



**TRAVAUX PRATIQUES MASTER 2021-2022**

Mention : Informatique

Parcours : Informatique Générale  
**Intitulé :**

**GESTION DES NOTES**

**Présenté par  :**

Mr Christin

Mr Kristian

Mr Toky

**Membre du jury**

* Président : Monsieur RATIARSON Venot,

Maître de Conférences

CURRICULUM VITAE



JEAN Christin

Age : 25 ans

Nationalité : Malagasy

Lot : 0202E0514 Antanimasaja Mahajanga

Téléphone : 0349487350/0325877717

e-mail : *[jeanchristin.ig.network@gmail.com](mailto:jeanchristin.ig.network@gmail.com" \t "/home/jckristin/Documents\\x/_top)*

**WEB DEVELOPPEUR**

**DIPLÔMES OBTENUS**

2015 : Diplôme de Technicien Supérieur (DTS) en Technologie Navale, Institut Supérieur de Technologie D’Antsiranana (ISTD) à la ville de Diégo Suarez

2013 : Baccalauréat Scientifique série C, Lycée PHILIBERT TSIRANANA Mahajanga

**FORMATIONS**

2018 – 2019 : Formation License niveau 2 en informatique générale(IG), Ecole Nationale d’Informatique Fianarantsoa

2017 – 2018 : Formation License niveau 1 en informatique générale(IG),

Ecole Nationale d’Informatique Fianarantsoa

2013 – 2015 : Formation technicien supérieur en Technologie Navale,

Institut Supérieur de Technologie d’Antsiranana (ISTD)

à la ville de Diego Suarez

**EXPERIENCES PROFESSIONNELLES**

Fin 2019 : Stage de fin d’année universitaire au DREN ETP à Ihorombe

2018 – 2019 : Projet de configuration de routeur CISCO mode console

2018 : Plusieurs travaux pratiques sont effectués sur les modules suivants :

Système LINUX , WINDOWS , Langage C et C++

Juin 2016 : Stage de perfectionnement au SECREN.SA à Antsiranana

2015 : Stage de mémoire – AMBININA Sarlu Mahajanga, stage de 3 mois

2014 : Stage d’ouvrier – S.E.M.S carénage Mahajanga, stage de 2 mois

**CONNAISSANCES LINGUISTIQUES**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Parler | | | | Ecrire | | | | Comprendre | | | | Lire | | | |
| LANGUES | TB | B | AS | P | TB | B | AS | P | TB | B | AS | P | TB | B | AS | P |
| FRANCAIS |  |  | \* |  |  | \* |  |  |  | \* |  |  |  | \* |  |  |
| ANGLAIS |  |  | \* |  |  |  | \* |  |  |  | \* |  |  |  | \* |  |

**TB** : très bien **B** : bien **AS** : assez bien **P** : passable

**CONNAISSANCES INFORMATIQUES ET LOGICIELS PROFESSIONNELS**

Cisco Packet Tracer  : logiciel de simulation de réseau

GNS3  : logiciel de simulation de réseau

Packs offices : Microsoft offices

Gimp et Inkscape : logiciels de dessin assiter par ordinateur comme photoshop

**CENTRES D’INTERETS**

Sports  : Natation, football

Association  : Mpanazava sy Tily Eto Madagasikara (MTEM),

Réseau d’Honnêteté et d’Intégrité Fianarantsoa (RHI)

Divers : Dynamique, Responsable, Courageux, Intègre

Je déclare sur l’honneur que les renseignements qui me concerne ci-dessus sont sincères et exactes.

**SOMMAIRE GENERALE**

[CURRICULUM VITAE I](#_Toc5274665)

[SOMMAIRE GENERALE III](#_Toc703647502)

[REMERCIEMENTS IV](#_Toc1544972400)

[LISTE DES FIGURES V](#_Toc139786330)

[LISTE DES TABLEAUX VII](#_Toc1948176462)

[LISTE DES ABREVIATIONS OU NOMENCLATURE VIII](#_Toc191210348)

[INTRODUCTION GENERALE 1](#_Toc1712731266)

[PARTIE I : PRESENTATIONS 2](#_Toc149407158)

[Chapitre 1. Présentation de l’Ecole Nationale d’Informatique (ENI) 3](#_Toc689839706)

[Chapitre 2. Présentation de la Direction Territoriale BIANCO Mahajanga 16](#_Toc337589212)

[Chapitre 3. Description du projet 18](#_Toc2017196517)

[PARTIE II : ANALYSE ET CONCEPTION 22](#_Toc170855347)

[Chapitre 4. Analyse Préalable 23](#_Toc519258483)

[Chapitre 5. Analyse conceptuelle 38](#_Toc1453420919)

[Chapitre 6. Conception détaillée (UML) 62](#_Toc779280820)

[PARTIE III : REALISATION 84](#_Toc617757407)

[Chapitre 7. Mise en place de l'environnement de développement ....................................85](#_Toc484871112)

[Chapitre 8. Développement de l'application 89](#_Toc811070899)

[CONCLUSION GENERALE 94](#_Toc400931835)

[REFERENCES BIBLIOGRAPHIQUES X](#_Toc1781742959)

[REFERENCES WEBOGRAPHIQUES XI](#_Toc826293346)

[GLOSSAIRES XII](#_Toc2022472942)

[TABLE DES MATIERES XIII](#_Toc475662738)

[RESUME XVI](#_Toc1773619195)

[ABSTRACT XVI](#_Toc1996774910)

REMERCIEMENTS

Je tiens à remercier Dieu tout puissant de m’avoir donné la force, la volonté, la connaissance et la compétence requises durant ce projet de mémoire de licence professionnelle en informatique.

Ce projet est le fruit de nos trois années d’études et de recherches au sein de l’Ecole Nationale d’informatique (ENI) et je m’adresse mes sincères remerciements à tous ceux qui m’ont fait ce que je suis aujourd’hui, en l’occurrence les professeurs, en particulier :

* Monsieur RAFAMANTANANTSOA Fontaine, Professeur, Président de l’Université de Fianarantsoa, de nous avoir reconnu comme étant un étudiant de l’université où il préside;
* Monsieur RAMAMONJISOA Bertin Olivier, Professeur Titulaire, Directeur de l’Ecole Nationale d’Informatique, qui supervise l’établissement de l’ENI;

LISTE DES FIGURES

[Figure 1. Organigramme de l’Ecole Nationale d’Informatique 6](#_Toc2122437821)

[Figure 2. Organigramme de la direction territoriale Bianco Mahajanga 17](#_Toc561675643)

[Figure 3. Déroulement de la réalisation du projet 21](#_Toc693746153)

[Figure 4. Flux des données du service éducation, prévention et investigation 24](#_Toc1250351329)

[Figure 5. Quatorze (14) types de diagrammes d’UML 2.2 28](#_Toc1087266902)

[Figure 6. Le processus de développement en Y 29](#_Toc1179030360)

[Figure 7. Logo de spring boot 31](#_Toc1894887806)

[Figure 8. Architecture du Spring Boot 31](#_Toc1869037380)

[Figure 9. Logo du Thymeleaf 33](#_Toc270683094)

[Figure 10. Logo de MariaDB 35](#_Toc1840388692)

[Figure 11. Architecture 3 niveaux 37](#_Toc1878926419)

[Figure 12. Diagramme de contexte dynamique du système de BIANCO 45](#_Toc1688134554)

[Figure 13. Diagramme de cas d’utilisation du système BMSI 47](#_Toc1364729782)

[Figure 14. Diagramme du domaine de modélisation du système 61](#_Toc232303884)

[Figure 15. Architercture MVC 62](#_Toc623218132)

[Figure 16. Diagramme de séquence de conception pour gérer les services 63](#_Toc1661855074)

[Figure 17. Diagramme de séquence de conception pour suivre le tableau de bord 63](#_Toc1321805785)

[Figure 18. Diagramme de séquence de conception pour planifier le PTA 64](#_Toc569539193)

[Figure 19. Diagramme de séquence de conception pour planifier le POT 65](#_Toc1892101922)

[Figure 20. Diagramme de séquence de conception pour gérer la quantité 66](#_Toc2119191368)

[Figure 21. Diagramme de séquence de conception pour gérer les réalisations 67](#_Toc2053831044)

[Figure 22. Diagramme de séquence de conception pour gérer la prévention 67](#_Toc1813147611)

[Figure 23. Diagramme de séquence de conception pour gérer la réalisation éducation 68](#_Toc1847295387)

[Figure 24. Diagramme de séquence de conception pour gérer la réalisation communication 69](#_Toc594613679)

[Figure 25. Diagramme de séquence de conception pour gérer la réalisation financière 70](#_Toc1011737335)

[Figure 26. Diagramme de séquence de conception pour gérer la réalisation MOFF 71](#_Toc1784664736)

[Figure 27. Diagramme de séquence de conception pour consulter le rapport 72](#_Toc439124602)

[Figure 28. Diagramme de séquence de conception pour suivre les dossiers 72](#_Toc1358137474)

[Figure 29. Diagramme de séquence de conception pour traiter un dossier 73](#_Toc1114468238)

[Figure 30. Diagramme de classe de conception pour gérer les services 74](#_Toc99809814)

[Figure 31. Diagramme de classe de conception pour planifier le PTA 74](#_Toc297243163)

[Figure 32. Diagramme de classe de conception pour planifier le POT 75](#_Toc1089422411)

[Figure 33. Diagramme de classe de conception pour gérer la quantité 75](#_Toc661485457)

[Figure 34. Diagramme de classe de conception pour gérer les réalisations 76](#_Toc990989316)

[Figure 35. Diagramme de classe de conception pour gérer la réalisation prévention 76](#_Toc192290092)

[Figure 36. Diagramme de classe de conception pour gérer la réalisation éducation. 77](#_Toc1748752359)

[Figure 37. Diagramme de classe de conception pour gérer la réalisation financière 77](#_Toc22536028)

[Figure 38. Diagramme de classe de conception pour gérer la réalisation MOFF 78](#_Toc2087177899)

[Figure 39. Diagramme de classe de conception pour gérer le rapport. 78](#_Toc1470306091)

[Figure 40. Diagramme de classe de conception pour suivre le tableau de bord 79](#_Toc293219122)

[Figure 41. Diagramme de classe de conception pour suivre les dossiers. 79](#_Toc1780082943)

[Figure 42. Diagramme de classe de conception pour traiter un dossier. 80](#_Toc1201748862)

[Figure 43. Classe de conception global 81](#_Toc1981353676)

[Figure 44. Diagramme de paquetages de l’application 82](#_Toc997329078)

[Figure 45. Diagramme de déploiement de l’application 83](#_Toc1434052747)

[Figure 46. Installation d’OpenJdk-11-jdk sur un terminal linux 85](#_Toc457088160)

[Figure 47. Installation et la configuration de l’environnement Eclipse IDE 86](#_Toc511700504)

[Figure 48. Configuration du chemin d’installation 86](#_Toc608374884)

[Figure 49. Configuration et installation du code source spring boot 87](#_Toc1026627353)

[Figure 50. Installation de mariadb-server et configuration de la base de données 88](#_Toc256318778)

[Figure 51. Architecture de l’application développer en framework spring boot 88](#_Toc580082604)

[Figure 52. Création de la base de données «bmsi» 89](#_Toc932974749)

[Figure 53. Configuration de la base de données Mariadb avec spring boot 89](#_Toc2069466389)

[Figure 54. Tables générées la par l’application dans la base de données «bmsi» 90](#_Toc279894344)

[Figure 55. Code du modèle de l’objet User avec le framework spring boot 90](#_Toc1527588428)

[Figure 56. Code de la partie contrôleur de login sur spring boot 91](#_Toc933720076)

[Figure 57. Code pour la partie vue du login avec template thymeleaf 91](#_Toc2064559080)

[Figure 58. Formulaire d’authentification de l’application 92](#_Toc1966713030)

[Figure 59. Tableau de bord du service STEP du bianco Mahajanga 92](#_Toc144373902)

[Figure 60. Tableau de bord du service STI du bianco Mahajanga 93](#_Toc1031543671)

[Figure 61. Gestion des utilisateurs du sytème d’information du Bianco Mahajanga 93](#_Toc2066522844)

**LISTE DES TABLEAUX**

[Tableau 1. Organisation du système de formation pédagogique de l’Ecole 7](#_Toc1628499288)

[Tableau 2. Architecture des études correspondant au système LMD 8](#_Toc181669271)

[Tableau 3. Liste des formations existantes à l’ENI 9](#_Toc1583708050)

[Tableau 4. Débouchés professionnels éventuels des diplômés 14](#_Toc608425473)

[Tableau 5. Caractéristiques de l’ordinateur utilisé lors de la réalisation du projet 19](#_Toc98498923)

[Tableau 6. Détails des packages utilisés lors de la réalisation du projet 19](#_Toc1178933840)

[Tableau 7. Liste des matériels du service éducation, prévention et investigation 24](#_Toc31790079)

[Tableau 8. Liste des logiciels utilisés par le service éducation, prévention et investigation 25](#_Toc1930658076)

[Tableau 9. Comparaison de la modélisation avec MERISE et UML 27](#_Toc1296871847)

[Tableau 10. Comparaison de framework Spring Boot et Symfony 30](#_Toc15222447)

[Tableau 11. Comparaison de la base de données MariaDB et Oracle 34](#_Toc1621541107)

[Tableau 12. Comparaison de l’architecture à 2 niveaux et de l’architecture à 3 niveaux 36](#_Toc841403427)

[Tableau 13. Dictionnaire des données 38](#_Toc947325848)

[Tableau 14. Messages échangés entre les acteurs et le système 41](#_Toc2121785616)

[Tableau 15. Identification des cas d’utilisation 45](#_Toc954326412)

[Tableau 16. Priorisation des cas d’utilisation 48](#_Toc289257664)

[Tableau 17. Diagramme des séquences système pour chaque cas d’utilisation 50](#_Toc20645285)

[Tableau 18. Plugins nécessaires pour dévelloper l’application avec Eclipse 87](#_Toc441051271)

LISTE DES ABREVIATIONS OU NOMENCLATURE

2TUP: 2 Tracks Unified Process

AN: Alpha-numérique

BD : Base de Données

BIANCO : Bureau Indépendant Anti-Corruption

BMSI : Bianco Mahajanga Système d’Information

BTS : Brevet de Téchnicien Supérieur

CARI : Colloque Africain sur la Recherche en Informatique

CITEF : Conférence Internationale des Ecoles de formation d’Ingénieurs et Technicien d’Expression Française

CNH : Commission Nationale d’Habilitation

CPTE : Conseiller Principale Territorial de l’Education

CPTP : Conseiller Principale Territorial de la Prévention

CPU : Central Processing Unit

D : Date

DEA : Diplôme d’Etudes Approfondie

DTS : Diplôme de Téchnicien Supérieur

DUT : Diplôme Universitaire de Téchnicien

HDD : Hard Drive Disk

HTML: HyperText Markup Language

IDE : Integrated development environment

INPG : Institut National Polytechnique de Grenoble

Integer : Nombre entier

JSP : Java Server Page

LCC : Lutte Contre la Corruption

LMD : Licence-Master-Doctoral

Long : Nombre entier

MESupRES : Ministères de l’Enseignement Supérieur et de la Recherche Scientifique

MVC: Model View Controller

MVP: Model View Presenter

MVT: Model View Template

MVVM: Model View View Model

MySQL: My Structured Query Language

N: Numérique

OS : Operating System

PAC : Pôles Anti-Corruption

PILCC : Politique Interne sur la Lutte Contre la Corruption

POT : Plan Opérationnel du Travail

PPM : Page Per Minute

PTA : Plan de Travail Annuel

PTH : Plan de Travail Hebdomadaire

RAM : Random Access Memory

SGBDR : Système de Gestion de Base de Données Relationnelles

SI : Système d’Information

SNLCC : Stratégie Nationale de Lutte Contre la Corruption

SQL: Structured Query Language

STEP : Service Territorial d’Education et de Prévention

STI : Service Territorial d’Investigation

String : Chaine de caractère

TIC : Technologies de l’Information et de la Communication

UML: Unified Modeling Language

USB : Universal Serial Bus

Wifi : Wireless Fidelity

INTRODUCTION GENERALE

Le système d’information (SI) est un concept qui permet de collecter, stocker,

traiter et distribuer les ressources de l’entité concernée de façon automatisées afin d’améliorer l’efficacité du fonctionnement interne de l’entité. [1]

1. Description du projet
2. Formulation

Début de projet le 17 décembre 2021 à l’ENI Fianarantsoa

1. Objectif et besoins de l’utilisateur

Gérer les notes des étudiants à fin de facilités la délivrance de résultat.

1. Moyens nécessaires à la réalisation du projet
2. Moyens Humains

Pour mener bien ce projet, les personnes suivantes sont nécessaire:

* Mr Venot qui est le maître d’ouvrage
* 3 informaticiens spécialiser en développement des web

1. Moyens matériels et logiciels

Pour pouvoir réaliser ce projet, nous avons utilisé ces matériels :

* 3 ordinateur portable

Le tableau 5 présente les caractéristiques de l’ordinateur utilisé à la réalisation du projet.

1. Caractéristiques de l’ordinateur utilisé lors de la réalisation du projet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modèle | CPU | Mémoire vive  (RAM) | Disque dur  (HDD) | Système d’exploitation (OS) |
| Toshiba Satellite Pro C660-13C | Intel Core i3  M 380 @  4x 2.533GHz | 6Go | 500Go | Linux 4.19.0-16-amd64  Debian 10 buster |

* des connexions wifi, pour télécharger les technologies,

les documents nécessaires à la réalisation du projet.

* une usb modem, pour permettre d’avoir

une connexion internet à domicile pour les recherches particulières

les logiciels (ou Packages) utilisés sont :

* Framework ExpressJS du NodeJs
* Vs Code : environnement de développement pour coder facilement
* Mysql : pour faire persister les données dans la base donnée avec le langage SQL
* PhpMyAdmin : pour gérer la base de données Mysql sur un navigateur web
* Firefox et Google Chrome : pour naviguer sur internet et tester l’application web
* GanttProject : pour organiser les tâches à exécuter pendant le réalisation du projet
* UMLet : pour réaliser la conception du système d’information
* Notes : pour noter des tâches spécifiques à réaliser
* Github: pour versionner le développement en parallèle

Le tableau 6 présente les détails des packages utilisés lors de la réalisation du projet.

1. Détails des packages utilisés lors de la réalisation du projet

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de package | Version | Propriétaire | Licences |
| Express | 4.7 |  | Free and Open Source |
| Vs Code |  |  | Free and open source |
| Mysql | 10.3.27 | Oracle | Free and Open Source |
| PhpMyAdmin | 5.1.0 | Free Software Foundation | Free and Open Source,  GNU GPL |
| Firefox ESR | 78.11.0esr (64-bit) | Mozilla | Mozilla Public License,  Free and Open Source |
| Google Chrome | 91.0.4472.77 (Official Build) (64-bit) | GOOGLE | Free and Open Source |
| Github |  |  | free |
| GanttProject | 3.0.3004 | BarD Software sro | GNU GPL |
| UMLet | 14.3.0 | com.umlet | Free and Open Source |
| Notes | 1.8.1 | The Xfce development team | Free,  GNU GPL |

1. Résultats attendus

Les résultats attendus du projet sont de fournir une application qui permet :

* de gérer les bulletins des étudiants
* de gérer les classement des étudiants pa odre de mérite

1. Chronogramme des activités

Nous allons utiliser le logiciel Ganttproject pour planifier la réalisation du projet système d’information

La figure 3 présente le déroulement de la réalisation du projet.

1. Déroulement de la réalisation du projet
2. ANALYSE ET CONCEPTION
3. Analyse Préalable
4. Analyse de l'existant
5. Organisation actuelle

Les résultats des notes des étudiants sont calculer sur un excel.

Les flux des données dans les trois (03) services sont présentés dans la figure 4.

Enseignants

Scolarité

reclamation

Notes

Etudiants

Résultat

reclamation

1. Flux des données
2. lnventaire des moyens matériels et logiciels

Seul les matériels utilisés dans le service éducation, prévention et investigation fait l’objet des matériels cités dans le tableau 7.

1. Liste des matériels du service éducation, prévention et investigation

|  |  |  |  |
| --- | --- | --- | --- |
| Matériels | Modèle | Caractéristiques | Nombre |
| Ordinateurs  portables | HP Stream  14 Laptop | CPU : 4 x 2.8 GHz  RAM : 4 Go DDR4  OS : Windows 10 | 1 |
| Ordinateurs  bureau | Dell | CPU : core 2 Duo  RAM : 2Go  OS : Windows 7 | 2 |
| Imprimante | Bizhub 306 | - Impression noir et blanc  - vitesse A4/A3 : 30/11 ppm  - supporte l’impression avec ethernet | 1 |

Le tableau 8 présente les logiciels utilisés dans le service scolarité

1. Liste des logiciels utilisés par le service scolarité

|  |  |  |
| --- | --- | --- |
| Nom du logiciel | Description | Licence |
| Packs Microsoft Office 2013 et 2016 | * Le microsoft word est utilisé pour saisir les rapports d’activité * Le microsoft excel est utilisé pour plannifier les activités à faire et de faire une suivi- évalution * Le microsoft powerpoint est utilisé lors de présentation lors d’une réunion | Payant |
| Google Chrome  Firefox | * Pour navigeur sur internet et consulter les emails | Gratuit |

1. Critique de l’existant

Le système utilisé par le service scolarité est penible et sensible aux erreurs vue l’air encombrant offrant par le travail sur le fichier excel.

Les personnels de ces services trouvent qu’ il est avantageux pour eux ainsi que pour leur travail de pouvoir manipuler leurs activités quotidiennes dans un système d’information qui répond à leur besoins vu que leur tâches sont assez répétitives chaque année. Et cela apportera un grand dynamisme dans leur mode de travail car ce système d’information va leur permet de dématérialiser et de sécuriser la plupart des documents de l’établissement et de faciliter le déplacement des flux de données qui circulent à l’intérieur. C’est-à-dire, plus la circulation des données est facile, plus elle sera rapide aux opérationnels et aux décideurs de mettre en oeuvre les activités définies et d’effectuer le suivi-évalution de ces activités.

Et pour favoriser et faciliter ces déplacements des flux de données, il est nécessaire d’ajouter un serveur physique à la scolarité pour centraliser les informations et les ressources à partager avec les personnels des services concernés.

1. Conception avant projet
2. Proposition de solutions

Pour satisfaire les besoins de la scolarité, quatre (04) solutions s’offrent à nous :

**Solution 1**: Engager un prestataire à concevoir une application desktop compatible avec les ordinateurs utilisés.

**Solution 2**: Engager un prestataire à concevoir une application web et centraliser les ressources.

**Solution 3**: Engager un stagiaire à concevoir une application desktop compatible avec les ordinateurs utilisés.

**Solution 4**: Engager un stagiaire à concevoir une application web et centraliser les ressources.

Avant de choisir la solution adéquate pour le système d’information, nous allons voir d’abord les contraintes et les exigences qui seront confrontées pour réaliser ce projet. [3]

Les contraintes et les exigences pour la réalisation de l’application sont :

* le manque budget
* l’expérience du développeur
* la durée de réalisation
* le nombre de maître d’œuvre
* la disponibilité des ressources
* le choix de technologie adéquat
* l’interopérabilité du système
* la sécurité des données

Ainsi pour complémenter les contraintes et les exigences du projet, les solutions suivantes doivent être adopter :

* du point de vue modélisation, nous allons choisir une modélisation orientée objet avec une approche itérative et incrémentale pour minimiser les risques comme la disponibilité des ressources nécessaires à la réalisation , ainsi que la compatibilité et l’intégration des technologies utilisées.
* du point de vue réalisation, nous avons opté pour un framework des langages de programmation. Celui-ci est dans le but de minimiser le temps de développement et pour bien sécuriser l’application.
* du point de vue déploiement, il est préférable d’utiliser un client léger pour éviter l’incompatibilité du système vis-à-vis des systèmes d’exploitations utilisés par les utilisateurs.
* du point de vue architecture, nous allons installer une architecture 3-tiers avec SGBDR

Vue les contraintes et les exigences ainsi que la complémentation qu’on vient de voir, la solution 4 est la solution adéquate pour le projet puisque celui-la répond bien à nos exigences ainsi que les ressources seront centralisées.

1. Comparaison et choix de modélisation

Durant notre cursus universitaire, nous n’avons appris que deux (02) façons de modélisation de conception dont avec la méthode de conception merise qui est la plus utilisée par les européens et avec le langage de modélisation UML, qui est le plus utilisé par les anglophones.

Le tableau 9 compare la modélisation avec MERISE et UML.

1. Comparaison de la modélisation avec MERISE et UML [4]

|  |  |  |
| --- | --- | --- |
| Point  de comparaison | MERISE | UML |
| Type | méthode de modélisation | langage de modélisation |
| Adéquation à la modélisation orienté objet | moins adéquat | plus adéquat |
| Niveaux d’abstraction | moins de notions  et de diagrammes | dispose plusieurs notions  et des diagrammes |
| Caractéristique | représente une démarche d’informtisation | facilite la représentation et  la compréhension de solutions objet |
| Nombre de processus de développement | un seul | plusieurs |
| Domaine d'application | entreprise, organisation | tous les domaines |
| Extensibilité  de la conception | difficile | dépend du cas d’utilisation |
| Outil gratuit | assez | plusieurs |

Après avoir comparer la méthode merise et le langage UML, en s’appuyant sur les solutions pour développer l’application, le choix mieux adapté pour notre application est la modélisation avec le langage de modélisation UML.

1. Présentation du langage modélisation UML

UML est un langage unifié pour la modélisation objet. Et d’après l’OMG, c’est un langage visuel dédié à la spécification,à la construction et à la documentation des artefacts d’un système logiciel. Son version 1.0 a vu le jour en Janvier 1997 et la dernière version est le 2.5 qui a été standardisée le mois d’Octobre 2012.[5]

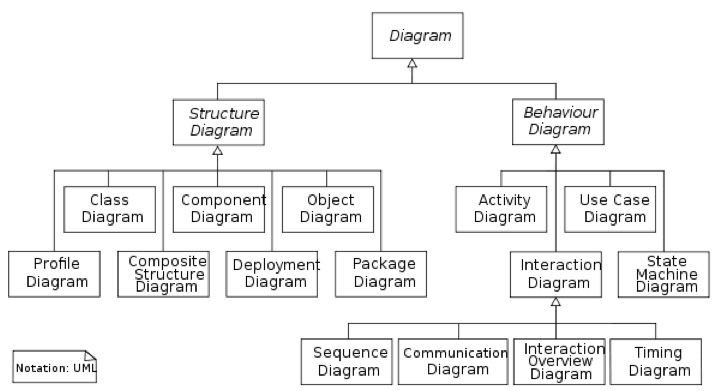
1. Caractéristiques d’UML [2]

* un travail d'expert
* utilise l’approche orientée objet normalisé, riche
* formel : sa notation limite les ambiguïté et les incompréhensions
* langage ouvert : indépendant du langage de programmation
* domaine d'application : permet de modéliser n'importe quel système
* supporté par plusieurs outils (AGL) : Objecteering, Open tools, Rational Rose, PowerAMC, WinDesign, ...

1. Les diagrammes UML

L’UML 2.2 comporte 14 types de diagrammes qui sont subdivisés en deux (02) catégories, il y a le diagramme de structure qui est utilisé pour la partie statique de la modélisation et le diagramme de comportement pour la partie dynamique.

La figure 5 présente les quatorzes (14) diagrammes de L’UML 2.2.



1. Quatorze (14) types de diagrammes d’UML 2.2 [2]

L’utilisation de ces diagrammes dépend de l’avancement de la modélisation et ils nous aident à clarifier nos conceptions.

1. Les processus unifiés [6]

UML comporte plusieurs types de processus unifiés:

* RUP ou Rational Unified Process
* XP ou eXtreme Programming
* 2TUP ou Two Tracks Unified Process

Vue qu’on n’a pas encore eu des expériences sur le processus RUP et XP donc nous allons utiliser le processus 2TUP.

1. Présentation du processus 2TUP

le processus 2TUP (2 Track Unified Process) est un processus qui suit deux chemins dont le branchement « fonctionnel » et le branchement « technique », qui correspondent aux deux axes de changement imposés au système informatique et il se fusionne à la phase de conception préliminaire pour associer les 2 branches en donnant la forme Y. [1]

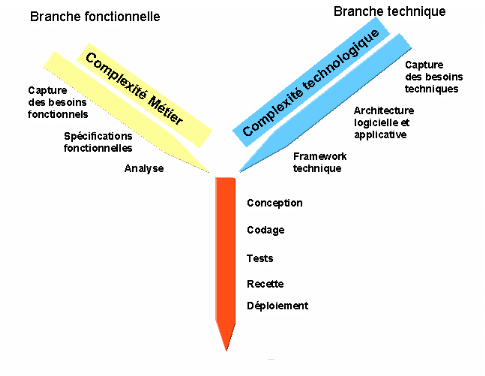
Caractéristique du 2TUP :

- le traitement des besoins fonctionnels et techniques sont séparés

- donne une place importante à la technologie et à la gestion du risque

- définit les profils des intervenants, les livrables, les plannings, les prototypes

La figure 6 présente le développement du processus Y.



1. Le processus de développement en Y [7]
2. Comparaison et choix des frameworks

Il est important de préciser à l’avance que l’application que nous allons élaborer est une application de type professionnel. Donc parmi les frameworks professionnels existants, nous allons choisir entre « Spring Boot » et « Symfony».

Le tableau 10 compare le framework Spring Boot et le framework Symfony.

1. Comparaison de framework Spring Boot et Symfony

|  |  |  |
| --- | --- | --- |
| **Point**  **de comparaison** | **Swing via NET URL ou ApacheClient** | **ExpressJS** |
| Langage  de programmation | JAVA | Javascript |
| Utilisation | Utilisé pour le développement  d’application web, Desktopet mobile | Utilisé pour le développement  d’application web seulement |
| Performance | Robuste, Rapide à l’exécution sauf  au démarrage | Rapide |
| Installation  et configuration | simple et auto-configuration |  |
| Platform | cross-platform | cross-platform |
| POO | Obligatoire | Optionnel |
| Extensibilité | Ouverte à l’extension et  fermée à la modification |  |
| Moteur de template | Thymeleaf, Freemaker | Pug |
| Documentation | Large | Large |

Après avoir comparé les deux (02) technologies, nous avons vu qu’ils ont une place important dans le développement de l’application web donc pour la partie frontend nous allons utiliser Swing via NET URL ou ApacheClient et pour le backend nous allons opté pour l’expressJS.

1. Présentation du Swing via NET URL ou ApacheClient
2. Formulation
3. Architecture

1. Avantage
2. Fonctionnement
3. Dépendances importants
4. Comparaison et choix de SGBD

Les bases de données sont des concepts technologiques, qui nous permettent de faire persister les données du client.

Aujourd’hui, on peut trouver plusieurs types de bases de données. On peut les classifier en deux catégories dont le SQL et le NoSQL. Et ils sont des langages standardisés pour stocker, manipuler et retirer des données d’une base de données.

Parmi le SQL, on trouve MariaDB, MySQL, Oracle, SQL Server, MS Access, Postgres, SQLite etc... Et parmi le NoSQL, on trouve CouchDB, Cassandra, HBase, Redis, Neo4J.

Vue le nombreux choix qu’on a sur la base donnée de notre application, nous allons pencher sur ce qu’on a déjà expérimenté durant notre cursus de formation.

Le tableau 11 compare les deux (02) bases de données que nous avons expérimenté.

1. Comparaison de la base de données MariaDB et Oracle [15]

|  |  |  |
| --- | --- | --- |
| Point de comparaison | Mysql | Postgresql |
| Propriétaire | Oracle |  |
| Plateforme | Multi-plateforme | Multi-plateforme |
| Domaine d’utilisation |  |  |
| Répartition de bases de données sur des serveurs |  |  |
| Point de sauvegarde |  |  |
| Développement |  |  |
| Installation |  |  |
| Licence |  |  |
| Perfomance |  |  |
| Documentation |  |  |

En

1. Présentation de base de données MariaDB
2. Formulation

MariaDB est un système de gestion de base de données relationnelle ([SGBDR](https://www.lemagit.fr/definition/Systeme-de-gestion-de-base-de-donnees))

utilisé par des nombreuses grandes entreprises, des distributeurs Linux etc.

Parmi les organisations qui utilisent MariaDB, citons Google, Wikipedia, Archlinux, RedHat, CentOS et Fedora.[15]

MariaDB est développé par l’entreprise MariaDB Corporation et il est open source.

La figure 10 montre le logo de MariaDB



1. Logo de MariaDB [16]
2. Caractéristiques du MariaBD [15]

* Offre une compatibilité ascendante
* Logiciels Open Source
* Il est basé sur la version de la communauté MySQL
* Nouveau moteur de stockage (PBXT, XtraDB, Maria, FederatedX)

1. Interopérabilité et client-serveur

Il y a plusieurs types d’architectures client-serveur qu’on peut choisir pour le déploiement de notre système. Mais chacun d’eux présente leur propre avantage et leur faiblesse.

Le tableau 12 compare deux types d’architecture du système.

1. Comparaison de l’architecture à 2 niveaux et de l’architecture à 3 niveaux

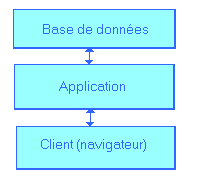
|  |  |  |
| --- | --- | --- |
| Point de comparaison | Architecture à 2 niveaux | Architecture à 3 niveaux |
| Structure | - Présentation et Application du coté client  - Base de données du coté serveur | - Présentation du coté client  - Application sur un serveur d’application  - Base de données du coté serveur |
| Type de client | - Client lourd | - Client léger |
| Scalalabilité  ou Extensibilité | - Une modification dans l’architecture entraine la modification sur le niveau | - Chaque niveau peut évoluer horizontalement |
| Performance | - Plus lente | - Plus rapide |
| Sécurité | - Moins sécurisé puisque le client communique directement à la base  de donnée | - Sécurisé puisque les ressources sont centralisées et le client ne communique qu’avec le serveur de traitement |
| Maintenance | Difficile | Facile |

Vue les contraintes et la comparaison qu’on vient de faire, il est raisonnable d’opter pour l’architecture à niveaux pour gagner plus de performance lors de traitement, de sécurité et assurer la facilité de maintenance et la compatibilité des systèmes d’exploitation et le système.

1. Présentation de l’architecture à 3 niveaux
2. Formulation

L’architecture à 3 niveaux (3 tiers) ou l’architecture distribuée est la 2è génération de l’architecture client-serveur et il est le mieux adapter pour les applications web vue qu’il sépare la couche présentation de la couche métier et de la couche donnée.

La figure 11 présente communication utilisée en architecture à 3 niveaux



1. Architecture 3 niveaux [17]
2. Avantage de l’architecture 3 niveaux [17]

Les avantages de l’architecture 3 niveaux sont :

* Toutes les données sont centralisées sur un seul serveur, physique ou virtuel, ce qui simplifie les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels.
* La complexité du traitement et la puissance de calculs sont à la charge du ou des serveurs, les utilisateurs utilisant simplement un client léger sur un ordinateur terminal qui peut être simplifié au maximum.
* Maintenance matériel minimisée.
* Grande vélocité sur des grands volumes de données et de traitements.

1. Inconvénients de l’architecture 3 niveaux [17]

Les avantages de l’architecture 3 niveaux sont :

* Si l'ordinateur central n'est plus disponible, plus aucun des clients ne fonctionne
* Les coûts de mise en place et de maintenance peuvent être élevés.
* Les clients ne peuvent communiquer directement entre eux, entrainant une concentration des flux sur l'ordinateur central.

1. Analyse conceptuelle

La conception est la phase la plus critique dans le processus de développement de logiciel. Pour avoir une bonne conception, il faut bien choisir le bon processus de modélisation. Et pour notre cas, nous allons choisir le processus 2TUP, puisque celui-là est adapté à notre projet.

1. Dictionnaire des données

Le dictionnaire des données est une collection des métadonnées ou des données de références, nécessaire à la conception d'une base de données relationnelle.

Le tableau 13 représente le dictionnaire des données.

1. Dictionnaire des données

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom**  **de la rubrique** | **Description** | **Domaine**  **de valeur** | | **Observation** |
| **Type** | **Taille** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Règles de gestion

Les règles de gestion sont des exigences imposées sur notre application afin de mieux gérer les contraintes liées au système.

Les règles de gestion sont :

**RG1** : Le système d’information est destiné pour le service scolarité

**RG2** : L’accès au système dépend du rôle de l’utilisateur

**RG3** :

**RG4** : Seul l’administrateur qui gèrent les utilisateurs

**RG5** : Le chef de service ont le droit de consulter les travaux dans son service en mode lecture seulement

**RG6** : Le directeur a le pouvoir de consulter le travail du services en mode lecture

**RG7** : Le code matière et numéro étudian de doit être unique

1. Représentation et spécification des besoins
2. Recueil des besoins (pré-étude)
3. Identifier les acteurs

Les acteurs externes du système d’information sont :

* Directeur Général (DG)
* Chef de service de scolarité (chef SCO)
* Secretaire
* Enseignant
* Etudiant

1. Identifier les messages

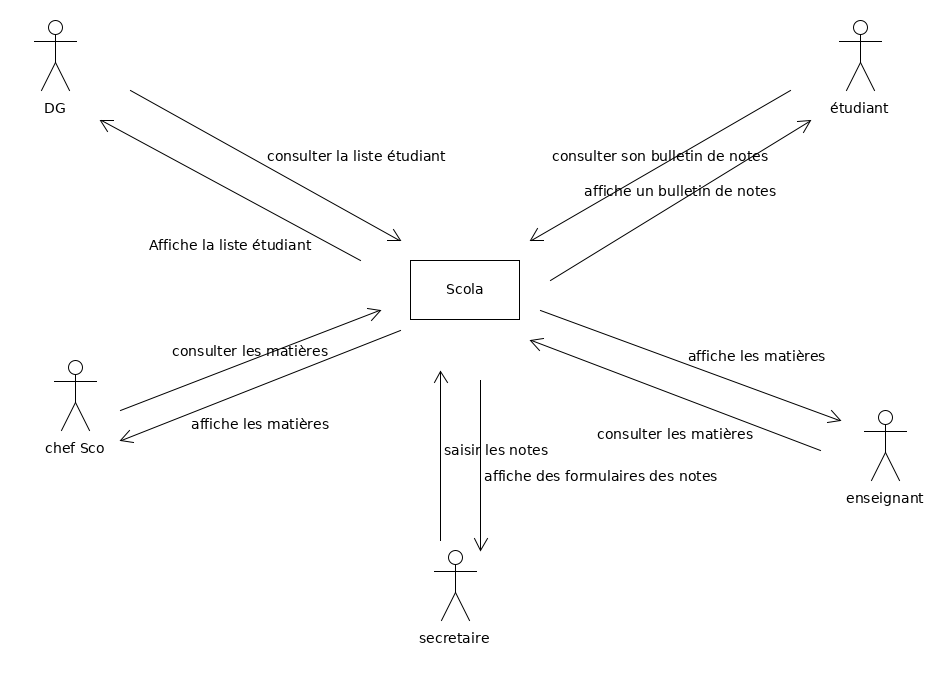
Le tableau 14 présente les messages échangés entre les acteurs et le système.

1. Messages échangés entre les acteurs et le système

|  |  |  |
| --- | --- | --- |
| Acteur | Emet | Reçoit |
| DG | - demande de connexion  - s’authentifier  - consulter la liste des étudiants  - trier la liste des étudiants  - consulter les matières  - consulter les notes par niveau et matière  - consulter un bulletin de notes  - consulter les classements | - demande d’authentification  - rediriger vers la page d’accueil  - Affiche la liste des étudiants  - affiche la liste des étudiants trier  - affiche les matières  - affiche les notes par niveau et matière  - affiche un bulletin de notes  - affiche les classements |
| Chef Sco | - demande de connexion  - s’authentifier  - consulter la liste étudiant  - trier la liste étudiant  - consulter les matières  - consulter les notes par niveau et matière  - consulter un bulletin de notes  - consulter les classements | - demande d’authentification  - rediriger vers la page d’accueil  - Affiche la liste étudiant  - affiche la liste étudiant trier  - affiche les matières  - affiche les notes par niveau et matière  - affiche un bulletin de notes  - affiche les classements |
| Secretaire | - demande de connexion  - s’authentifier  - consulter la liste étudiant  - trier la liste étudiant  - ajouter , modifier , supprimer des étudiants  - consulter les matières  - ajouter , modifier , supprimer les matières  - consulter les notes par niveau et matière  - saisir et modifier les notes  - consulter un bulletin de notes  - consulter les classements  - publier le résultat | - demande d’authentification  - rediriger vers la page d’accueil  - Affiche la liste étudiant  - affiche la liste étudiant trier  - affiche formulaire d’ajout ,de modifocation, de suppression d’étudiant  - affiche les matières  - affiche formulaire d’ajout ,de modifocation, de suppression des matières  - affiche les notes par niveau et matière  - affiche des formulaires des notes  - affiche un bulletin de notes  - affiche les classements  - affiche le résultat |
| Enseignant | - demande de connexion  - s’authentifier  - consulter la liste étudiant  - trier la liste étudiant  - consulter les matières  - consulter les notes par niveau et matière  - consulter un bulletin de notes  - consulter les classements  - envoyer les notes | - demande d’authentification  - rediriger vers la page d’accueil  - Affiche la liste étudiant  - affiche la liste étudiant trier  - affiche les matières  - affiche les notes par niveau et matière  - affiche un bulletin de notes  - affiche les classements  - affiche les notes envoyer |
| Etudiant | - demande de connexion  - s’authentifier  - consulter son notes  - consulter son bulletin de notes  - consulter les classements  - demande de reclamation | - demande d’authentification  - rediriger vers la page d’accueil  - affiche son notes  - affiche un bulletin de notes  - affiche les classements  - reclamation traiter |

1. **Modéliser le contexte**

Nous allons présenter une simple communication entre le système d’information et les acteurs avec à aide d’un diagramme de communication.



Le figure 12 montre le diagramme de contexte dynamique du système .

1. Diagramme de contexte dynamique du système [1]
2. Identifier les cas d’utilisation

Nous allons identifier les cas d’utilisation de notre système dans le tableau 15.

1. Identification des cas d’utilisation

|  |  |  |
| --- | --- | --- |
| **Cas d’utilisation** | **Acteur principal**  **et secondaires** | **Message(s) émis / reçus**  **par les acteurs** |
| S’authentifier | secretaire | émis: demande de connexio  reçus: demande d’authentification  émis: s’authentifier  reçust : rediriger vers la page d’accueil |
| chef sco |
| DG |
| Enseignant |
| Etudiant |
| Gérer les étudiants | sécretaire | émis : consulter la liste des étudiants  reçus : affiche la liste des étudiants  émis : trier la liste des étudiants  réçu: affiche la liste des étudiants trier  émis : ajouter , modifier , supprimer des étudiants  reçus : affiche formulaire d’ajout ,de modifocation, de suppression d’étudiant |
| chef sco |
| Enseignant |
| DG |
| Gérer les matières | sécretaire | émis : consulter les matières  reçus :affiche les matières  émis : ajouter , modifier , supprimer les matières  réçus : affiche formulaire d’ajout ,de modifocation, de suppression des matières |
| chef sco |
| DG |
| Enseignant |
| Gérer les notes | sécretaire | émis:consulter les notes par niveau et matière  réçus: affiche les notes par niveau et matière  émis : saisir et modifier les notes  réçus: affiche la formulaire des notes  émis : envoyer les notes  réçus : affiche les notes envoyer |
| enseignant |
| chef sco |
| DG |
| Suivre le résultat | sécretaire | émis : consulter un bulletin de notes  réçus : affiche un bulletin de notes  émis : publier le résultat  réçus : affiche le résultat |
| étudiant |
| chef sco |
| DG |
| Enseignant |
| Suivre le classement | sécretaire | émis : consulter les classements  réçus : afficher les classements |
| DG |
| Etudiant |
| Enseignant |
| chef sco |
| Reclamer un note | Etudiant | émis: demande de reclamation  réçus : reclamation traiter |
| sécretaire |
| chef sco |
| DG |
| Enseignant |

2. Diagrammes des cas d'utilisation

Le diagramme de cas d’utilisations est un diagramme qui nous permet de voir le fonctionnalité de notre système ainsi que son acteur.

La figure 13 montre le diagramme de cas d’utilsation de notre système

1. Diagramme de cas d’utilisation du système BMSI
2. Priorisation des cas d'utilisation

La priorisation des cas d’utilisation est un moyen utile pour identifier les cas d’utilisation nécessaire à la fonctionnalité du programme afin qu’il soit opérationnel.

Le tableau 16 représente l’ordre de priorisation des cas d’utilisation.

1. Priorisation des cas d’utilisation

|  |  |  |
| --- | --- | --- |
| **Acteurs** | **Cas d’utilisation** | **Priorité** |
|  |  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |  |
|  |  |
|  |  |  |
|  |  |
|  |  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. Diagramme des séquences systèrne pour chaque cas d'utilisation

Nous allons représenter dynamiquement l’intéraction entre l’acteur et le système à chaque cas d’utilisation à l’aide d’un diagramme de séquence.

Le tableau 17 présente le diagramme des séquences système pour chaque cas d’utilisation

1. Diagramme des séquences système pour chaque cas d’utilisation

|  |  |
| --- | --- |
| **Cas d’utilisation** | **Diagramme de séquence** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. Spécification des besoins techniques

Dans le processus 2TUP, les besoins techniques sont traités parallèlements aux besoins fonctionnels de l’application.

Alors pour que notre application soit opérationnelle, elle doit répondre aux spécifications techniques suivantes :

* supporter une implantation sur l’architecture à 3 niveaux avec SGBDR
* coder en framework expressJS
* capable de gérer l’interopérabilité du système
* assurer l’intégrité des données
* permettre l’authentification des utilisateurs
* privilégier le rôle des utilisateurs
* héberger localement dans une zone démilitarisée DMZ
* avoir une présentation intuitif pour les opérationnels et les décideurs

1. Modélisation du domaine

La modélisation du domaine est un moyen d’identifier les modèles significatifs du monde réel de notre conception de façon abstrait.

La figure 14 définie les modèles du domaine de notre système.

1. Diagramme du domaine de modélisation du système
2. Conception détaillée (UML)

La conception détaillée est la partie qui va nous permettre de construire et de documenter avec précision les classes, les interfaces, les tables ainsi que les méthodes qui constituent le codage de la solution.

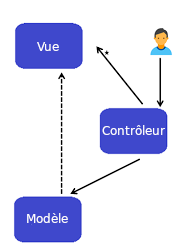
1. Architecture du système

Il existe plusieurs façons de structurer l’architecture système en utilisant les design pattern comme MVC, MVT, MVVM MVP., Mais pour résoudre notre problème d’architecture système, nous allons utiliser le design pattern MVC puisque celui-ci est aussi mieux adapté à la technologie qu’on a choisi durant la conception de l’avant projet.

Le design pattern MVC va nous permettre de structure notre application en 3 composants:

* **Modèle** : qui représente la couche logique ou métier de l’application et l’accès à la base de donnée
* **Vue** : qui définit l’affichage à rendre sur le côté client
* **Contrôleur** : qui fait l’intermédiaire entre le modèle et la vue en traitant les requêtes entrantes avant de les transmettre au modèle et retourne les résultats du modèle à la vue.

La figure 15 représente l’architecture MVC.

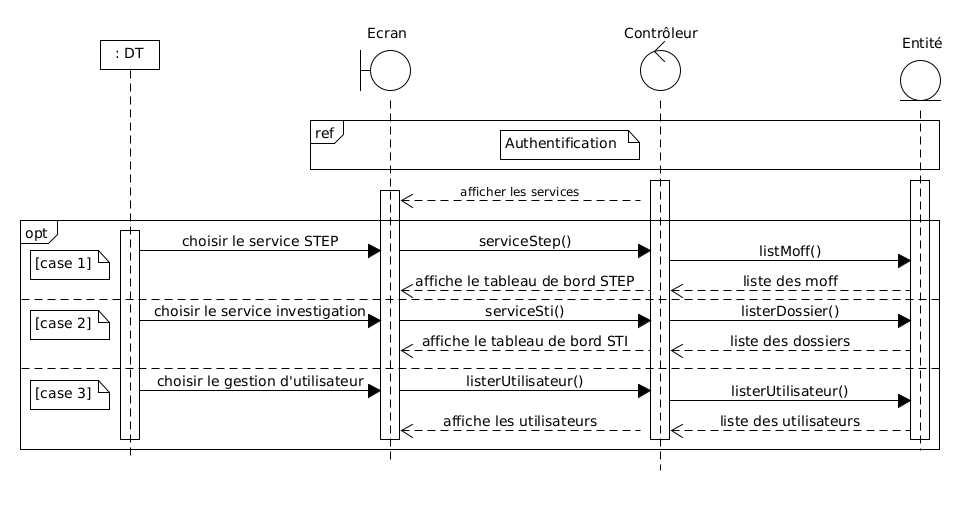


1. Architercture MVC [18]
2. Diagramme de séquence de conception pour chaque cas d'utilisation

Nous allons voir maintenant le Diagramme de séquence de conception pour chaque cas d'utilisation.

Cas d’utilisation «**Gérer les services**»

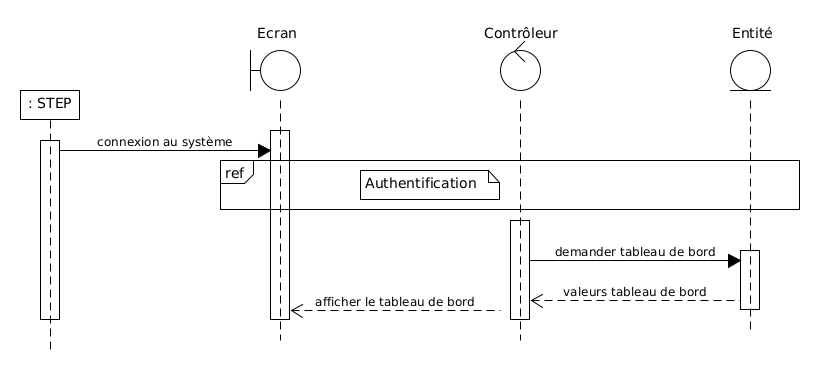
La figure 16 présente le diagramme de séquence de conception pour gérer les services.



1. Diagramme de séquence de conception pour gérer les services

Cas d’utilisation « **Suivre le tableau de bord** »

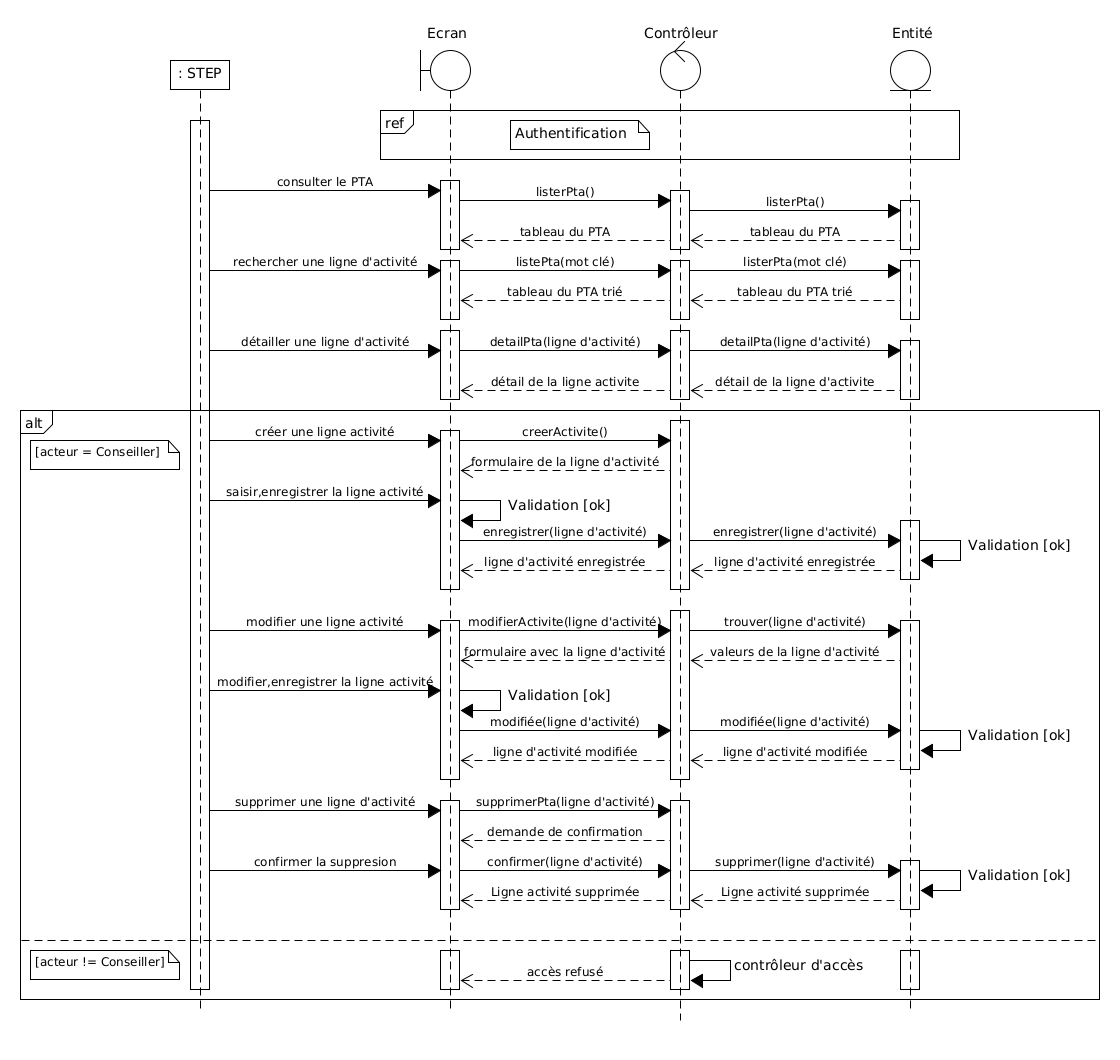
La figure 17 présente le diagramme de séquence de conception pour suivre le tableau de bord.



1. Diagramme de séquence de conception pour suivre le tableau de bord

Cas d’utilisation « **Planifier le PTA** »

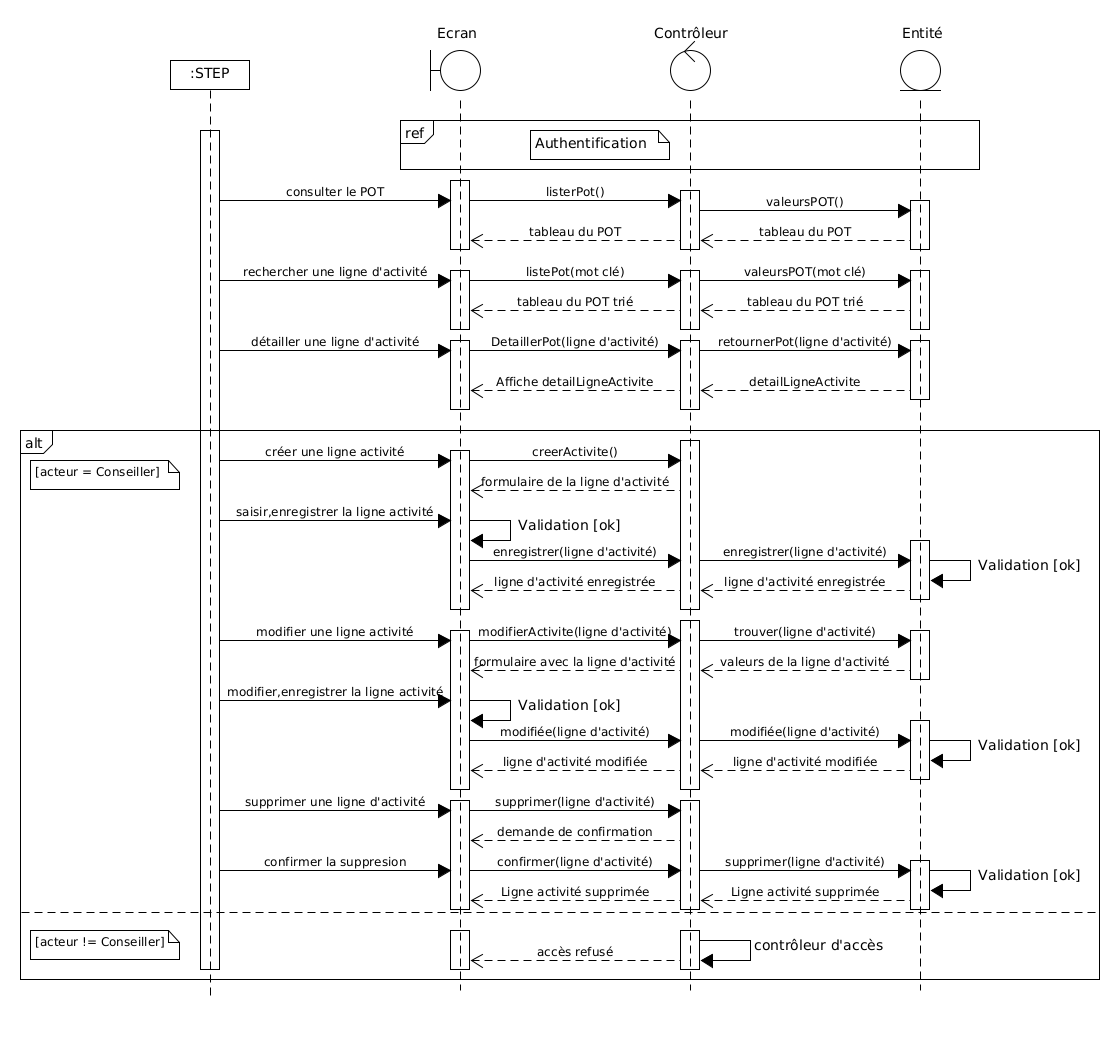
La figure 18 présente le diagramme de séquence de conception pour planifier le PTA.



1. Diagramme de séquence de conception pour planifier le PTA

Cas d’utilisation « **Planifier le POT** »

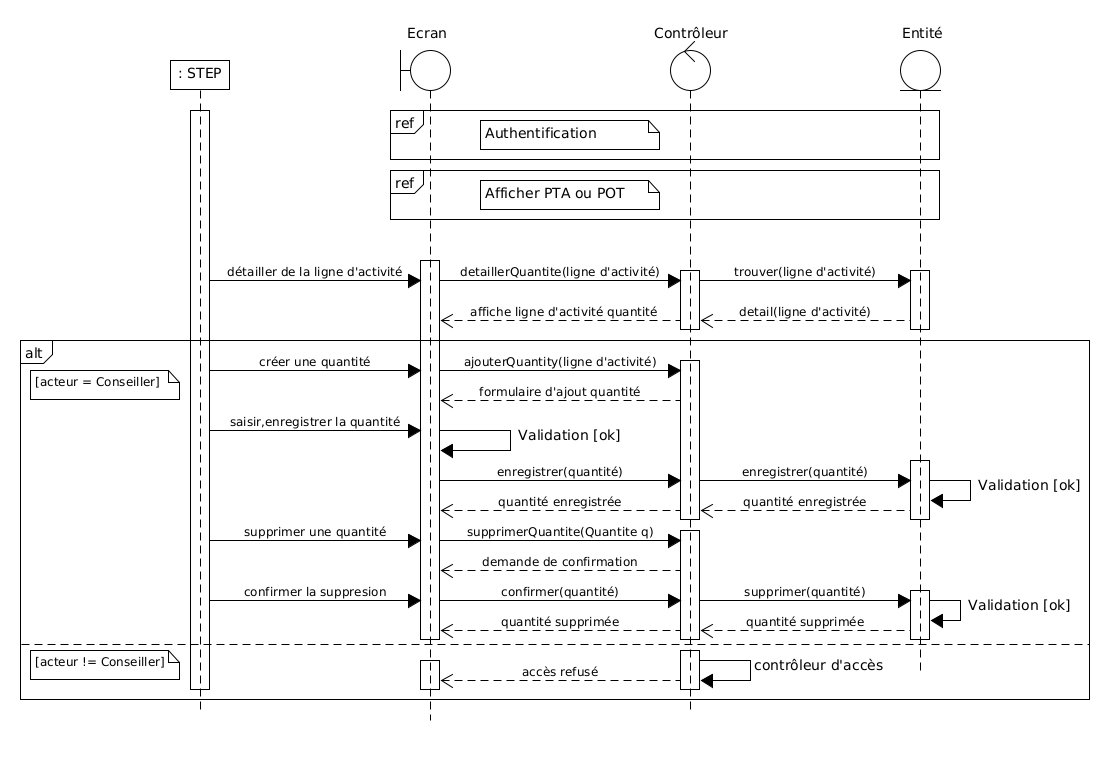
La figure 19 présente le diagramme de séquence de conception pour planifier le POT.



1. Diagramme de séquence de conception pour planifier le POT

Cas d’utilisation « **Gérer la quantité** »

La figure 20 présente le diagramme de séquence de conception pour gérer la quantité.

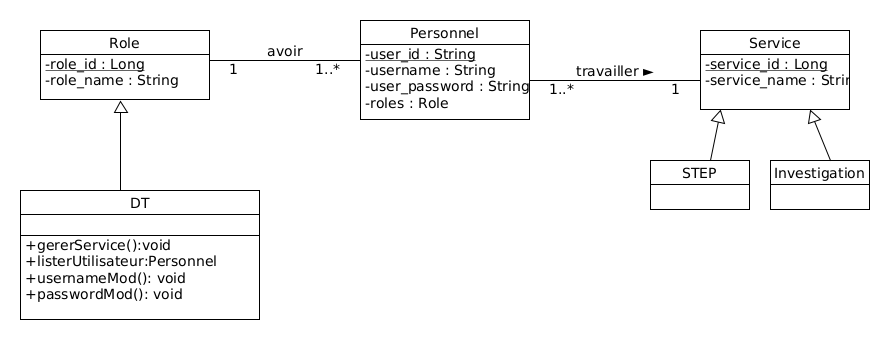


1. Diagramme de séquence de conception pour gérer la quantité
2. Diagramme de classe de conception pour chaque cas d'utilisation

Nous allons voir la classe de conception de chaque cas d’utilisation pour montrer les relations entre les classes participantes à la réalisation.

Cas d’utilisation « **Gérer les services** »

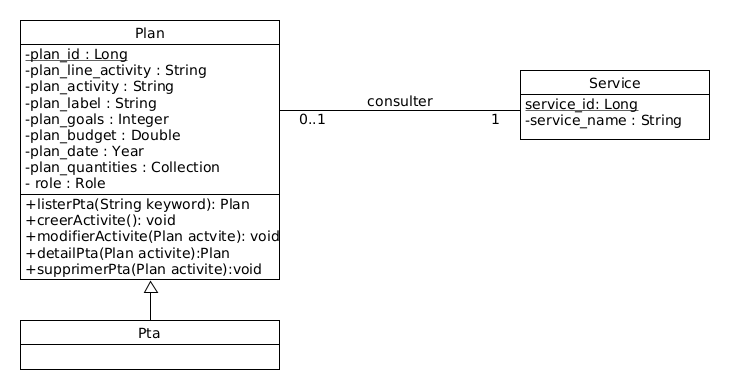
La figure 30 présente le diagramme de classe de conception pour gérer les services.



1. Diagramme de classe de conception pour gérer les services

Cas d’utilisation « **Planifier le PTA** »

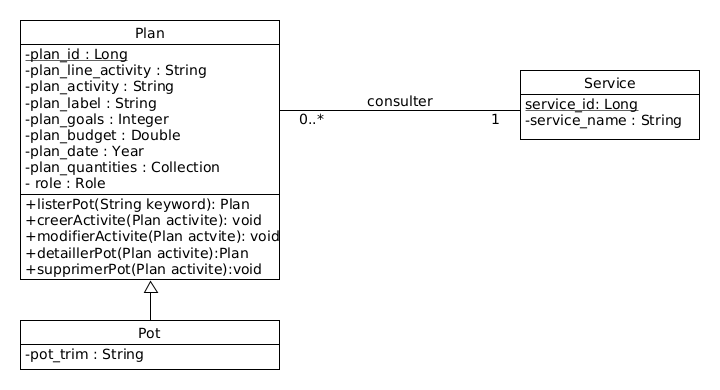
La figure 31 présente le diagramme de classe de conception pour planifier le PTA.



1. Diagramme de classe de conception pour planifier le PTA

Cas d’utilisation « **Planifier le POT** »

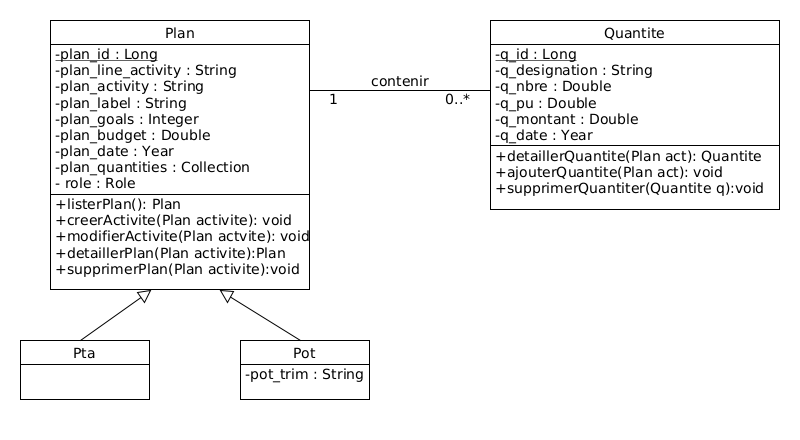
La figure 32 présente le diagramme de classe de conception pour planifier le POT.



1. Diagramme de classe de conception pour planifier le POT

Cas d’utilisation « **Gérer la quantité** »

La figure 33 présente le diagramme de classe de conception pour gérer la quantité.



1. Diagramme de classe de conception pour gérer la quantité

Cas d’utilisation « **Gérer les réalisations** »

La figure 34 présente le diagramme de classe de conception pour gérer les réalisations.

1. Diagramme de classe de conception global

Le diagramme de classe de conception global va nous montrer la synthèse globale entre les classes définies dans chaque cas d’utilisation.

La figure 43 présente la classe de conception global

1. Classe de conception global
2. Diagramme de paquetages

Maintenant, on va regrouper les classes dans un package pour mieux leur structures.

La figure 44 montre le diagramme de paquetages de l’application.

1. Diagramme de paquetages de l’application
2. Diagramme de déploiement

Le diagramme de déploiement va nous permettre devoir notre application lors de son déploiement dans l’architecture matériel.

La figure 45 montre le diagramme de déploiement de notre application

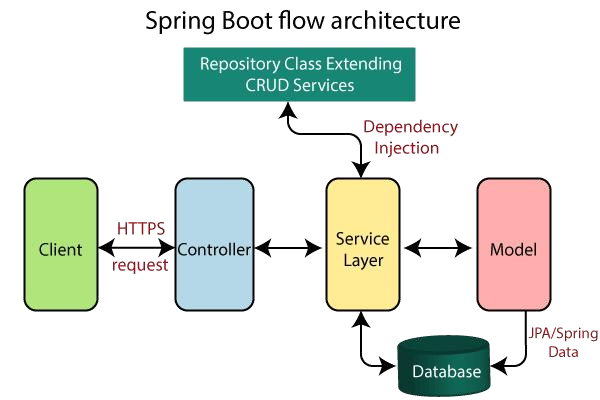
1. Diagramme de déploiement de l’application

1. REALISATION
2. Mise en place de l'environnement de développement
3. lnstallation et configuration des outils

Nous allons voir les installations et les configurations des outils nécessaires pour réaliser ce projet en sachant que nous utilisons un système d’exploitation linux comme environnement du travail.

1. Installation et configuration

1. Architecture de l'application



1. Développement de l'application
2. Création de la base de données

Pour commencer le développement de l’application, nous allons créer la base de données «bmsi» pour permettre à nos données de persister.

1. Codage de l’application
3. Présentation de l’application

CONCLUSION GENERALE

REFERENCES BIBLIOGRAPHIQUES

[1]. Pascal Roques et Franck Vallée, février 2007. UML 2 en action. Edition Eyrolles, 397p

**REFERENCES WEBOGRAPHIQUES**

GLOSSAIRES

**Acteur** : type stéréotypé représentant une abstraction qui interagit avec un système en

échangeant des informations.

**Analyse d’existant**: Examen méthodique permettant de distinguer les différentes parties d’un

problème et de définir leurs rapports.

**Base de données** : Une base de données est une technologie, dans laquelle il est possible de

stocker des données de façon structurée et avec le moins de redondance possible.

**Cas d’utilisation** : C’est un concept graphique qui représente les besoins de l’utilisateur du système.

**Design Pattern :** C’est une technique utilisée pour résoudre les problèmes dans la réalisation d’un système d’information.

**Diagramme de classe :** Représentation statique d’un concept graphique d’un ou plusieurs objets en relation.

**Diagramme de séquence :** C’est un concept graphique qui représente l’interaction entre l’utilisateur et le système.

**Framework** : C’est un cadre de travail ou cadre d’application offrant un ensemble d’outils et de briques logicielles, qui permettent d’automatiser certains parties du développement d’application.

**IDE** : ensemble d’outils pour augmenter la productivité des programmeurs qui

développent des logiciels.

**Mode lecture** : On peut seulement consulter sans modifier le contenu

**MVC** : C’est un design pattern qui permet de résoudre le problème d’architecture système.

**POO** : Programmation Orientée Objet. Consiste en la définition et l'assemblage de

briques logicielles appelées objet, un objet représente un concept, une idée ou une entité

quelconque.

**POT :** Document pour planifier les tâches à faire pendant trois (03) mois

**PTA** : Document pour planifier les tâches à faire pendant une année.

**Serveur** : C’est un ordinateur performant capable de stocker des milliards d’informations.

**Template** : c’est un modèle ou gabarit d’une page web sans contenu dont les éléments statiques sont déjà positionnés et mis en forme.[20]

TABLE DES MATIERES

[CURRICULUM VITAE II](#_Toc855339578)

[SOMMAIRE GENERALE IV](#_Toc854852539)

[REMERCIEMENTS V](#_Toc167095678)

[LISTE DES FIGURES VI](#_Toc1575671624)

[LISTE DES TABLEAUX VIII](#_Toc565417654)

[LISTE DES ABREVIATIONS OU NOMENCLATURE IX](#_Toc1163980223)

[INTRODUCTION GENERALE 1](#_Toc1927729618)

[Chapitre 1. Description du projet 2](#_Toc1747494718)

[3.1. Formulation 2](#_Toc739648847)

[3.2. Objectif et besoins de l’utilisateur 2](#_Toc1639093082)

[3.3. Moyens nécessaires à la réalisation du projet 2](#_Toc985387315)

[3.3.1. Moyens Humains 2](#_Toc216554379)

[3.3.2. Moyens matériels et logiciels 2](#_Toc424075339)

[3.4. Résultats attendus 4](#_Toc2007526519)

[3.5. Chronogramme des activités 4](#_Toc1655109344)

[PARTIE I : ANALYSE ET CONCEPTION 6](#_Toc1027058986)

[Chapitre 2. Analyse Préalable 7](#_Toc731048534)

[4.1. Analyse de l'existant 7](#_Toc428724068)

[4.1.1. Organisation actuelle 7](#_Toc195046444)

[4.1.2. lnventaire des moyens matériels et logiciels 7](#_Toc966339808)

[4.2. Critique de l’existant 8](#_Toc1791269112)

[4.3. Conception avant projet 8](#_Toc1595852555)

[4.3.1. Proposition de solutions 8](#_Toc450502052)

[4.3.2. Comparaison et choix de modélisation 10](#_Toc1845756577)

[4.3.3. Présentation du langage modélisation UML 10](#_Toc842745831)

[4.3.4. Présentation du processus 2TUP 12](#_Toc235494682)

[4.3.5. Comparaison et choix des frameworks 13](#_Toc1275275892)

[4.3.6. Présentation du Swing via NET URL ou ApacheClient 13](#_Toc2121534050)

[4.3.7. Comparaison et choix de SGBD 14](#_Toc1337797525)

[4.3.8. Présentation de base de données MariaDB 15](#_Toc871431879)

[4.3.9. Interopérabilité et client-serveur 15](#_Toc1570524622)

[4.3.10. Présentation de l’architecture à 3 niveaux 16](#_Toc45653455)

[Chapitre 3. Analyse conceptuelle 18](#_Toc1726284419)

[5.1. Dictionnaire des données 18](#_Toc1737620300)

[5.2. Règles de gestion 20](#_Toc1621325080)

[5.3. Représentation et spécification des besoins 21](#_Toc144218425)

[5.3.1. Recueil des besoins (pré-étude) 21](#_Toc754116875)

[5.3.2. Identifier les cas d’utilisation 23](#_Toc1401571050)

[5.3.3. Diagrammes des cas d'utilisation 23](#_Toc1891713144)

[5.3.4. Priorisation des cas d'utilisation 24](#_Toc1493765723)

[5.3.5. Diagramme des séquences systèrne pour chaque cas d'utilisation 26](#_Toc893180484)

[5.4. Spécification des besoins techniques 30](#_Toc729616811)

[5.5. Modélisation du domaine 31](#_Toc1710320102)

[Chapitre 4. Conception détaillée (UML) 32](#_Toc1317255823)

[6.1. Architecture du système 32](#_Toc589659683)

[6.2. Diagramme de séquence de conception pour chaque cas d'utilisation 33](#_Toc1217945798)

[6.3. Diagramme de classe de conception pour chaque cas d'utilisation 36](#_Toc196831162)

[6.4. Diagramme de classe de conception global 39](#_Toc1320708217)

[6.5. Diagramme de paquetages 41](#_Toc1646669867)

[6.6. Diagramme de déploiement 42](#_Toc391877606)

[PARTIE II : REALISATION 43](#_Toc139564378)

[Chapitre 5. Mise en place de l'environnement de développement 44](#_Toc1290455331)

[7.1. lnstallation et configuration des outils 44](#_Toc1987730161)

[7.1.1. Installation et configuration 44](#_Toc590066430)

[7.2. Architecture de l'application 44](#_Toc988728260)

[Chapitre 6. Développement de l'application 45](#_Toc682992344)

[8.1. Création de la base de données 45](#_Toc825561113)

[8.2. Codage de l’application 45](#_Toc116520505)

[8.3. Présentation de l’application 45](#_Toc657042747)

[CONCLUSION GENERALE 46](#_Toc15874990)

[REFERENCES BIBLIOGRAPHIQUES X](#_Toc987952384)

[REFERENCES WEBOGRAPHIQUES XI](#_Toc80083721)

[GLOSSAIRES XII](#_Toc61528446)

[TABLE DES MATIERES XIII](#_Toc566753155)

[RESUME XVI](#_Toc1817704021)

[ABSTRACT XVI](#_Toc1682853526)

RESUME

Notre projet consiste à concevoir une application pour les 03 volets (éducation, prévention, investigation) de la direction territoriale du Bianco Mahajanga à partir d’un tableau de bord pour leur permettre de suivre les activités définies dans le PTA et d’effectuer le suivi-évaluation de ces activités (techniquement, financièrement).

Dans la partie conception avant projet, nous avons pris la solution 4 qui est d’engager un stagiaire à concevoir une application web et la déployée sur une architecture à 3 niveaux.

A la conception du projet, nous avons utilisé le langage de modélisation UML avec le processus 2TUP pour déterminer les besoins des utilisateurs du système.

A la réalisation, nous avons codé en JAVA avec le framework spring boot et stocké les ressources dans une base de données MariaDB et les déployé dans un serveur linux.

**Mots clés :** Volets, Tableau de Bord, PTA, Conception, Application web , Architecture à 3 niveaux, UML, 2TUP, JAVA, Framework, Spring Boot, Ressources, MariaDB, déployé, Serveur linux

ABSTRACT

**Keywords:**