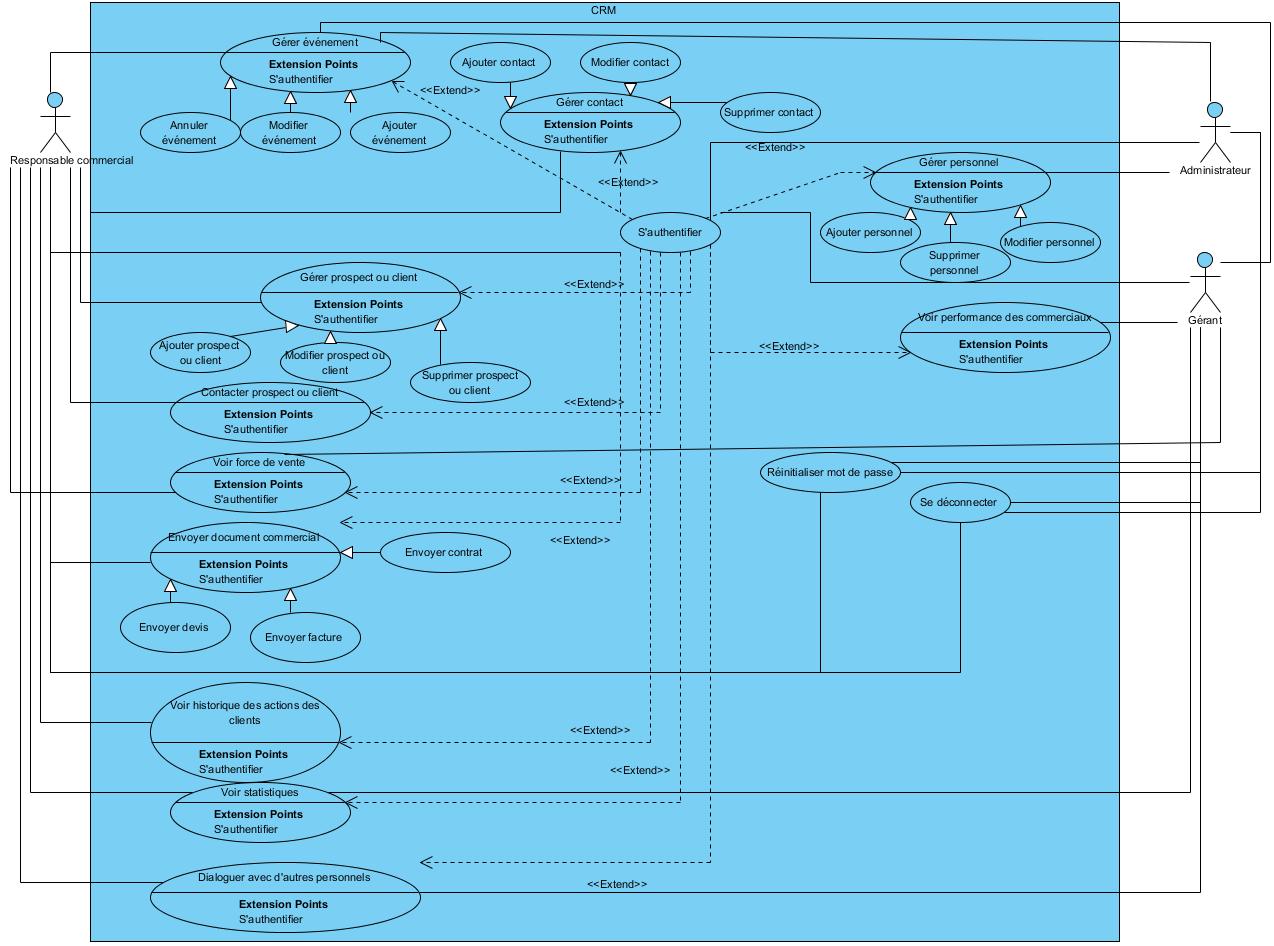


Figure 17. Diagramme des cas d'utilisation global du système.

### Modélisation du domaine

#### Diagramme de séquence système de chaque cas d’utilisation

[44.a] Les diagrammes de séquence systèmes sont des diagrammes d’interaction qui détaillent la manière dont les opérations sont effectuées. Comme les diagrammes de séquence peuvent être utilisés pour capturer l’interaction entre les objets dans le contexte d’une collaboration, l’un des principaux objectifs des diagrammes de séquence est de passer des exigences exprimées en tant que cas d’utilisation au prochain niveau de raffinement, plus formel.

Les diagrammes de séquence sont centrés sur le temps et ils indiquent visuellement l’ordre de l’interaction en utilisant l’axe vertical du diagramme pour représenter l’heure à laquelle les messages sont envoyés et à quel moment.

Les diagrammes de séquence capturent les interactions à différents niveaux de granularité :

* Interactions de haut niveau entre l’utilisateur du système et le système, entre le système et d’autres systèmes ou entre sous-systèmes (parfois appelés diagrammes de séquence systèmes) ;
* Interaction qui a lieu dans une collaboration qui réalise un cas d’utilisation ou une opération (diagrammes d’instance ou diagrammes génériques) ;
* Les objets de représentation interagissent dans (modèle, vue, contrôleur) modèle MVC du framework logiciel.

La figure 12 illustre le diagramme de séquence système du cas d’utilisation « S’authentifier ».

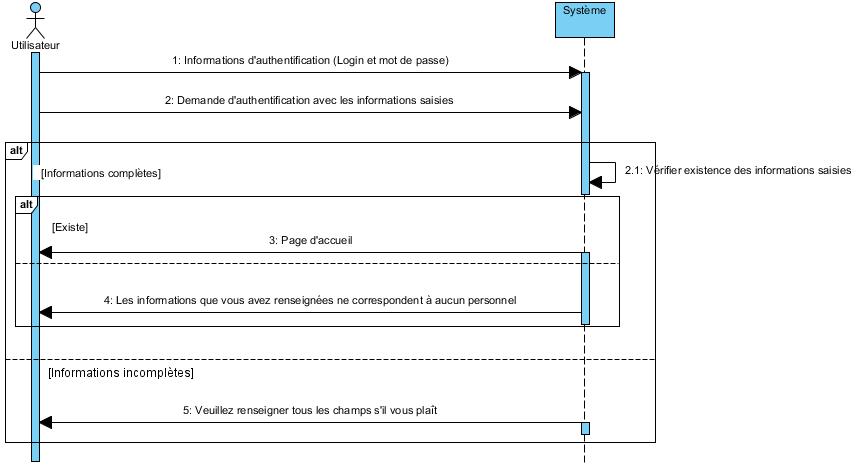


Figure 18. Diagramme de séquence système du cas d'utilisation « S'authentifier ».

La figure 13 illustre le diagramme de séquence système du cas d’utilisation « Se déconnecter ».

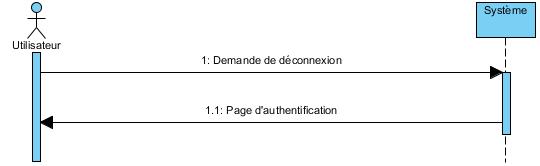


Figure 19. Diagramme de séquence système du cas d'utilisation « Se déconnecter ».

La figure 14 illustre le diagramme de séquence système du cas d’utilisation « Ajouter client ou prospect ».

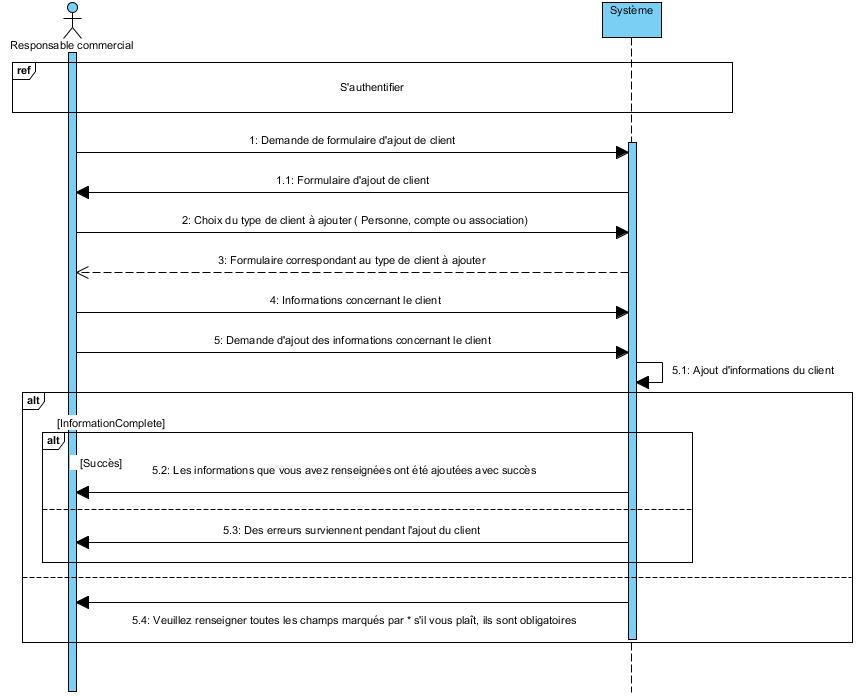


Figure 20. Diagramme de séquence système du cas d'utilisation « Ajouter client ou prospect ».

La figure 15 illustre le diagramme de séquence système du cas d’utilisation « Modifier client ou prospect ».

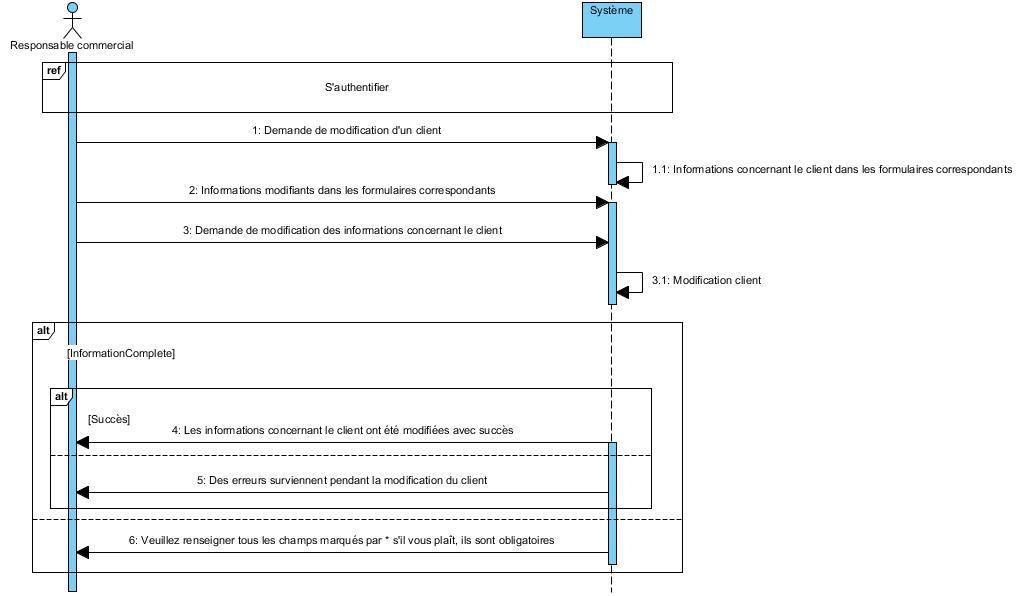


Figure 21. Diagramme de séquence système du cas d'utilisation « Modifier client ou prospect ».

La figure 16 illustre le diagramme de séquence système du cas d’utilisation « Supprimer client ou prospect ».

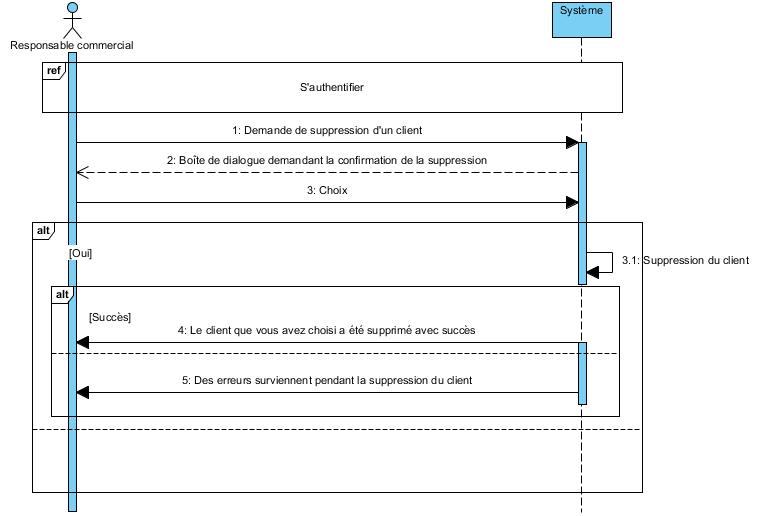


Figure 22. Diagramme de séquence système du cas d'utilisation « Supprimer client ou prospect ».

La figure 17 illustre le diagramme de séquence système du cas d’utilisation « Ajouter contact ».

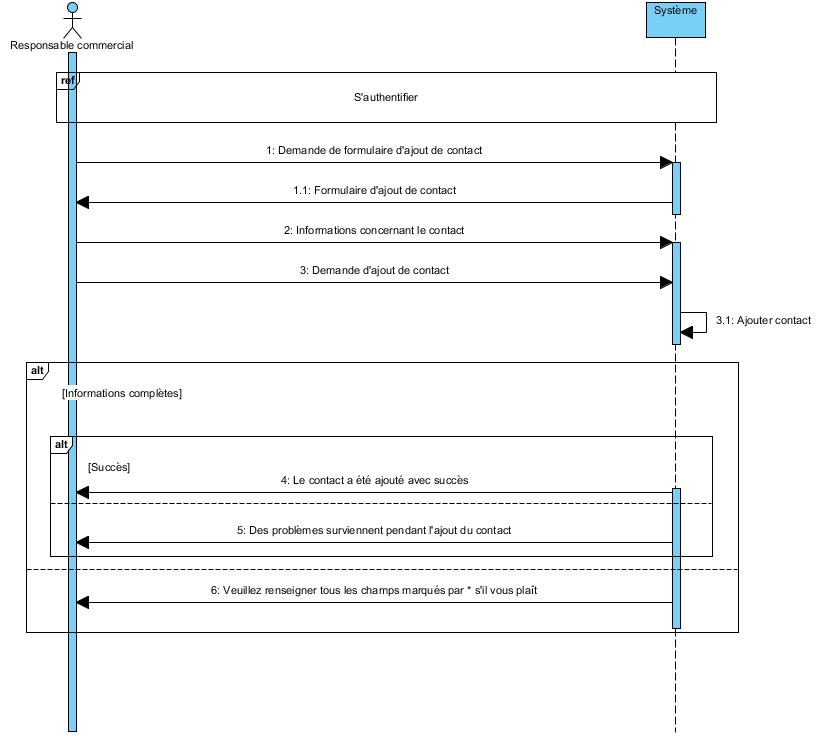


Figure 23. Diagramme de séquence système du cas d'utilisation « Ajouter contact ».

La figure 18 illustre le diagramme de séquence système du cas d’utilisation « Modifier contact ».

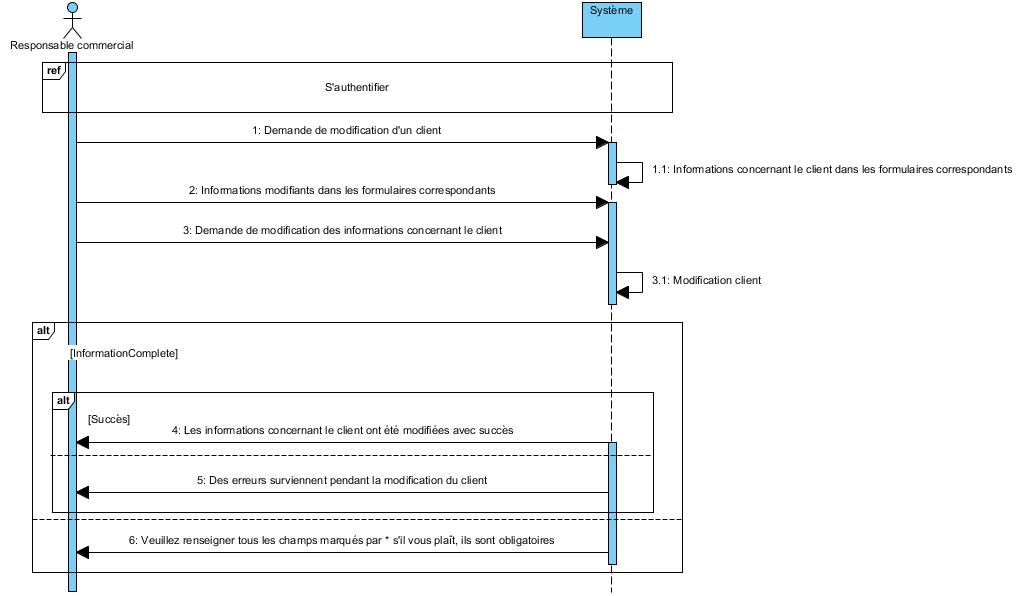


Figure 24. Diagramme de séquence système du cas d'utilisation « Modifier contact ».

La figure 19 illustre le diagramme de séquence système du cas d’utilisation « Supprimer contact ».

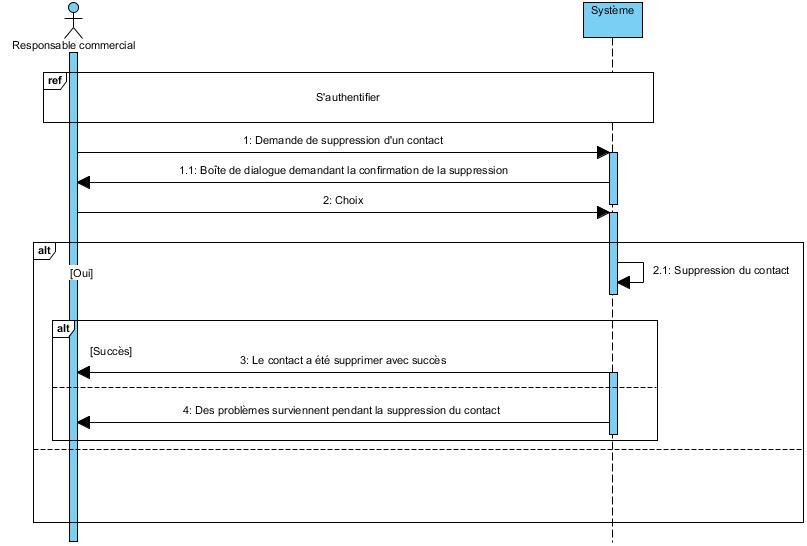


Figure 25. Diagramme de séquence système du cas d'utilisation « Supprimer contact ».

La figure 20 illustre le diagramme de séquence système du cas d’utilisation « Envoyer document commercial ».

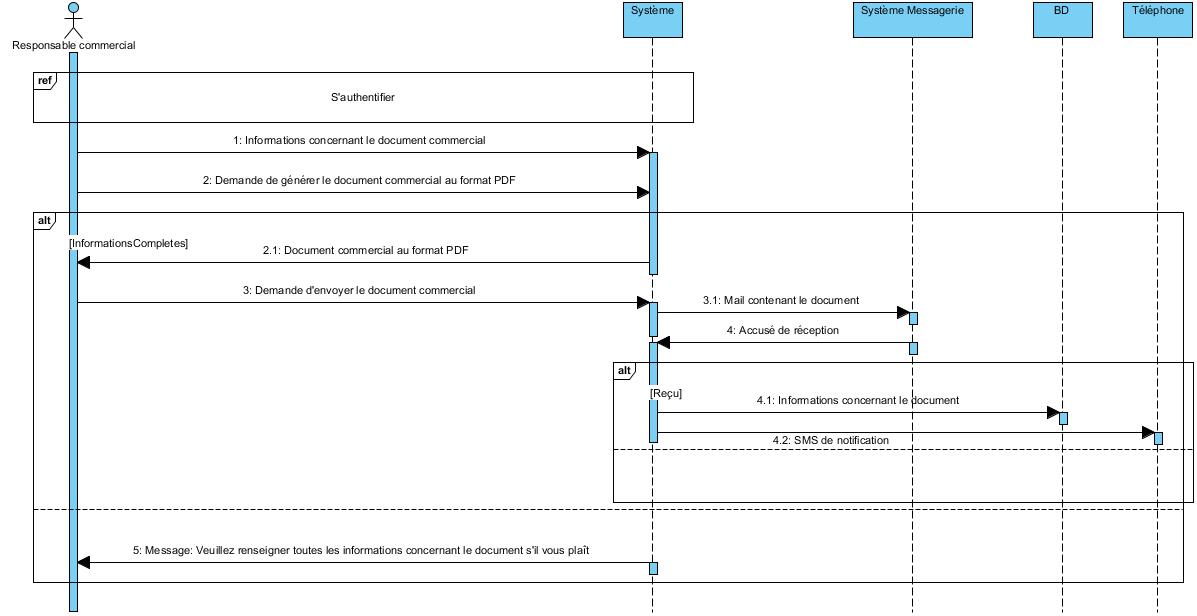


Figure 26. Diagramme de séquence système du cas d'utilisation « Envoyer document commercial ».

La figure 21 illustre le diagramme de séquence système du cas d’utilisation « Ajouter événement ».

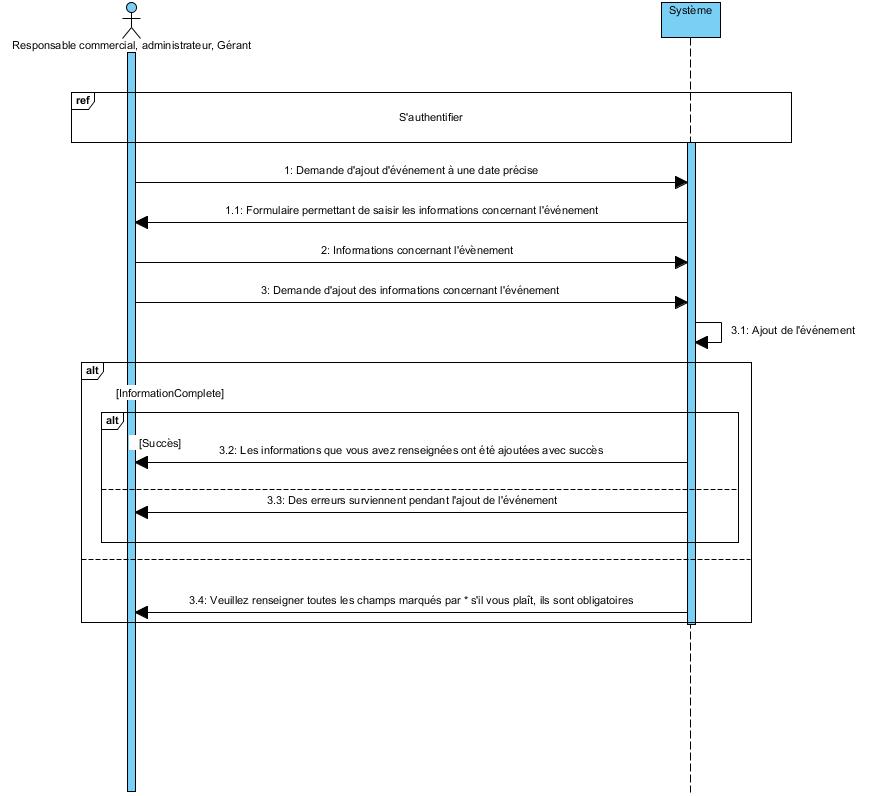


Figure 27. Diagramme de séquence système du cas d'utilisation « Ajouter événement ».

La figure 22 illustre le diagramme de séquence système du cas d’utilisation « Modifier événement ».

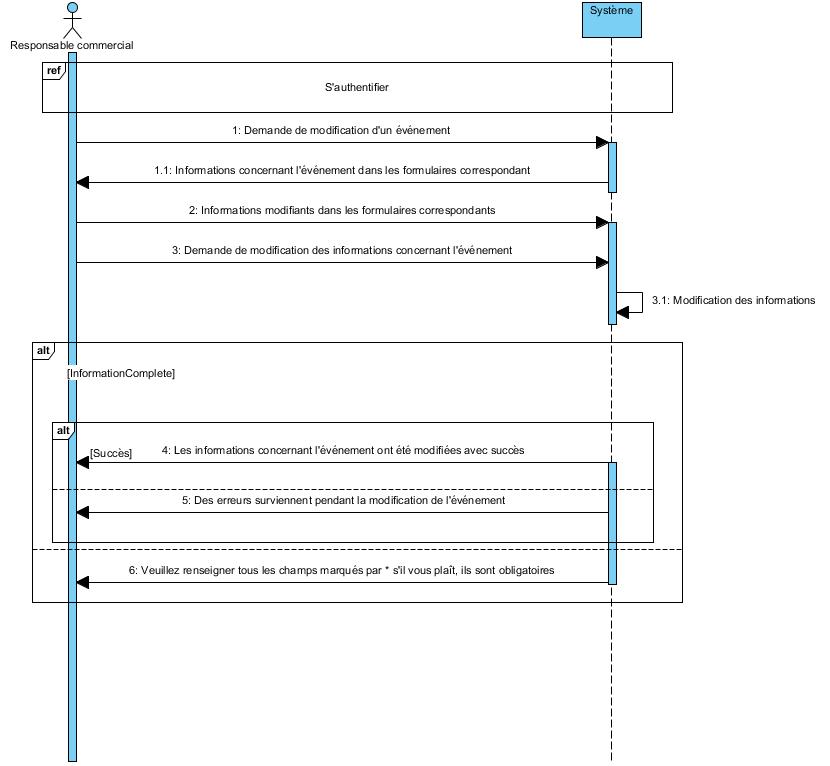


Figure 28. Diagramme de séquence système du cas d'utilisation « Modifier événement ».

La figure 23 illustre le diagramme de séquence système du cas d’utilisation « Supprimer événement ».

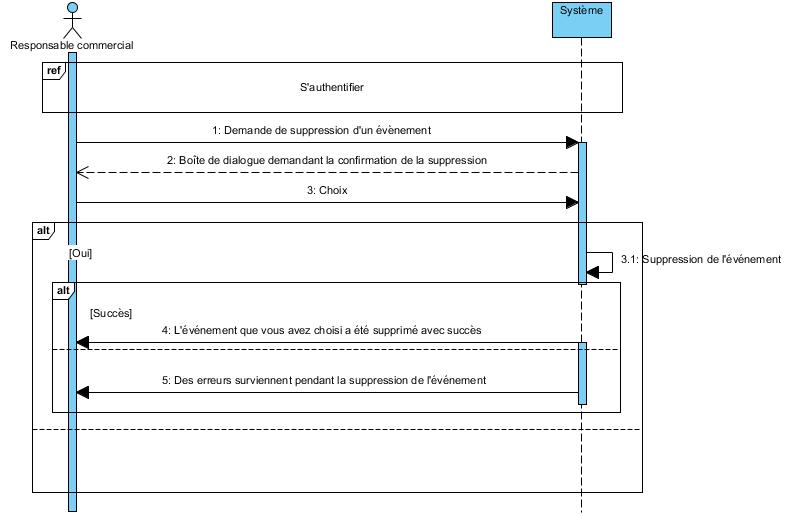


Figure 29. Diagramme de séquence système du cas d'utilisation « Supprimer événement ».

La figure 24 illustre le diagramme de séquence système du cas d’utilisation « Voir force de vente ».

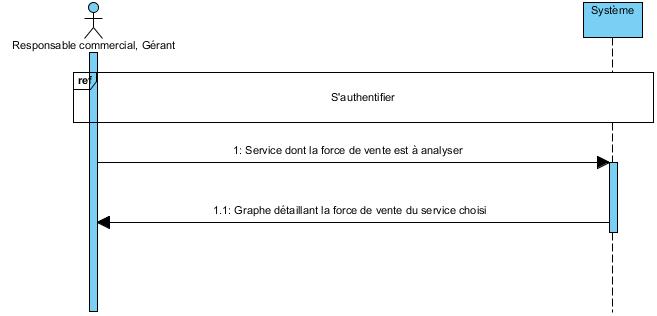


Figure 30. Diagramme de séquence système du cas d'utilisation « Voir force de vente ».

La figure 25 illustre le diagramme de séquence système du cas d’utilisation « Voir historique des actions avec les clients ».

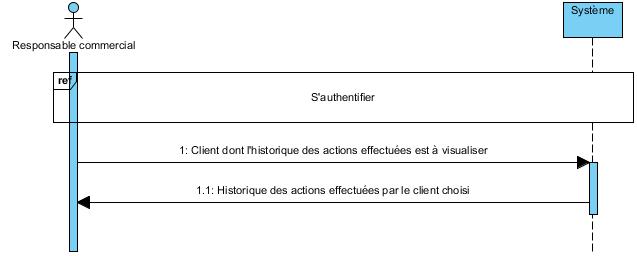


Figure 31. Diagramme de séquence système du cas d'utilisation «Voir historique des actions avec les clients».

La figure 26 illustre le diagramme de séquence système du cas d’utilisation « Se dialoguer entre personnel ».

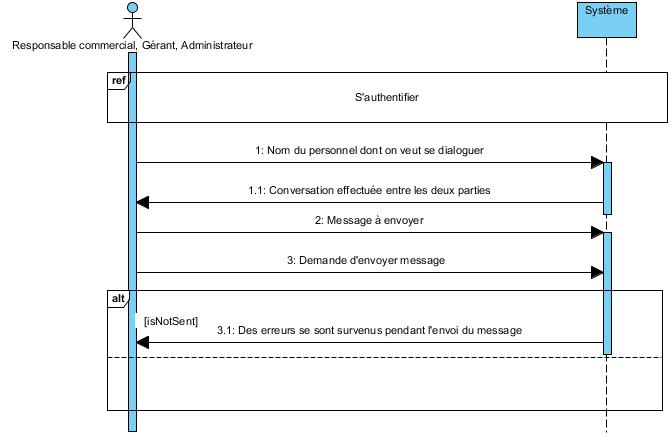


Figure 32. Diagramme de séquence système du cas d'utilisation « Se dialoguer entre personnel ».

La figure 27 illustre le diagramme de séquence système du cas d’utilisation « Contacter client ou prospect ».

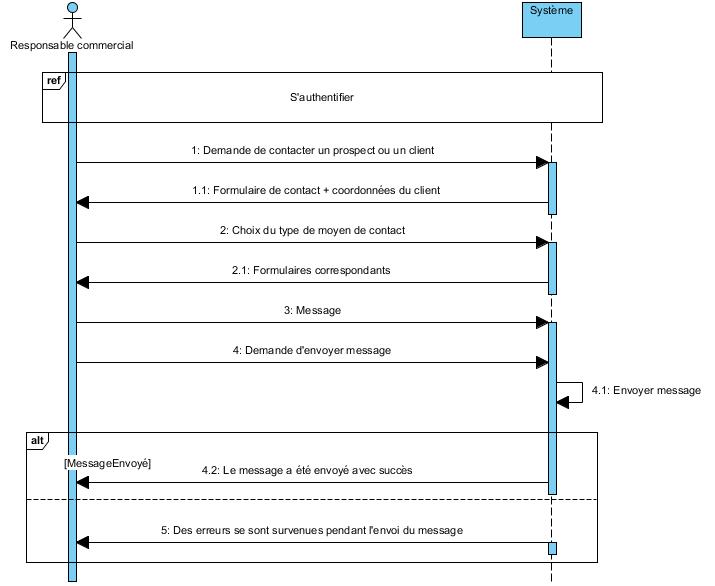


Figure 33. Diagramme de séquence système du cas d'utilisation « Contacter prospect ou client ».

La figure 28 illustre le diagramme de séquence système du cas d’utilisation « Voir statistiques ».

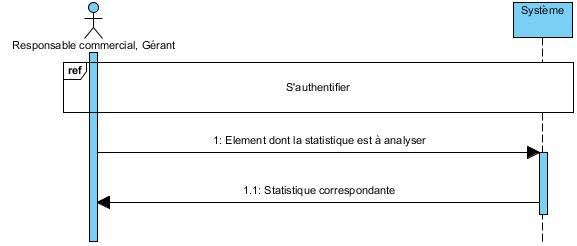


Figure 34. Diagramme de séquence système du cas d'utilisation « Voir statistiques ».

La figure 29 illustre le diagramme de séquence système du cas d’utilisation « Voir performance des commerciaux ».

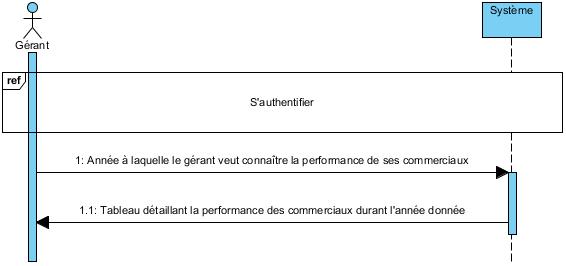


Figure 35. Diagramme de séquence système du cas d'utilisation « Voir performance des commerciaux ».

La figure 30 illustre le diagramme de séquence système du cas d’utilisation « Réinitialiser mot de passe ».

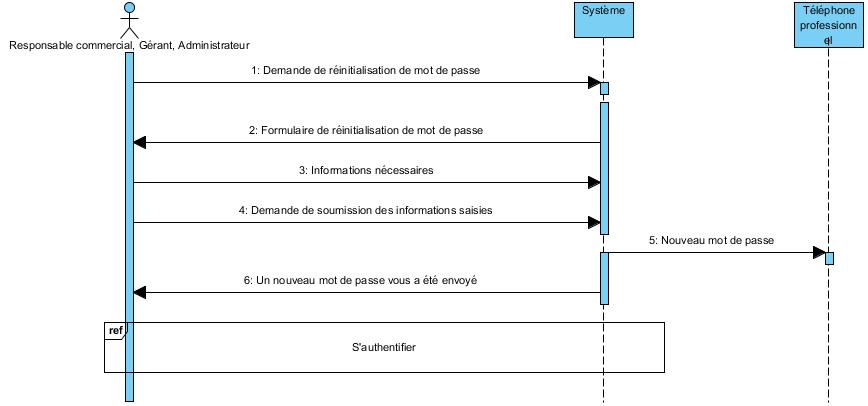


Figure 36. Diagramme de séquence système du cas d'utilisation « Réinitialiser mot de passe ».

#### Modèle de domaine

Un modèle de domaine est une représentation des différentes entités qui compose le présent système d’information. Ce modèle est aussi à la fois la représentation schématique de la future base de données qui sera utilisée pour mettre en place l’application et la première version qui inspire la création de diagramme de classes.

La figure 31 représente le modèle de domaine du système.

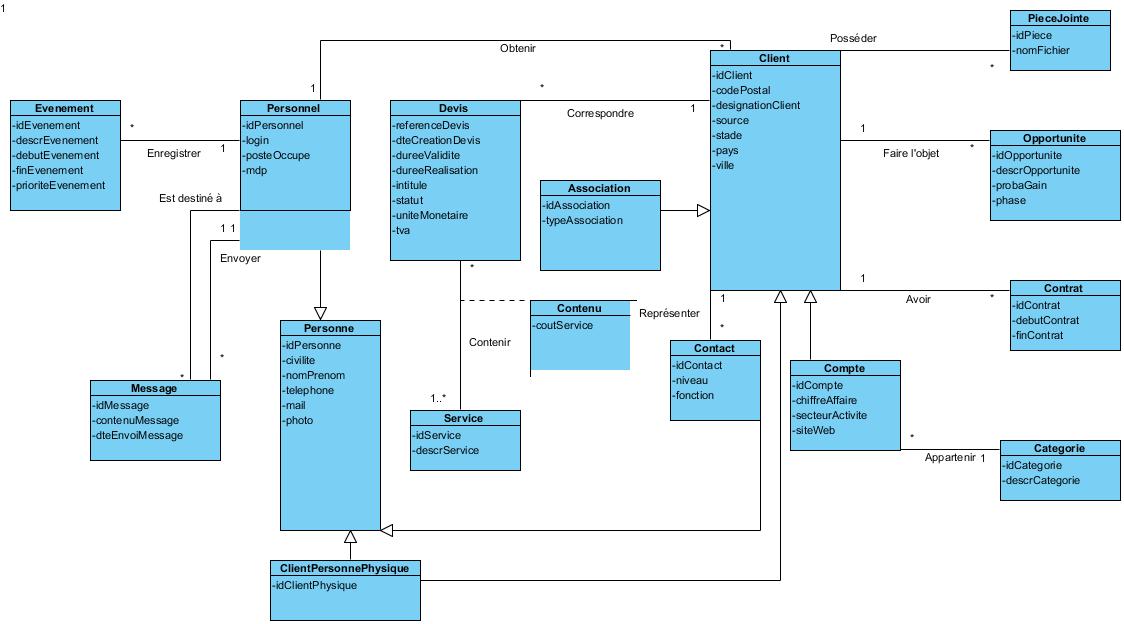


Figure 37. Modèle de domaine du projet.

## Conception détaillée

### Architecture de l’application

Au niveau conceptuel, l’architecture d’une application est généralement définie selon le cadre d’application choisi. Ces cadres d’application possèdent souvent leurs propres patterns qui font leurs architectures. Dans notre cas, puisque nous avons opté CodeIgniter comme framework pour mettre en place l’application, donc l’architecture est le MVC ou Modèle-Vue-Contrôleur.

[45.a] Le modèle MVC décrit une manière d’architecturer une application informatique en la décomposant en trois sous-parties :

* La partie Modèle ;
* La partie Vue ;
* La partie Contrôleur.

Ce modèle de conception ou design pattern a été imaginé à la fin des années 1970 pour le langage Smalltalk afin de bien séparer le code de l’interface graphique de la logique applicative. Il est utilisé dans de très nombreux langages : bibliothèques Swing, JSP, frameworks PHP, ASP.Net MVC, etc. La partie Modèle d’une architecture MVC encapsule la logique métier ou business logic ainsi que l’accès aux données. Il peut s’agir d’un ensemble de fonctions (modèle procédural) ou de classes (modèle orienté objet). La partir Vue s’occupe des interactions avec l’utilisateur : présentation, saisie et validation des données. La partie Contrôleur gère la dynamique de l’application. Elle fait le lien entre l’utilisateur et le reste de l’application.

La figure 32 illustre les relations entre les composants d’une architecture MVC.



Figure 38. Architecture MVC. [46.a]

La demande de l’utilisateur (exemple : requête http) est reçue et interprétée par le contrôleur. Celui-ci utilise les services du modèle afin de préparer les données à afficher. Ensuite, le contrôleur fournit ces données à la vue, qui les présente à l’utilisateur (par exemple sous la forme d’une page HTML). Dans le cas d’une application web, l’utilisation du modèle MVC permet aux pages HTML (qui constituent la partie Vue) de contenir le moins possible de code serveur, étant donné que le Scripting est regroupé dans les deux autres parties de l’application.

### Diagramme de séquence de conception de chaque cas d’utilisation

Un diagramme de séquence de conception est un diagramme de séquence beaucoup plus détaillé qu’un diagramme de séquence système tant en termes d’acteur intervenant au système qu’aux fréquences des messages échangés.

La figure 33 représente le diagramme de séquence de conception du cas d’utilisation &« S’authentifier ».

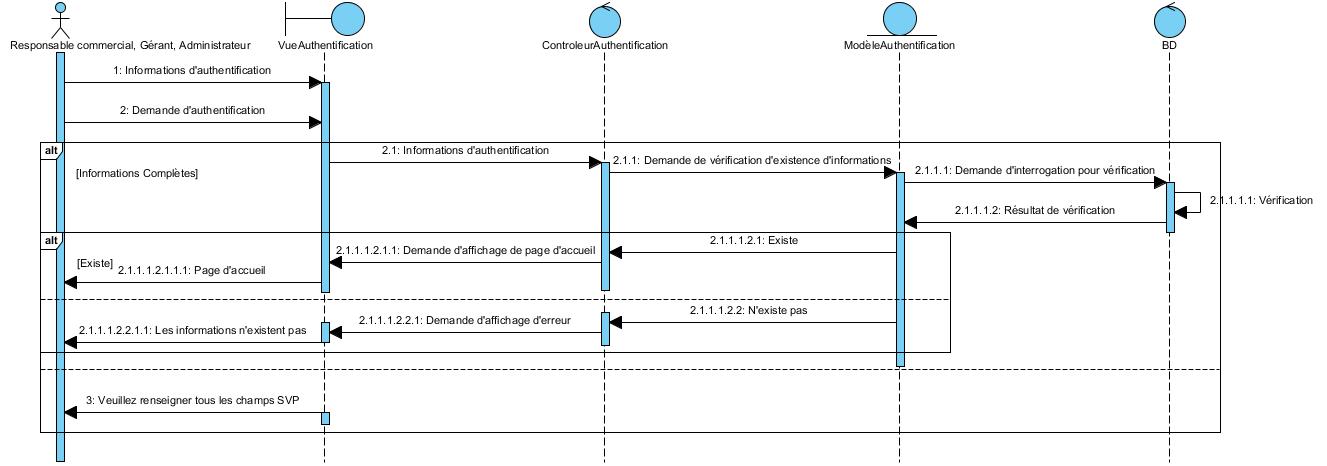


Figure 39. Diagramme de séquence de conception « S'authentifier ».

La figure 34 représente le diagramme de séquence de conception du cas d’utilisation « Ajouter client ou prospect ».

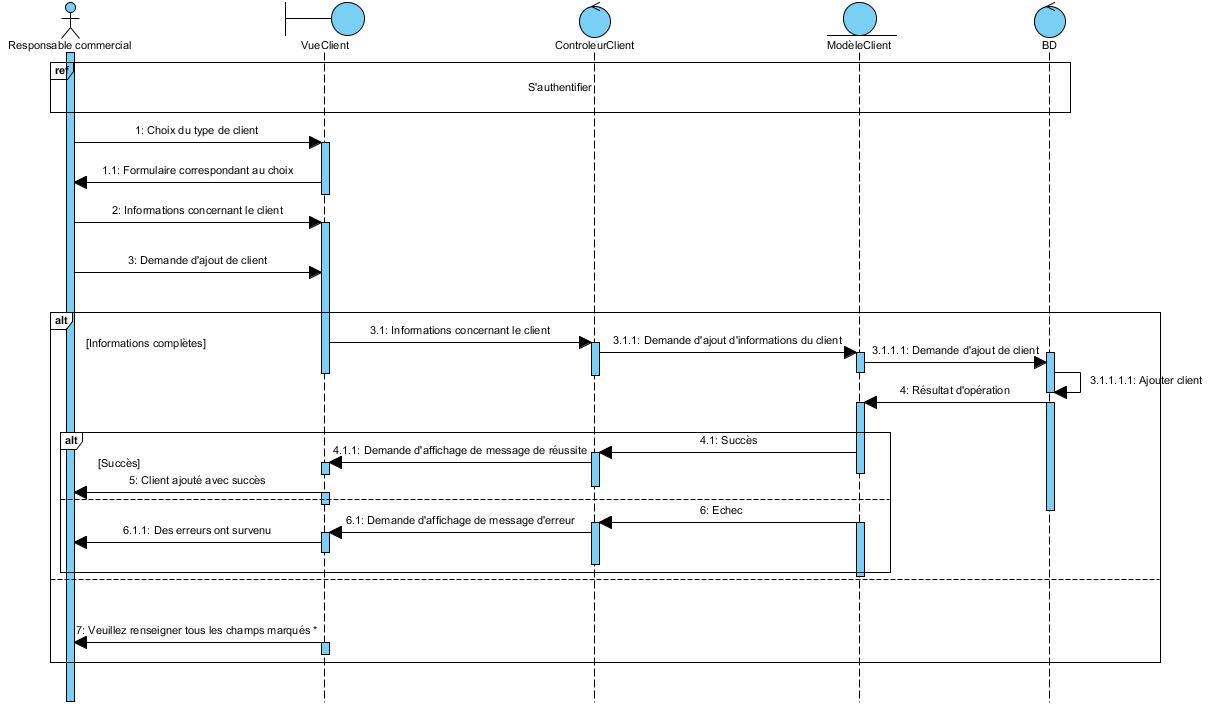


Figure 40. Diagramme de séquence de conception « Ajouter client ou prospect ».

La figure 35 représente le diagramme de séquence de conception du cas d’utilisation « Modifier client ou prospect ».

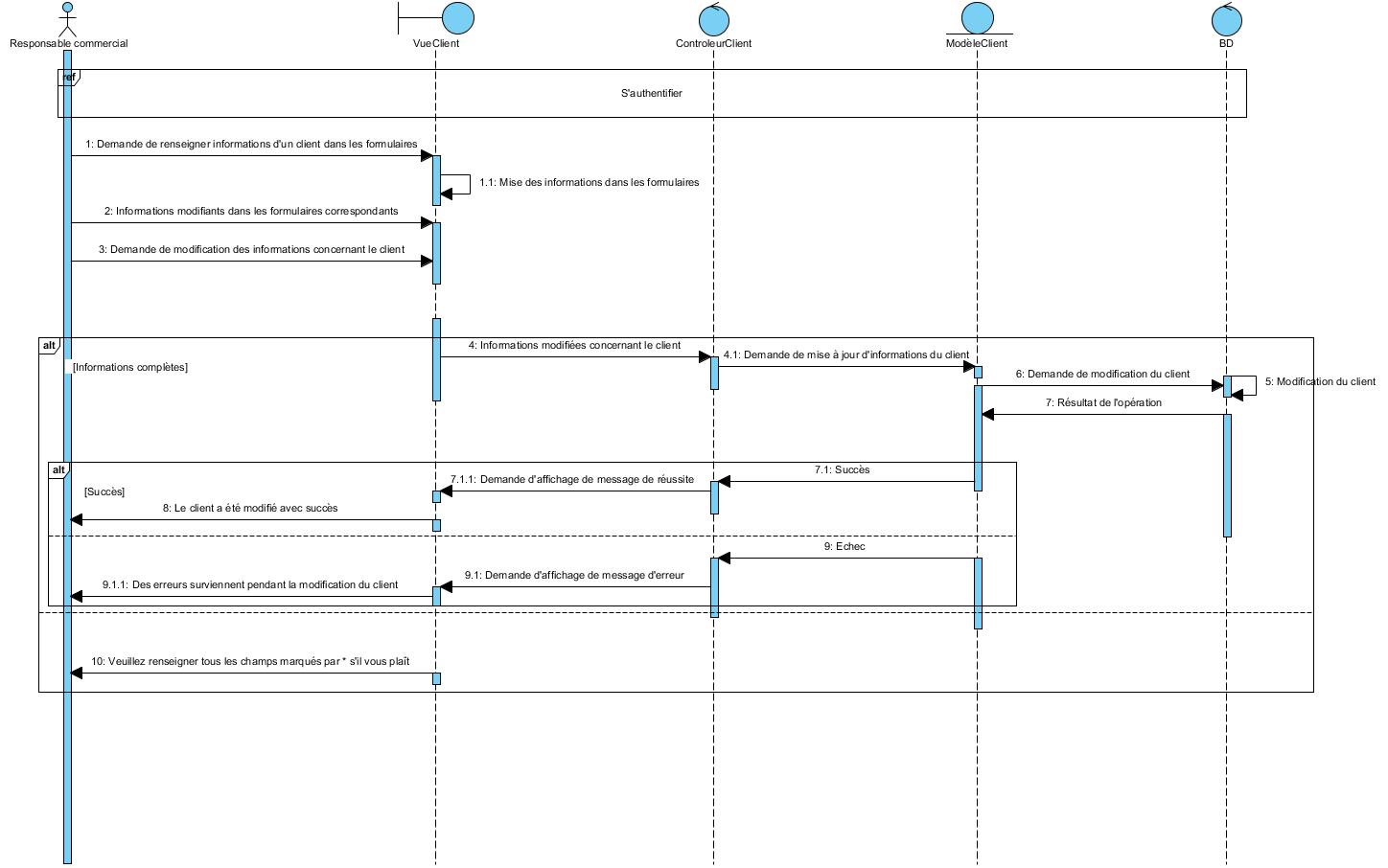


Figure 41. Diagramme de séquence de conception « Modifier client ou prospect ».

La figure 36 représente le diagramme de séquence de conception du cas d’utilisation « Supprimer client ou prospect ».

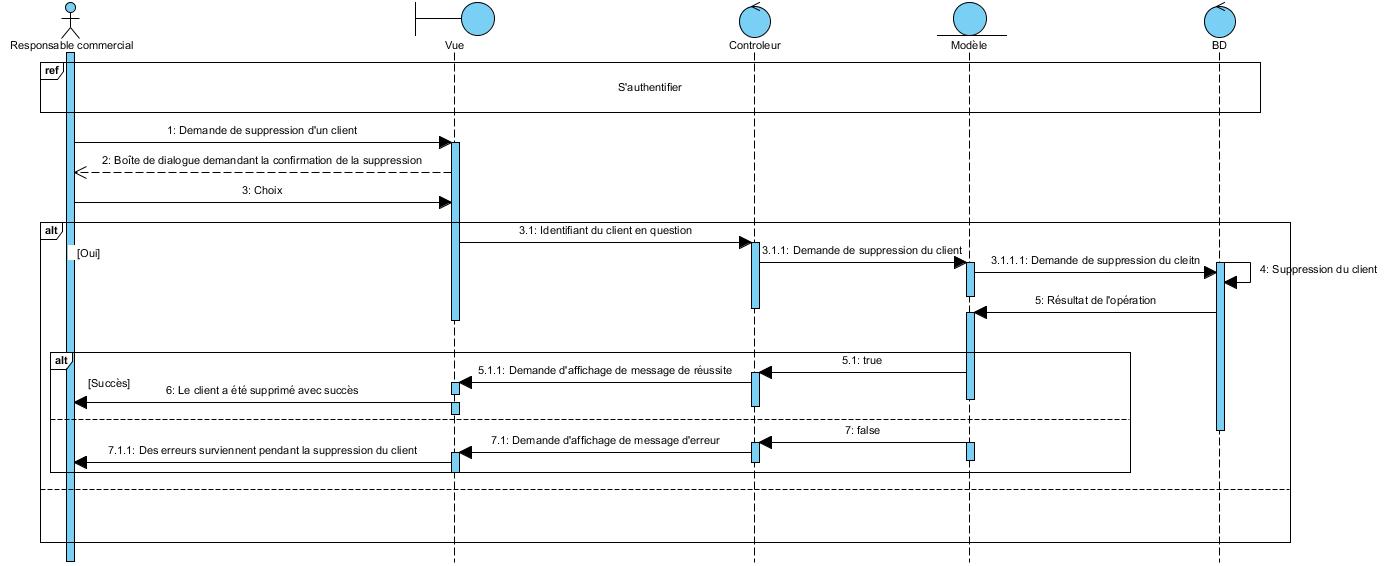


Figure 42. Diagramme de séquence de conception « Supprimer client ou prospect ».

La figure 37 représente le diagramme de séquence de conception du cas d’utilisation « Contacter client ou prospect ».

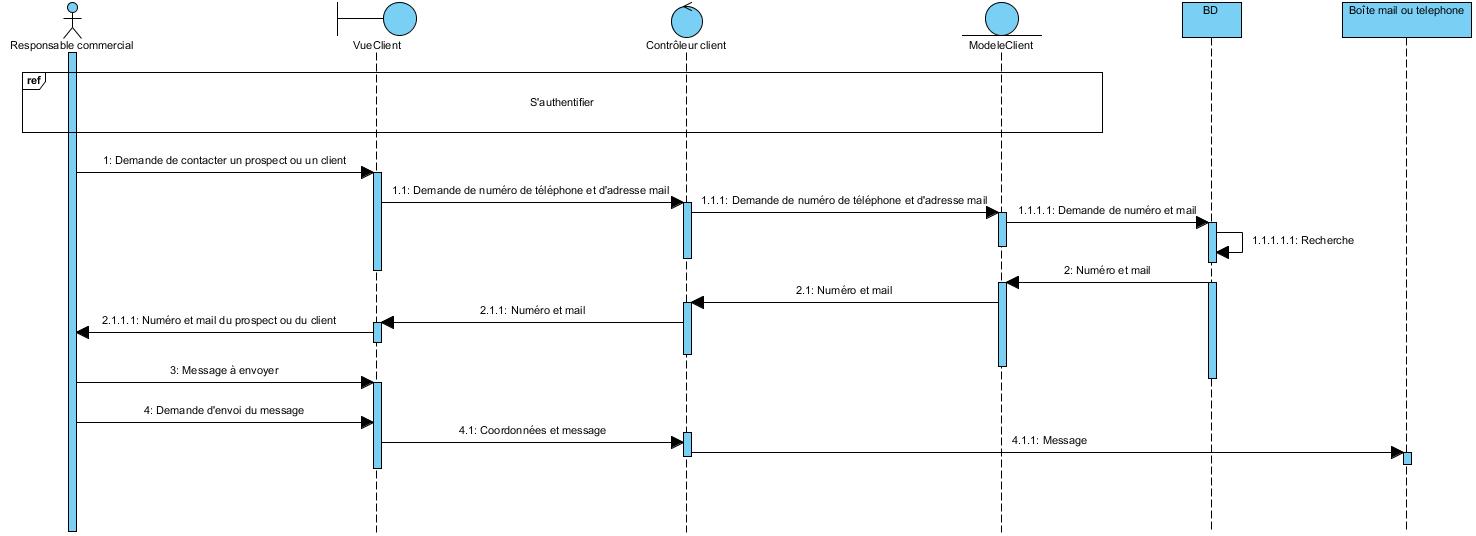


Figure 43. Diagramme de séquence de conception « Contacter client ou prospect ».

La figure 38 représente le diagramme de séquence de conception du cas d’utilisation « Ajouter événement ».

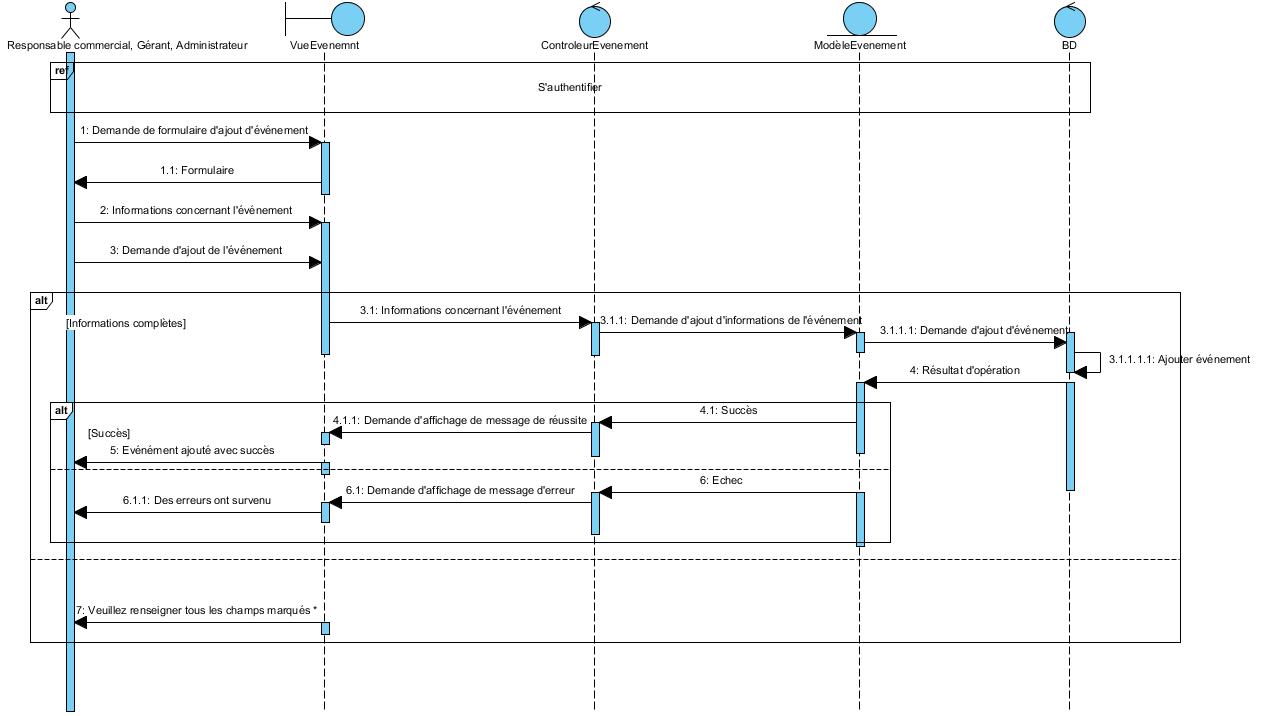


Figure 44. Diagramme de séquence de conception « Ajouter événement ».

La figure 39 représente le diagramme de séquence de conception du cas d’utilisation « Modifier événement ».

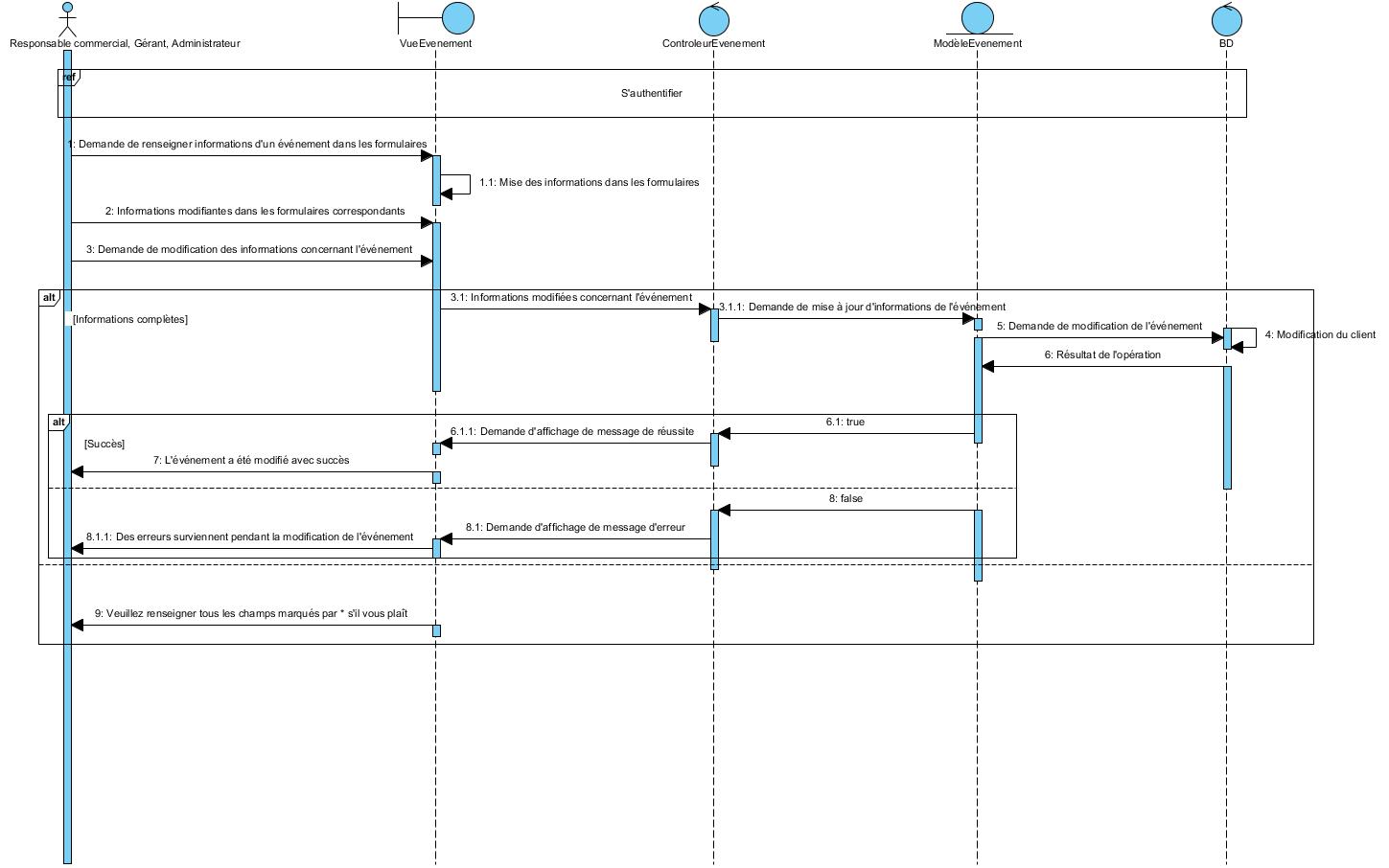


Figure 45. Diagramme de séquence de conception « Modifier événement ».

La figure 40 représente le diagramme de séquence de conception du cas d’utilisation « Annuler événement ».

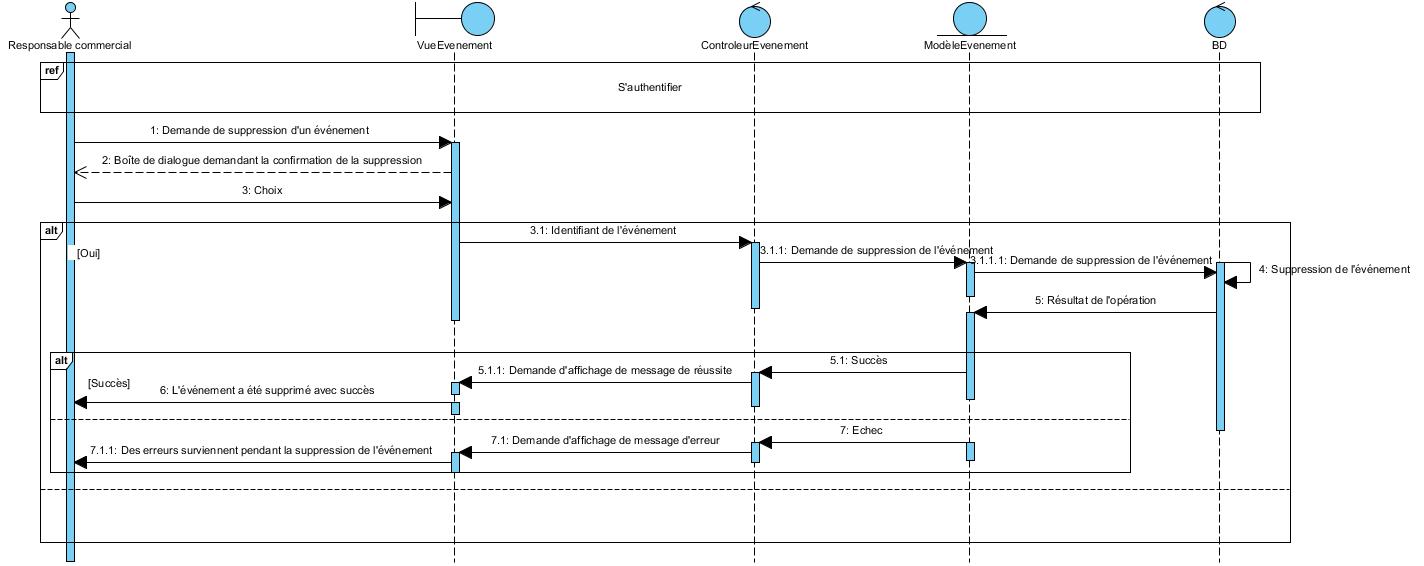


Figure 46.. Diagramme de séquence de conception « Annuler événement ».

La figure 41 représente le diagramme de séquence de conception du cas d’utilisation « Envoyer document commercial ».

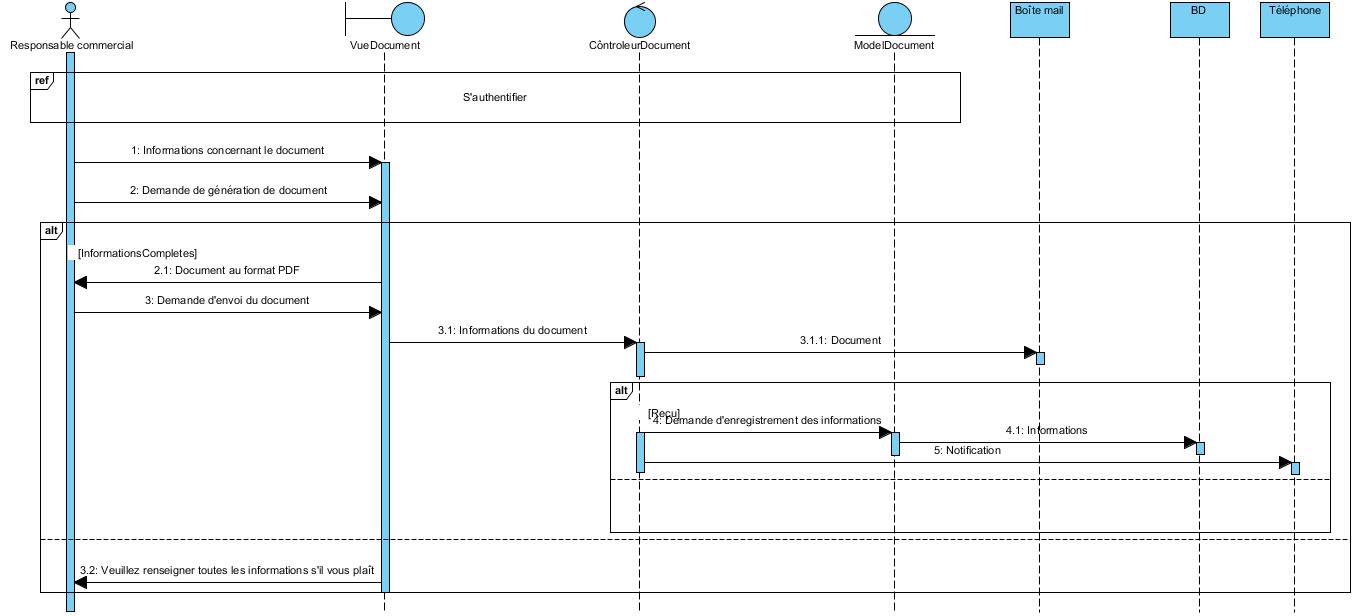


Figure 47. Diagramme de séquence de conception « Envoyer document commercial ».

La figure 42 représente le diagramme de séquence de conception du cas d’utilisation « Voir force de vente ».



Figure 48. Diagramme de séquence de conception « Voir force de vente ».

La figure 43 représente le diagramme de séquence de conception du cas d’utilisation « Voir historique des actions avec les clients ».

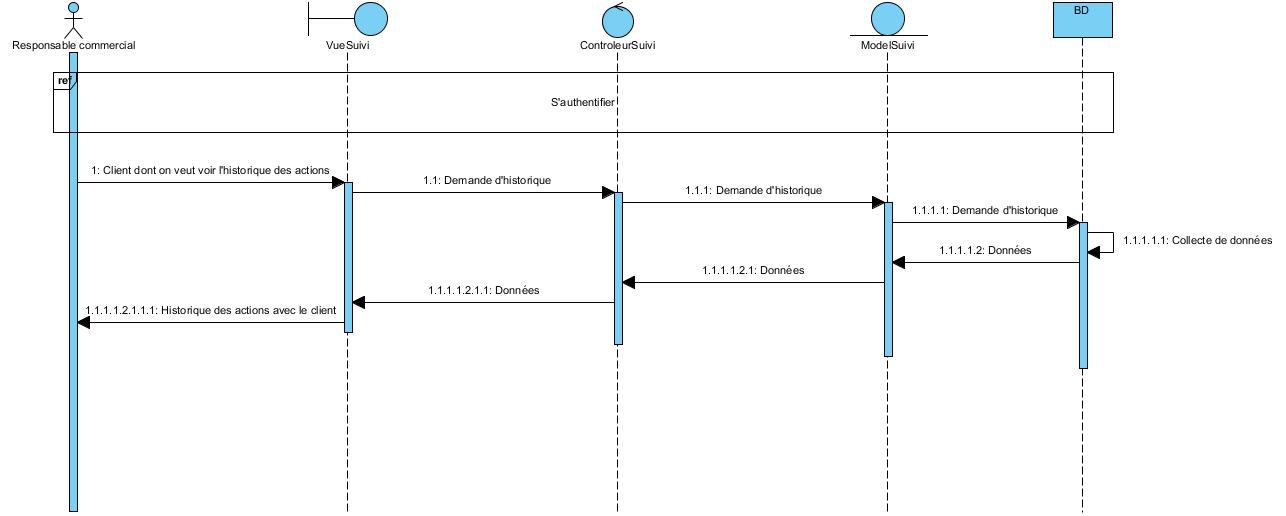


Figure 49. Diagramme de séquence de conception « Voir historique des actions avec les clients ».

La figure 44 représente le diagramme de séquence de conception du cas d’utilisation « Voir statistiques ».

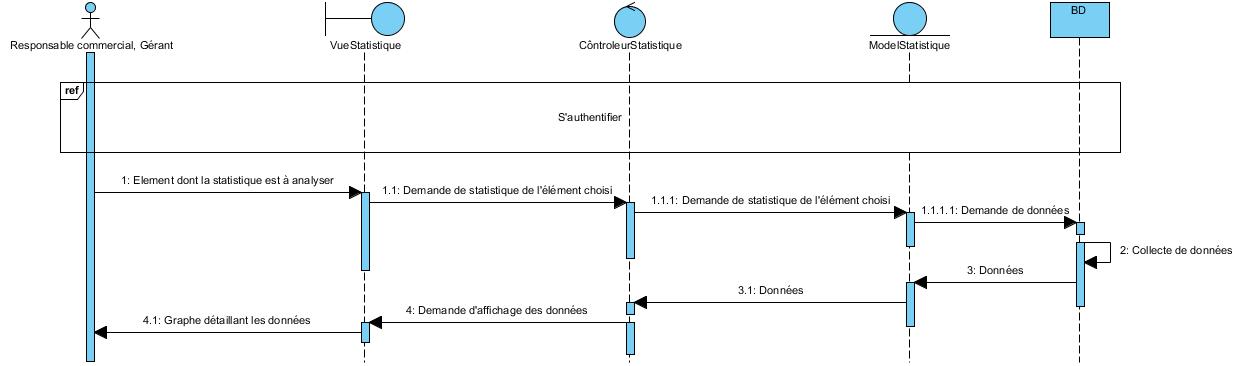


Figure 50. Diagramme de séquence de conception « Voir statistiques ».

La figure 45 représente le diagramme de séquence de conception du cas d’utilisation « Se dialoguer entre personnel ».

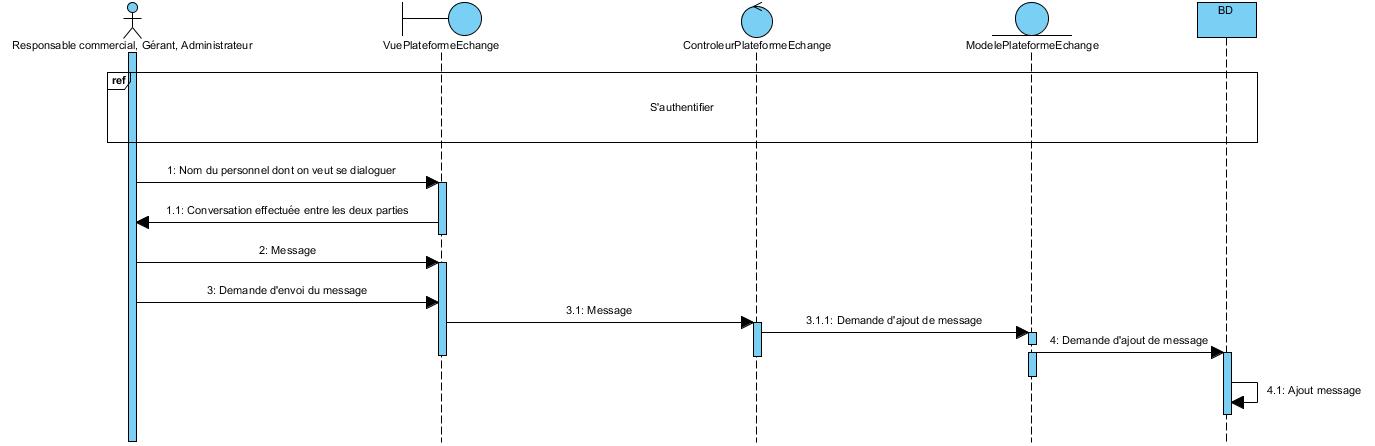


Figure 51. Diagramme de séquence de conception « Se dialoguer entre personnel ».

La figure 46 représente le diagramme de séquence de conception du cas d’utilisation « Voir performance des commerciaux ».

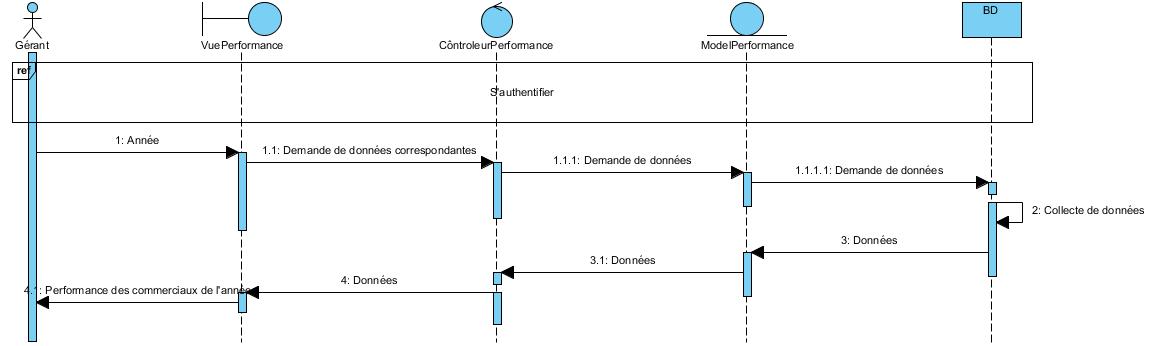


Figure 52. Diagramme de séquence de conception « Voir performance des commerciaux ».

La figure 47 représente le diagramme de séquence de conception du cas d’utilisation « Ajouter contact ».



Figure 53. Diagramme de séquence de conception « Ajouter contact ».

La figure 48 représente le diagramme de séquence de conception du cas d’utilisation « Modifier contact ».

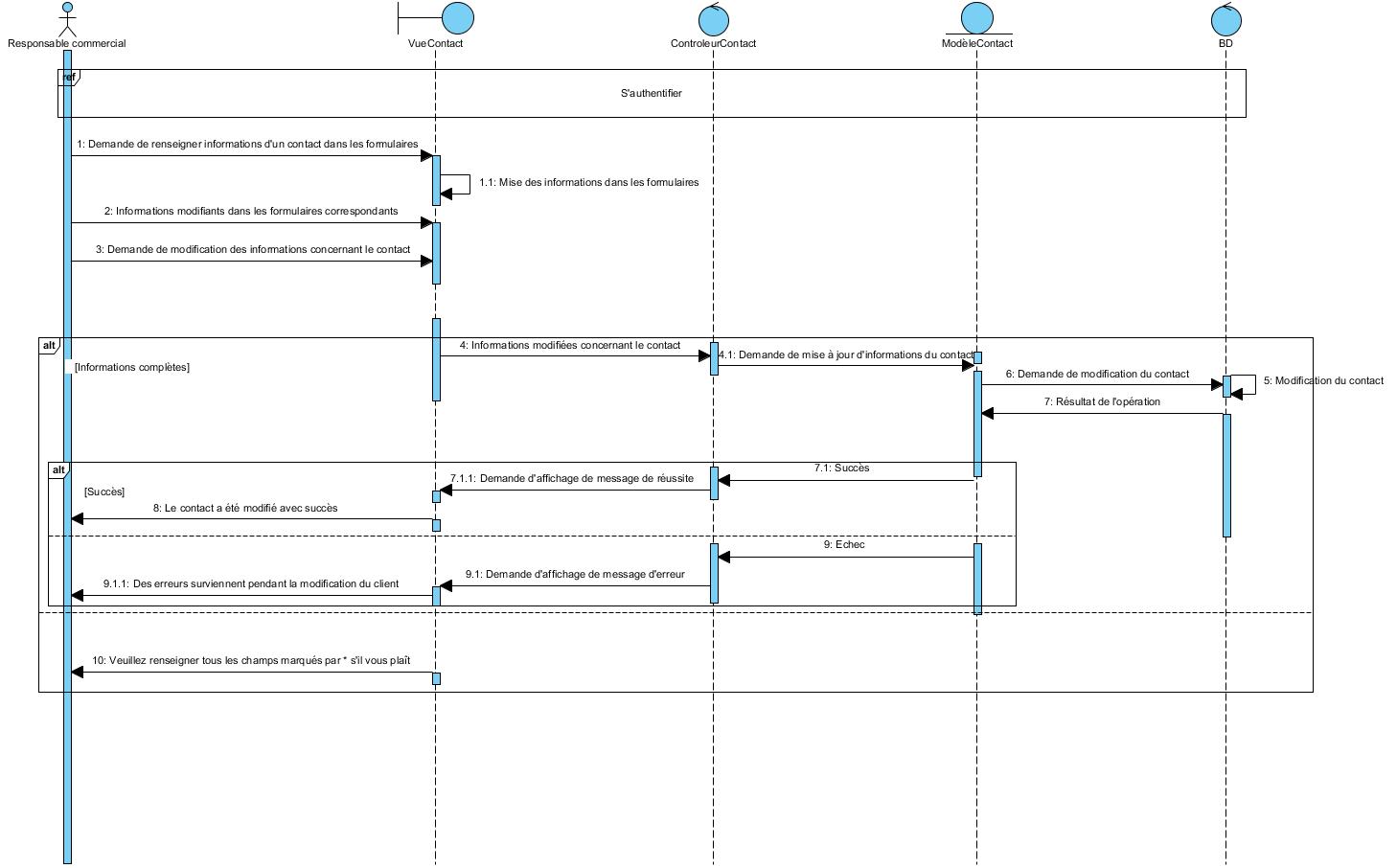


Figure 54. Diagramme de séquence de conception « Modifier contact ».

La figure 49 représente le diagramme de séquence de conception du cas d’utilisation « Supprimer contact ».

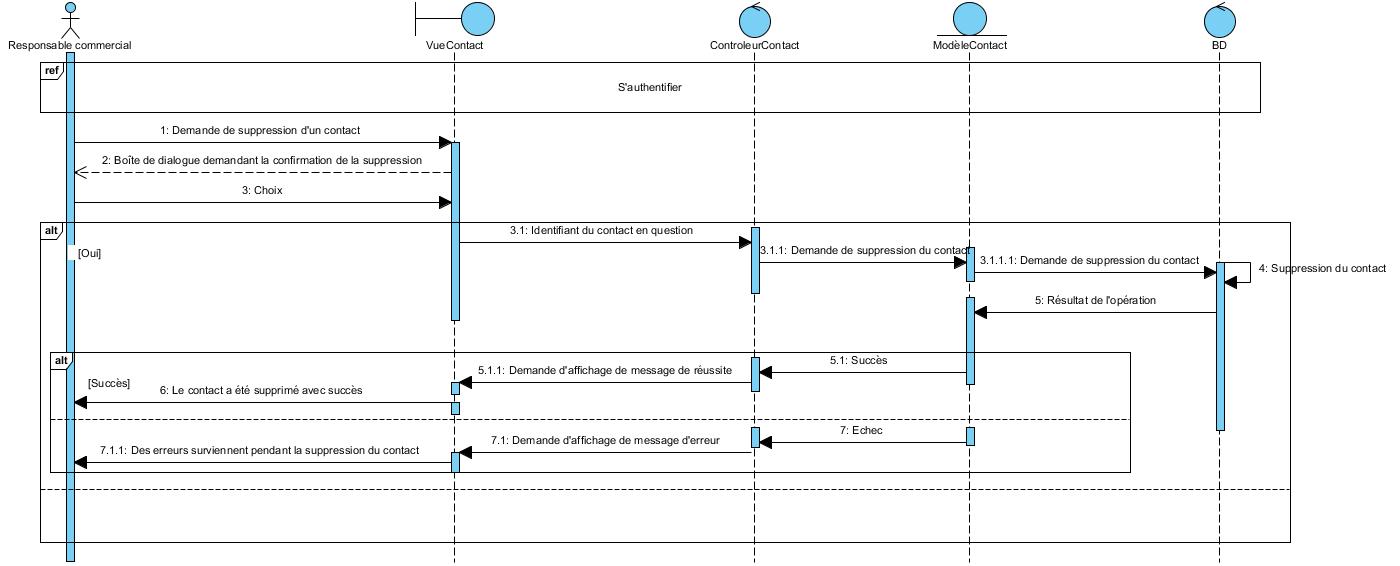


Figure 55. Diagramme de séquence de conception « Supprimer contact ».

La figure 50 représente le diagramme de séquence de conception du cas d’utilisation « Ajouter personnel ».

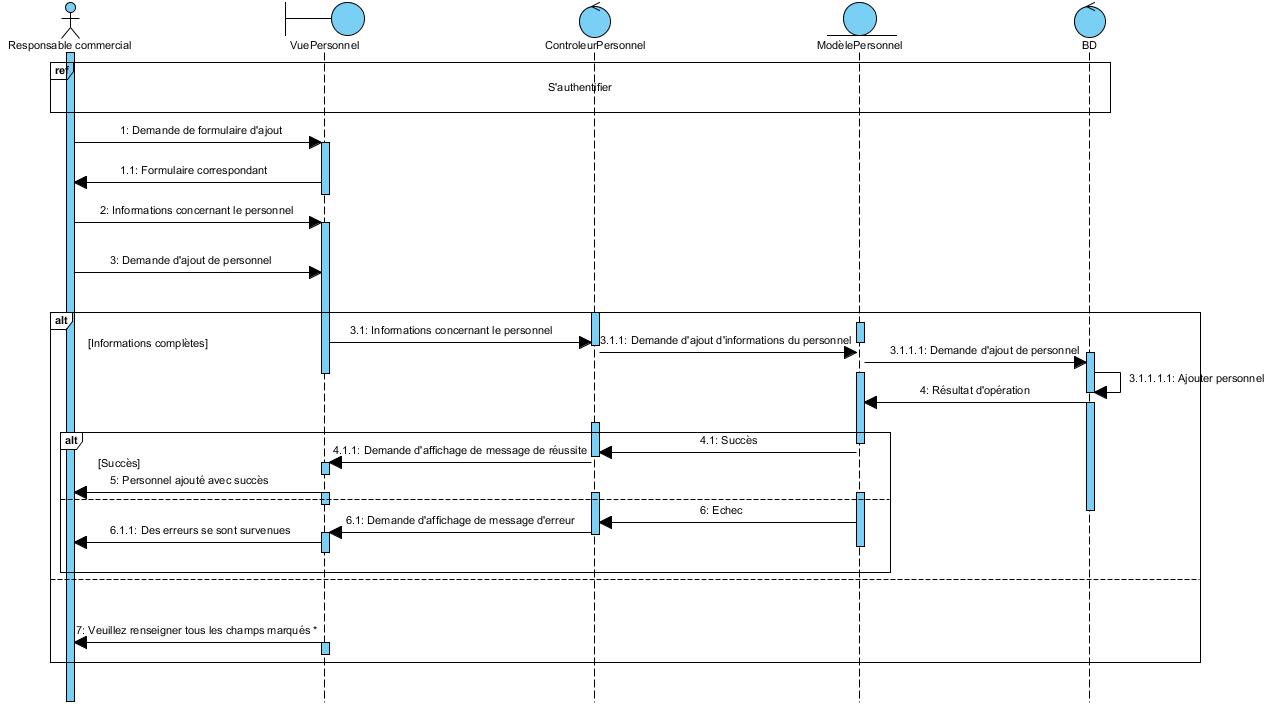


Figure 56. Diagramme de séquence de conception « Ajouter personnel ».

La figure 51 représente le diagramme de séquence de conception du cas d’utilisation « Modifier personnel ».

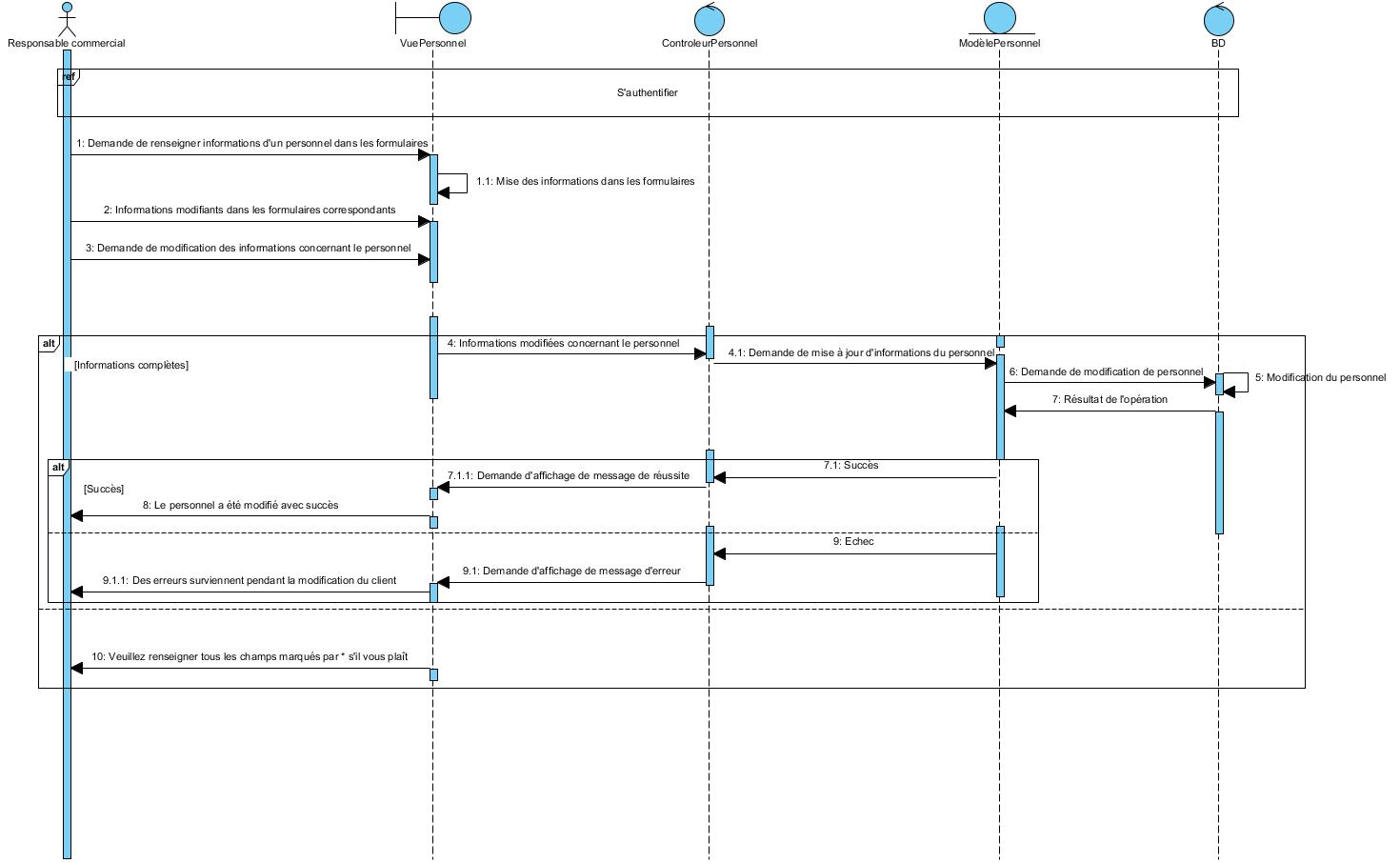


Figure 57. Diagramme de séquence de conception « Modifier personnel ».

La figure 52 représente le diagramme de séquence de conception du cas d’utilisation « Supprimer personnel ».

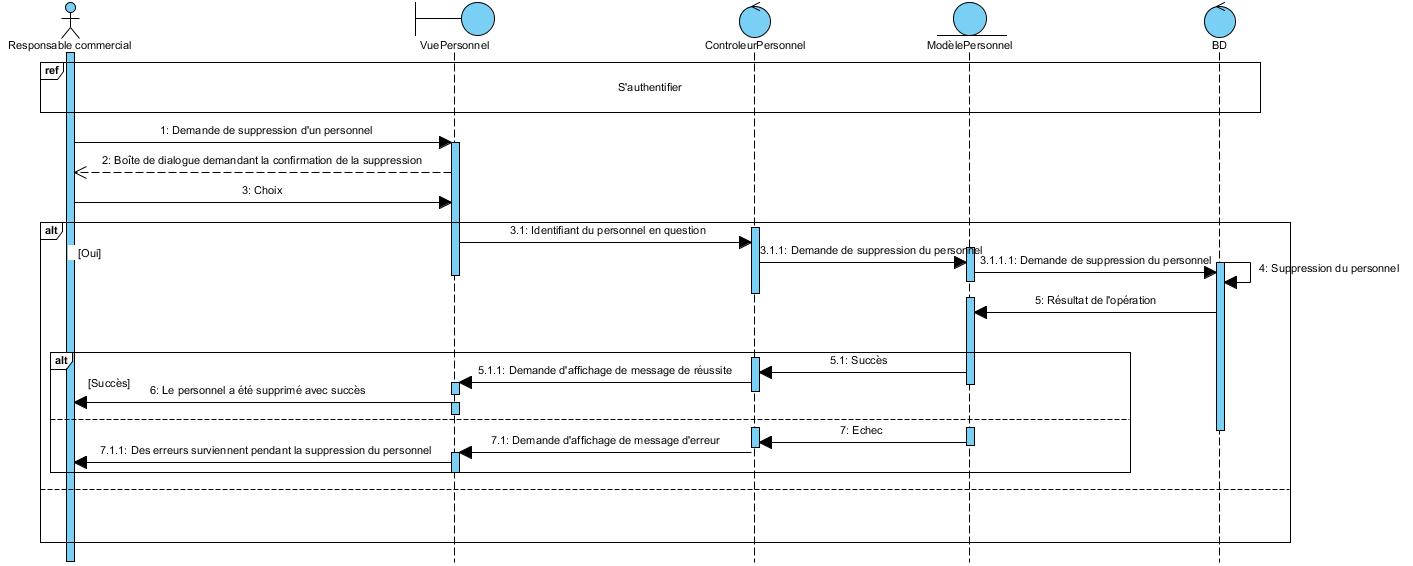


Figure 58. Diagramme de séquence de conception « Supprimer personnel ».

La figure 53 représente le diagramme de séquence de conception du cas d’utilisation « Se déconnecter ».

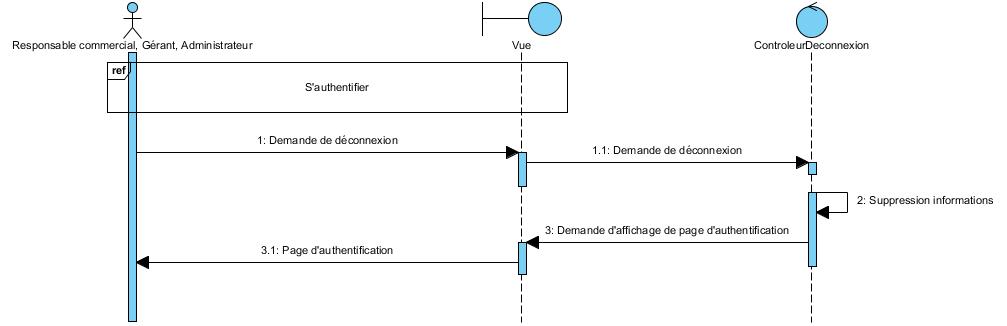


Figure 59. Diagramme de séquence de conception « Se déconnecter ».

La figure 54 représente le diagramme de séquence de conception du cas d’utilisation « Réinitialiser mot de passe ».

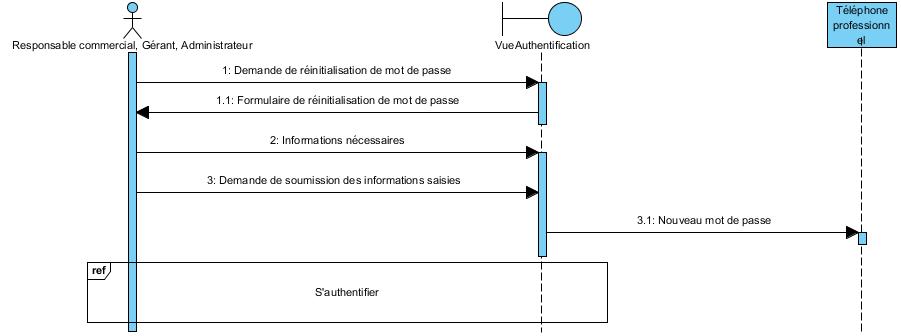


Figure 60. Diagramme de séquence de conception « Réinitialiser mot de passe ».

### Diagramme des classes de conception de chaque cas d’utilisation.

Comme les précédents diagrammes de séquence de conception, un diagramme de classe de conception pour un cas d’utilisation est la représentation de l’ensemble des classes candidates à ce cas d’utilisation tout en prenant en considération les éventuelles associations existantes.

La figure 55 représente le diagramme des classes de conception de « « Gérer prospect ou client ».

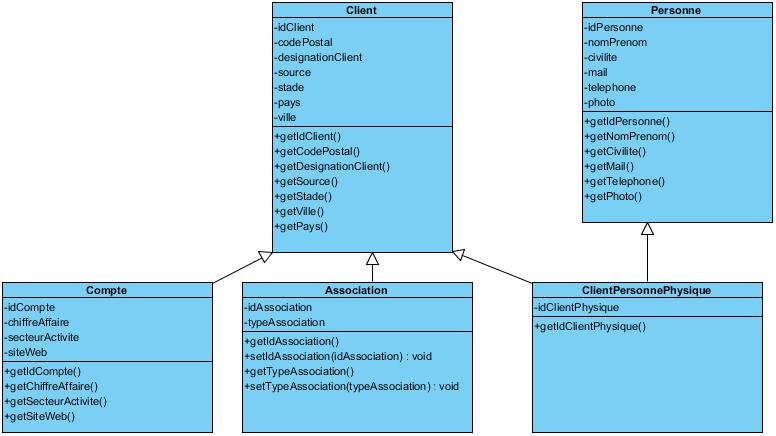


Figure 61. Diagramme des classes de conception « Gérer prospect ou client ».

La figure 56 représente le diagramme des classes de conception de « « Gérer événement ».

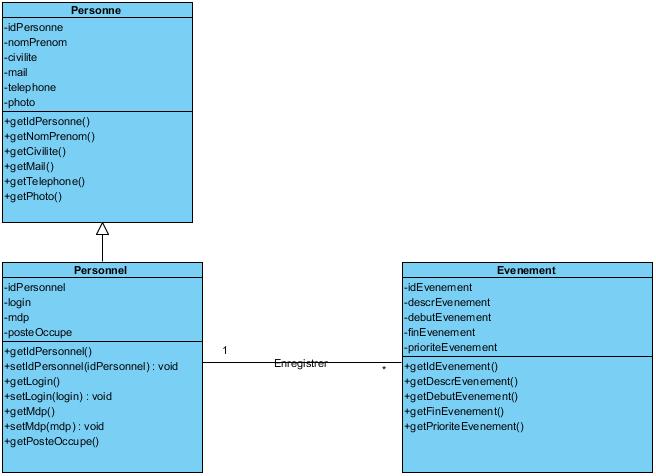


Figure 62. Diagramme des classes de conception « Gérer événement ».

La figure 57 représente le diagramme des classes de conception de « « Envoyer document commercial ».

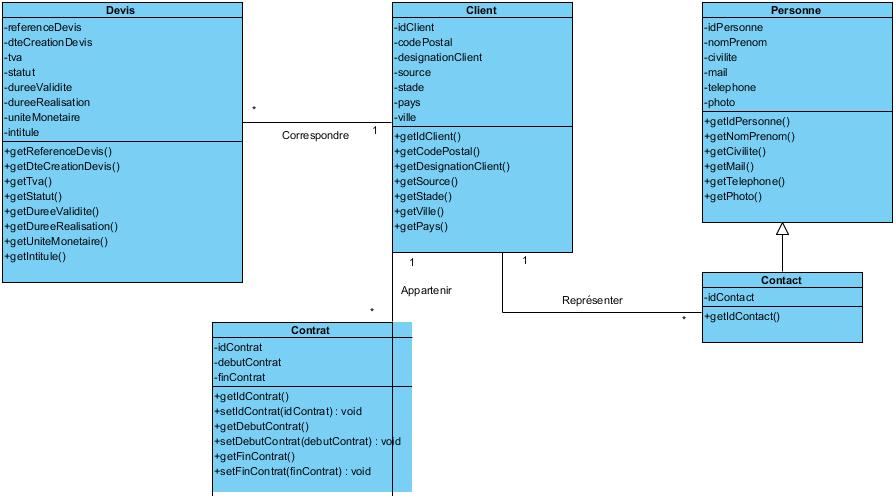


Figure 63. Diagramme des classes de conception « Envoyer document commercial ».

La figure 58 représente le diagramme des classes de conception de « « Voir force de vente, Voir statistiques, Voir performance des commerciaux ».

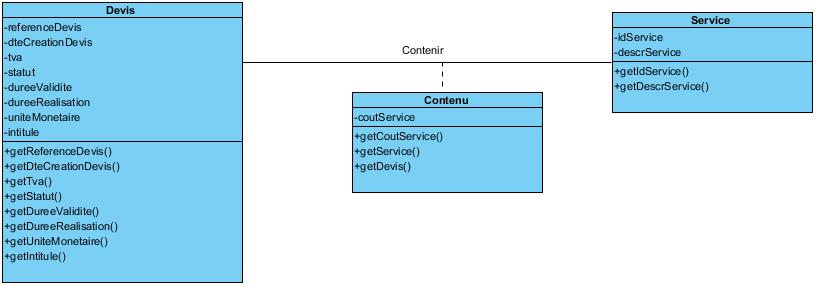


Figure 64. Diagramme des classes de conception « Voir force de vente, Voir statistiques, Voir performance des commerciaux ».

La figure 59 représente le diagramme des classes de conception de « « Gérer personnel, Réinitialiser mot de passe, S’authentifier, Se déconnecter ».

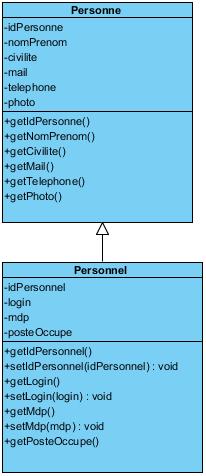


Figure 65. Diagramme des classes de conception « Gérer personnel, Réinitialiser mot de passe, S'authentifier, Se déconnecter ».

### Diagramme des classes global

Le diagramme des classes global est un diagramme de classe issu de l’assemblage des précédents diagrammes de classe de chaque cas d’utilisation.

La figure 60 représente le diagramme des classes global du projet.

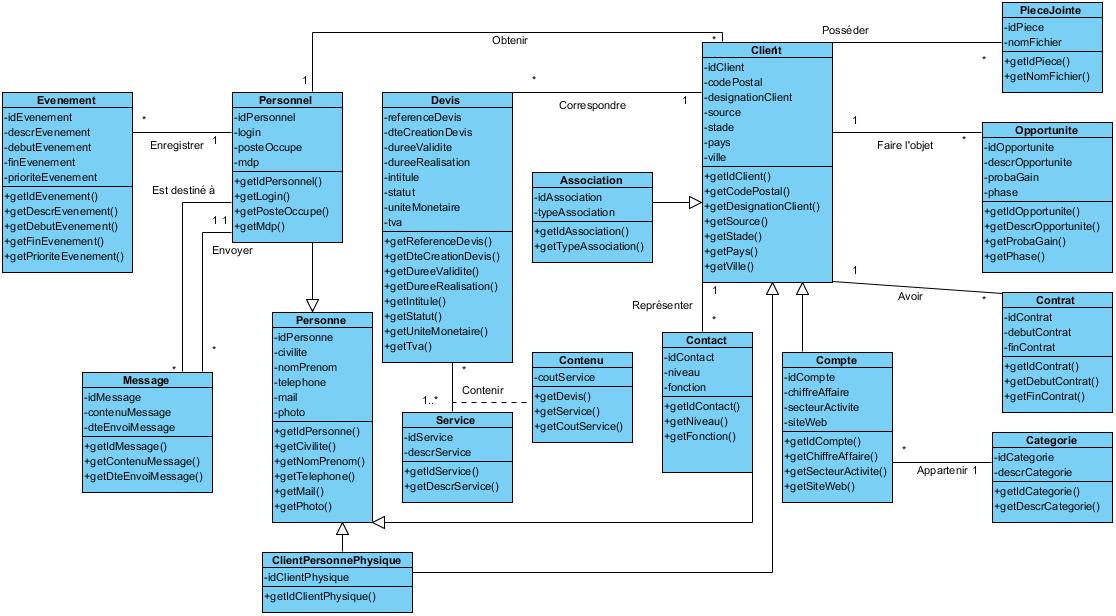


Figure 66. Diagramme des classes global.

### Diagramme des paquetages

Un paquetage est un regroupement des différents éléments d’un système. Ces éléments peuvent être des classes, des diagrammes, des fonctions ou bien même des interfaces. Ce regroupement permet de clarifier le modèle en l’organisant.

[47.a] Un diagramme de paquetages montre la disposition et l’organisation des éléments de modèle dans les projets de moyenne à grande échelle, pouvant être utilisés pour montrer à la fois la structure et les dépendances entre sous-systèmes ou modules.

La figure 61 représente le diagramme des paquetages du projet.

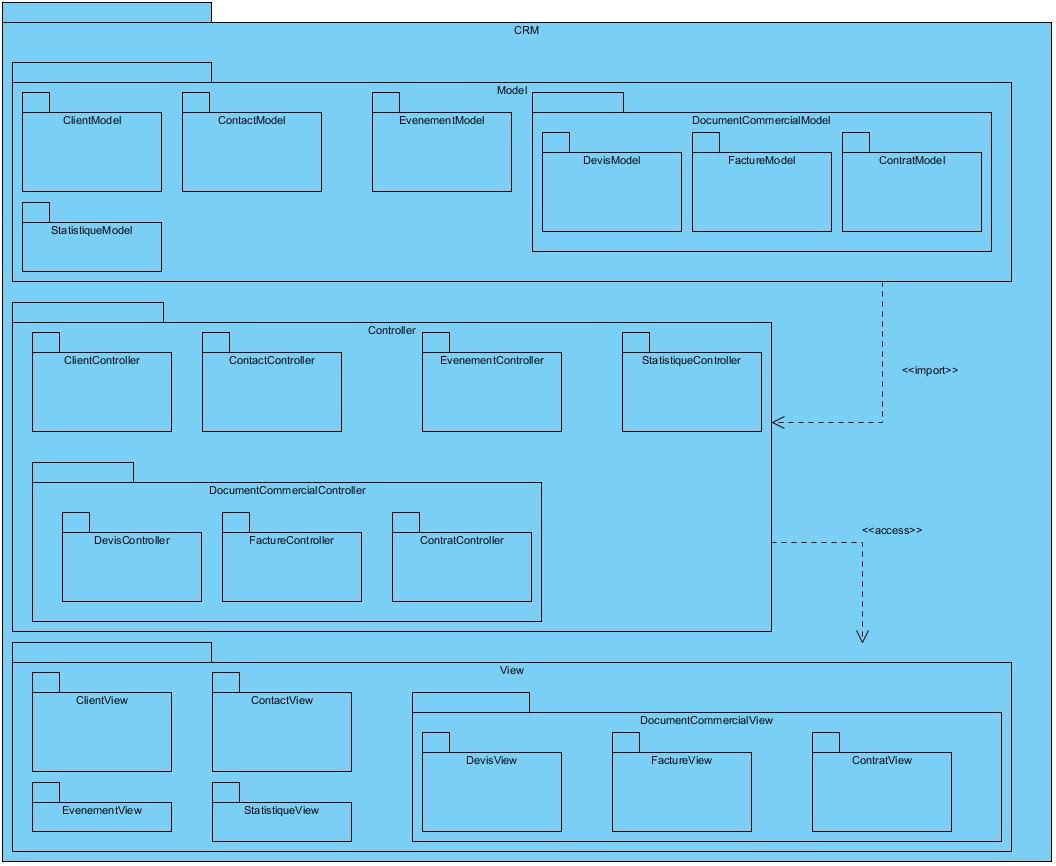


Figure 67. Diagramme des paquetages.

### Diagramme de déploiement

Un diagramme de déploiement est un diagramme permettant de visualiser l’aspect statique de ces nœuds physiques et de leurs relations, ainsi que de spécifier leurs détails pour la construction.

[48.a] Les diagrammes de déploiement sont l’un des deux types de diagramme utilisés pour modéliser les aspects physiques d’un système orienté objet. Un diagramme de déploiement présente la configuration des nœuds de traitement d’exécution et des composant qui y résident. Les diagrammes de déploiement traitent de la vue de déploiement statique d’une architecture. Ils sont liés aux diagrammes de composants, en ce sens qu’un nœud comprend généralement un ou plusieurs composants.

La figure 62 représente le diagramme de déploiement du projet.

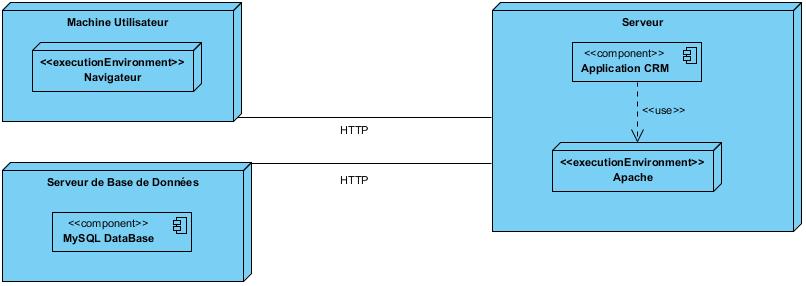


Figure 68. Diagramme de déploiement.

# PARTIE III : REALISATION

## Mise en place de l’environnement de développement

Ce chapitre se consacre à la mise en œuvre de la solution. On y présente les installations et configurations effectuées ainsi que l’architecture de l’application.

### Installation et configuration des outils

#### Visual Studio

C’est une suite logicielle de développement conçue par Microsoft. La version utilisée au cours de ce projet est la version 2022. Visual Studio est un ensemble complet d'outil de développement permettant de générer des applications web, des services web, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE).

Visual C#, l’outil permettant de développer des applications en C# apparait pour la première fois dans la version 2002 de Visual Studio [6].

#### Installation et configuration de CodeIgniter

Pour pouvoir télécharger le Framework, il faut se rendre sur le site officiel de CodeIgniter [*www.codeigniter.com*](http://www.codeigniter.com), et plus précisément dans les rubriques downloads. Une fois que le Framework est téléchargé, il faut décompresser l’archive et placer le dossier portant le même nom que la version dans le répertoire web du projet *(www).* Il est beaucoup plus recommandé de renommer le dossier une fois dans le répertoire d’application pour éviter toute sorte de confusion de nom. Dans notre cas, le nom du dossier est *crm*.

La première chose à configurer, une fois le framework installé, est de définir s’il est actuellement en période de développement, de test ou de production. Pour ce faire, il faut ouvrir le fichier index.php et procéder à cette définition. Par défaut, la constante est déjà sur développement.

Une fois cela fait, nous avons normalement accès à la page d’accueil du framework au moyen du lien [*http://localhost/crm*](http://localhost/crm) et qui donne généralement un message de bienvenu.

La figure 64 illustre la page d’accueil de CodeIgniter.

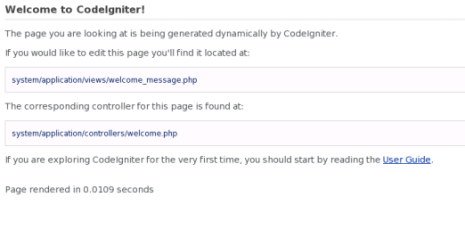


Figure 69. Message de bienvenue de CodeIgniter.

#### Configuration de CodeIgniter avec l’architecture REST

Avant toutes configurations, il faut tout d’abord se rendre au pour télécharger le Rest Server. Ensuite, une fois le fichier téléchargé, il faut s’assurer de bien vouloir suivre l’enchaînement des procédures suivantes :

* Extraire le répertoire d’archive téléchargé et le renommer selon nos besoins, par défaut le nom du fichier est codeigniter-restserver-master ;
* Copier le fichier codeigniter-restserver-master\application\config\rest.php dans le répertoire crm\application\config, crm est le répertoire d’application en question ;
* Copier le fichier codeigniter-restserver-master\application\libraries\Rest\_Controller.php dans le dossier crm\application\libraries ;
* Copier le fichier codeigniter-restserver-master\application\libraries\REST\_Controller\_Definitions.php dans le dossier crm\application\libraries ;
* Copier le fichier codeigniter-restserver-master\application\libraries\Format.php dans le répertoire crm\application\libraries ;
* Copier le fichier codeigniter-restserver-master\application\language\english\rest\control\_lang.php dans le dossier crm\application\language\english de l’application.
* Dans le fichier crm\application\libraries\REST\_Controller.php, il faut ajouter l’instruction « require APPPATH.’libraries/REST\_Controller\_Definitions.php’ ; » pour inclure le REST\_Controller\_Definitions.

#### Installation de Postman

Postman est gratuitement disponible sur son site web officiel [*www.getpostman.com*](http://www.getpostman.com). Comme tout autre logiciel, Postman se trouve parmi ceux qui sont faciles à installer. Son installation requiert 3 clics de bouton et ne dépasse pas les 3 minutes à cause de sa disposition de guide d’installation. Une fois installé, Postman n’aura plus besoin d’aucune configuration et on pourra déjà tester nos API REST à travers de sa barre de navigation.

La figure 65 indique Postman testant un API REST.

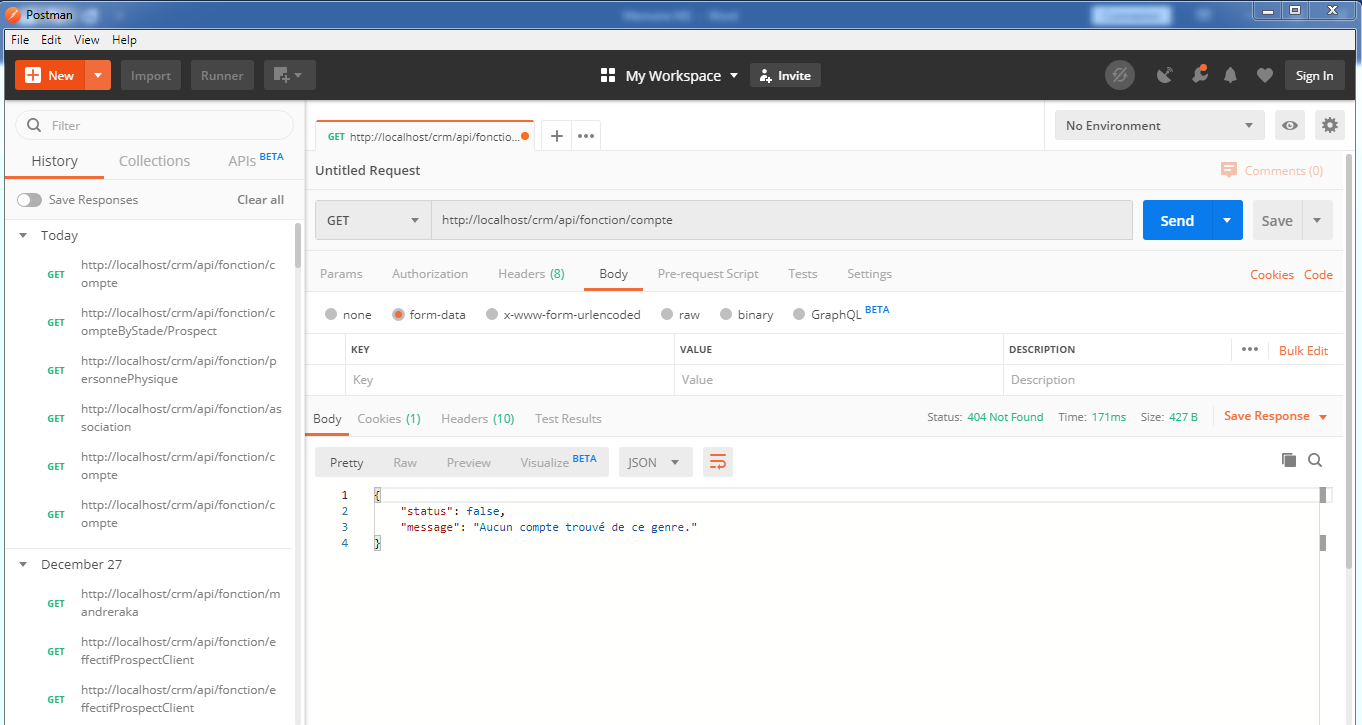


Figure 70. Postman testant un API REST.

#### Installation de Sublime Text

Tout comme Postman, installer Sublime Text exige peu de ressource énergétique. Il faut obligatoirement disposer d’un fichier exécutable pour pouvoir l’installer. Un tel fichier est disponible sur [*www.sublimetext.com*](http://www.sublimetext.com). Ensuite, pour l’installation, il faut se laisser aidé par l’installateur du produit en question.

La figure 66 représente une capture d’écran de Sublime Text.

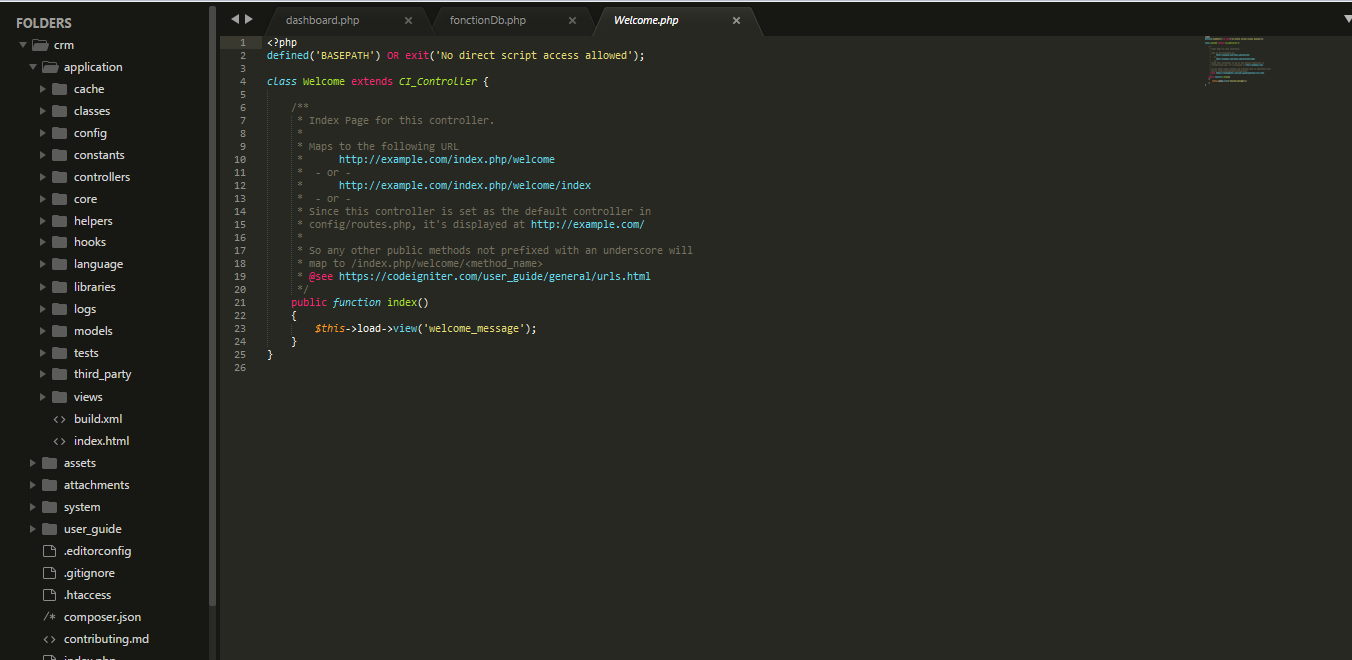


Figure 71. Capture d'écran de Sublime Text.

### Architecture de l’application

Au niveau de développement, l’architecture d’une application est représentée par l’ensemble de dossiers et fichiers qui composent cette application. L’architecture de notre application est illustrée par l’arborescence des dossiers et fichiers représentés par la figure 67.

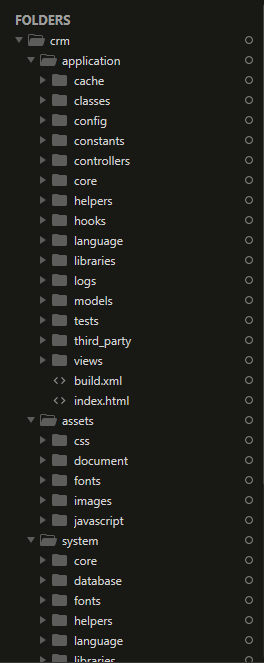
Le dossier parent de l’application est le répertoire *crm*. A son tour, il est composé de quatre dossiers et d’autres fichiers supplémentaires. Parmi ces quatre dossiers se trouvent le répertoire *application, assets, system* et *user\_guide*.

Figure 72. Architecture de l'application.

Dans le répertoire *application*, le sous-répertoire *config* contient les fichiers permettant de configurer CodeIgniter ou une bibliothèque. Certains sont inclus automatiquement, d’autres seulement lorsque on les demande. Le dossier *controllers* contient les contrôleurs. Le répertoire *errors* contient les pages d’erreurs. Les pages de ce dit dossier sont personnalisables selon nos propres designs. Le dossier *helper* contient tous nos helpers. Il est important de noter que les helpers fournis avec CodeIgniter sont situés dans le dossier *system*. Le répertoire *hooks* est un dossier qui contient des fichiers assez complexes. Ils permettent d’exécuter des scripts à différents moments du processus d’exécution de CodeIgniter. Le dossier *language* contient tous nos fichiers de langue dans le cas où on souhaite un site internationalisé. Le répertoire *libraries* contient les bibliothèques, *models* contient les modèles et *views* contient les vues.

Le dossier *assets* contient toutes les ressources qui seront manipulées par l’application à savoir les fichiers tels que les fichiers css, javascript ainsi que les fichiers images. Le troisième répertoire system contient les sources de CodeIgniter qui est fortement recommandé de ne pas toucher. Le dernier dossier user\_guide contient quelques documentations à propos des démarches à suivre pour manipuler le framework CodeIgniter.

## Développement de l’application

### Création de la base de données

Pour le développement de l’application, notre base de données sera nommée « stage\_crm\_bdd » et représentera l’intégralité du modèle de domaine. La création de la base de données et de quelques tables sont illustrées par la séquence d’instructions SQL :

CREATE DATABASE stage\_crm\_bdd;

USE stage\_crm\_bdd;

CREATE TABLE client (

idClient INT (4) NOT NULL AUTO\_INCREMENT,

designationClient VARCHAR (30) NOT NULL,

codePostal VARCHAR (5) NOT NULL,

stade VARCHAR (8) NOT NULL,

ville VARCHAR (25) NOT NULL,

pays VARCHAR (20) NOT NULL,

source VARCHAR (10) NOT NULL,

PRIMARY KEY (idClient)

);

CREATE TABLE compte (

idCompte INT (4) NOT NULL,

chiffreAffaire VARCHAR(15) NOT NULL,

secteurActivite VARCHAR (15) NOT NULL,

siteWeb VARCHAR (25) NOT NULL,

PRIMARY KEY (idCompte),

FOREIGN KEY (idCompte) REFERENCES client (idClient) ON DELETE CASCADE

);

### Codage de l’application

La partie consacrée au codage de l’application sert à présenter quelques morceaux d’algorithmes qui ont été utilisés pendant le développement de l’application.

La figure 68 montre un algorithme de création d’un client de type société.



Figure 73. Algorithme de création d'un client de type société.

La figure 69 montre un algorithme de suppression d’un événement.



Figure 74. Algorithme de suppression d'un événement.

La figure 70 montre un algorithme de récupération d’un ou de plusieurs événements.

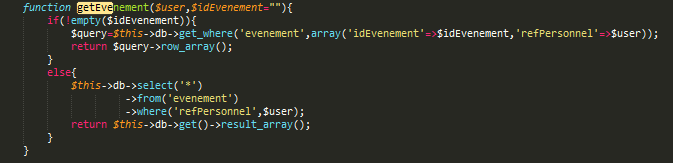


Figure 75. Algorithme de récupération d'un ou de plusieurs événements.

La figure 71 montre un algorithme JavaScript d’envoi d’une requête AJAX.

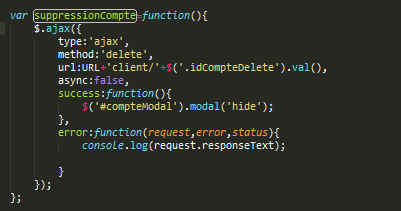


Figure 76. Algorithme JavaScript d'envoi d'une requête AJAX.

### Présentation de l’application

En ce qui concerne cette partie qui se focalise sur la présentation de l’application, elle se concentre sur la présentation de ladite application développée. Cette présentation se fait tout simplement par des captures d’écrans de quelques fonctionnalités fournies par l’application.

Pour pouvoir accéder à l’application, il suffit tout simplement de se rendre à l’adresse [http : localhost/crm](http://stage.inovatic.mg/crm) au moyen d’un navigateur internet. Ensuite, la page d’authentification de l’application devrait normalement s’afficher.

La figure 72 représente la page d’authentification de l’application.

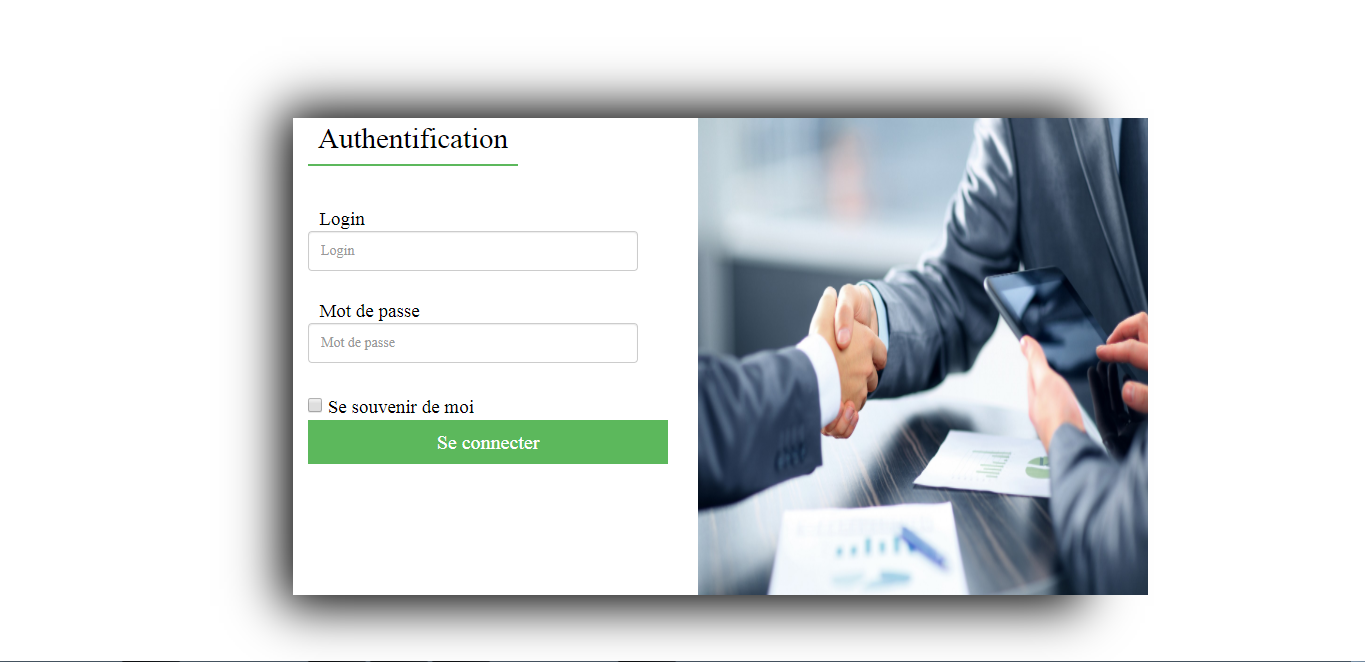


Figure 77. Page d'authentification de l'application.

Cette page d’authentification sert à contrôler l’accès à l’application ainsi qu’à ses différentes fonctionnalités. En effet, pour utiliser l’application, l’utilisateur devrait saisir son login ainsi que son mot de passe dans les formulaires correspondants et demande ensuite la connexion. Si tout se passe bien, l’utilisateur est authentifié et le système lui renvoie la page destinée au tableau de bord ou Dashboard. Dans le cas échéant, le système oblige l’utilisateur à entrer correctement ses informations de connexion tout en restant sur la page d’authentification. Au cas où il aurait oublié son mot de passe, il est possible de le réinitialiser. Pour se faire, il doit cliquer sur « Mot de passe oublié ? » et de suivre les instructions fournies par le système.

La figure 73 illustre la page consacrée au tableau de bord ou dashboard de l’application qui accueille l’utilisateur une fois authentifié.

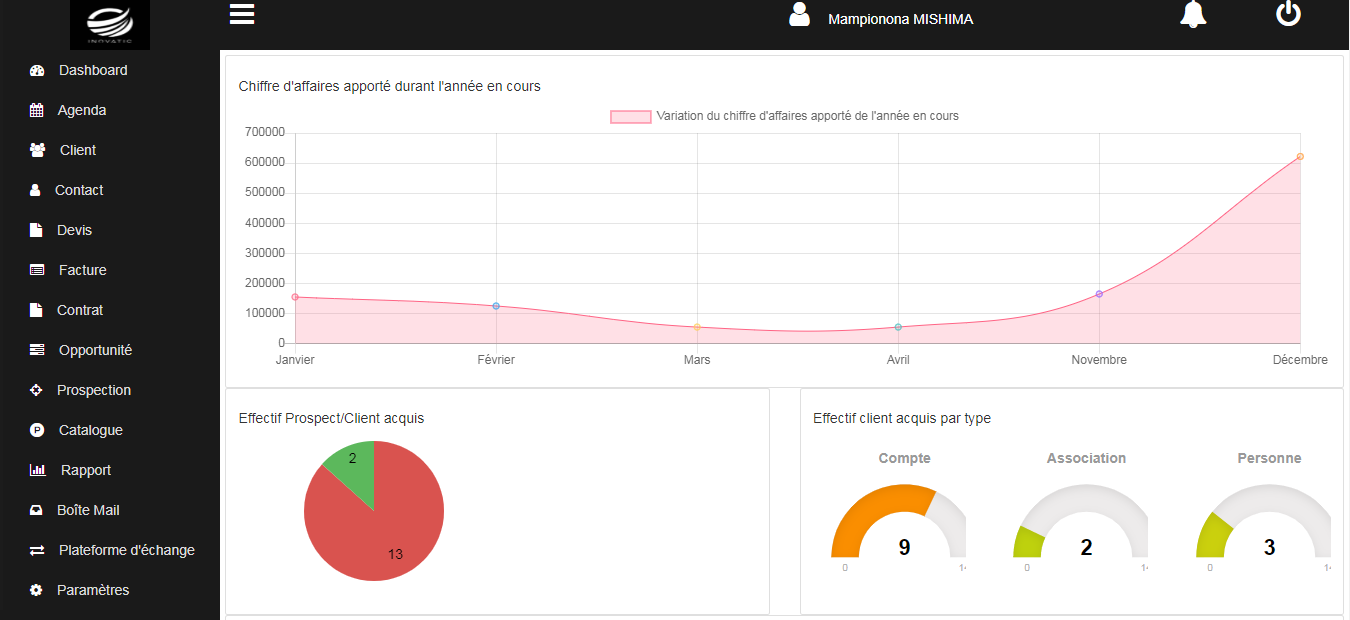


Figure 78. Tableau de bord de l'application.

Le CRM comporte plusieurs menus qui ont principalement leurs contenus respectifs. Parmi ces derniers, on distingue agenda, client, contact, etc.

La figure 74 représente la page d’agenda du CRM.

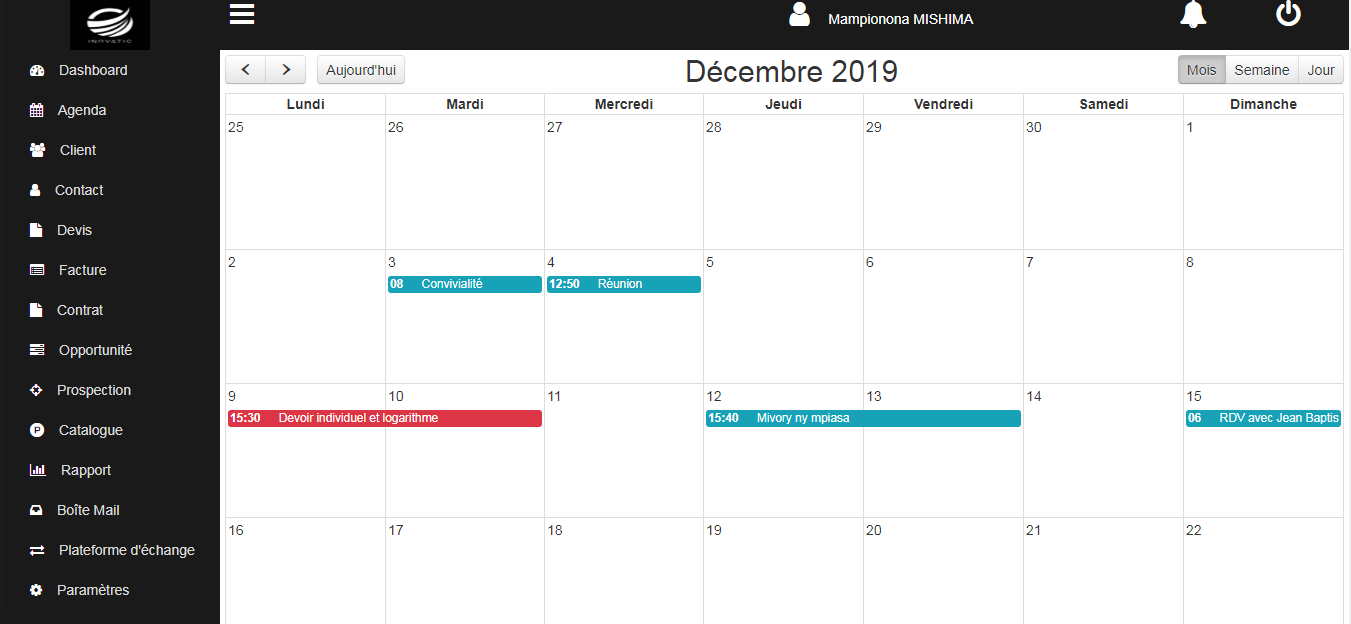


Figure 79. Page d'agenda du CRM.

L’agenda qui est illustré par cette figure présente quelques évènements qui ont leurs dates respectives. Les évènements qui sont en couleur rouge sont ceux qui sont urgents, en bleu, ce sont les évènements de niveau de priorité moyen et le reste pour les évènements de niveau de priorité bas.

La figure 75 illustre la page de création de devis.

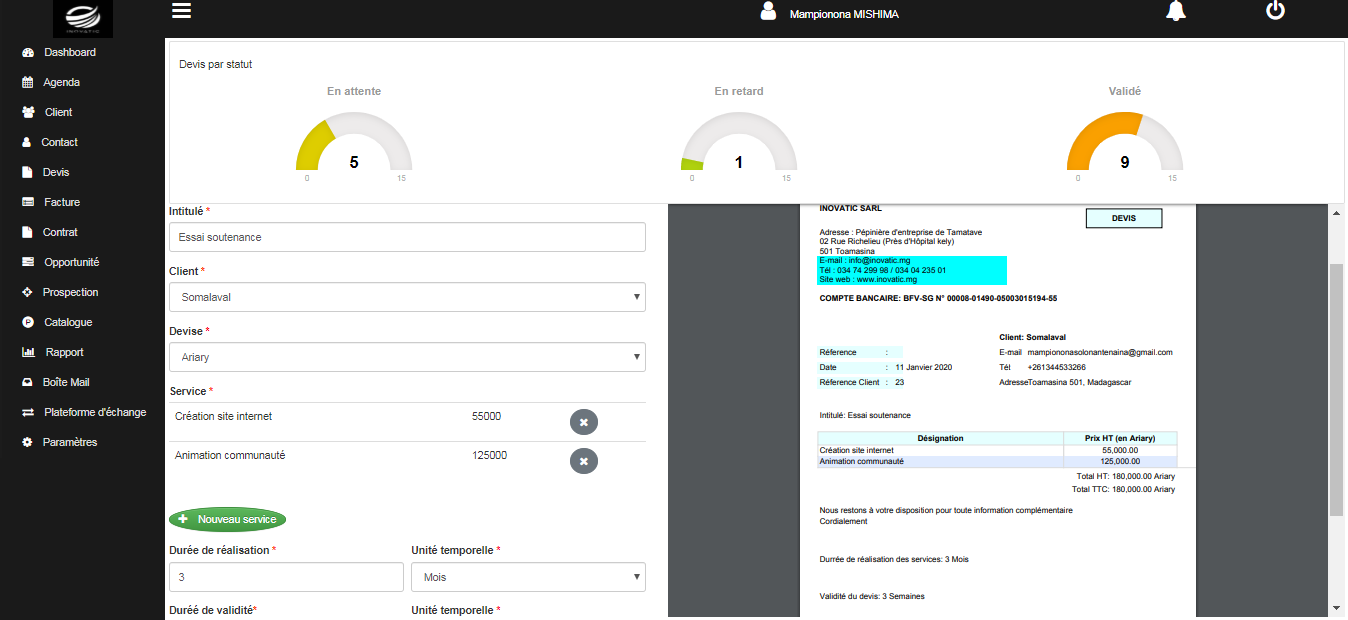


Figure 80. Création de devis.

En haut de la page, ces croquis représentent respectivement de gauche à droite, le nombre de devis en attente, le nombre de devis en retard et le nombre de devis validé.

La partie gauche de la seconde section représente les formulaires de création. Pour créer un devis, le responsable commercial doit entrer toutes les informations nécessaires à la production d’un tel document. Plus précisément, il doit obligatoirement mentionner, le client à qui le devis va être restituer, la devise utilisée dans le devis (Ariary, €, CAD, USD, etc.), les services demandés par les clients et ses prix respectifs, la durée de validité du devis et la durée totale de réalisation des services par les personnels impliqués. L’inclusion du TVA est optionnel.

Quant à la partie droite, elle illustre le devis crée au format PDF.

Le responsable commercial peut instantanément mettre à jour le devis au moyen du bouton « Mette à jours le devis » et une fois qu’il est satisfait, il doit envoyer le devis en cliquant sur le bouton « Envoyer le devis ». Cet envoi consiste à expédier un mail au client cible affirmant que son devis est déjà prêt. Le devis crée est en pièce jointe du mail envoyé.

La figure 76 représente le mail reçu par le client et le devis correspondant.



Figure 81. Mail reçu par le client et le devis correspondant.

Une fois le mail contenant le devis dans la boîte de réception du client, il se peut qu’il ne le consulte pas dans l’immédiat, c’est pourquoi le système envoye un SMS de notification au client à chaque fois qu’un devis lui est envoyé. La figure 77 représente le SMS envoyé au client à chaque envoi de devis.



Figure 82. SMS envoyé au client à chaque envoi de devis.

Au cas où la réalisation de tous les services demandés par le client est terminée et qu’il est satisfait, on peut lui transmettre une facture. La figure 78 illustre la facture crée à partir d’un devis validé.

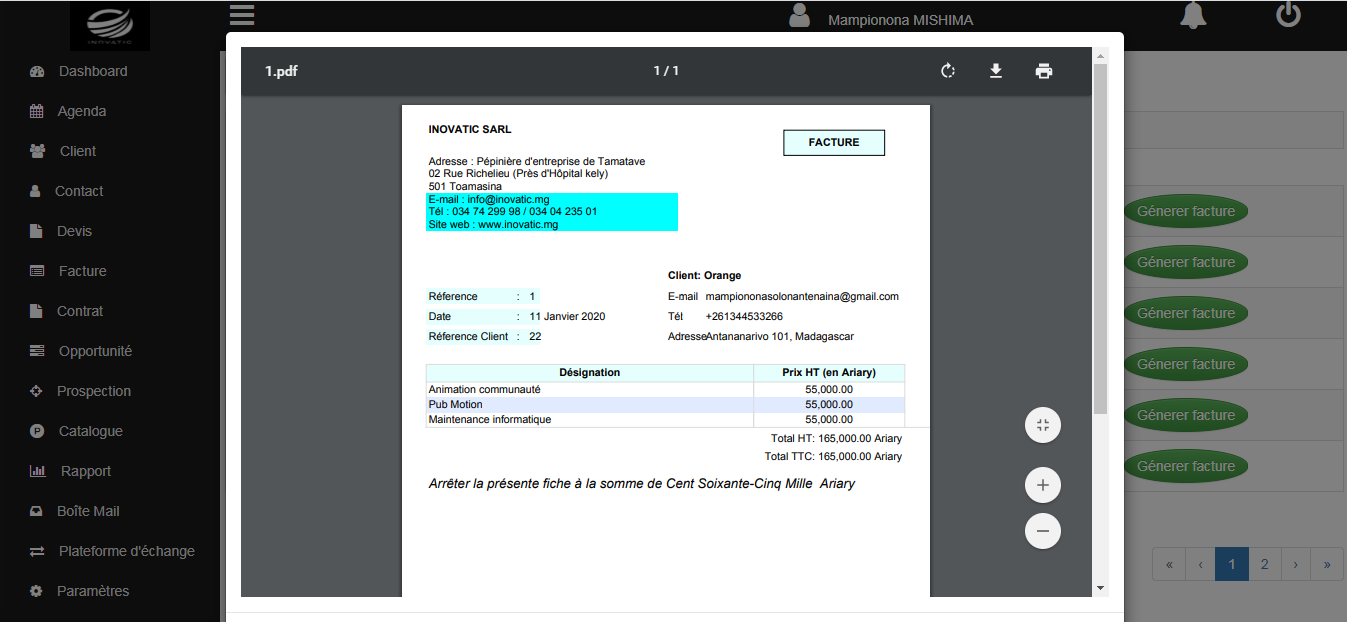


Figure 83. Facture créée à partir d'un devis validé.

# Conclusion

Etant une entreprise informatique, notre établissement d’accueil Inovatic se spécialise sur la promotion de service numérique envers ses clients. Clients avec qui, l’entretien de relation ainsi que la gestion de ses informations se font encore à l’aide de multiples dispositifs logiciels. Dans ce projet, nous avons conçu et réaliser une application web dynamique, plus précisément un CRM qui est capable de pallier les difficultés rencontrées en utilisant ces outils.

En parlant de création d’application, multiples sont les outils qui ont été d’une vraie nécessité pour la mise en place du projet. Nous avons utilisé les méthodes agiles comme méthode de gestion de projet plus précisément SCRUM. 2 TUP pour la méthode de conception, Visual Paradigm comme outil de modélisation, PHP pour le langage de programmation tout en utilisant le framework CodeIgniter. En plus de CodeIgniter, nous avons aussi utilisé d’autres cadres d’application comme JQuery et Bootstrap. Le Système de Gestion de Base de données étant le SGBDR MySQL, le service web utilisé est REST, Sublime Text pour l’environnement de développement, Postman comme outil de test d’API et Filezilla comme logiciel de déploiement.

L’application permet aux personnels de la société de gérer l’ensemble des informations des clients de la société. En plus, elle est aussi capable d’entretenir des relations avec ces dits clients. En un mot, l’application gère l’ensemble de la relation client de la société.

Cette application est désormais fonctionnelle et répond aux besoins de la société. Néanmoins, dans la perspective, nous envisageons de l’améliorer en y intégrant d’autres fonctionnalités comme effectuer de la téléprospection en passant des appels téléphoniques, utiliser un système automatique de paiement en ligne pour les règlements et rendre à l’application la capacité de l’utiliser en mode déconnecté.

Certes, ce stage nous a permis pour la troisième fois, de familiariser avec le monde professionnel, il a aussi enrichi nos connaissances et qualités d’aptitude tant en termes de développement informatique qu’au niveau d’approfondissement de culture professionnelle.

# Références bibliographiques

|  |  |
| --- | --- |
| [1] | Nicolas Larousse, 2009. Création de bases de données. Edition Pearson, France, 205 pages. |
| [2] | Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde. 2012, The Scrum primer. France, 224 pages. |
| [3] | Claude Aubry. 2009, SCRUM, Pour une pratique vivante de l’agilité. 5è édition. Edition Dunod, France, 314 pages. |
| [4] | Christophe Legrenzi. 2011, Scrum, ou les principes du rugby appliqués au développement applicatif. France, 112 pages. |
| [5] | Pierre-Alain Muller, 1997. Modélisation objet avec UML. Edition Eyrolles, France, 449 pages. |
| [6] | Pascal Roques, 2007. UML 2 en action, De l’analyse à la conception. Edition Eyrolles, France, 394 pages. |
| [7] | Joseph Gabay, David Gabay, 2008. UML 2, de l’analyse à la conception. Edition Dunod, France, 242 pages. |
| [8] | Pierre-Alain Muller, 1997. Modélisation objet avec UML. Edition Dunod, France, 449 pages. |
| [9] | Pascal Roques, 2008. Les cahiers du programmeur, UML 2, Modéliser une application web, 4è édition. Edition Eyrolles, France, 264 pages. |
| [10] | Myriam Bakayoko, Jeremy Michel, André Noel, Guillaume Gebavi, Arthur Ageron, Fabien Simon. 2011, Les IDE de conception. France, 15 pages. |
| [11] | Gilles Dowek, 1997. Le langage mathématique et les langages de programmation. La Villette, Paris, 227 pages. |
| [12] | Luke Welling, Laura Thomson, 2012. PHP & MySQL, 4è édition, Pearson, 961 pages. |
| [13] | Nicolas Larousse, 2009. Création de bases de données. Edition Pearson, France, 205 pages. |
| [14] | Fabien Celaia, 2003. Quel SGBD choisir ? France, 23 pages. |
| [15] | Philippe Rigaux, 2009.Pratique de MySQL et PHP. 4è édition, Dunod, Paris, 557 pages. |
| [16] | Benoît Philibert, 2015. Bootstrap 3, le Framework 100% web design, Edition Eyrolles, 37 pages. |
| [17] | Jonathan Chaffer, Karl.Swedberg, 2009. JQuery, Simplifiez et enrichissez vos développements Javascript, Edition Pearson, France, 428 pages. |
| [18] | Ryan Benedetti, Ronan Cranley, 2011. JQuery, Tête la première, Edition Dunod, France, 30 pages. |
| [19] | Thomas Bodin, 2017. Bootstrap. France, 68 pages. |
| [20] | Benoît Philibert, 2015. Bootstrap 3, le Framework 100% web design, Edition Eyrolles, France, 37 pages. |
| [21] | Xavier Fournier-Morel, Pascal Grojean, Guillaume Plouin, Cyril Rognon. 2012, SOA : le guide de l’architecte SI, 2è édition, Edition Dunod, 372 pages. |
| [22] | Pascal roques. 2011. UML 2 par la pratique, Etude de cas et exercices corrigés. 5è édition, Edition Eyrolles, 364 pages. |
| [23] | Guillaume Harry, Marie-Claude Quidoz. 2011. Décrire sa base de données. France, 64 pages. |
| [24] | Claude Belleil, 2012. Le langage UML 2.0, Diagramme de contexte. France, 12 pages. |
| [25] | Pierre-Alain Muller, 1997. Modélisation objet avec UML. Edition Eyrolles, France, 449 pages. |

# Références webographiques

|  |  |
| --- | --- |
| [1.a] | <https://stph.scenari-community.org/idl-bd/1/co/pri1c23.html>, Introduction générale aux bases de données, Consulté le 15 Octobre 2019. |
| [2.a] | <https://www.visual-paradigm.com/tutorials/data-flow-diagram-dfd.jsp>, Qu'est-ce qu'un diagramme de flux de données (DFD) ? Comment dessiner DFD ? Consulté le 27 Septembre 2019. |
| [3.a] | <http://www.comparatif-logiciels.fr/logiciel-saas/microsoft-dynamics-crm/>, Présentation de Microsoft Dynamics CRM, Consulté le 27 Septembre 2019. |
| [4.a] | <https://www.appvizer.fr/magazine/relation-client/customer-relationship-management-crm/exemples-de-logiciel-crm>, Exemples de logiciel CRM, Consulté le 27 Septembre 2019. |
| [5.a] | <https://www.planzone.fr/blog/methodologies-gestion-projet>, 7 méthodologies de gestion de projet à connaître absolument, Consulté le 16 Octobre 2019. |
| [6.a] | <https://agiliste.fr/introduction-methodes-agiles/>, Introduction aux méthodes agiles et SCRUM, Consulté le 16 Octobre 2019. |
| [7.a] | <https://www.visual-paradigm.com/scrum/extreme-programming-vs-scrum/>, Extreme Programming (xP) Vs SCRUM, Consulté le 24 Décembre 2019. |
| [8.a] | <https://www.mountaingoatsoftware.com/blog/differences-between-scrum-and-extreme-programming>, Différences entre SCRUM et Extreme Programming, Consulté le 24 Décembre 2019. |
| [9.a] | <https://www.geeksforgeeks.org/difference-between-scrum-and-xp/>, Différence entre SCRUM et xP, Consulté le 24 Décembre 2019. |
| [10.a] | <https://interworks.com.mk/why-scrum-simply-works/>, Why SCRUM simply works? , Consulté le 11 Janvier 2020. |
| [11.a] | <http://pf-mh.uvt.rnu.tn/816/1/conception-developpement-bureau-virtuel-WebTop.pdf>, Méthode 2TUP, Consulté le 3 Septembre 2019. |
| [12.a] | <https://www.researchgate.net/figure/La-methode-2TUP-6_fig6_303497462>, La méthode 2TUP, Consulté le 3 Septembre 2019. |
| [13.a] | <https://about.draw.io/uml-diagrams/>, Diagrammes UML – quel diagramme utiliser et pourquoi, Consulté le 11 Janvier 2020. |
| [14.a] | <http://argouml-fr.tigris.org/>, Bienvenue à ArgoUML, Consulté le 3 Janvier 2020. |
| [15.a] | <https://www.visualparadigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html>, Visual Paradigm, présentation du produit, Consulté le 17 Décembre 2019. |
| [16.a] | <https://www.techno-science.net/definition/11378.html>, Langage de programmation, Consulté le 27 Août 2019. |
| [17.a] | <https://slideplayer.fr/slide/3011987/>, Etude comparative Php, ASP, JSP, Consulté le 26 Décembre 2019. |
| [18.a] | <https://waytolearnx.com/2019/04/difference-entre-jsp-et-php.html>, Différence entre PHP et JSP, Consulté le 26 Décembre 2019. |
| [19.a] | <http://www-igm.univ-mlv.fr/~dr/XPOSE2001/Jourdan/>, Présentation de JSP (un peu de PHP), Consulté le 26 Décembre 2019. |
| [20.a] | <https://www.gladir.com/LEXIQUE/ARTICLES/comparelanguagewebphpvsaspnetvsjsp.htm>, Comparatif des langages web, Consulté le 26 Décembre 2019. |
| [21.a] | <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203633-sgbd-systeme-de-gestion-de-base-de-donnees-definition-traduction-et-acteurs/>, SGBD (Système de Gestion de Base de données) : définitions, traductions et acteurs, Consulté le 20 Septembre 2019. |
| [22.a] | <https://www.techno-science.net/definition/1471.html>, Définitions et généralités des frameworks, Consulté le 27 Août 2019. |
| [23.a] | <https://www.educba.com/laravel-vs-symfony/>, Laravel vs Symfony, Consulté le 2 Janvier 2020. |
| [24.a] | <https://www.educba.com/laravel-vs-codeigniter/>, Laravel vs CodeIgniter, Consulté le 2 Janvier 2020. |
| [25.a] | <https://capitainewp.io/formations/wordpress-creer-blocs-gutenberg/differences-react-jquery/>, Différences entre React et JQuery, Consulté le 2 Janvier 2020. |
| [26.a] | <https://www.supinfo.com/articles/single/6956-differences-cles-angularjs-reactjs>, Différences clés entre AngularJs et ReactJs, Consulté le 3 Janvier 2020. |
| [27.a] | <https://blog.arca-computing.fr/javascript-comparatif-angular-react-vue/>, Le comparatif Javascript frameworks : Angular Vs React Vs Vue, Consulté le 3 Janvier 2020. |
| [28.a] | <https://w3cmasters.wordpress.com/2012/11/12/beaucoup-davantages-et-peu-dinconvenients/>, Beaucoup d’avantages et peu d’inconvénients …, Consulté le 3 Janvier 2020. |
| [29.a] | <https://www.supinfo.com/articles/single/2694-comparatif-10-frameworks-utilises-front-end>, Comparatif de 10 des frameworks utilisés en Front-end, Consulté le 29 Décembre 2019. |
| [30.a] | <https://codeigniter.com/user_guide/general/welcome.html>, Bienvenue sur CodeIgniter, Consulté le 25 Octobre 2019. |
| [31.a] | <https://www.ibm.com/support/knowledgecenter/fr/SS5JSH_9.1.2/org.eclipse.jst.ws.doc.user/concepts/cws.html>, Présentation des services web, Consulté le 20 Septembre 2019. |
| [32.a] | <https://www.journaldunet.fr/web-tech/developpement/1202749-soap-vs-rest-les-principales-differences/>, SOAP vs REST : Les principales différences. Consulté le 13 Septembre 2019. |
| [33.a] | <https://waytolearnx.com/2019/04/difference-entre-soap-et-rest.html>, Différence entre SOAP et REST, Consulté le 17 Septembre 2019. |
| [34.a] | <https://www.mobizel.com/definition-cest-quoi-un-ide/>, [DEFINITION] C'EST QUOI...UN IDE ? Consulté le 31 Octobre 2019. |
| [35.a] | <https://stackshare.io/stackups/phpstorm-vs-sublime-text>, PHP Storm Vs Sublime Text, Consulté le 26 Décembre 2019. |
| [36.a] | <https://www.jetbrains.com/phpstorm/buy/#commercial?billing=yearly>, Abonnement à ToolBox, Consulté le 28 Décembre 2019. |
| [37.a] | <https://www.slant.co/versus/40/812/~sublime-text_vs_phpstorm>, Sublime Text Vs PHP Storm, Consulté le 26 Décembre 2019. |
| [38.a] | [https://www.supinfo.com/articles/single/5302-sublime-text,](https://www.supinfo.com/articles/single/5302-sublime-text,%20Sublime%20Text) , Sublime Text, Consulté le 15 Novembre 2019. |
| [39.a] | <https://www.capterra.fr/compare/169838/188497/ibm-api-management/vs/postman>, IBM API Management vs Postman, Consulté le 15 Novembre 2019. |
| [40.a] | <https://www.capterra.fr/compare/152298/188497/dell-boomi/vs/postman>, Dell Boumi et Postman, Consulté le 15 Novembre 2019. |
| [41.a] | <https://www.getpostman.com/>, La plateforme de collaboration pour le développement des API, Consulté le 15 Novembre 2019. |
| [42.a] | <https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/comparaison-de-clients-ftp-dexcellence-pour-lhebergement-dun-site-web/>, 10 clients FTP/logiciels FTP d'excellence pour l'hébergement web, Consulté le 26 Décembre 2019. |
| [43.a] | <https://filezilla-project.org/client_features.php>, Caractéristiques de Filezilla, Consulté le 2 Janvier 2020. |
| [44.a] | <https://online.visual-paradigm.com/diagrams/tutorials/sequence-diagram-tutorial/>, Tutoriel sur les diagrammes de séquence, Consulté le 25 Octobre 2019. |
| [45.a] | <https://www.techno-science.net/definition/1471.html>, Définitions et généralités des frameworks, Consulté le 27 Août 2019. |
| [46.a] | <http://www.webcky.fr/blog/2016/09/16/developpement-site-web-avec-larchitecture-mvc-en-isere/>, Blog – webcky. Développement de site internet et application mobile, Consulté le 11 Janvier 2020. |
| [47.a] | <https://online.visual-paradigm.com/diagrams/tutorials/package-diagram-tutorial/>, Tutoriel sur les diagrammes de paquetage, Consulté le 25 Octobre 2019. |
| [48.a] | <https://online.visual-paradigm.com/diagrams/tutorials/deployment-diagram-tutorial/>, Didacticiel du diagramme de déploiement, Consulté le 25 Octobre 2019. |

# Glossaire

|  |  |
| --- | --- |
| B2B | Ensemble des activités commerciales nouées entre deux entreprises. Le terme anglais complet est Business to Business. Il peut aussi s’écrire également BtoB. De manière génerale, le B2B concerne tous les moyens utilisés pour mettre en relation des sociétés et faciliter les échanges de produit, de services et d’informations entre elles. |
| B2C | Activité commerciale et marketing qui s’opère à destination des consommateurs particuliers. Le marketing B2C regroupe l’ensemble des actions marketing qui visent et s’adressent aux consommateurs particuliers. Il est composé par des techniques, actions et campagnes marketing les plus visibles et les mieux connues et identifiées par le grand public. |
| Backlog | Destiné à accueillir tous les besoins du client que l’équipe du projet doit réaliser. Il contient donc la liste des fonctionnalités intervenant dans la constitution d’un produit, ainsi que tous les éléments nécessitant l’intervention de l’équipe projet. |
| Big Data | Appelé aussi mégadonnées, le Big Data désigne l’ensemble des données numériques produites par l’utilisation des nouvelles technologies à des fins personnelles ou professionnelles. Ces dernières recoupent les données d’entreprise (courriel, document, base de données, historique de processus métiers, etc.) aussi bien que des données issues des capteurs, des contenus publiés sur le web (images, vidéos, sons, textes), des transactions de commerce électronique, des échanges sur les réseaux sociaux, des données transmises par des objets connectés (étiquettes électroniques, compteurs intelligents, smartphones…), des données géolocalisées, etc. |
| Blueprint | Société créée en 2006 par Olav Frihagen Bjorkoy. |
| Chiffre d’affaires | Somme de vente de biens ou de services facturées sur un exercice. Il peut être exprimé Hors Taxes (HT) ou Toutes Taxes Comprises (TTC) s’il inclut la TVA. Il est également possible de déterminer un montant brut ou net (en déduisant les RRR ou Rabais, Remises et Ristournes accordées aux clients. |
| Compte | Synonyme de société, un compte désigne une entité personne morale qui exerce habituellement des activités à but lucratif et qui dispose généralement d’une personnalité juridique. |
| Contact | Personne qui travaille dans une organisation (compte ou association) et qui fait l’intermédiaire entre cette dernière et d’autres personnes incluses ou non dans d’autres institutions pour une opération marketing ou pour d’autres types d’opération. |
| Contrat | Document contenant des conventions par laquelle une ou plusieurs personnes (physiques ou morales) s’obligent à donner, à faire ou à ne pas faire. |
| CRM | Stratégie de gestion des relations et interactions d’une entreprise avec ses clients ou prospects. Il regroupe l’ensemble des dispositifs ou opérations de marketing ou support ayant pour but d’optimiser la qualité de la relation client, de fidéliser et de maximiser le chiffre d’affaires ou la marge par client. Un système CRM aide les entreprises à interagir en permanence avec les clients, à rationaliser leur processus et à améliorer leur rentabilité. |
| Data Binding | Technique générale qui lie les sources de données du fournisseur et du consommateur et les synchronise. |
| Devis | Document écrit et présenté par un fournisseur proposant de vendre un bien à un prix qu’il s’engage à ne pas changer tant que l’acheteur n’a pas souhaité de renoncer à en faire la demande. On parle plus simplement d’une proposition de contrat dans le cas d’une offre de travaux. |
| Diagramme | Représentation visuelle simplifiée et structurée des concepts, des idées et des constructions. Il est employé dans tous les aspects des activités humaines pour visualiser et éclaircir une matière. |
| Echéance | Data à laquelle est exigible l’exécution d’une obligation, moment où quelque chose doit arriver, et qui marque la fin d’un délai ou bien même d’une période. |
| Fidélisation client | Processus adopté par une entreprise et les moyens mis en place pour fidéliser ses clients sur le long terme, et les inciter à renouveler leurs achats. |
| Lead | Suspect avec qui un début de conversation a démarré. Ce début de conversation peut être direct ou indirect. Direct quand les responsables commerciaux sont en relation avec le Lead et indirect quand on leur a parlé ou mis en relation avec le Lead. |
| Logiciel libre | Selon la FSF, un logiciel libre est défini par 4 libertés dont la liberté d’exécuter le programme pour tous les usages, la liberté d’étudier le fonctionnement du programme, de l’adapter aux besoins, la liberté de redistribuer des copies, la liberté d’améliorer le programme et de publier les améliorations. |
| Logiciel on-premise | Logiciel dont l’architecture est physiquement présente au sein de l’entreprise (c’est-à-dire installée sur les serveurs de l’entreprise) qui s’est acquittée d’une licence auprès de l’éditeur. |
| Logiciel propriétaire | Logiciel écrit, pour la plupart du temps, par une entreprise. Il est distribué uniquement sous forme binaire et disponible sous une licence assez restrictive. On ne connaît pas sa recette car elle reste la propriété de l’entreprise. |
| Logiciel SaaS | Solution logicielle utilisable en ligne (on parle aussi de cloud ou de on demand). Le logiciel est délocalisé et accessible à distance par un réseau, moyennant un abonnement auprès de l’éditeur/hébergeur, sur n’importe quel navigateur. |
| Modèle | Représentation de la réalité. Il en existe deux types en général dont les maquettes et les prototypes. Les maquettes peuvent être à échelle réduite (différente échelle possible). Elles sont idéales si l’on souhaite montrer des projets ou bien même visualiser des designs. Par contre, les prototypes, sont réalisés à taille réelle et sont généralement uniques puisque ce sont les premiers exemplaires d’un produit. |
| Opportunité commerciale | Appelée aussi occasion d’affaires, une opportunité commerciale est une tendance potentiellement favorable que l’on observe dans l’environnement, qui peut donner lieu à une action commerciale bénéfique pour l’entreprise. |
| Porte-à-Porte | Appelée aussi vente à domicile ou vente traditionnelle, Porte-à-Porte est une forme de vente directe où le vendeur se rend chez l’acheteur pour proposer ses produits et ses services. La vente à domicile peut être individuelle ou collective. La vente individuelle s’effectue le plus souvent sous forme de démarchage direct, sans rendez-vous préalable, tandis que la vente collective sera toujours organisée à l’avance. |
| Product Owner | Responsabe de la définition et de la conception d’un produit. Il est chargé de mener à terme un projet en utilisant la méthode SCRUM. |
| Propale | Appelée aussi proposition technique et commerciale, une propale est en quelque sorte un « devis amélioré ». Plus complète, elle a pour objectif de définir de manière précise les modalités d’exécution d’une prestation de service. Elle permet au prestataire de formaliser son offre et de la présenter au client. Ce document est souvent composé de trois parties dont la partie technique qui contient les réponses techniques aux besoins du client, la partie organisationnelle qui mentionne les ressources humaines allouées et la partie commerciale qui détaille les postes de facturation, les tarifs appliqués et les conditions de règlement, etc. |
| Prospect | Futur client potentiel, soit une personne ou une organisation qui correspond aux critères des acheteurs et avec qui une vente sera possible. Il est identifié au cours d’un certain nombre d’actions, comme l’annonce dans les médias, le publipostage, les appels téléphoniques, la participation aux foires et aux salons. La transformation du prospect en client est définitive lorsqu’à la suite de plusieurs relances, le prospect décide d’acheter le produit proposé et conclut la vente. |
| Prospection | Technique efficace permettant de promouvoir une entreprise et de mettre en avant ses actions commerciales en présentant ses prestations. Il existe plusieurs formes de prospection commerciale parmi lesquelles la prospection téléphonique ou démarchage téléphonique ou phoning, la prospection physique ou meeting, le marketing direct par envoi d’emails ou mailing, le faxing, etc. |
| Publipostage | Campagne de marketing direct qui consiste à envoyer une proposition commerciale par voie postale ou bien même par courrier électronique à un ensemble d’individus soigneusement ciblés. L’objectif d’un publipostage est de provoquer une réponse de la part du prospect. |
| Relance | Opération de marketing direct destinée à des prospects déjà contactés par l’entreprise. Dans une vente à distance, les relances consistent généralement en un envoi de mail ou un appel téléphonique auprès des prospects déjà détenteurs de catalogue général. |
| Relation client | Ensemble des actions visant à augmenter les ventes par la création et le suivi d’une relation avec la clientèle. |
| Scrum Master | Responsable d’un projet SCRUM. Son rôle est d’assurer que les règles du SCRUM sont respectées. Il a aussi pour rôle de faciliter la communication de l’équipe. |
| Snippets | Morceaux de code qui seront interprétés par un environnement de développement et qui permettront de taper moins de code pour en écrire plus. |
| Suivi commercial | Technique et méthode qui permettent d’accompagner un prospect depuis le premier contact avec le commercial jusqu’à la conclusion de vente. |
| Suspect | Personne ou entité dont les caractéristiques correspondent à un client idéal potentiel pour l’entreprise. |
| Téléprospection | La téléprospection désigne l’activité de prospection par téléphone. Les activités de téléprospection se font le plus souvent à partir d’un centre d’appels spécialisé mais peuvent également être menée par un commercial isolé. L’objectif d’une action par téléprospection est rarement un objectif de vente direct au téléphone, le plus souvent, il s’agit d’obtenir des rendez-vous ou de qualifier des prospects qui seront ensuite rappelés et contactés par un autre canal. |

# Annexes

1. **Cahier des charges fonctionnels (cdc).**

**Contexte et définition du problème**

La société Inovatic œuvre sur le développement des sites internet ainsi que sur la mise en place des applications informatiques quelques soient web, lourdes ou bien même mobiles. Elle se spécialise aussi sur le design de divers genres et sur tout autre service touchant le monde du numérique. Dans ce cadre, pour accomplir et bien organiser ses tâches professionnelles, il est impératif de posséder un outil permettant de gérer les relations avec les clients qui sont les principaux rouages de la société. En parlant de gestion de relation client, comme moyens utilisés actuels, Inovatic a aujourd’hui de plus en plus d’appels techniques à gérer et que ces derniers sont consignés dans des fichiers Excel. Sur ce, les personnels administratifs ne peuvent pas travailler dessus à plusieurs.

**Objectif principal du projet**

L’objectif suprême dans ce projet est de mettre en place un CRM ou Customer Relationship Management ou encore Gestion de la Relation Client et qui sera capable d’aider les responsables commerciaux ainsi que la Direction.

**Périmètre du projet**

Le projet devra se concentrer sur les clients nationaux et internationaux. Parmi les clients étrangers, France et Canada sont les pays dont la société a répétitivement d’opportunités d’affaires.

**Utilisateurs de l’application une fois mise en place**

Les responsables commerciaux de la société Inovatic ainsi que la Direction.

**Description fonctionnelle des besoins**

Une fois mise en place, le CRM doit inclure au moins les fonctionnalités renseignées dans le tableau 46.

Tableau 29. Description des besoins fonctionnels.

|  |  |
| --- | --- |
| **Fonctions** | **Sous-fonctions** |
| Gestion des clients/prospects | * Ajouter des clients ; * Modifier les informations concernant un client déjà existant ; * Supprimer un client ; |
| Historiser les actions des clients ou prospects | * Lister les actions des clients ou prospects. |
| Analyse de la force de vente | * Analyse des résultats commerciaux |
| Gestion de devis | * Génération automatique de devis et envoi au format PDF |
| Gestion de facture | * Génération de facture à partir d’un devis validé par un client et son envoi au client cible. * Consulter la liste des factures déjà générées. |
| Gestion de contrat | * Créer des contrats * Consulter la liste des contrats |
| Mise en place d’un agenda | * Ajouter un événement ; * Modifier un événement ; * Supprimer un événement de l’agenda. |
| Gestion d’opportunités | * Consulter les opportunités |
| Gestion de catalogue de service | * Ajouter, modifier ou supprimer un service |
| Plateforme d’échange | * Envoyer des messages entre Personnel |
| Reporting | * Dégager les statistiques commerciales annuelles ou bien même mensuelles. |

**Langue de l’application**

L’application doit être pour le moment disponible uniquement en langue française.

**Délai de mise en fonction**

Environ 5 mois

1. **Devis dans l’ancien système.**

La figure 79 représente un exemple de devis déjà utilisé par la société avant l’automatisation de la relation client.



Figure 84. Devis utilisé dans l'ancien système.

1. **Facture dans l’ancien système.**

La figure 80 représente un exemple de facture utilisée par la société dans l’ancien système.

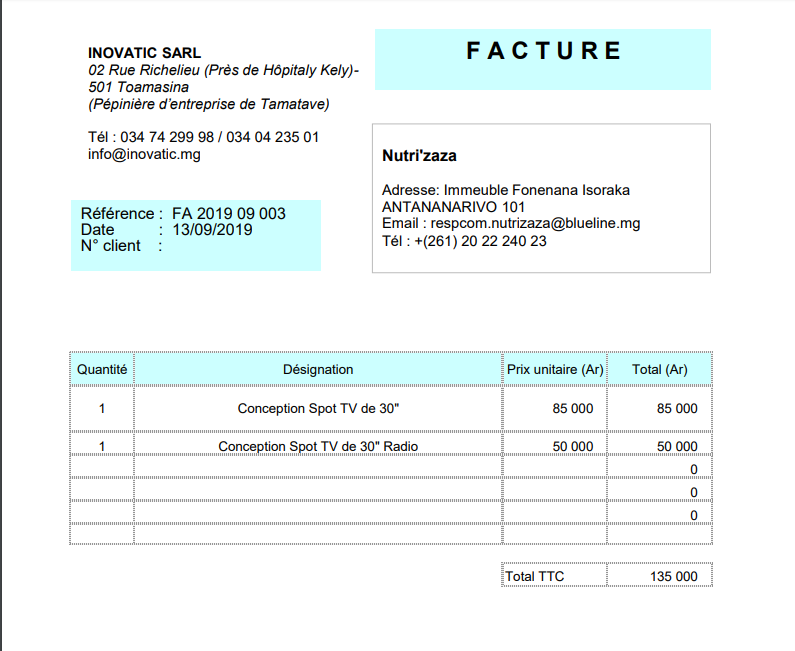


Figure 85. Facture dans l'ancien système.

1. **Fichier de prospection dans l’ancien système.**

La figure 81 représente un exemple de fichier Excel de prospection, utilisé auparavant par les commerciaux pour suivre les clients.

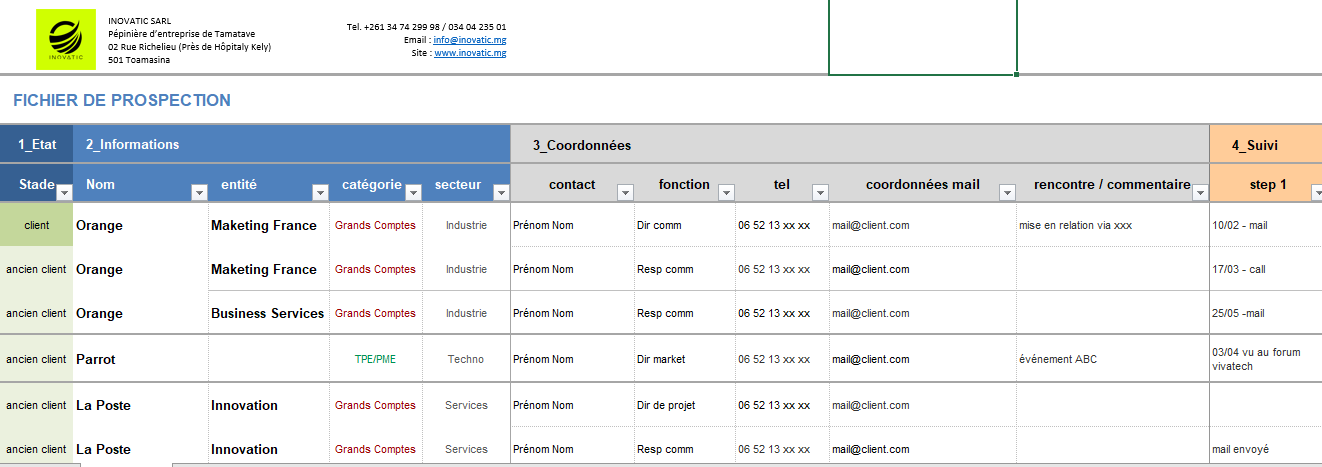


Figure 86. Fichier de prospection dans l'ancien système.

1. **Dépendances fonctionnelles**

|  |  |  |
| --- | --- | --- |
| idEvenement | 🡪 | descrEvenement, debutEvenement, finEvenement, prioriteEvenement |
| idOpportunite | 🡪 | descrOpportunite, probaGain, phase |
| idService | 🡪 | descrService |
| idCategorie | 🡪 | descrCategorie |
| idContrat | 🡪 | debutContrat, finContrat |
| idPiece | 🡪 | nomFichier |
| idPersonne | 🡪 | civilite, nomPrenom, mail, telephone, photo |
| idClient | 🡪 | codePostal, designationClient, source, stade, ville, pays |
| referenceDevis | 🡪 | dteCreationDevis, dureeValidite, dureeRealisation, intitule, statut, uniteMonetaire, tva |
| idCompte | 🡪 | chiffreAffaire, secteurActivite, siteWeb |
| idAssociation | 🡪 | typeAssociation |
| idPersonnel | 🡪 | login, mdp, posteOccupe |
| idContact | 🡪 | fonction, niveau |
| idMessage | 🡪 | contenuMessage, dteEnvoiMessage |

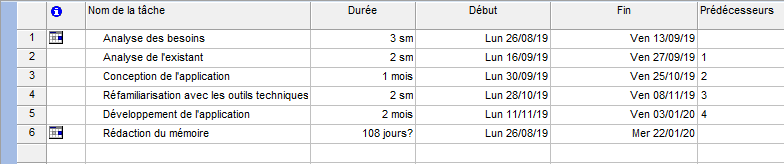
|  |  |  |
| --- | --- | --- |
| idEvenement | 🡪 | idPersonnel |
| idOpportunite | 🡪 | idClient |
| idCompte | 🡪 | idCategorie |
| referenceDevis | 🡪 | idClient |
| idContrat | 🡪 | idClient |
| idPiece | 🡪 | idClient |
| idClient | 🡪 | idPersonnel |
| idContact | 🡪 | idClient |
| idMessage | 🡪 | idPersonnel |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| referenceDevis, idService | 🡪 | coutService |

1. **Chronogramme des travaux**

Le tableau 47 représente le chronogramme des travaux.

Tableau 30. Chronogramme des travaux.



# Table des matières

[Curriculum Vitae I](#_Toc30668165)

[Sommaire général V](#_Toc30668166)

[Remerciements VII](#_Toc30668167)

[Liste des figures VIII](#_Toc30668168)

[Liste des tableaux XI](#_Toc30668169)

[Liste des abréviations ou nomenclature XIII](#_Toc30668170)

[Introduction générale 1](#_Toc30668171)

[PARTIE I : PRESENTATIONS 2](#_Toc30668172)

[Chapitre 1. Présentation de l’Ecole Nationale d’Informatique (ENI) 3](#_Toc30668173)

[1.1. Informations d’ordre général 3](#_Toc30668174)

[1.2. Missions et historique 3](#_Toc30668175)

[1.3. Organigramme institutionnel de l’ENI 4](#_Toc30668176)

[1.4. Domaines de spécialisation 6](#_Toc30668177)

[1.5. Architecture des formations pédagogiques 6](#_Toc30668178)

[1.6. Relations de l’ENI avec les entreprises et les organismes 9](#_Toc30668179)

[1.7. Partenariat au niveau international 10](#_Toc30668180)

[1.8. Débouchés professionnels des diplômés 11](#_Toc30668181)

[1.9. Ressources humaines 12](#_Toc30668182)

[Chapitre 2. Présentation de la société Inovatic 14](#_Toc30668183)

[2.1 Création et description 14](#_Toc30668184)

[2.2 Localisation 14](#_Toc30668185)

[2.3 Spécialisations 14](#_Toc30668186)

[2.4 Contacts 15](#_Toc30668187)

[Chapitre 3. Description du projet 16](#_Toc30668188)

[3.1 Formulation 16](#_Toc30668189)

[3.2 Objectif et besoins de l’utilisateur 16](#_Toc30668190)

[3.2.1 Objectif 16](#_Toc30668191)

[3.2.2 Besoins de l’utilisateur 16](#_Toc30668192)

[3.3 Moyens nécessaires à la réalisation du projet 16](#_Toc30668193)

[3.3.1 Moyens humains 16](#_Toc30668194)

[3.3.2 Moyens matériels 17](#_Toc30668195)

[3.4 Résultats attendus 17](#_Toc30668196)

[PARTIE II : ANALYSE ET CONCEPTION 18](#_Toc30668197)

[Chapitre 4. Analyse préalable 19](#_Toc30668198)

[4.1 Analyse de l’existant 19](#_Toc30668199)

[4.1.1 Organisation actuelle 19](#_Toc30668200)

[4.1.2 Inventaire des moyens matériels et logiciels 21](#_Toc30668201)

[a. Moyens matériels 21](#_Toc30668202)

[b. Moyens logiciels 22](#_Toc30668203)

[4.2 Critique de l’existant 22](#_Toc30668204)

[4.3 Proposition de solution 23](#_Toc30668205)

[4.4 Conception avant-projet 25](#_Toc30668206)

[4.4.1 Méthode de gestion de projet 25](#_Toc30668207)

[a. Etude comparative 26](#_Toc30668208)

[b. Choix et justification 27](#_Toc30668209)

[c. Présentation de SCRUM 27](#_Toc30668210)

[4.4.2 Méthode de conception 29](#_Toc30668211)

[a. Etude comparative 29](#_Toc30668212)

[b. Choix et justification 30](#_Toc30668213)

[c. Présentation de la méthode de conception 2TUP 30](#_Toc30668214)

[4.4.3 Présentation de la notation UML 32](#_Toc30668215)

[4.4.4 Outil de conception 33](#_Toc30668216)

[a. Etude comparative 33](#_Toc30668217)

[b. Choix et justification 34](#_Toc30668218)

[c. Présentation de Visual Paradigm 35](#_Toc30668219)

[4.4.5 Langage de programmation 35](#_Toc30668220)

[a. Etude comparative 35](#_Toc30668221)

[b. Choix et justification 36](#_Toc30668222)

[c. Présentation du langage de programmation PHP 37](#_Toc30668223)

[4.4.6 Système de Gestion de Base de Données (SGBD) 37](#_Toc30668224)

[a. Etude comparative 38](#_Toc30668225)

[a. Choix et justification 40](#_Toc30668226)

[b. Présentation de MySQL 40](#_Toc30668227)

[4.4.7 Cadres d’application ou Frameworks 40](#_Toc30668228)

[a. Etude comparative 40](#_Toc30668229)

[b. Choix et justification 43](#_Toc30668230)

[c. Présentation des Frameworks retenus 44](#_Toc30668231)

[4.4.8 Service web 45](#_Toc30668232)

[a. Etude comparative 46](#_Toc30668233)

[b. Choix et justification 47](#_Toc30668234)

[c. Présentation du service web REST 47](#_Toc30668235)

[4.4.9 Environnement de développement 47](#_Toc30668236)

[a. Etude comparative 48](#_Toc30668237)

[b. Choix et justification 48](#_Toc30668238)

[c. Présentation de Sublime Text 49](#_Toc30668239)

[4.4.10 Outil de test d’API 49](#_Toc30668240)

[a. Etude comparative 49](#_Toc30668241)

[b. Choix et justification 50](#_Toc30668242)

[c. Présentation de l’outil de test d’API Postman 51](#_Toc30668243)

[4.4.11 Outil de déploiement 51](#_Toc30668244)

[a. Etude comparative 51](#_Toc30668245)

[b. Choix et justification 52](#_Toc30668246)

[c. Présentation de Filezilla 52](#_Toc30668247)

[Chapitre 5. Analyse conceptuelle 53](#_Toc30668248)

[5.1 Dictionnaire des données 53](#_Toc30668249)

[5.2 Règles de gestion 55](#_Toc30668250)

[5.3 Etude préliminaire 56](#_Toc30668251)

[5.3.1 Identification des acteurs 57](#_Toc30668252)

[5.3.2 Identification des messages 57](#_Toc30668253)

[5.3.3 Diagramme de contexte dynamique 59](#_Toc30668254)

[5.3.4 Capture des besoins fonctionnels 59](#_Toc30668255)

[5.4 Spécification des besoins fonctionnels 61](#_Toc30668256)

[5.4.1 Diagramme des cas d’utilisation 61](#_Toc30668257)

[a. Formalisme 62](#_Toc30668258)

[b. Diagramme des cas d’utilisation du système 63](#_Toc30668259)

[5.4.2 Description textuelle de chaque cas d’utilisation 64](#_Toc30668260)

[5.5 Spécification des besoins techniques 74](#_Toc30668261)

[5.6 Diagramme des cas d’utilisation global du système 76](#_Toc30668262)

[5.7 Modélisation du domaine 77](#_Toc30668263)

[5.7.1 Diagramme de séquence système de chaque cas d’utilisation 77](#_Toc30668264)

[5.7.2 Modèle de domaine 89](#_Toc30668265)

[Chapitre 6. Conception détaillée 91](#_Toc30668266)

[6.1 Architecture de l’application 91](#_Toc30668267)

[6.2 Diagramme de séquence de conception de chaque cas d’utilisation 92](#_Toc30668268)

[6.3 Diagramme des classes de conception de chaque cas d’utilisation. 108](#_Toc30668269)

[6.4 Diagramme des classes global 110](#_Toc30668270)

[6.5 Diagramme des paquetages 111](#_Toc30668271)

[6.6 Diagramme de déploiement 112](#_Toc30668272)

[PARTIE III : REALISATION 113](#_Toc30668273)

[Chapitre 7. Mise en place de l’environnement de développement 114](#_Toc30668274)

[7.1 Installation et configuration des outils 114](#_Toc30668275)

[7.1.1 Installation et configuration de WampServer 114](#_Toc30668276)

[7.1.2 Installation et configuration de CodeIgniter 115](#_Toc30668277)

[7.1.3 Configuration de CodeIgniter avec l’architecture REST 115](#_Toc30668278)

[7.1.4 Installation de Postman 116](#_Toc30668279)

[7.1.5 Installation de Sublime Text 116](#_Toc30668280)

[7.2 Architecture de l’application 117](#_Toc30668281)

[Chapitre 8. Développement de l’application 119](#_Toc30668282)

[8.1 Création de la base de données 119](#_Toc30668283)

[8.2 Codage de l’application 119](#_Toc30668284)

[8.3 Présentation de l’application 121](#_Toc30668285)

[Conclusion 126](#_Toc30668286)

[Références bibliographiques VI](#_Toc30668287)

[Références webographiques VIII](#_Toc30668288)

[Glossaire XI](#_Toc30668289)

[Annexes XV](#_Toc30668290)

[Table des matières XXI](#_Toc30668291)

[Résumé XXV](#_Toc30668292)

[Abstract XXV](#_Toc30668293)

# Résumé

Dans ce présent projet, nous avons conçu et réalisé une application web destinée à gérer la relation client de la société Inovatic Toamasina Madagascar. Ce projet est né uniquement dans le but de disposer d’une masse importante d’informations concernant les clients et de leur entretenir des relations solides et durables en utilisant un outil à la fois sûr et actif. L’application a été construite en utilisant la méthode de conception 2TUP qui est pilotée par UML, l’outil de conception Visual Paradigm, le langage de programmation PHP, les frameworks CodeIgniter, Bootstrap et JQuery, le SGBDR MySQL, le service web REST, l’outil de test d’API Postman, l’éditeur de code Sublime Text et l’outil de déploiement Filezilla. L’application répond aux attentes des utilisateurs en raison de ses principales fonctionnalités telles que la gestion des clients et des contacts ainsi que l’envoi de documents commerciaux. Certes, sa plus grande importance réside sur sa capacité de gestion et de suivi de statut des activités commerciales exercées au sein de la société.

Mots-clés : 2TUP, activités commerciales, application, conception, relation client, UML.

# Abstract

In this present project, we designed and produced a web application intended to manage Inovatic customer relationship. This project was born solely for the purpose of having a large mass of information concerning the customers and of maintaining them solid and lasting relationships by using a tool that is both safe and active. The application was built using 2TUP design method which is driven by UML, Visual Paradigm as design tool, PHP as programming language, CodeIgniter, JQuery and Bootstrap as frameworks, MySQL as database system management, REST as WebService, Postman as API test tool, Sublime Text as editor and Filezilla as deployment tool. The application meets user expectation due to its main functionalities such as customer and contact management as well as sending business documents. Of course, its greteast importance lies in its ability to manage and monitor the status of commercial activities carried out within the company.

Keywords: 2TUP, application, commercial activities, customer relationship, design, UML