



**UNIVERSITE DE FIANARANTSOA**

**ECOLE DE MANAGEMENT ET D’INNOVATION TECHNOLOGIQUE**

**Mention**: **INFORMATIQUE**

**Parcours**: **Développement d’Application Intranet - Internet (D.A.I.I)**

**MEMOIRE DE FIN DE CYCLE**

Pour l’obtention du Diplôme de Licence en Informatique

**MISE EN PLACE D’UNE APPLICATION WEB DE GESTION D’UN RAPPORT DE LA PRODUCTION ELECTRICITE**

Présenter par : **MAMINJARA JUNNOS**

Sous l’encadrement de : **RAKOTOARISON TSIORINANTENAINA Réné**

Année Universitaire : 2023 – 2024

**CURRICULUM VITAE**



MAMINJARA Junnos

Né le 29 Novembre 2003 à Mananjary

209 011 043 624

Célibataire

Email : [junnos.317@gmail.com](mailto:junnos.317@gmail.com)

N° Tel : 034 33 562 78

Etudiant en Troisième Année de Licence en

Développement d’Application Internet Intranet

**FORMATIONS ET DIPLÔME :**

* 2023-2024: Troisième Année de Licence Professionnelle à l’Ecole de Management et  
  d’Innovation Technologique (EMIT) de l'Université de Fianarantsoa ;
* 2022-2023 : Deuxième Année de Licence Professionnelle à l’Ecole de Management et  
  d’Innovation Technologique (EMIT) de l'Université de Fianarantsoa ;
* 2022-2023 : Participants au salon d’entreprise 1.0 ;
* 2022-2023 : Participants au salon de Métiers ;
* 2021-2022 : Formation de BOCASAY dans l’IT Connecte 2.0 Fianarantsoa ;
* 2021-2023 : 1 ère année à l’EMIT mention DAII ;
* 2019-2020 : BAC Technique option génie civil (Mention Assez-bien) -LTP Mananjary ;

**EXPERIENCES PROFESSIONNELLES :**

* Juin 2024 : Stage professionnel au sein du JIRAMA Centrale Fianarantsoa durée (03) trois mois, par une thème « Mise en place d’application web pour la production électrique » au sein de JIRAMA Tsianolondroa Fianarantsoa ;

Juin 2023 : Stage professionnel au sein du JIRAMA Centrale Fianarantsoa durée (02) deux mois, a pour thème « Conception et Réalisation d’une application de gestion d’absence

* des employés » au sein de JIRAMA Tsianolondroa Fianarantsoa ;
* . Conception et réalisation d’un logiciel pour la « Gestion de vents ordinateurs » en JAVA, PHP, C# ;
* Conception et réalisation d’un logiciel pour la « Gestion de frais de scolarité de CUFP » en langage scripte ;
* Conception et réalisation d’un logiciel pour la « Gestion de note EMIT » en Access.

**CONNAISSANCES EN INFORMATIQUE :**

* Systèmes d’Exploitation : Windows, Linux ;
* Méthode et langage de conception : MERISE 2 ;
* Outils de conception : Microsoft Access ;
* Outils de développement : NetBeans, Visual Studios code, Code Blocks ;
* Programmation Web : HTML&CSS, PHP (laravel), JS(React) ;
* Langages de Programmation : C, C#, Java, Python ;
* SGBD : MS Access, MySQL ;

**CONNAISSANCES LINGUISTIQUES :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Langues** | **Ecrire** | **Rédiger** | **Parler et Communiquer Oralement** |
| **Malagasy** | Très Bien | Très Bien | Très Bien |
| **Française** | Bien | Assez-Bien | Assez-Bien |
| **Anglaise** | Assez-bien | Passable | Assez-bien |

**DIVERS :**

* Sport : Football, Volley-ball, Natation.
* Sens du travail d’équipe, sens de la communication.
* Lire des livres, regarder des documentaires et films.

# **AVANT – PROPOS**

Le terme principal de la formation Universitaire est de préparer les étudiants dans le domaine du travail. Ce futur métier qui attend les étudiants requiert le savoir-faire, la bonne volonté, le dynamisme, et le professionnalisme. Le futur cadre doit, donc, être à la portée de ces exigences. L’adaptation des méthodes avec l’évolution de la technologie dans le monde devient de plus en plus nécessaire de nos jours. Avec les possibilités qui accompagnent cet essor inévitable, nous pouvons envisager l’informatisation de tous systèmes de travail.

Dans mon cadre de nos études dans la motivation Développement d’Application Internet Intranet (DAII) à l’Ecole Mangement et d’Innovation Technologique (EMIT), les étudiants en troisième année de licence doivent effectuer un stage de trois (03) mois au sein de l’entreprise en vue d’obtient du diplôme de licence professionnelle. Ce stage aura pour objectif la mise en pratique des connaissances théoriques acquises en classe, mais également l’adaptation au milieu du travail et confirmer l’adéquation des connaissances aux demandes des marches réel.

C’est la raison pour laquelle nous avons effectué le stage au sein de l’entreprise JIRAMA Centrale à Fianarantsoa Tsianolondroa. Nous avons donc expliqué toutes les démarches que nous avons suivie ainsi pour la mise en place du système d’exploitation en gérant le rapport d’une production électricité au sein de JIRAMA. Lors du stage, il est également demandé de rédiger un ouvrage portant sur le stage effectué en guide de mémoire de fin d’études de cycle de formation Licence.

Notre thème se pose que « Mise en place d’une application web pour la production électricité au sein de JIRAMA Fianarantsoa ». Il s’insère donc la mise en place d’application et recèle les opérations effectuées au cours du stage.

# **REMERCIEMENTS**

Il nous est particulièrement agréable avant de présenter notre travail, d’exprimer toute notre gratitude envers les personnes qui de près ou de loin nous ont apporté leur sollicitude. Ce présent ouvrage a été le fruit d’une collaboration de différents intervenants à qui nous tenions présenté tous nos sincères remerciements. Plus particulièrement à :

* Professeur **HAJALALAINA Aimé Richard**, Président de l’université Fianarantsoa, de nous avoir accueillir et permis dans son université.
* Docteur HDR **RAKOTANIRAINY Hasina Lalaina**, Directeurs de L’Ecole de Mangement et d’Innovation Technologique (EMIT) de nous avoir accueillir et permis d’étudier dans son établissement.
* Madame **RABEZANAHARY Hoby**, notre responsable de Mention, qui nous a permis d’acquérir des connaissances dans la mention Développement d’application Internet Intranet (DAII).
* Madame **IHARIMALALA EVA-LISI**, chef de service de ressource humaine.
* Monsieur **ANDREA** **Ben Alphonse**, notre encadreur professionnel qui on nous a guider et soutenue avec ses conseils et ses formations.
* A tous les enseignements et les personnels de L’EMIT-UF

Ces remerciements vont également à nos très chères et adorables familles qui sont toujours été présents, avec leurs soutiens et leurs précieuses aides, et à tous ceux qui sont concourus pour accumuler cet ouvrage.

Merci à vous tous !

# **LISTE DES FIGURES**

[Figure 1.1 : organigramme d’EMIT 11](#_Toc181010125)

[Figure 2.1: Structure organisationnelles 17](#_Toc181010126)

[Figure 4.1 : Modèle de cycle de vie d’une application web 33](#_Toc181010127)

[Figure 4.2 : Modèle en cascade [Wikipédia/cycle en cascade] 37](#_Toc181010128)

[Figure 4.3 : Schéma de modèle en V 38](#_Toc181010129)

[Figure 4.4 : Modèle en spirale [Wikipedia.org/wiki/spiral model] 39](#_Toc181010130)

[Figure 4.5 : Modèle par incrément ou itératif 40](#_Toc181010131)

[Figure 4.6 : Modèle incrémentale 41](#_Toc181010132)

[Figure 4.7: Evolution de l’UML 44](#_Toc181010133)

[Figure 4.8 : Les différents diagrammes UML 49](#_Toc181010134)

[Figure 4.9 : Formalisme du diagramme de cas d’utilisation. 50](#_Toc181010135)

[Figure 4.10 : Formalisme du diagramme de séquence. 52](#_Toc181010136)

[Figure 4.11 : Formalisme de diagramme de classe 54](#_Toc181010137)

[Figure 4.12 : Formalisme du diagramme d’activité 55](#_Toc181010138)

[Figure 4.13 : Formalisme du diagramme de déploiement 55](#_Toc181010139)

[Figure 6.1 : Diagramme global de cas d’utilisation 69](#_Toc181010140)

[Figure 6.2 : Diagramme de séquence de création d'échantillon 71](#_Toc181010141)

[Figure 6.3 : Diagramme d’activité de l’authentification 72](#_Toc181010142)

[Figure 6.4 : Diagramme d’Activité Globale de l’application 72](#_Toc181010143)

[Figure 6.5 : Diagramme de classe de l’application 73](#_Toc181010144)

[Figure 7.1 : historique de version laravel 83](#_Toc181010145)

[Figure 8.1: Les deux parties principales 85](#_Toc181010146)

[Figure 8.2: les instructions de logiciel 85](#_Toc181010147)

[Figure 8.3 : la partie matérielle des ordinateurs 86](#_Toc181010148)

[Figure 8.4 : Architecture matérielle 87](#_Toc181010149)

[Figure 9.1 : interface générale de l’application 93](#_Toc181010150)

[Figure 9.2 : menu principale 93](#_Toc181010151)

[Figure 9.3 : interface de choix de rapport qu’on remplir 94](#_Toc181010152)

[Figure 9.4 : formulaire du rapport de la production électricité 94](#_Toc181010153)

[Figure 1: installation de Wampserver xix](#_Toc181010154)

[Figure 2: Interface de PhpMyAdmin xx](#_Toc181010155)

# **LISTE DES TABLEAUX**

[Tableau 1.1 : Tableau récapitulatif 3](#_Toc179482519)

[Tableau 2.1: Centre d’exploitation de la DIR/FR 19](#_Toc179482520)

[Tableau 3.1 : Risque du projet 22](#_Toc179482521)

[Tableau 3.2 : déroulement de stage 23](#_Toc179482522)

[Tableau 4.1 : Les avantages et les inconvénients d’UML 47](#_Toc179482523)

[Tableau 4.2 : Classement des diagrammes ; 49](#_Toc179482524)

[Tableau 4.3 : Comparaison des outils de conception d’UML 56](#_Toc179482525)

[Tableau 5.1 : liste des matériels informatiques existants 61](#_Toc179482526)

[Tableau 6.1 : Dictionnaire de données de notre application 67](#_Toc179482527)

[Tableau 6.2 : Raffinement du cas d’utilisation « S’authentifier » 70](#_Toc179482528)

[Tableau 7.1: Comparaison du SGBD 79](#_Toc179482529)

[Tableau 7.2 : Comparaison des différents serveurs de développement 79](#_Toc179482530)

# **LISTE DES ABREVATIONS**

**AES** Administration Economique et Sociale

**AEOI** l’Association des Electriciens des Iles de l’Océan Indien

**BD** Base de Données

**CETE** Centre d’Etude Technique de l’Equipement

**CFP**  Centre Formation Professionnel

**CMN** Communications et Médias Numériques

**CUFP** Centre Universitaire de Formation Professionnalisante

**CTI** Centre de Technique d’Information

**DASI** Développement d’Application et Système d’Information

**M2I** Master Professionnel en Modélisation et Ingénierie en informatique

**JIRAMA** JIro sy RAno Malagasy

**LMD** Licence, Master et Doctorat

**MD**  Management Décisionnel

**MERISE** Méthode d’Etude et de Réalisation Informatique pour les Systèmes d’Entreprise

**RPC**  Master Professionnel en Relations Publiques et Communications

**SEC**  Société d’Etat Civil

**SINEE** Société d’Intérêt Nationale de l’Eau et d’Electricité

**SGBD** Système de Gestion de Base de Données

**SMEE** Société Malgache de l’Eau et de l’Electricité

**SI**  Système d’Information

**SIGD** Système d’Information Géomatique et Décisionnel

**SINEE** Société d’Intérêt Nationale de l’Eau et d’Electricité

# **GLOSSAIRE**

Le glossaire vise à définir les termes techniques ou spécifiques utilisés dans le cadre du projet.

**Application web** : Un programme informatique accessible via un navigateur web, qui permet aux utilisateurs d'effectuer diverses tâches en ligne.

**Base de données** : Ensemble structuré d’informations, souvent organisées en tables, permettant de stocker et de gérer des données.

**MySQL :** est un système de gestion de bases de données relationnelles SQL open source développé et supporté par Oracle.

**UML** (Unified Modeling Language) : Langage utilisé pour modéliser et représenter graphiquement les différentes composantes et interactions d’un système.

**Interface utilisateur** (UI) : La partie visible d’une application avec laquelle l’utilisateur interagit.

**Interface de programmation d'application** (API) : Un ensemble de protocoles permettant à deux systèmes logiciels de communiquer entre eux.

**Requête** **:** désigne une interrogation ou une action que l’on adresse à une base de données, c’est grâce à elle que l’on peut obtenir des listes précises.

**Système d'information** : Un ensemble organisé de ressources (personnes, données, processus) qui permet de collecter, stocker, traiter et distribuer de l'information.

**Serveu**r : Ordinateur ou logiciel qui fournit des services, comme l’hébergement de l'application ou la gestion de la base de données, à d'autres ordinateurs.

**Serveur SGBD :** Serveur SGBD Système de Gestion de Base de Données. Logiciel  
permettant de stocker les données, de les mettre à jour et de les consulté.

# **TABLE DES MATIERES**

[**AVANT – PROPOS** iv](#_Toc182465615)

[**REMERCIEMENTS** v](#_Toc182465616)

[**LISTE DES FIGURES** vi](#_Toc182465617)

[**LISTE DES TABLEAUX** vii](#_Toc182465618)

[**LISTE DES ABREVATIONS** viii](#_Toc182465619)

[**GLOSSAIRE** ix](#_Toc182465620)

[**TABLE DES MATIERES** x](#_Toc182465621)

[**INTRODUCTION** 1](#_Toc182465622)

[**Chapitre 1 : Présentation de L’EMIT** 3](#_Toc182465623)

[**1.1.** **Historique** 3](#_Toc182465624)

[**1.2.** **Partenariat** 4](#_Toc182465625)

[**1.3.** **Cycle de licence** 5](#_Toc182465626)

[**1.3.1.** **Licence professionnelle en développement d’application Internet Intranet (DA2I)** 5](#_Toc182465627)

[**1.3.1.1.** **Condition générale** 5](#_Toc182465628)

[**1.3.1.2.** **Stages** 5](#_Toc182465629)

[**1.3.1.3.** **Diplôme** 5](#_Toc182465630)

[**1.3.1.4.** **Compétences** 5](#_Toc182465631)

[**1.3.2.** **Licence professionnelle en administration économique et sociale (AES)** 6](#_Toc182465632)

[**1.3.2.1.** **Condition générale** 6](#_Toc182465633)

[**1.3.2.2.** **Stages** 6](#_Toc182465634)

[**1.3.2.3.** **Diplômes** 6](#_Toc182465635)

[**1.3.2.4.** **Compétences** 6](#_Toc182465636)

[**1.3.3.** **Licence professionnelle en relations publiques et multimédias (RPM)** 7](#_Toc182465637)

[**1.3.3.1.** **Condition générale** 7](#_Toc182465638)

[**1.3.3.2.** **Stages** 7](#_Toc182465639)

[**1.3.3.3.** **Diplômes** 7](#_Toc182465640)

[**1.3.3.4.** **Compétences** 7](#_Toc182465641)

[**1.4.** **Cycle de master** 8](#_Toc182465642)

[**1.4.1.** **Master professionnel en management décisionnel** 8](#_Toc182465643)

[**1.4.1.1.** **Condition d’accès** 8](#_Toc182465644)

[**1.4.1.2.** **Durée de formation** 8](#_Toc182465645)

[**1.4.1.3.** **Objectifs** 8](#_Toc182465646)

[**1.4.1.4.** **Insertion professionnelle** 8](#_Toc182465647)

[**1.4.2.** **Master professionnel en système d’Information géomatique-télédétection et décision** 9](#_Toc182465648)

[**1.4.2.1.** **Condition d’accès** 9](#_Toc182465649)

[**1.4.2.2.** **Durée de formation** 10](#_Toc182465650)

[**1.4.2.3.** **Objectifs** 10](#_Toc182465651)

[**1.5.** **Présentation de service** 10](#_Toc182465652)

[**1.6.** **Organigramme de l’EMIT** 10](#_Toc182465653)

[**Chapitre 2 : Présentation de JIRAMA** 12](#_Toc182465654)

[**2.1** **Historique** 12](#_Toc182465655)

[**2.1.1** **Avant 1975** 12](#_Toc182465656)

[**2.1.2** **De 1975 jusqu’à nos jours** 12](#_Toc182465657)

[**2.1.3** **Statique juridique** 13](#_Toc182465658)

[**2.1.4** **Les Objectifs** 13](#_Toc182465659)

[**2.2** **Les activités de la JIRAMA** 14](#_Toc182465660)

[**2.2.1** **La Branche électricité** 14](#_Toc182465661)

[**2.2.2** **La production** 14](#_Toc182465662)

[**2.2.3** **La Commercialisation** 15](#_Toc182465663)

[**2.2.4** **Le transport et la distribution** 15](#_Toc182465664)

[**2.2.5** **Branche d’eau** 16](#_Toc182465665)

[**2.2.6** **L’exploitation** 16](#_Toc182465666)

[**2.2.7** **La commercialisation** 16](#_Toc182465667)

[**2.2.8** **La Distribution** 16](#_Toc182465668)

[**2.3** **Structure organisationnelle de la JIRAMA** 16](#_Toc182465669)

[**2.4** **Les missions de chaque service** 18](#_Toc182465670)

[**2.5** **Présentation des centres d’exploitation de la DIR/FR** 19](#_Toc182465671)

[**Chapitre 3 : Présentation de Projet** 21](#_Toc182465672)

[**3.1** **Introduction** 21](#_Toc182465673)

[**3.2** **Origine de projet** 21](#_Toc182465674)

[**3.3** **Problématique** 21](#_Toc182465675)

[**3.4** **Conduite de projet** 22](#_Toc182465676)

[*a Maintenance évaluative* 25](#_Toc182465677)

[*b Formation de l’utilisateur* 25](#_Toc182465678)

[**3.5** **Cahier de charge** 26](#_Toc182465679)

[**3.5.1** **Définition** 26](#_Toc182465680)

[**3.5.2** **Analyse de besoins** 26](#_Toc182465681)

[**3.5.3** **Besoin Fonctionnels** 26](#_Toc182465682)

[**3.5.4** **Besoins non fonctionnels** 27](#_Toc182465683)

[**3.6** **Objectif** 28](#_Toc182465684)

[**Chapitre 4 : Méthodes et notations utilisées** 31](#_Toc182465685)

[**4.1** **Méthode utilisée** 31](#_Toc182465686)

[**4.2** **Généralité sur le génie logiciel** 31](#_Toc182465687)

[**4.3** **Définition d’un modèle** 32](#_Toc182465688)

[**4.3.1** **Modèle de cycle de vie d’un projet web** 32](#_Toc182465689)

[**4.3.2** **Modèle de cycle de vie d’un logiciel** 34](#_Toc182465690)

[**4.3.3** **Les différents modèles de cycle de vie** 35](#_Toc182465691)

[**4.3.3.1** **Cycle de vie en Cascade** 36](#_Toc182465692)

[**4.3.3.2** **Cycle de vie en V** 37](#_Toc182465693)

[**4.3.3.3** **Modèle de cycle de vie en spirale** 38](#_Toc182465694)

[**4.3.3.4** **Modèle de cycle de vie par incrément ou itératif** 39](#_Toc182465695)

[**4.3.3.5** **Cycle de vie incrémentale** 40](#_Toc182465696)

[**4.3.4** **Comparaison de modèle de cycle de vie** 41](#_Toc182465697)

[**4.3.5** **Choix et justification** 42](#_Toc182465698)

[**4.4** **Méthodologie de conception et technologies utilisées** 43](#_Toc182465699)

[**4.5** **Présentation de UML** 43](#_Toc182465700)

[**4.5.1** **L’aspect choisi** 43](#_Toc182465701)

[**4.5.2** **L’UML** 44](#_Toc182465702)

[**4.5.2.1** **Historique** 44](#_Toc182465703)

[**4.5.2.2** **Définitions** 45](#_Toc182465704)

[**4.5.2.3** **Les caractéristiques de ses diagrammes** 46](#_Toc182465705)

[**4.5.2.4** **Ses Objectifs** 46](#_Toc182465706)

[**4.5.2.5** **UML face à la programmation** 47](#_Toc182465707)

[**4.5.2.6** **Les points forts et points faibles d’UML** 47](#_Toc182465708)

[**4.5.2.7** **Les grandes lignes essentielles de l’UML** 48](#_Toc182465709)

[***a. Diagramme de cas d’utilisation UML*** 50](#_Toc182465710)

[***b. Diagramme de séquence*** 51](#_Toc182465711)

[***c. Diagramme de classes*** 52](#_Toc182465712)

[***d. Diagramme d’activités*** 54](#_Toc182465713)

[***d. Diagramme de déploiement*** 55](#_Toc182465714)

[**4.5.2.8** **Choix des outils pour la modélisation en UML** 56](#_Toc182465715)

[**Chapitre 5 : Analyse du projet et étude de l’existant** 57](#_Toc182465716)

[**5.1** **Analyse des besoins** 57](#_Toc182465717)

[**5.1.1** **Besoin fonctionnel et non fonctionnel** 58](#_Toc182465718)

[**5.2** **Analyse de faisabilité** 59](#_Toc182465719)

[**5.3** **Analyse de l’existant** 60](#_Toc182465720)

[**5.3.1** **Existant Matériels** 61](#_Toc182465721)

[**5.3.2** **Existant Logiciel** 62](#_Toc182465722)

[**5.5** **Solution proposée** 63](#_Toc182465723)

[**5.6** **Solution retenue** 64](#_Toc182465724)

[**Chapitre 6 : Conception du projet** 65](#_Toc182465725)

[**6.1** **. Présentation de l’outil de conception** 65](#_Toc182465726)

[**6.1.1** **WinDesign** 65](#_Toc182465727)

[**6.1.2** **Autres outils de conception** 65](#_Toc182465728)

[**6.2** **Choix d’outils de conception** 65](#_Toc182465729)

[**6.3** **Règle de gestion** 66](#_Toc182465730)

[**6.4** **Dictionnaire de données** 67](#_Toc182465731)

[**6.5** **Modélisation** 68](#_Toc182465732)

[**6.5.1** **Schéma de diagramme de cas d’utilisation globale** 68](#_Toc182465733)

[**6.5.2** **Spécification détaillée des cas d’utilisation d’un système** 69](#_Toc182465734)

[**6.6** **Diagrammes de séquences** 71](#_Toc182465735)

[**6.6.1** **Diagramme de séquence de cas d’utilisation «S’Authentifier»** 71](#_Toc182465736)

[**6.7** **Diagramme d’activité** 71](#_Toc182465737)

[**6.8** **Diagramme de classe du projet** 73](#_Toc182465738)

[**Chapitre 7 : Spécification des Outil de Réalisation** 75](#_Toc182465739)

[**7.1** **Présentation des technologies utilisées** 75](#_Toc182465740)

[**7.1.1** **Le langage de programmation PHP** 75](#_Toc182465741)

[**7.1.2** **Framework PHP** 76](#_Toc182465742)

[**a. *Définition*** 76](#_Toc182465743)

[***b. Avantages et Inconvénients*** 76](#_Toc182465744)

[**7.2** **SGBD** 78](#_Toc182465745)

[**7.2.1** **Définition d’une base de données** 78](#_Toc182465746)

[**7.2.2** **Utilité d’une base de données** 78](#_Toc182465747)

[**7.2.3** **Choix du SGBD** 79](#_Toc182465748)

[**7.3** **Caractéristique des SGBD** 80](#_Toc182465749)

[**7.4** **Points forts et faibles des SGBD** 80](#_Toc182465750)

[**7.4.1** **PostgreSQL (SGBD choisi) :** 80](#_Toc182465751)

[**7.4.2** **MySQL :** 81](#_Toc182465752)

[**7.4.3** **XAMPP :** 81](#_Toc182465753)

[**7.5** **Présentation du Framework Laravel** 82](#_Toc182465754)

[**7.5.1** **Introduction** 82](#_Toc182465755)

[**7.5.2** **Historique** 82](#_Toc182465756)

[**7.5.3** **Historique des versions** 82](#_Toc182465757)

[**7.5.4** **Fonctionnalités** 83](#_Toc182465758)

[**Chapitre 8 : Mise en œuvre et implémentation** 84](#_Toc182465759)

[**8.1** **Architecture logicielle** 84](#_Toc182465760)

[**8.2** **Architecture matérielle** 87](#_Toc182465761)

[**8.3** **Architecture microservice** 88](#_Toc182465762)

[**8.3.1** **Introduction** 88](#_Toc182465763)

[**8.3.2** **La différence entre Architecture monolitiques et Architecture microservices.** 88](#_Toc182465764)

[**8.3.3** **Les avantages clés des microservices** 89](#_Toc182465765)

[**8.3.4** **Les inconvénients des microservices** 89](#_Toc182465766)

[**8.3.5** **Conclusion d’architecture microservice** 90](#_Toc182465767)

[**Chapitre 9 : Présentation de l’application** 91](#_Toc182465768)

[**9.1** **Présentation des interfaces** 91](#_Toc182465769)

[**9.1.1** **Front office** 91](#_Toc182465770)

[**9.1.2** **Back office** 91](#_Toc182465771)

[**9.2** **Présentation de l’application web** 91](#_Toc182465772)

[**9.2.1** **Interface de l’application** 92](#_Toc182465773)

[**9.2.2** **Liste des agents dans la centrale** 93](#_Toc182465774)

[**9.2.3** **Interface de choix de rapport** 94](#_Toc182465775)

[**9.2.4** **Formulaire du rapport** 94](#_Toc182465776)

[**CONCLUSION** 96](#_Toc182465777)

[**BIBLIOGRAPHIQUE** xv](#_Toc182465778)

[**WEBOGRAPHIE** xvi](#_Toc182465779)

[**ANNEXE** xvii](#_Toc182465780)

[**1** **Installation Laravel** xvii](#_Toc182465781)

[**a.** **Prérequis :** xvii](#_Toc182465782)

[**b.** **Étapes pour installer Laravel :** xvii](#_Toc182465783)

[**2** **Qu’est-ce que WampServer ?** xix](#_Toc182465784)

[**3** **Comment installer WampServer ?** xix](#_Toc182465785)

[**4** **Comment Administrer une base de données dans WampServer ?** xx](#_Toc182465786)

[**RESUME** xxi](#_Toc182465787)

[**ABSTRACT** xxi](#_Toc182465788)

# **INTRODUCTION**

Des maintenant, l’informatique c’est l’une de principe de développement technologique qui s’évolue chaque jour et devient le plus utiles dans les sociétés, l’entreprise ainsi que dans les Universités.

Pour obtenir un diplôme de licence en informatique, les étudiants sont obligés de faire un stage trois mois dans une entreprise. Leurs tâches sont non seulement de concevoir et de développer une application, mais aussi d'appliquer les cours théoriques en cours dans le monde professionnel.

Le En tant qu’une société en plein développement, elle veut profiter des avantages apportés par les nouvelles technologiques, notamment pour da gestion. Ce procédé sert généralement à des fins décisionnelles afin de réduire les temps dépens. Il existe déjà des outils comme Microsoft Excel pour effectuer ce rôle, mais au vu de la spécification des exigences, la révolution informatique nous offre des options le plus poignantes pour suive la gestion de rapport de production par exemple « Comment rendre automatique la production des rapports ? ».

Afin de mieux décortique le sujet et ainsi répondre aux questions axées dessue, cet ouvrage se présente en trois grandes parties, à savoir : Une Présentation Générale, qui présenter en détails les entités concernés par les présent d’ouvrage ; Une partie Analyse et Conception, qui explique en quoi consiste les besoins et les infrastructure au sein de rapport électrique mais également comment les donnée ont été traitées pour aboutir à une base de donnée et un cahier des charges ; enfin une partie réalisation qui argumente sur le choix de technologique, la mise en œuvre et implémentation de l’application suive de sa représentation.

**PARTIE I – PRESENTATION GENERALE**

# **Chapitre 1 : Présentation de L’EMIT**

* 1. **Historique**

L’Ecole de Management et d’Innovation Technologique (EMIT) est une école universitaire publique pluridisciplinaire, rattachée à l’Université de Fianarantsoa. La grande maturité au niveau de l’enseignement et la compétence des étudiants sortant de l’établissement ont permis aux dirigeants sous l’approbation du Ministère la conversion du Centre en Ecole au sein de l’Université de Fianarantsoa par le Décret N°2016-1394 du 15

Novembre 2016.L’EMIT prépare d’une part le diplôme de Master en deux mentions en trois parcours et d’autre part le diplôme de License en trois mentions en cinq parcours.

Auparavant, elle a été connue sous le nom du Centre Universitaire de Formation Professionnalisante (CUFP), crée par le Décret N°2005-205 du 26 Avril 2005 et dispensait le diplôme de License professionnelle en Administration ainsi qu’en Informatique. Mais avant cela, elle a été Rectoral N°99-23/UF/R du 10 Mars 1999 qui formait de diplôme de Technicien Supérieur.

L’EMIT a été sélectionnée « Meilleur Etablissement » pendant le Salon de la Recherche organisé par l’Organisation Internationale du Travail les 5 et 6 Juillet 2017. Depuis l’année universitaire 2013-2014, l’école est basculée totalement vers le système Licence,

Master et Doctorat (LMD). Toutes les offres de formation dispensée à l’EMIT sont habilitées par le Ministère de l’Enseignement Supérieur et de la Recherche Scientifique. L’école propose huit parcours repartis en trois mentions, représentés sur le tableau1.1 :

Tableau 1.1 : Tableau récapitulatif

|  |  |  |  |
| --- | --- | --- | --- |
| **Cycle** | **Mention Management** | **Mention Informatique** | **Mention Relation Publiques et Multimédia** |
| Licence | Administration Economique et Sociale | Développement d’Application Internet/Intranet | Communication Multimédia |
|  |  | Conception, Intégration et Gestion des Systèmes d’information | Relation Publiques et Communication Organisationnelle |
| Master | Management Décisionnel | System d’Information, Géomatique et Décision | Relation Publiques et Multimédia |
| Modélisation Ingénierie Informatique |

* 1. **Partenariat**

L’EMIT travail en collaboration avec plusieurs laboratoires de recherche d’entreprises et d’autres écoles et universités. Parmi les organisations partenaires, citons à titre d’exemple les laboratoires de recherche tels que le LIMAD, LIMOS, IRD, CNRE, SPAD, Espace Dev, LRI et l’UPR-Green à travers le CIRAD.

Pour ce qui est des écoles et des universités partenaires, il y a entre autres : l’EDMI, l’Université de Toulouse Paul Sabatier, Université de Montpellier 2, l’ENI,

GOUVSOMU, IOGA, l’Université de Clermont Auvergne, ESMIA, Université de Mahajanga, ISSTM et l’Université de Fianarantsoa.

L’Ecole est également en partenariat avec plusieurs entreprises, notamment dans le cadre des stages à travers chaque parcours, telles que les entreprises Etech consulting, Orange,

Lazan’i Betsileo, STAR, Alliance Française de Fianarantsoa, TELMA, BFV-SG, Bank of Africa(BOA), BNI Madagascar, Nelli Studio, YMAGOO, PREMIYA, JIRAMA, les assurances NY HAVANA, MAMA et ARO.

Des organismes gouvernementaux sont également partenaires de l’EMIT : la Région Haute Mahatsiatra, le Ministère de l’enseignement Supérieur et de la Recherche Scientifique, le Ministère de l’Education Nationale, le Ministère des Travaux Publics, les Ministère des Finances et du Budget, le Ministère du Tourisme, le Ministère des Transports et de la Météorologie, le Ministère de la Poste, de la Télécommunication et des Technologies

Numériques, la Banque Centre de Madagascar, le Foibe Taosaritanin’i Madagasikara(FTM), l’INSTAT.

* 1. **Cycle de licence**

L’école possède trois (03) mentions pour (05) parcours en cycle Licence. Les étudiants toutes mentions confondues doivent effectuer un voyage d’études d’insertion en entreprise pour les premières années de Licence(L1), un stage de réalisation en entreprise avec un rapport de stage soutenu pour les étudiants en deuxième année de Licence(L2) et un stage de fin d’études suivi de la soutenance d’un mémoire pour les étudiants en troisième année de Licence(L3). La formation sure trois années universitaires et à la fin de la formation, les étudiants obtiennent des diplômes de Licence.

* + 1. **Licence professionnelle en développement d’application Internet Intranet (DA2I)** 
       1. **Condition générale**

La condition d’accès en première année de Licence se fait par voie de concours pour les élevés titulaires du diplôme de Baccalauréat General Scientifique (Série C et Série D), Baccalauréat Technique Professionnelle et Baccalauréat Technique Technologique (Filière Industrielle, Maintenance Automobile, Ouvrage Métallique, Génie Civile).

* + - 1. **Stages**

Les étudiants dans la Formation de Licence en Développement d’Application Internet Intranet doivent effectuer :

* Un stage d’insertion en entreprise en L1 ;
* Un stage de réalisation suivi d’un rapport de stage en L2 et d’une soutenance ;
* Un stage de fin d’études suivi de la soutenance d’un mémoire en L3.
  + - 1. **Diplôme**

A la fin de la Formation, les étudiants obtiennent un diplôme de Licence en Développement d’Application Internet Intranet (DA2I).

* + - 1. **Compétences**

A l’issue de la formation, les étudiants sont compétents en :

* Administration des bases de données ;
* Administration des réseaux et systèmes informatiques ;
* Développement d’application client/serveur ;
  + 1. **Licence professionnelle en administration économique et sociale (AES)** 
       1. **Condition générale**

Dans la Formation de Licence Professionnelle en Administration Economique et Sociale, la condition d’admission se fait par un test de niveau. Ainsi, les élèves titulaires du diplôme de Baccalauréat General de toute série (A, C, D), Baccalauréat technique G1 et G2 peuvent être accéder dans la formation. Enfin, la formation a pour durer de 03 années universitaires.

* + - 1. **Stages**

Les étudiants dans la Formation Administration Economique et Sociale doivent effectuer :

* Un stage d’insertion en entreprise ou voyage d’études en première année de Licence (L1) ;
* Un stage de réalisation suivi d’un rapport de stage en deuxième année de Licence(L2) ;
* Un stage de fin d’études suivi de la soutenance d’un mémoire en troisième année de Licence(L3).
  + - 1. **Diplômes**

A la fin de la Formation, les étudiants obtiennent un diplôme de Licence en Administration Economique et Sociale.

* + - 1. **Compétences**

A l’issue de la Formation, les étudiants ont les compétences ;

* Assister le Directeur General, le Directeur des Ressources Humaines et le Directeur Administratif et Financier.
* Gérer des Ressources Humaines.
* Gérer une Entreprise et un projet.
  + 1. **Licence professionnelle en relations publiques et multimédias (RPM)** 
       1. **Condition générale**

Dans la Formation de Licence Professionnelle en Relations Publiques et multimédia :

* La condition d’admission : Test de niveau ;
* La condition d’accès : Baccalauréat général de toute série (A, C, D), BAC G1 et G2 ;

La durée de la formation : 03 années universitaires

* + - 1. **Stages**

Les étudiants dans la Formation Relations Publiques et Multimédia doivent effectuer :

* Un stage d’insertion en entreprise ou voyage études en première année de Licence (L1) ;
* Un stage de réalisation suivi d’un rapport de stage en deuxième année de Licence (L2) ;
* Un stage de fin d’études suivi de la soutenance d’un mémoire en troisième année de Licence (L3).
  + - 1. **Diplômes**

A la fin de la Formation, les étudiants obtiennent un diplôme de Licence en Relations Publiques et Multimédia.

* + - 1. **Compétences**

A l’issue de la Formation, les étudiants ont les compétences :

* Rédiger un article dans un journal.
* Occuper une poste d’un technicien de presse.
* Travailler dans la revue de presse.
  1. **Cycle de master**

L’Ecole de Management et d’Innovation Technologique (EMIT) de L’université de Fianarantsoa possède deux (02) cycles de Master Professionnel dont :

* Le Master Professionnel en Management Décisionnel (MD) ;
* Le Master Professionnel en Système D’Information-Géomatique-Télédétection et Décision (SIGDTD).
  + 1. **Master professionnel en management décisionnel**

Au niveau de l’EMIT, il existe le Second Cycle (MASTER) en Management Décisionnel.

* + - 1. **Condition d’accès**
* En S7 : Sur sélection de dossier après Licence en Administration Economique et Sociale, en Gestion ou en Economie.
* En S9 : Apres validation des crédits déjà obtenus par les responsables
  + - 1. **Durée de formation**

La durée de formation est quatre semaines (Deux années universitaires).

* + - 1. **Objectifs**

Le Master Management décisionnel a pour objectif d’équiper l’apprenant avec les outils de prise de décisions en matière de management et de leur donner les compétences requises dans ce domaine. Comme le management a besoin de se conformer en permanence aux diverses nouvelles exigences du marché, l’enseignement doit alors toujours viser à mettre à jour les connaissances de l’apprenant par la formulation de programmes de cours qui tiennent compte de ces nouveautés. Ainsi l’objectif majeur du parcours est d’avoir des acteurs de haut niveau en management décisionnel. Enfin, les parcours préparent des cadres capables de gérer et créer un projet de développement économique régional et national.

* + - 1. **Insertion professionnelle**

Les sortants de la mention Management Décisionnel travaillent dans les secteurs privés et publics des différentes régions de Madagascar en tant que chefs de conduire de travaux d’enquêtes communautaires, concepteurs de projets, chef de services ou de bureaux, chefs ou directeurs d’entreprises. Parmi ces secteurs où ils occupent des postes en conformité avec leur formation, on peut citer :

* Les Organismes (Organisation Non Gouvernemental) ;
* Les Ministères (Ministre de Finances et des Budgets, Ministères du Tourisme,
* Ministère de l’Education Nationale, …) ;
* Les Directions Régionales (Direction Régionale de l’Environnement et de Direction Régionale de l’Eau, …) ;
* Les Communes Urbaines ;
* Les sociétés d’assurances (NyHavana, ARO et MAMA) ;
* Les Sociétés Commerciales et Industrielles.

Enfin, les étudiants obtenus le diplôme de master en Management Décisionnel sont capables de :

* Créer une petite entreprise ;
* Monter un projet de développement rural ;
* Gere un grand projet.
  + 1. **Master professionnel en système d’Information géomatique-télédétection et décision**

Au niveau de CUFP, il existe aussi le Second Cycle (MASTER) en Système d’information géomatique-télédétection et décision.

* + - 1. **Condition d’accès**
* En S7 : Sur sélection de dossier après Licence en Informatique, Gestion, Economie, Mathématiques et Physique.
* En S9 : Apres validation des crédits déjà obtenus par les responsables.
  + - 1. **Durée de formation**

La durée de formation est quatre semestres (Deux années universitaires).

* + - 1. **Objectifs**

Le parcours Système d’Information, Géomatique et Décision a pour objectif de donner un panorama des recherches actuelles et émergeantes dans le domaine des systèmes d’aide à la décision. Les systèmes informatiques et la géomatique connaissent en effet actuellement une forte évolution (voire mutation) avec d’une part les grilles de calcul et d’autre part, les multiples appareils mobiles intégrant des systèmes informatiques de plus en plus performants et complexes. Ces systèmes informatiques intégrant au parallélisme massif ou /et une mobilité des composants sont un défi pour le génie logiciel qui doit apporter de nouvelles méthodes et des outils de production de logiciel pour la description de l’architecture de ces systèmes complexes et pour leur validation et /ou certification. En plus, la montée en puissance des techniques en géomatique va permettre de prendre une bonne décision à partir des données spatiales et temporelles dans différents domaines.

* 1. **Présentation de service**

A part des formations, il y a aussi des prestations de service, telles que :

* Conception et réalisation ;
* Conception et développement d’application informatique sur mesure ;
* Installation et maintenance des Systèmes informatiques : matériels et logiciels ;
* Assistance et conseil en matière Informatique ;
* Bureautique.
  1. **Organigramme de l’EMIT**

L’organigramme de l’Ecole de Management et d’Innovation Technologique montre des conseils (établissement, scientifique), une direction, un collège des enseignants, des chefs de mention et services existants dans l’école, il s’est présenté par la figure 1.1

*.*

CONSEIL

D’ETABLISSEMENT

DIRECTION

CONSEIL SCIENTIFIQUE

COLLEGE DES

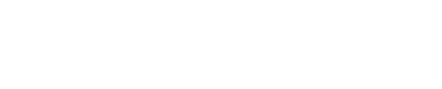
ENSEIGNANTS

MENTIONS

SECRETARIAT PRINCIPAL

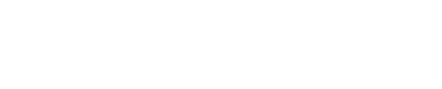
GRADE DE LICENCE

GRADE DE MASTER



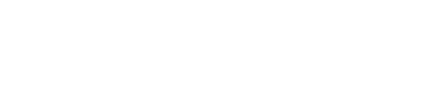
MENTION

AES



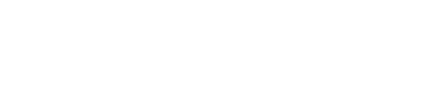
MENTION

RPM



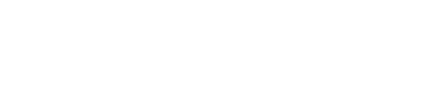
MENTION

DASI



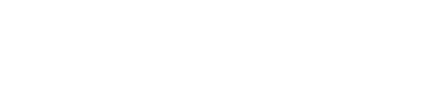
MENTION

MD



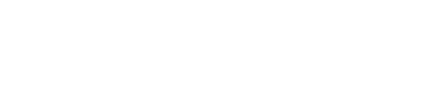
MENTION

SIGTO



SERVICE

SCOLARITE



SERVICE

COMPTABLE

Figure 1.1 : organigramme d’EMIT

# **Chapitre 2 : Présentation de JIRAMA**

* 1. **Historique**

A Madagascar, l’exploitation et la distribution d’eau et d’électrique a connue deux (02) périodes bien distincts celle d’avant et celle après la nationalisation de la société en 1975.

* + 1. **Avant 1975**

La première distribution publique d’eau et d’électricité s’est fait Antananarivo vers 1899. En 1905, une convention a été signée par le Général Gallieni accordant à la Société d’Etat Civil (SEC) le privilège d’adduction d’eau et d’électricité de Madagascar (SEEM) qui, en plus d’Antananarivo, ravitaille en électrique les régions de Toamasina, de Fianarantsoa, d’Antsirabe et de Diego.

Le 29 septembre 1952, à côte de SEEM la Société d’Eau et d’Energie de Madagascar a été créé suite à un accord entre l’Etat de SEM sur la distribution d’eau. De ce fait, ce deux (02) société s’entraine mutuellement de sorte qu’à l’intérieure d’une ville, l’une fournissait l’eau et d’autre l’électricité.

Le 04 Février 1974, l’Etat a confirmé son monopole en créant la Société Malgache de l’Eau et de l’Electricité (SMEE) qui prend en changer la totalité des emplois de la SEEM. Plus tard, l’eau a été regroupé dans une nouvelle Société d’Intérêt Nationale de l’Eau et d’Electricité (SINEE) qui prend en charge la totalité des emplois de la SEEM.

* + 1. **De 1975 jusqu’à nos jours**

Le 31 Octobre 1975, la SINEE a été ses anciennes attributions ont été confiées à la Jiro sy Rano Malagasy (JIRAMA). Elle a été conçue conformément à l’ordonnance n°75024 du 17 Octobre 1975. Le 29 Juin 1977, l’Etat a fusionné le SEEM, la SEM et la Gérance Nationale de l’Eau (GNE) au sein de la JIRAMA. A cette époque, la JIRAMA était membre fondateur de l’association des Electriciens des Iles de l’Océan Indien (AOIE).

En 1983, La JIRAMA a signé un accord avec les bailleurs de fonds étrangers au titre de projet ENERGIE I pour les grandes réalisations telles qu’Andekaleka, Namorona et les travaux d’équipement comme le renforcement en eau à Mandroseza.

En 1996, pour le développement de son service, la JIRAMA a signé un nouvel accord avec le Banque Mondial pour assure le projet ENERGIE II. Elle viser comme objectif L’amélioration de perspective de croissance durable de Madagascar en assurant un approvisionnement en électricité adéquat à longue terme, ainsi qu’a un meilleur service pour la population urbaine et rurale.

En 2005, un courant de mise en place d’une Task-Force dont la mission était de parvenir à l’i dentification de l’option de gestion de mieux adaptée pour que JIRAMA puisse répondre au grand enjeu de développement de Madagascar a été signer par l’Etat malgache avec une société d’origine allemande nommée LAHMAYER le contrat a pris in de mars 2007.

* + 1. **Statique juridique**

La JIRAMA est une Société Anonyme de droit commun dont le capital est détenu entièrement par l’Etat Malgache. Elle est dirigée par un conseil d’Administration est composé des représentants de l’Etat notamment de ministère de tutelle et des représentants des employés et le DG est nommée par le Ministère chargé de l’énergie.

Son siège social se trouve : 19 Rue Rainandriamampandry Ambohijatovo, BP 200, Tana (101). Sa dénomination sociale est Jiro sy Rano Malagasy. Cette société est identifiable fiscalement sous le Numéro d’Identification Fiscal (NIF) : 10500311 et inscrire aux Registre de Commerce des Société (RCS) n°204 BOO 533. Elle est encore dans le cadre de la direction publique, son siège social est joignable par le téléphone sous ne numéro 2220031. Quant à la Direction Interrégionale JIRAMA de Fianarantsoa, elle se situe à Tsianolondroa sous la BP56 JIRAMA FIANARANTSOA. Elle est joignable au téléphone n° 020 75 505 22/020 75 505 23 ou aussi par une site web : [www.jirama.mg](http://www.jirama.mg)

* + 1. **Les Objectifs**

La JIRAMA œuvre pour la réalisation des objectifs gouvernementaux en matière d’alimentation en eau potable et d’électrification dans tout Madagascar tout en respectant les normes en vigueur du pays. Elle constitue un besoin fondamental et vital sur le plan économique et sur le plan social.

* + - 1. **Objectif général**

L’objectif général vise à assumer un développement rapide et une croissance soutenue par ses activités principalement en termes de nombre d’abonnés telles que : les productions et les vents, les installations de bases par la satisfaction simultanée et de manière équilibrée de l’ensemble de ses partenaires.

* + - 1. **Objectifs Spécifiques**

La Société a pour objectifs spécifique les suivants : d’abord, offrir à ses clients la meilleure qualité avec une garantie de certification, et récupération les gains de productivités et l’amélioration des prix. Ensuite, donner des réelles possibilités d’épanouissement et d’évolution aux personnels ; payer des dividendes aux harmonisant les actions avec ses partenaires nationaux et régionaux tels que la Société civile et les collectivités.

* 1. **Les activités de la JIRAMA**

La société Jiro si Rano Malagasy exerce deux branches d’activités principales bien distinctes à savoir la branche électricité et la branche eau. Alors ses principales activités concernent la production, transport et la distribution d’énergie électrique, vente eau et électricité ainsi que la production, le traitement et la distribution d’eau potable.

* + 1. **La Branche électricité**

La société exerce dans ce domaine une activité complète dont la production, la commercialisation, le transport et la distribution de l’énergie électrique.

* + 1. **La production**

De toutes les diverses sources d’énergie le JIRAMA produit de l’énergie électrique en provenance des deux types de centrale, d’une part la centrale hydroélectrique : c’est une groupe turbine qui transforme l’énergie hydraulique en énergie électrique. Exemple : le Centrale Namorona (Ranomafana), Manandray (Talata Ampano) et d’autre part, la centrale thermique : elle produit de l’énergie électrique par l’intermédiaire d’une machine thermique motrice alimentée soit par gasoil soit par du fuel lourd. La qualification des productions des courants se fait par trois niveaux dont :

* Pour la Haute TENSION (HT) : 35 000 V à 50 000 V ;
* Pour la moyenne Tension (MT) : 5 000 V à 35 000 V et ;
* Pour la Base Tension (BT) : 220 V à 380 V

Le type de centrale thermique à monteurs diesel est plus utilisé par la société et pour des raisons thermiques et environnementales, les centrales sont installées dans des endroits éloignés des agglomérations, Exemple : centrale thermique Ampopoka et Ankidona.

* + 1. **La Commercialisation**

La société JIRAMA exerce une activité de vente de l’énergie électrique qu’elle produit. Elle a affecté d’une côte le service commercial pour la prise en charge de la clientèle et le recouvrement, et de l’autre côté, le service technique électrique pour garantie de branchement jusqu’à la réalisation du contrat pour le débranchement Le computeurs est un instrument qui sert à enregistrer de consommation.

* + 1. **Le transport et la distribution**

Après la production, l’énergie électrique de haut tension (HT) est convertie en moyenne tension (MT) dans une sous- station (Poste de Transformation et de répartition) et ensuit en basse tension (BT) par des postes de transformation MT/BT, avant de la distribuer. Les centres d’exploitation acheminent l’électricité produit à l’aide d’une ligne de transport. Cette distribution d’énergie est assurée par le service électricité (SDE). Elle consiste à la réalisation des travaux de branchement en électricité. Les niveaux de tension qui existe actuellement à la JIRAMA sont de 5k V – 15k V – 30k V – 35k V – 63k V. La JIRAMA propose deux lignes électriques de réseau de distribution à savoir les lignes aériennes ou en espace (suspendue sur des poteaux électrique) et les lignes souterraines (enfouies sous terre).

* + 1. **Branche d’eau**

Comme pour la branche électricité, nous allons considérer les activités d’exploitation de traitement, de commercialisation et de distribution en eau potable.

* + 1. **L’exploitation**

Pour suivre, tous les êtres vivants ont besoin d’eau. C’est en ce sens que la Société JIRAMA exploite l’eau dans le but de satisfaire la demande de fourniture d’eau. Le monde d’exportation de l’eau se fait en trois étapes, d’abord la recherche de source d’eau soit l’eau en surface, soit l’eau souterraine ; ensuite le mode d’adduction par système granulaire ou par le système de pompage et enfin le système de traitement de purification et de désinfection (traitement physico-chimique, ou traitement bactériologique).

* + 1. **La commercialisation**

La société garantit le vent en eau potable en attribuant les tâches au service commerciale et au service technique eau. En effet, ces services sont responsables de la prise en charge des demandes des fournitures en eau, et de gouvernement jusqu’à réalisation du contrat Toutefois, les procédures des prises en charge et de recouvrement sont pareilles tant pour l’eau que pour l’électricité.

* + 1. **La Distribution**

La Distribution d’eau est assurée par les services exploitation eau de la société. Elle conçoit à l’exécution des demandes de fourniture d’eau et des travaux des branchements du château d’eau jusqu’à l’installation de computeur au domicile des consommateurs. Les monde de transport de l’eau est sous terrain.

* 1. **Structure organisationnelle de la JIRAMA**

Actuellement, comme le montre la figure 2., la société est organisée en différents organes pour offrir la qualité de service que les clients attendent.

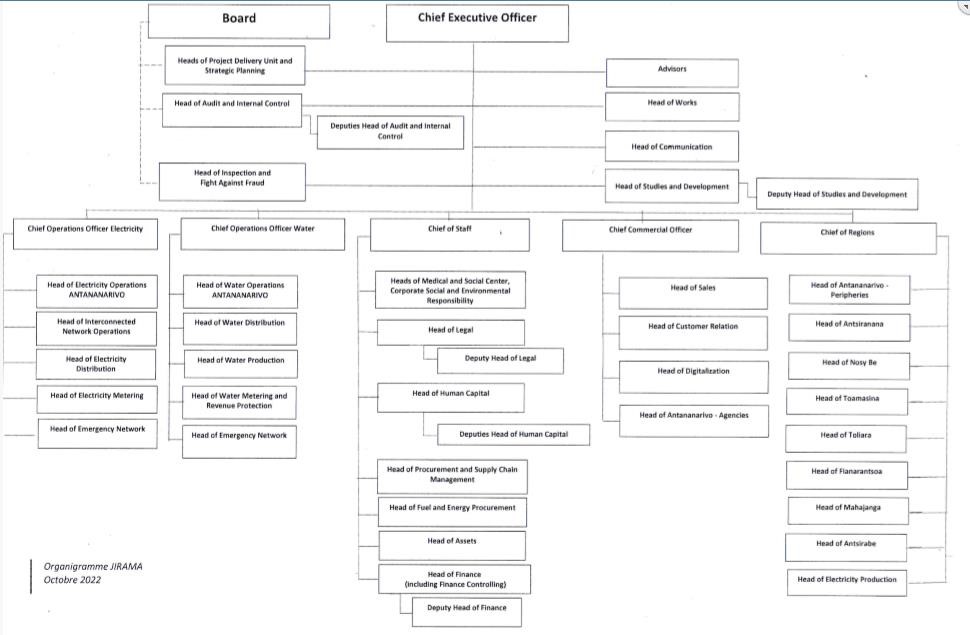


Figure 2.1: Structure organisationnelles

Source : [www.jirama.mg](http://www.jirama.mg)

La Direction Interrégionale de la JIRAMA Fianarantsoa est rattachée directement à la Direction Générale située à Tananarive, et assure le ravitaillement de production, distribution et commercialisation de l’eau et de l’électricité dans les différents sous groupement et secteur existants. Toutes les décisions concernant les ressources humaine telles que, les recrutements, l’affectation, la formation, le reclassement et le licenciement sont prises par le Direction Interrégionale. En outre, les chefs de sous groupements et secteurs doivent communiquer à la Direction Interrégionale toutes les opérations réalisées dans leurs zone respectives telles que l’encaissement, les dépenses, la cession d’immobilisation ainsi que les problèmes liés aux activités de l’entreprise.

* 1. **Les missions de chaque service**

Les Entreprise JIRAMA consistent des plusieurs missions à chaque leurs services comme la mission ci-dessous :

* **Le Centre Médical-Social (CMS)** consiste à couvrir les soins de tous les travailleurs de la JIRAMA ainsi que leurs familles. De plus, il assura la santé de stagiaire en cas de maladie grave.
* **Le Service Ressources Humaines (SRH)** assurer l’administration et la gestion du personnel c’est-à-dire gère le différent mouvement du personnel, l’embauche, la mutation, le licenciement, le retraite, le décès, le démissions, la sanction, etc.
* **Le service Logistique et Environnement (SLE)** occupe les affaires logistiques contentieux et responsable des bâtiments et des véhicules.
* **Le Service Télécom (STEL)** occupe la n radio, le téléphone et le dépannage électronique.
* **Le Service Système Informatique (SSI)** assure le traitement des toutes les activités de la direction et la maintenance des machines de la société.
* **Le Services Contrôle Coordination (SCC)** contrôle la réception technique du branchement eau et électricité sans compteur et contrôle aussi toutes les signatures.
* **Le Service Environnement et Contrôle Carburant (SECC)** assure une cadre propice pour les déchets hydrocarbure fin de ne pas polluer l’environnement.
* **Le Service Commerciale (SCOM)** assure la gestion des ventes d’eau et de l’électricité, c’est-à-dire la vente envers les clients particulier et ordinaires et la vents inter-service.
* **Le Services Distribution Electricité (SDE)** occupe des réseaux de distribution de l’électricité y compris les travaux d’entretien et des branchement neufs pour nouveau clients.
* **Le Service Exploitation Eau (SEXO)** assure la protection et la distribution d’eau potable.
* **Le Service Production Electricité (SPE)** occupe la protection d’électricité et la maintenance, l’entretien des groupes électrogènes, des groupes hydrauliques ainsi que les réseaux de transports et les réseaux interconnectés.
* **Le Service Financier (SFI)** ou comptabilité règle le compte et toutes les affaires administratives.
* **Le Services Approvisionnement (S. APPRO)** assure l’achat et la gestion stock. Dans ce service, il y a quatre chefs dont : Le chef de Service assure la bonne marche des services d’approvisionnement de la direction interrégional ; le chef de division assiste le chef de service, le remplace en cas d’absence, fait le suivi du traitement de la demande d’achat et des articles en stock ; le chef de groupe magasin gère les stocks, contrôle et fait le suivi des mouvements, établit mensuels, prépare les inventaires.
  1. **Présentation des centres d’exploitation de la DIR/FR**

Les centre d’exploitation de DIR/FR se divisent en 22 secteurs et en 22 centres d’exploitation dont 15 centre mixte (eau et électricité) et 6 centrale électrique compris les zones périphérique et 1 centre eau.

Tableau 2.1 : Centre d’exploitation de la DIR/FR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DIR** | **SOUS GROUPEMENTS** | **SECTEURS** | **SITE** | **ACTIVITES** |
| F  I  A | IMPUTATION DIRECTE | Ranomafana  Ambalavao  Ambohimasoa  Ikalamavony  Ampopoka/Ankidona | A 10  A 84  A 45  B 11  A 15 | Mixte  Mixte  Electricité  Mixte  Mixte |
| N | MANAKARA | Manakara  Vohipeno  Ikongo | A 79  A 59  B12 | Mixte  Mixte  Mixte |
| A  R  A | IHOSY | Ihosy  Betroka  Iakora  Ranohira  Ivohibe | A 97  B 37  B 19  B 25  A 85 | Mixte  Mixte  Electricité  Electricité  Mixte |
| N  T  S | FARAFANGANA | Farafangana  Vangaindrano  Midongy Sud  Vondrozo  Tangainony  Befotaka | A 78  A 66  A 98  A 67  A 99  A 18 | Mixte  Mixte  Mixte  Electricité  Eau  Mixte |
| O  A | MANANJRARY | Mananjary  Nosy Varika  Ifanadina | A 09  B 09  A 83 | Mixte  Electricité  Electricité |

# **Chapitre 3 : Présentation de Projet**

* 1. **Introduction**

Dans ce chapitre, nous allons décrire l’origine du projet, nous poserons la problématique, nous exposerons les objectifs et les besoins des utilisateurs de notre travail puis en fin rédigeons le cahier de charge.

* 1. **Origine de projet**

Les pratiques ont amélioré la situation de rapport électrique au sein de la JIRAMA Centrale. Un projet est un processus unique qui consiste à n ensemble d’activités coordonnées et maîtriser, comportant des dates de début et de fin, entreprise dans le but d’atteindre un objectif conforme à des exigences spécifique, telle que des contraintes de délai, de coût et de ressource.

La collection de la production électrique au sein de JIRAMA est la tâche qui provoque des grandes pertes de temps pour les responsables et entrainement aussi la lenteur de leur travail. L’accomplissement de leur travail est mal géré surtout à propos de ce rapport. Par conséquent, cette inefficacité de la saisie du rapport électrique va provoquer des problèmes pendant que les responsables effectuent leurs services. Une solution informatique s’avère donc nécessaire. C’est pour cette raison que nous avons proposé le réaliser cette application web afin de pouvoir faciliter la gestion de la production électrique.

* 1. **Problématique**

Plusieurs problématiques ont convoqué pendant le traitement de la production surtout l’insertion de la donnée dans l’application Microsoft Office (Excel), l’absence de système de la gestion du rapport électrique au sein de JIRAMA Centrale. Par conséquent, le temps du traitement s’avère un peu plus long vue le volume de données manipulées. De cette maniéré, une liste des problèmes est dégagée :

* Une grande perte de temps lors d’un rapport électrique
* L’absence du logiciel de la gestion de rapport de production électrique chaque jour, ils utilisent Microsoft office (Excel) à cette gestion mais il est assez difficile et on assiste à une insécurité des données car il y a des différents utilisateurs peu accéder à cette application.
* L’absence de consultation des données
  1. **Conduite de projet**

La conduit de projet ou gestion d’une projet aussi appelé management de projet est une démarche qui a pour but de structurer et assurer le bon déroulement d’un projet. Dans cette partie, nous allons voire les différentes étapes d’un projet.

#### **3.4.1** **Phase préparatoire**

Cette phase nous permet de prendre conscient du projet. Cette partie met en œuvre les différentes étapes à préparer avant la réalisation de notre projet, en partant de l’analyse de besoin du client, le rapport détaillé de notre projet. Voici les solutions que nous avons proposées à notre client :

* Créer une application web de gestion du rapport de la production électrique au sein de JIRAMA Centrale (architecture client/serveur).
* Installation d’un logiciel de gestion du rapport de la production électrique au sein de JIRAMA Centrale (architecture client/ serveur).

#### **3.4.2 Etude d’opportunité**

En quoi le projet est-il intéressant ? en se basant sur cette optique, l’étude d’opportunité vise à évaluer les avantages du projets une fois terminé. Elle répond à la question : Quels seront les bénéfices lorsque le logiciel sera place (lorsque le projet sera fini) ? Premièrement, le logiciel facilite le traitement du rapport électrique, il permet aussi de gagner assez de temps et de perfectionner les services (sans faute de calcule, etc…).

#### **3.4.3 Risque du projet :**

Pour mener à bien le projet, il faut d’abord éviter ce risque dans le tableau 3.1

Tableau 3.1 : Risque du projet

|  |  |  |  |
| --- | --- | --- | --- |
| **Les risques** | **Types** | **Impact** | **Action correctives** |
| Cahiers de charges | Risques non bloquant | Créer une ambiguïté ce qui influencera probablement la date de la livraison | Prévoir des réunions des point de validation avec l’encadrant au fur et à mesure du l’avancement du projet |
| Les pannes inattendues du matériel | Risque non bloquant | Ralentissement des travaux | Utilisé les autres matériaux disponibles.  Recours à une réparation rapide |
| Absence ou malade | Risque non bloquant | Ralentissement des travaux | Doubler l’effort et travailler un temps extra |

#### 

#### **3.4.4 Planning du projet et déroulement du stage**

Facteurs clé, le temps de réalisation d’un projet doit être géré à l’avance afin d’éviter le risque d’échec du projet ou de désaccord entre le maître d’œuvre et le maître d’ouvrage. Il peut être retranché le différent période avec l’étapes pratiquée.

Tableau 3.2 : déroulement de stage

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Juin | | Juillet | | | | Aout | | | |  | Septembre | | | Octobre | |
| Activités | S3 | S4 | S  1 | S  2 | S  3 | S4 | S  1 | S  2 | S  3 | S  4 | S  1 | S  2 | S3 | S  4 | S  1 | S  2 |
| Délimitation de thème | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Discussion et choix du projet |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Mise au point du projet |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Recherche documentaire |  |  | X | X | X |  |  |  |  |  |  |  |  |  |  |  |
| Rencontre des personnes |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |
| Enquêtes et entretient |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |
| Rédaction du mémoire |  |  |  |  | X | X | X |  | X | X | X | X |  |  |  |  |
| Conception et réalisation |  |  |  |  |  |  | X | X | X | X | X | X |  |  |  |  |
| Elaboration des essais |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |
| Préparation de soutenance |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |
| Soutenance de mémoire |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |

#### 

#### **3.4.5 Phase de création**

Phase opérationnelle de création de l’ouvrage, du lancement du projet et du développement par rapport au choix du besoin du client.

Dans cette partie, nous allons préparer les différents matérielles, logiciel, et autre ressource utile à la réalisation de notre projet.

Pour finir le projet à temps, nous devons répartir les différentes tâches. Alors avant des décider la réparation de ces tâches, nous allons déduire les différentes étapes qui peuvent se faire sans attendre, ni dépendre d’une autre tâche. Il faut donc récolter les différentes tâches utilises. On a rétabli les tâches comme suit :

* Délimitation de thème
* Discussion et choix du projet
* Mise au point du projet
* Recherche documentaire
* Rencontre des personnes
* Enquêtes et entretient
* Rédaction du mémoire
* Conception et réalisation
* Elaboration des essais

#### **3.4.6 Phase de fin du projet**

Mise en production de l’ouvrage. Avant d’entamer la fin du projet, nous devons mettre au point quelques étapes, comme l’installation du logiciel dans son environnement réel (mise en production), son utilisation, et sa maintenance.

#### **3.4.6.1 Mise en production**

Une fois terminé, il faut procéder à l’installation de notre logiciel dans son environnement réel. Il faut montre les différentes ressources d’installation, la configuration requise de l’ordinateur, ainsi ses différentes perspectives et informer, préparer, former et encadrer les utilisateurs.

#### **3.4.6.2 Maintenance**

Une fois installer, nous devons faire quelque vérification de la fonctionnalité de notre logiciel, de faire quelque maintenance pour son bon fonctionnement si besoin. Il faut donc faire un test de fonctionnalité pendant quelques jours et repérer les éventuels problèmes puis les résoudre.

## *a Maintenance évaluative*

La maintenance évolutive est utile pour l’évolution future de notre logiciel, par rapport à l’avancement technologique, ce qui implique que notre logiciel doit être maniable et souple pour s’adapter à la future évaluation pour plus option.

## *b Formation de l’utilisateur*

La manipulation de notre logiciel est un peu difficile pour notre client, alors il faut former les futurs utilisateurs pour qu’ils puissent profiter au maximum les capacités de notre logiciel et aussi pour éviter quelques problèmes concernant la manipulation du logiciel.

* 1. **Cahier de charge**
     1. **Définition**

Le cahier de charge résumer de manière contractuelle les diverses charges et conditions entre le maître d’œuvre et le maître d’ouvrage lors d’un projet.il spécifie entre autres les besoins des entreprises transforme en fonctionnalités. Dans cette partie, nous allons montrer les besoins fonctionnels et non fonctionnelle de la future application, de préciser les livrable et les risques les plus critiques du projet.

Le Cahier de charge est un document qui est attendu du maître d’œuvre par le maître d’ouvrage. Il s’agit d’un document de façon plus précise avec un vocabulaire simple. Dans la mesure où le maître d’œuvre est compétent pour proposer une solution technique, le cahier de charge fait apparaître les besoins d’une manière fonctionnelle, indépendamment de toute solution technique. Il s’agit aussi d’une part un document permettant de garantir au maître d’ouvrage que les livrables seront conformes à ce qui est écrit et d’autre part, le maître d’ouvrage modifie son souhait au fur et à mesure du projet et demande au maître d’œuvre des nouvelles fonctionnalités non prévues initialement.

* + 1. **Analyse de besoins**

La collection de donnée de fait chaque jour et devient difficile par le chef d’exploitation ou des Agents. La création d’application de gestion du rapport de production électricité de JIRAMA est un moyen de facilité cette gestion. Donc le rapport électricité a besoin d’une application web pour gérer ce rapport. L’analyse de besoins est un processus qui permet de transformer une idée floue en spécification précis des besoins servant de support à la spécification et de ses interfaces avec l’environnement.

Dans cette partie, nous allons intéresser aux besoins des utilisateurs à traves les spécification fonctionnelles et non fonctionnelles pour aboutir à une application de qualité selon les besoins des utilisateurs liés à ce projet.

* + 1. **Besoin Fonctionnels**

Les besoins fonctionnels représentent les fonctionnalités du système. Ce sont les besoins spécifiant un comportement d’entrée et sortie du système. Cette application doit permettre :

* **S’authentifier**
* **Les tâches de gestion de puissance installée**: Ajouter, Modifier, Suppression, Recherche puissance installée.
* **Le tâches de gestion de puissance dispo** : Ajouter, Modification, Suppression, Recherche puissance dispo.
* **Le tâche de gestion production** : Ajouter, Modification, Suppression, Recherche de production.
* **Les tâche de gestion d’Agent**: Ajout, Modification, Suppression, Recherche de pointe.
* **Les tâche de gestion de consommation gasoil**: Ajout, Modification, Suppression, Recherche de Consommation Gasoil
* **Les tâche de gestion de delestage**: Ajout, Modification, Suppression, Recherche d’entreprise
* **Les tâches de gestion de rapport**: Ajout, Modification, Suppression, Recherche
  + 1. **Besoins non fonctionnels**

Les besoins non fonctionnels sont les besoins en matière de performance, ils sont importants car ils décrivent les objectifs liés aux performances du système et aux contrainte de son environnement.

Ses exigences techniques sont exprimées sons forme d’objectifs spécifiques que doivent atteindre le système. Les besoins non fonctionnels sons importants car ils s’agissent de façons indirectes sur les résultats et sur le rendement de l’utilisateurs.

Ses exigences de qualité sont une facilité d’apprentissage, l’interface graphique de l’application doit être simple et intuitive. Les formulaires à remplir doivent utiliser des termes techniques métier et présentent certains choix pour les valeurs fréquentes.

Ses exigences de performance et sécurité devra être hautement sécurisée car les informations ne devront pas être accessibles à tous les personnels sauf à son utilisateur spécifié qui sont les destinateurs. Les recherches effectuées sur l’application ne doivent pas prendre beaucoup de temps, ainsi que le calcul de temps. Toutes activités sur l’application doivent être facilement atteintes mais il faut répondre aux exigences suivantes :

* **Fiabilité** : l’application doit fonctionner de façon cohérente sans erreur,
* **Les** **erreurs** : l’application doit les signalées par des messages d’erreur ;
* **Efficacité**: l’application doit permettre l’accomplissement de la tâche avec le minimum de manipulation ;
* **Sécurité**: l’application doit être sécurisé au niveau des donnée, authentification et contrôle d’accès et chaque utilisateur pour accéder à l’application, obligé de s’authentifier pour un nom d’utilisateur et un mot de passe, il ne pourra accéder qu’ils sont permis par son profil aux droits qu’ils sont affectés par l’administrateurs ;
* **La maintenabilité**: le code doit être compréhensible par simple lecteur, notamment en respectant les règles de gestion et nomes de développement.
* **Exploitabilité**: les impacts en termes de performances doivent être pris en compte lors de développement, ainsi que la consommation des ressource (mémoire, etc.) qui être minimisée.
* **Capacité de fonctionnelle et de convivialité** : les composantes développées doivent respecter les spécifications fournies par la JIRAMA. Le système doit être facilement utilisable et disposer d’interface conviviable, notamment par le respect des règles d’ergonomie de la JIRAMA.
  1. **Objectif**

Les développements de technologique du rapport de production électricité et la micro-informatique permet actuellement à toute entreprise, quelle que soit sa taille, d’informatiser ses opérations et sa gestion, Ceci à l’avantage de faciliter de rapport, le système manuel est encombrant, moins rapide et présent souvent des erreurs de manipulation. L’application devra tout d’abord être extrêmement faible. En effet, son domaine d’application concerne le cœur du système du rapport de la production électricité de JIRAMA et son utilisation quotidienne ne devra pas laisser placer à l’éventuel point faible.

L’objectif principale est la gestion du production électricité au sein du JIRAMA. Le développement d’une application web pour la gestion de rapport de la production électrique au sein de JIRAMA permettra l’amélioration et la spécification du travail administratif au sein de la production électrique, participe ainsi au progrès informatique

L’application devra notamment :

* Permettre d’insérer de nouveaux rapports
* Permettre de faire la gestion des rapports de la production électricité
* Permettre de contrôler le démarrage du rapport
* Faciliter la conception et l’automatisation du rapport
* Permettre de faciliter le travail du responsable de la production électrique (saisie de rapport)

Ainsi l’intérêt de ce travail est de créer une application web pour la gestion d’un rapport de la production électricité au sein de JIRAMA.

Parmi ces objectifs, le rapport de la production électrique consiste à développer et implémentation d’une application de gestion au sein de rapport qui permettra essentiellement l’amélioration et la simplification technique de travail.

* Manuel
* Automatisé

**PARTIE II – ANALYSE ET CONCEPTION DU PROJET**

# **Chapitre 4 : Méthodes et notations utilisées**

Avant l’analyse et la conception, il faut d’abord présenter la méthode utilisée pour obtenir un rendu réaliste qui aboutira à une base de données conforme aux attentes. Dans ce chapitre, nous allons présenter d’une part, les différents cycles de vie d’un logiciel et d’autre part, la méthode qui sera utilisée à la modélisation de notre système. Pour tous les logiciels, il existe un modèle pour vérifier si un logiciel est de qualité ou pas ce qui nous amène dans le domaine du génie logiciel. Définition : un domaine de l’ingénierie qui permet la conception, la réalisation et la maintenance des systèmes logiciels de qualité.

En principe, le « cycle de vie d’un logiciel » (en anglais \*software life cycle\*), désigne toutes les étapes du développement d’un logiciel, de sa conception à sa disparition. L’objectif d’un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement logiciel, c’est-à-dire la conformité du logiciel avec les besoins exprimés, et la vérification du processus de développement, c’est-à-dire l’adéquation des méthodes mises en œuvre.

Avant de créer une application web, il est strictement nécessaire de choisir un modèle pour le réaliser, cela dépend du projet à réaliser.

* 1. **Méthode utilisée**

Dans cette partie, nous allons voir les différentes méthodes utilisées dans un cycle de développement d’un logiciel ainsi que UML.

* 1. **Généralité sur le génie logiciel**

C’est « l’ensemble des méthodes, des techniques et outils concourant à la production d’un logiciel, au-delà de la seule activité de programmation » [Université Nongo Conakry, 2016]. [b]. Le génie logiciel est un domaine de recherche qui a été défini du 07 au 11 octobre 1968, à Garmisch-Partenkirchen, sous le parrainage de l’OTAN. Il a pour objectif de répondre à un problème qui s’énonçait en deux constatations : d’une part, le logiciel n’était pas fiable, d’autre part, la réalisation de logiciels satisfaisant leur cahier des charges dans les délais prévus était incroyablement difficile. Le génie logiciel est « l’application pratique de la connaissance scientifique dans la conception et l’élaboration de programmes informatiques et de la documentation associée nécessaire pour les développer, les mettre en œuvre et les maintenir » [Barry W., 1988] [1]. Le génie logiciel consiste à appliquer des méthodologies, c’est-à-dire à.… développer et à utiliser des méthodes et des outils dans les bus de produire des logiciels de quantité en respectant des contraintes de temps et de coût.

* 1. **Définition d’un modèle**

Un modèle est une abstraction de la réalité, Il s’agit d’un processus qui consiste à identifier les caractéristiques intéressantes d’une entité, en vue d’une utilisation précise. L’abstraction désigne aussi le résultat de ce processus, c'est-à-dire l’ensemble de la caractéristique essentielle d’une entité, retenue par un observateur. Un modèle est une vue subjective mais pertinente de la réalité. Il définit une frontière entre la réalité et la perspective de l’observateur. Ce n’est pas « la réalité », mais une vue très subjective de la réalité. Bien qu’un modèle ne représente pas une réalité absolue, il reflète des aspects importants de la réalité, il en donne donc une vue juste et pertinente. Le caractère abstrait d’un modèle doit notamment permettre :

* De soutenir la compréhension du système étudié : il réduit la complexité du système étudié ;
* De simuler le système étudié : il représente le système étudie et reproduit ses comportements.
  + 1. **Modèle de cycle de vie d’un projet web**

Comme toute application informatique, une application web nécessite d’être défini, puis conçu, avant d’être réalisé, sans oublier qu’il existe aussi un « après projet », qui devra être préparé : exploitation, promotion et animation.

Début du projet

Etude

Préalable

Rédaction du plan

projet

Constitution de

L’équipe

Analyse de besoins

Etude de faisabilité

Etude de marché

Type des objectifs

généraux

Plan du projet et

estimation

Définition des

contrats de

réparation

Conception de

l’application

Aspect de

l’application

(

Template)

Arborescence

Structure des âges

Présentation

UML

Réalisation

de l’ossature

Intégration

statique

Intégration

dynamique

Test et recette

Model page

Rapport de test

Correction

Fin du projet

Fin du projet

PHASE 1 :

Définition

du projet

PHASE2:

Concep-

tion de

l’applica

tion

PHASE 3:

Réalisa-

tion de

l’applica

-

tion

PHASE4 :

Mise en

ligne de

l’

applica-

tion

Figure 4.1 : Modèle de cycle de vie d’une application web

* + 1. **Modèle de cycle de vie d’un logiciel**

Un modèle est une représentation abstraite et simplifiée de la réalité en vue de la décrire. Le modèle exclut certains détails de la réalité qui n'influencent pas de manière significative sur le phénomène étudié. Modéliser un système revient donc à représenter de façon abstraite ce que le concepteur du système croit important pour la compréhension du phénomène réel étudié [Djallel B., 2016] [2]. Il existe deux approches de modélisation :

* L'approche fonctionnelle (dérivée des langages de procéduraux) ;
* L'approche objet (le logiciel est considéré comme un ensemble d'objets possédant des caractéristiques spécifiques et qui interagissent entre eux).

Les modèles de cycle de vie ne sont pas identiques. Mais éventuellement sont formés par les phases suivantes d'analyse, de conception, d'implémentation, de test et d'installation. Le but de ce découpage est de maîtriser les risques, maîtriser au mieux les délais et les coûts, et d'obtenir une qualité conforme aux exigences. L'origine de ce découpage provient du constat que les erreurs ne sont d'autant plus élevées si elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs le plus tôt et ainsi de maîtriser la qualité du logiciel, les délais de sa réalisation et les coûts associés.

On distingue deux types de cycle de vie :

* Le cycle de vie des produits, s'appliquant à tous types de produits, et pouvant être considéré comme un outil de gestion ;
* Le cycle de vie de développement des logiciels, on l'appelle souvent abusivement cycle de vie des logiciels [Anne-Marie H., 2002] [3]. La plupart du temps, les logiciels sont réalisés en équipe. La modélisation permet de répartir efficacement les tâches entre les membres de l'équipe et même d'automatiser certaines de ces tâches. Un modèle de développement de logiciel est donc une représentation abstraite d'un processus.

Le modèle de cycle de développement est éventuellement formé par les phases suivantes : analyse, conception, implémentation, validation, test et installation.

Un modèle permet d'assurer la bonne compréhension du fonctionnement du système et de maîtriser la complexité de celui-ci. Par ailleurs, les coûts, les délais et la qualité du logiciel sont maîtrisés.

Le cycle de vie du logiciel comprend généralement plusieurs étapes :

* La définition des objectifs qui consiste à définir la finalité du projet et son inscription dans une stratégie globale ;
* L’analyse des besoins et faisabilité qui est l’expression, le recueil et la formalisation des besoins du demandeur et de l’ensemble des contraintes ;
* L’estimation de la faisabilité de ces besoins ;
* La conception générale qui s’agit de l’élaboration des spécifications de l’architecture générale du logiciel ;
* La conception détaillée qui consiste à définir précisément chaque sous-ensemble du logiciel ;
* Le codage, l’implémentation ou la programmation d’où la traduction dans un langage de programmation des fonctionnalités définies lors de la phase de conception ;
* Les tests unitaires permettant de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux spécifications ;
* L’intégration, qui a pour objectif de s’assurer de l’interfaçage des différents éléments du logiciel ;
* La qualification ou recette : c’est-à-dire la vérification de la conformité du logiciel aux spécifications initiales ;
* La documentation qui vise à produire les informations nécessaires pour l’utilisation du logiciel et pour des développements ultérieurs ;
* La mise en production qui est le déploiement sur site du logiciel ;
* La maintenance qui comprend toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.
  + 1. **Les différents modèles de cycle de vie**

Il existe une multitude de MCV. Face à cette multitude de modèles, nous avons fait le choix de faire la comparaison sur cinq (05) cycles de vie à savoir :

* Le modèle en Cascade ;
* Le modèle en V ;
* Le modèle en Spirale ;
* Le modèle par Incrément ou itératif ;
* Le modèle incrémental ;
  + - 1. **Cycle de vie en Cascade**

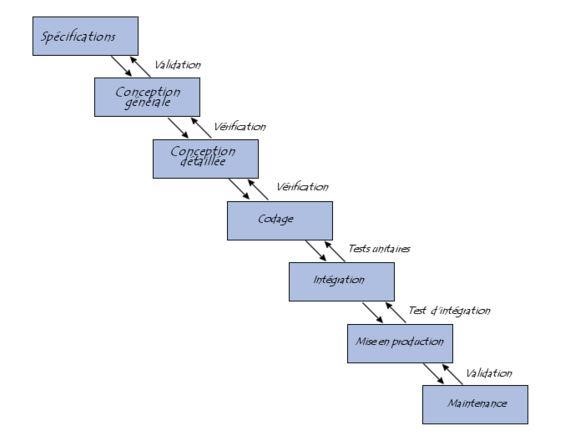
Le [modèle en Cascade](https://fr.wikipedia.org/w/index.php?title=Mod%C3%A8le_en_cascade&action=edit&redlink=1) est hérité de l'industrie du [BTP (](https://fr.wikipedia.org/wiki/BTP)c'est-à-dire de l’industrie lourde). Ce modèle repose sur les hypothèses suivantes :

* On ne peut pas construire la toiture avant les fondations ;
* Les conséquences d'une modification en amont du cycle ont un impact majeur sur les coûts en aval.

Les phases traditionnelles de développement sont effectuées simplement les unes après les autres, avec un retour sur les précédentes, voire au tout début du cycle. Le processus de développement utilisant un cycle en cascade exécute des phases qui ont pour caractéristiques :

* De produire des livrables définis au préalable ;
* De se terminer à une date précise ;
* De ne se terminer que lorsque les livrables sont jugés satisfaisants lors d'une étape de validation-vérification.

Pour mieux comprendre ce modèle, on peut dire qu’il a une facilitée de comprendre, d’utiliser et adapter à une équipe inexpérimentée. Les limites de chaque étape sont visibles. En plus, ce modèle facilite un management du projet et la définition des besoins est non-évolutive. Mais, tous les besoins doivent être bien spécifiés au départ. Il donne une fausse impression de l’avancée des travaux, pas d’interaction entre les phases de développement, pas de retour en arrière d’une phase à l’autre et l’intégration n’a lieu qu’à la fin du cycle, donc le client peut se retrouver non satisfait.

Figure 4.2 : Modèle en cascade [Wikipédia/cycle en cascade]

* + - 1. **Cycle de vie en V**

Le modèle du [Cycle en V](https://fr.wikipedia.org/wiki/Cycle_en_V) a été imaginé pour pallier le problème de réactivité du modèle en cascade. Ce modèle est une amélioration du modèle en cascade qui permet en cas d'anomalie, de limiter un retour aux étapes précédentes. Ses phases de la partie montante doivent renvoyer de l'information sur les phases en vis-à-vis lorsque des défauts sont détectés afin d'améliorer le logiciel. De plus, le cycle en V met en évidence la nécessité d'avancer et de préparer dans les étapes descendantes les « attendus » des futures étapes montantes : ainsi les attendus des tests de validation sont définis lors des spécifications, les attendus des tests unitaires sont définis lors de la conception. Mais tous les tests sont effectués et celui du contrôle se fait progressivement à chaque étape. Et sa phase de validation sont prises en main très tôt dans le processus de développement. Or, le processus n’est pas itératif et une mauvaise prise en compte des évènements concurrents et des changements de la spécification des besoins. En plus, il ne contient pas les activités d’analyses de risques. Malgré cela, il est excellent pour les systèmes requérant une grande sûreté.

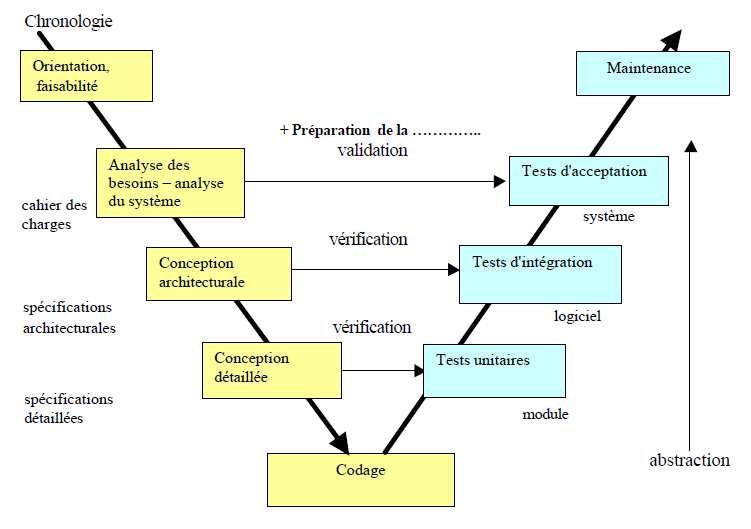


Figure 4.3 : Schéma de modèle en V

* + - 1. **Modèle de cycle de vie en spirale**

Le développement reprend les différentes étapes du cycle en V. Par l'implémentation de versions successives, le cycle recommence en proposant un produit de plus en plus complet et robuste. Le cycle en spirale met cependant plus l'accent sur la gestion des risques que le cycle en V. En effet, le début de chaque itération comprend une phase d'analyse des risques. Celle-ci est rendue nécessaire par le fait que, lors d'un développement cyclique, il y a plus de risques de défaire, au cours de l'itération, ce qui a été fait au cours de l'itération précédente. Alors, les fonctions critiques à haut risque sont développées en premier lieu, la conception ne doit pas forcement être terminée. Son développement se fait en interaction avec les clients et l’évolution du coût est sous contrôle. Mais le temps consacré à l’évaluation des risques est trop élevé pour des petits projets. De plus, le temps mis à planifier, fixer les objectifs, les prototypes peut être excessif et la spirale peut être infinie. Donc, il est difficile de définir les objectifs et les points de validation intermédiaires entre les différentes étapes.

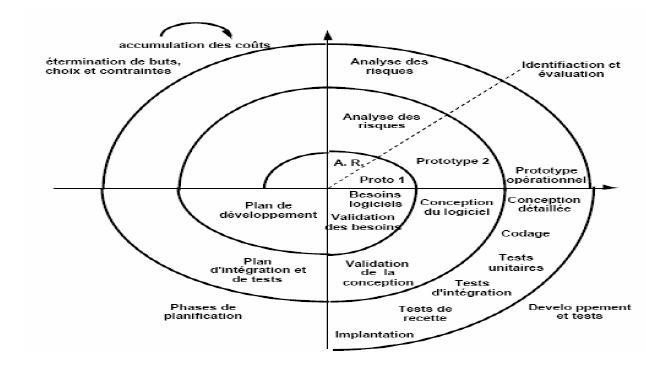


Figure 4.4 : Modèle en spirale [[Wikipedia.org/](http://www.en.wikipedia.org/)wiki/spiral model]

* + - 1. **Modèle de cycle de vie par incrément ou itératif**

Le cycle de vie par incrément, on sépare les activités des artéfacts. Un artéfact étant le produit issu d'une activité. Ainsi, on applique un cycle de type roue de Deming sur la production d'une documentation, d'un composant, d'un test, etc.

Rapportée à une activité de type gestion de projet, la première phase sera celle de

* La faisabilité : l'acceptation d'un nouveau besoin ;
* L’élaboration : on imagine comment on va le réaliser ;
* La fabrication : construction ;
* La transition : tout est mis en œuvre pour livrer au client.

L’avantage d’utiliser ce modèle est que le client peut valider chaque étape de processus. La délivrance du produit est rapide c’est-à-dire qu’un produit exploitable peut être délivré à tout moment et le risque du changement des besoins est minimal. Mais l’inconvénient, ce modèle requière une bonne planification, une bonne conception et la définition complète des fonctionnalités du système pour une définition des différents incréments. Et les différentes interfaces doivent être bien définies.

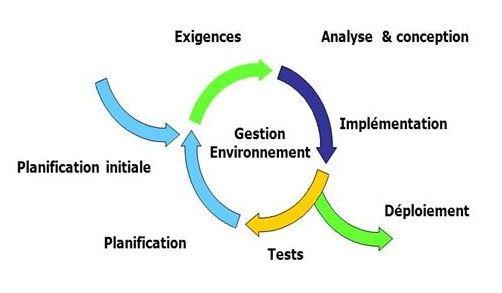


Figure 4.5 : Modèle par incrément ou itératif

* + - 1. **Cycle de vie incrémentale**

Dans les modèles précédents, un logiciel est décomposé en composants développés séparément et intégrés à la fin du processus. Dans les modèles par incrément un seul ensemble de composants est développé à la fois : des incréments viennent s’intégrer à un noyau de logiciel développé au préalable. Chaque incrément est développé selon l’un des modèles précédents. En bref, le modèle par incrément découpe le système en domaines qui sont traités individuellement sur le modèle en cascade.

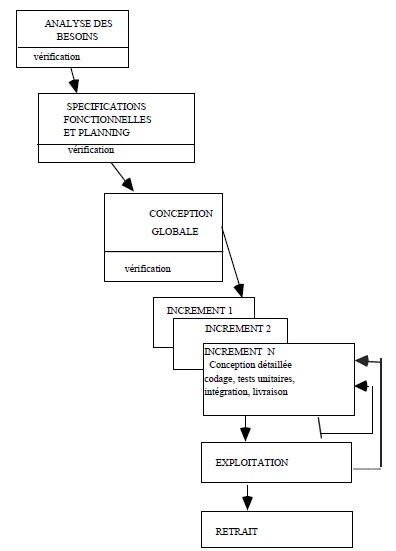


Figure 4.6: Modèle incrémentale

* + 1. **Comparaison de modèle de cycle de vie**

Nous venons de faire la description de ces cinq (05) MCV. Nous allons maintenant fait la comparaison entre le quatre (04) premiers selon le critère du moment d’utilisation.

En parlant du modèle en cascade, on l’utilise quand la phase de spécification a été très bien faite, quand la définition du produit est stable. Il ne s’agit d’une nouvelle version d’un produit existant. En plus, quand l’implantation d’un produit existant sur une nouvelle plateforme et une bonne maitrise de la technologie. Mais pour le modèle en V, les spécifications de besoins doivent être bien faites, la solution à développer et la technologie à utiliser doivent être connues. Et les changements doivent être faits avant l’analyse.

Si cela qu’on a besoin pour les deux modèles, quels sont donc les deux suivants ? On utilise le modèle en spirale quand les coûts et l’évaluation des risques sont importants. Et aussi pour des projets à risque au moins moyennement élevé et à long terme dont les financements peuvent varier. En outres, quand les utilisateurs ne définissent pas clairement leurs besoins et que la spécification des besoins soit complexe. Et en fin, pour le modèle par incrément, on l’utilise sur un projet utilisant de nouvelles technologies, sur des projets ayant une durée de développement assez longue. Quand la plupart des fonctionnalités doivent être connues au départ mais certaines ne seront utilisées que plus tard et que les risques, le financement, la complexité du logiciel, l’élaboration ou les besoins pour les bénéfices sont maximum dès le départ.

* + 1. **Choix et justification**

En disant comme résultat après avoir les comparer, il n’est pas facile de prendre une décision concernant ces différents cycles de vie. Chacun a ses forces, ses faiblesses et un cadre d’utilisation bien déterminé. Malgré tout, nous constatons une plus grande utilisation du Cycle en V par la plupart des équipes de développement.

Les principaux avantages de l’utilisation du modèle en V sont les suivants :

* Il permet l’organisation du travail et des équipes ;
* Il favorise la décomposition hiérarchique fonctionnelle ;
* Il propose des étapes clés (documentation, revues) qui entraine un bon suivi du projet ;
* Permet de garantir une certaine qualité (plan assurance qualité) ;
* Adapté à de grands projets ;
* Il est un modèle éprouvé car reproduit sur la production industrielle classique. Nous avons présenté les modèles de cycle de vie d’un logiciel dans la section ci-dessus par le biais de les exposer, de les comparer et avec une justification de choix du modèle.

Mais dans la section suivante, nous présentons le choix des technologies et méthodes de conception utilisée dans la réalisation du projet.

* 1. **Méthodologie de conception et technologies utilisées**

Pour faire face à la complexité croissante des systèmes d’information, de nouvelles méthodes et outils ont été créées. La principale avancée des quinze dernières années jusqu’aux années 2005 réside dans la programmation orientée objet.

Face à ce nouveau mode de programmation, les méthodes de modélisation classique (telle MERISE) ont rapidement montré certaines limites et ont dû s’adapter. De très nombreuses méthodes ont également vu le jour comme Booch, OMT (Object Modelling Technique) Dans ce contexte et devant l’affluence de nouvelles méthodes de conception « orientée objet », l’Object Management Group (OMG) a eu comme objectif de définir une notation standard utilisable dans les développements informatiques basés sur l’objet. C’est ainsi qu’est apparu UML, qui est issu de la fusion des méthodes Booch, OMT et OOSE (Object Oriented Software Engineering).

Issu du terrain et fruit d'un travail d'experts reconnus, UML est le résultat d'un large consensus. De très nombreux acteurs industriels de renom ont adopté UML et participent à son développement. En l'espace d'une poignée d'années seulement, UML est devenu un standard incontournable.

Alors, cette section présente le choix des technologies et méthodes de conception utilisée dans la réalisation du projet de notre futur système.

* 1. **Présentation de UML**
     1. **L’aspect choisi**

De la même façon qu’il vaut mieux dessiner une maison avant de la construire, il vaut mieux modéliser un système avant de la réaliser.

Il est nécessaire que nous choisissions UML comme langage de modélisation puisque nous allons utiliser le concept de l’orienter objet pour développer cette application.

UML est plus expressif, plus propre et plus uniforme que Booch, OMT, OOSE, et les autres méthodes. Cela signifie qu'il y a un bénéfice à passer à UML, parce qu'elle permettra aux projets de modéliser des choses qui n'auraient pas pu l'être avant.

Les utilisateurs de la plupart des autres méthodes et langages de modélisation auront avantage à utiliser UML, puisqu'elle supprime toutes les différences non nécessaires de notation et de terminologie qui obscurcissent les similarités de bases de ces différentes approches. Ainsi, la méthodologie de conception adoptée se base sur le choix de diagrammes UML adéquats. Nous avons utilisé cinq diagrammes : diagramme de cas d’utilisation, diagramme d’activités, diagramme de séquence, diagramme de classes, diagramme de déploiement.

* + 1. **L’UML** 
       1. **Historique**

Né de la fusion des 3 méthodes objet dominantes (OMT, Booch et OOSE), puis normalisé par l'OMG en 1997, UML est rapidement devenu un standard incontournable [Rumbaugh J., Jacobson I. et Booch G., 1999]. UML n'est pas à l'origine des concepts objet, mais il en donne une définition plu formelle et apporte la dimension méthodologique qui faisait défaut à l'approche objet. Le but de cette présentation n'est pas de faire l'apologie d'UML, ni de restreindre UML à sa notation graphique, car le véritable intérêt d'UML est ailleurs ! En effet, maîtriser la notation graphique d'UML n'est pas une fin en soi. Ce qui est primordial, c'est d'utiliser les concepts objet à bon escient et d'appliquer la démarche d'analyse correspondante.

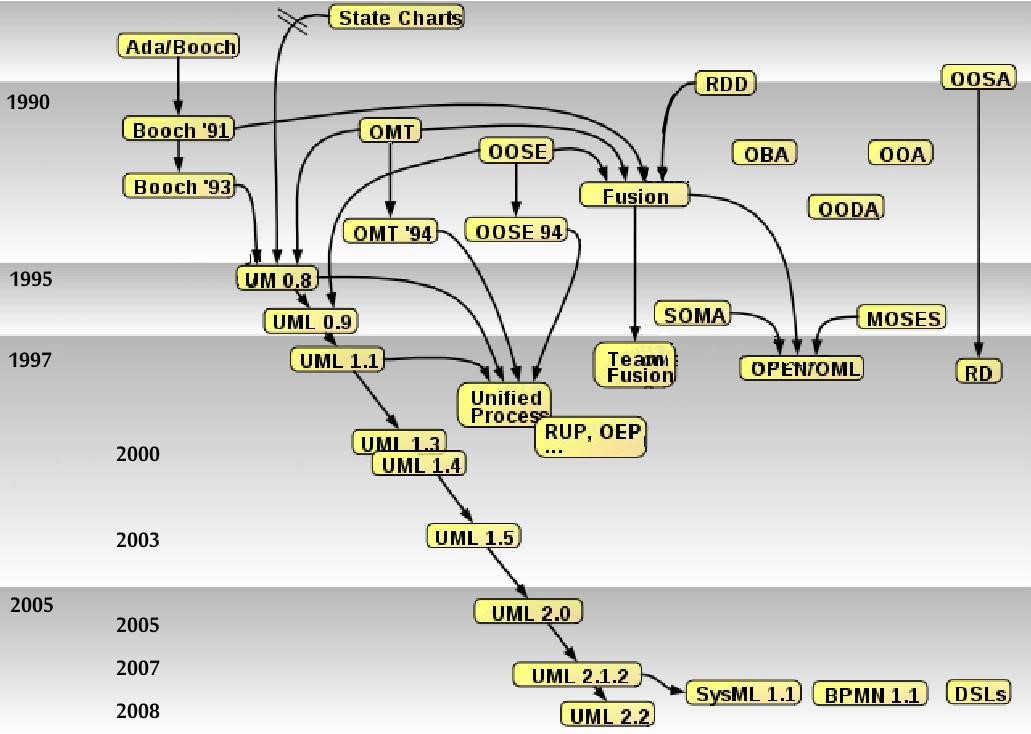


Figure 4.7: Evolution de l’UML

* + - 1. **Définitions**

L’UML traduit « langage de modélisation objet unifié » est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 80 [Rumbaugh J., Jacobson I. et Booch G., 1999].

* **Object Modeling Technique(OMT)** : a été développée par **James RUMBAUGH** dans le centre de Recherche et Développement de la société Générale Electric à la fin des années 80. Cette méthode connaît un certain succès en France et en Europe. OMT propose une triple perception du système d’information : une dimension statique qui décrit le schéma objet du système, une dimension dynamique qui décrit le changement d’états des objets en fonction d’événements, une dimension fonctionnelle qui décrit les processus de transformation des informations.
* **Object Oriented Development (OOD)** : Pionner de l’orienté objet, Grady BOOCH publie en 1981 un article intitulé « Objet Oriented Development » qui présent une méthode de modélisation objet. Cette méthode applique le modèle objet pour synthétiser la structure logique, physique, statique et dynamique d’une application. Pour cela, elle utilise une notion expressive et clairement définie permettant de capturer des points de vue différents. La méthode Booch2 propose des diagrammes de classes, diagrammes d’instancié et diagrammes d’état transitions. Le niveau physique contient les diagrammes de modules et diagramme de processus.
* **Object Oriented Software Engineering(OOSE)** : la méthode OOSE de Ivar JACOBSON couvre tout le cycle de développement du logiciel, créée dans un centre de recherche d’Ericson (Suède), Elle repose essentiellement sur l’analyse des besoins des utilisateurs à l’aide des diagrammes de cas d’utilisation formalisés par Ivar JACOBSON dans sa méthode puis introduits dans UML qui décrit, sous la forme d’actions et de réactions, le comportement d’un système du point de vue de l’utilisateur, définissent les limites du système et les relations entre le système et l’environnement. Ils permettent d’introduire une manière spécifique d’utiliser un système. C’est l’image d’une fonctionnalité du système, déclenchée en réponse à la stimulation d’un acteur externe.
  + - 1. **Les caractéristiques de ses diagrammes**

Pour un système donné, les diagrammes de conception de l’UML doit satisfaire les caractéristiques suivants [Laurent A., 2010]:

* Combinés, les différents types de diagramme UML offrent une vue complète des aspects statiques et dynamiques d’un système ;
* Par extension et abus de langage, un diagramme UML est aussi un modèle ;
* UML permet de définir et de visualiser un modèle, à l’aide de diagrammes ;
* Un diagramme UML est une représentation graphique, qui s’intéresse à un aspect précis du modèle ; c’est une perspective du modèle, pas « le modèle » ;
* Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis) ;
* Un type de diagramme UML véhicule une sémantique précise (un type de diagramme offre toujours la même vue d’un système).
  + - 1. **Ses Objectifs**

UML est avant tout un support de communication performant, qui facilite la représentation et la compréhension de solutions objet. Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.

L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions. Il a été conçu comme un ensemble de notation qui vise à être utilisées dans différentes étape de la modélisation objet, de l’expression des besoins jusqu’à l’implantation. Il s’agit de la notation synthétique et intuitive et accessible par différentes intervenantes de la modélisation.

UML est donc un langage de modélisation avec plusieurs objectifs qui font un véritable outil de communication [Laurent A., 2010]:

* Comprendre et décrire les besoins ;
* Spécifier un système ;
* Établir l’architecture logicielle.
  + - 1. **UML face à la programmation**

Pour programmer une application, il ne convient pas de se lancer tête baissée dans l’écriture du code, mais il faut organiser ses idées, les documenter puis organiser la réalisation en définissant les modules et les étapes

* + - 1. **Les points forts et points faibles d’UML**

Dans le tableau suivant en va voire le point faible et forts d’UML

Tableau 4.1 : Les avantages et les inconvénients d’UML

|  |  |
| --- | --- |
| **Points forts** | **Faiblesses** |
| UML est un langage formel et normalisé :   * Permet une grande précision, * Assure la pérennité, * Facilite l’utilisation d’outils UML est un outil qui facilite la communication ; * Propose un cadre d’analyse, * Permet la représentation d’éléments abstraits et complexes, * Offre une grande expressivité grâce à sa polyvalence et à sa souplesse. | UML nécessite un apprentissage et de l’expérience :   * UML n’est pas à l’origine de l’approche objet mais en précise les concepts sous-jacents,   UML n’est pas une méthode (et n’en propose pas) :   * Cela va probablement changer ultérieurement, |
| UML fournit un moyen astucieux permettant de représenter diverses projections d’une même représentation grâce à ces différents diagrammes ; | UML n’est pas à l’origine de concept objets, mais en constitue une étape majeure, car il unifie les différentes approches et donne une définition plus formelle. |
| Les diagrammes d’UML permettant de spécifier, de construire, de documenter, de visualiser et de manipuler les systèmes informatiques et ils permettent d’isoler certaines parties des modèles pour les rendre plus aisément compréhensibles. |  |

* + - 1. **Les grandes lignes essentielles de l’UML**

L’UML, « Langage de modélisation d’objet unifié », a pour but de clarifier et de simplifier la conception et la représentation d'un système donné et de son fonctionnement. Particulièrement, UML est un langage de choix, la schématisation simple compréhensible par l’homme et la machine de ses étapes essentielles.

Comme étape de modélisation d’UML, les différents diagrammes persistent. Les étapes que la modélisation contient distinguent en deux types des vues:

* Les vues statiques ;
* Les vues dynamiques.

Les vues statiques représentent le système physiquement qui comprend les diagrammes suivants :

* **Diagrammes d’objets,** il sert à représenter les instances de classes (objets) utilisées dans le système ;
* **Diagrammes de classes**, il représente les classes intervenant dans le système ;
* **Diagrammes de cas d'utilisation**, il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système ;
* **Diagrammes de composants**, il permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données…) ;
* **Diagrammes de déploiement**, utilisés pour représenter l'architecture physique d'un système en utilisant des nœuds et des connexions entre ces nœuds ;

Les vues dynamiques montrent le fonctionnement du système comprends les diagrammes suivants :

* **Diagrammes de séquence**, représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs ;
* **Diagrammes de collaboration**, Scénario d’un cas d’utilisation : activités des objets et des messages échangés ;
* **Diagrammes d'état** **transitions,** permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants ;
* **Diagrammes d'activités,** permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

Tableau 4.2 : Classement des diagrammes ;

|  |  |
| --- | --- |
| **CLASSIFICATION** | **DIAGRAMME** |
| Besoins des utilisateurs | * Diagramme des cas d’utilisation |
| Structure statique | * Diagramme de Classe |
|  | * Diagramme d’objet |
| Dynamique des objets | * Diagramme états-transition |
|  | * Diagramme d’activités |
| Interactions entre objets | * Diagramme de séquence |
|  | * Diagramme de collaboration |
| Réalisation et déploiement | * Diagramme de composants |
|  | * Diagramme de déploiement |

Les différents diagrammes UML qui sont les diagrammes structurels et comportementaux sont représentés par la figure 4.8 suivante :

Diagramme UML

Diagramme

Structurel

Diagramme

Comportemental

Diagramme

d’Activité

Diagramme de cas

d’Utilisation

Diagramme états-

transitions

Diagramme de

déploiement

Diagramme de

paquetage

Diagramme de

Composant

Diagramme d’Objet

Diagramme de

Classe

Diagramme

d’interaction

Diagramme de

structure composite

Diagramme de

Communication

Diagramme global

d’interaction

Diagramme de

temps

Diagramme de

Séquence

Figure 4.8 : Les différents diagrammes UML

***a. Diagramme de cas d’utilisation UML***

Les diagrammes de cas d’utilisation permettent de recueillir, d’analyser et d’organiser les besoins, et de recenser les grandes fonctionnalités d’un système. Dans notre cas nous allons utiliser deux types de diagrammes : le diagramme préliminaire et les diagrammes détaillés. Le diagramme de cas d’utilisation préliminaire permet de représenter les principaux cas d’utilisation du système, ces cas d’utilisation seront ensuite détaillés dans les Cas d’utilisation détaillés. Le diagramme de cas d’utilisation détaillés permet donc de détailler les cas d’utilisation du cas d’utilisation préliminaire, ces principaux cas d’utilisation seront traités sous forme de processus [Laurent A., 2010].

* **Acteur**

Un acteur représente l’abstraction d’un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

* **Cas d’utilisation**

C’est un graphe dont les nœuds sont des acteurs ou des cas d’utilisation et les arcs sont des relations de communication entre acteurs et cas d’utilisation ou des relations standardisées entre cas d’utilisation.

* **Les relations entre cas d’utilisation**

UML définit trois types de relations standardisées entre cas d’utilisation détaillées ci-après :

* Une relation de généralisation
* Une relation d’inclusion, formalisée par le mot-clé « include » ;
* Une relation d’extension, formalisée par le mot-clé « extend ».
* **Représentation graphique**

La figure 4.9 représente le formalisme du diagramme de cas d’utilisation.

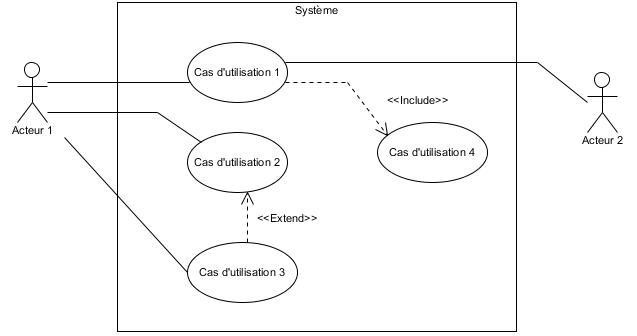


Figure 4.9 : Formalisme du diagramme de cas d’utilisation.

***b. Diagramme de séquence***

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur ; il indique les objets que l'acteur va manipuler, et les opérations qui font passer d'un objet à l'autre [Laurent Audibert, 2010].

Il permet de cacher les interactions d'objets dans le cadre d'un scénario d'un Diagramme de cas d’utilisation Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent par le biais de messages. La représentation se concentre sur l'expression des interactions et met l'accent sur la chronologie des envois de messages.

* **La ligne de vie**

La ligne de vie représente un participant à une interaction objet ou acteur.

* **Le message**

Le message définit la communication particulière entre des lignes de vie objets ou acteurs. Différents types de message (ou actions) peuvent transiter entre les acteurs et objets :

* **Message simple** : message qui ne spécifie aucune caractéristique d'envoi ou de réception particulière ;
* **Message minuté** (timeout) ou **message avec durée de vie** : Il bloque l'expéditeur pendant un temps donné en attendant la prise en compte du message par le récepteur. L'expéditeur est libéré si la prise en compte n'a pas eu lieu pendant le délai spécifié ;
* **Message synchrone** : Il bloque l'expéditeur jusqu'à la prise en compte du message par le destinataire. Le flot de contrôle passe de l'émetteur au récepteur à la prise en compte du message c’est-à-dire l'émetteur devient passif et le récepteur actif ;
* **Message asynchrone** : Il n'interrompt pas l'exécution de l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré ;
* **Message dérobant** : Il n'interrompt pas l'exécution de l'expéditeur et le message est mis en attente dans une liste d'attente de traitement chez le récepteur.

* **Représentation graphique**

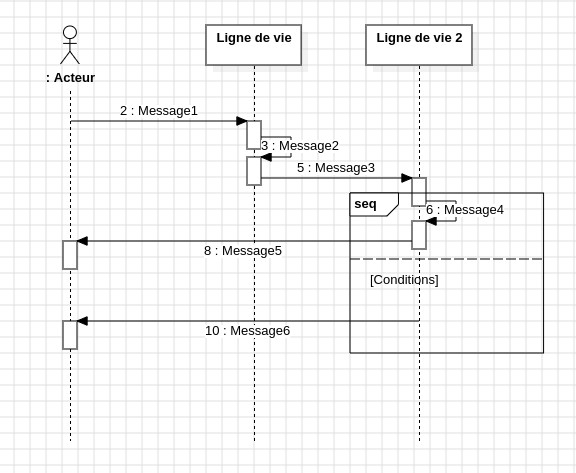
La figure 4.10 représente le formalisme de diagramme de séquences type.

Figure 4.10 : Formalisme du diagramme de séquence.

***c. Diagramme de classes***

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d’une telle modélisation. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d’utilisation. Il s’agit d’une vue statique car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d’application ainsi que les concepts internes créés de toutes pièces dans le cadre de l’implémentation d’une application. Les diagrammes de classe appropriés représentent les classes participant à la réalisation d’un cas d’utilisation et la relation entre elle. Les outils de cette notion peuvent gérer des interfaces graphiques pour le schéma de la conception, ils peuvent aussi établir les classes essentielles de l’application et enfin, ils peuvent générer automatiquement les codes écrits en langage spécifié par l’utilisateur à partir des classes déjà construites [Laurent Audibert, 2010].

* **Définition des éléments**

Les éléments d'un diagramme des Classes sont les classes et les relations qui les lient. Nous allons expliquer ci-dessous chacun de ces éléments :

**Les classes** sont les modules de base de la programmation orientée objet. Une classe est représentée en utilisant un rectangle divisé en trois sections. La section supérieure est le nom de la classe. La section centrale définit les propriétés de la classe et la section du bas énumère les méthodes de la classe.

**Une association** est une relation générique entre deux classes. Elle est modélisée par une ligne reliant les deux classes. Cette ligne peut être qualifiée avec le type de relation, et peut également comporter des règles de multiplicité (par exemple un à un, un à plusieurs et plusieurs à plusieurs) pour la relation.

**Une relation de composition** est indiquée par une ligne avec un "diamant" rempli. Si une classe ne peut pas exister par elle-même, mais doit être un membre d'une autre classe, alors elle possède une relation de composition avec la classe contenante.

**Une relation de dépendance** est représentée par une flèche pointillée. Quand une classe utilise une autre classe, par exemple comme membre ou comme paramètre d'une de ces fonctions, elle "dépend" ainsi de cette classe.

**Les agrégations** indiquent une relation de contenant-contenu. Elle décrite par une relation "possède". Une relation d'agrégation est représentée par une ligne avec un "diamant" creux.

**Une relation de généralisation** est l'équivalent d'une relation d'héritage en terme orientés objet (relation "est-un "). Une relation de généralisation est indiquée par une flèche creuse se dirigeant vers la classe "parent".

* **Représentation graphique**

C’est le formalisme du diagramme de classe qui est représenté par la figure 4.11:

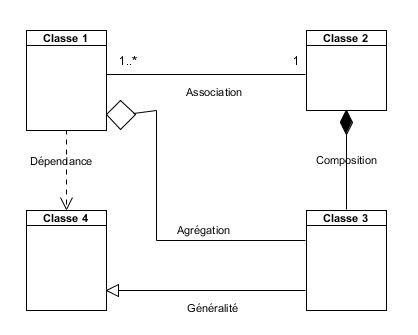


Figure 4.11: Formalisme de diagramme de classe

* **Cardinalités**

La cardinalité précise le nombre d'instances qui participent à une relation. L’expression des cardinalités d'une relation en UML :

* n : exactement "n" (n, entier naturel > 0) ;
* n..m : de "n" à "m" (entiers naturels ou variables, m > n) ;
* \* : plusieurs (équivalent à "0..n" et "0..\*") ;
* n..\* : "n" ou plus (n, entier naturel ou variable).

***d. Diagramme d’activités***

Les diagrammes d’activités permettent de mettre l’accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d’une méthode ou le déroulement d’un cas d’utilisation. Nous pouvons rattacher un diagramme d’activités à n’importe quel élément de modélisation afin de visualiser, spécifier, construire ou documenter le comportement de cet élément. Dans ce projet les diagrammes d’activités sont attachés aux diagrammes de cas d’utilisation détaillés [Laurent Audibert, 2010].

* **Représentation graphique**

C’est le formalisme du diagramme d’activité qui est représenté par la figure 4.12:

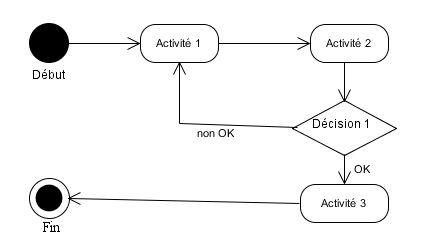


Figure 1: Formalisme du diagramme d’activité

***d. Diagramme de déploiement***

Un diagramme de déploiement est un diagramme de classes ou un diagramme d’objets représentant les nœuds ou les instances de nœuds sur lesquels le système s’exécute. Il propose une vision statique de la topologie du matériel sur lequel s’exécute le système et il montre les associations (connexions) existant entre les nœuds du système.

Un nœud est un élément physique matériel sur lequel le système s’exécute. Il peut être un processeur, un périphérique ou un réseau [Laurent Audibert, 2010].

* **Représentation graphique**

C’est le formalisme du diagramme de déploiement qui est représenté par la figure 4.13 :

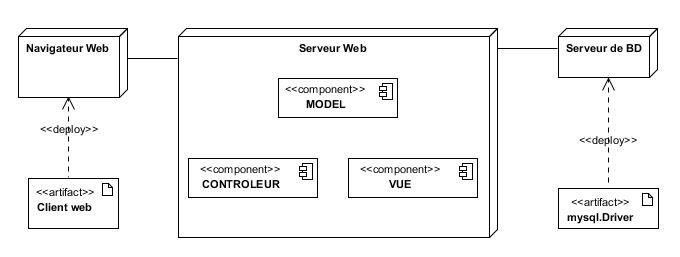


Figure 4.13 : Formalisme du diagramme de déploiement

* + - 1. **Choix des outils pour la modélisation en UML**
* **Les outils**

Il y a plusieurs outils avec lesquels on peut faire les diagrammes d’UML, ce sont :

* Win’ Design ;
* Visual Paradigm ;
* UML Designer ;
* BO UML ;
* Magic Draw.
* **Comparaison des outils**

Chaque logiciel sera évalué selon sa qualité à traduire différents éléments d’un diagramme de classes.

Le tableau 4.3 nous montre la comparaison des quelques outils de conception les plus utilisées

Tableau 4 : Comparaison des outils de conception d’UML

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Outils** | **Fiabilité** | **Efficacité** | **Open Source** | **Coût** | **Plateforme/ SE** |
| **Win’ Design** | Oui | Oui | Non | Payante | Windows |
| **BO UML** | Oui | Oui | <5.0 Oui  > 5.0 Non | <5.0 Gratuit  >5.0 Payante | Linux,  Windows et  Mac OS X |
| **Visual**  **Paradigm for**  **UML** | Oui | Oui | Non | Payante | Multiplateforme |
| **Magic Draw** | Oui | Oui | Non | Payante | Multiplateforme |
| **UML Designer** | Oui | Oui | Oui | Payante | Multiplateforme |

Dans la conception de notre projet, nous choisissons Win’Design pour l’outil de modélisation

# **Chapitre 5 : Analyse du projet et étude de l’existant**

L’analyse du projet est la première étape dans le processus de développement d’un logiciel afin de concevoir une application qui répond aux besoins et aux exigences de la société en respectant les diverses contraintes. Ainsi elle permet d’adapter la problématique à la modélisation conceptuelle pour apporter la solution appropriée. Cette partie consiste essentiellement à collecter et à analyser les rapports de la production électrique au lancement du projet, on peut y trouver toutes sortes de rapports concernant au service d’exploitation. Dans un premier temps, il est nécessaire de constater l’état des choses sur le terrain, ainsi à travers des interviews et des réunions, nous pourrons déterminer : « Que voulons-nous accomplir ? Qu’est-ce qui ne va pas ? Qu’est-ce que nous pouvons faire pour y remédier ? ».

* 1. **Analyse des besoins**

Jusqu’à nos jours, l’Entreprise JIRAMA Centrale a mis en place un projet de la production électrique qui sert de référentiel à ce qui devrait être, c’est-à-dire les objectifs à atteindre par la JIRAMA dans les années à venir et chercher une nouvelle méthode pour l’améliorer.

Le principal besoin est l’extension des activités payantes. Lorsqu’il y a l’existence d’amélioration des méthodes, des outils, des ressources financières, capacité des personnels responsables. Le projet se base sur l’idée venant des membres de tous les services via une analyse qui contient entre autres les points forts et les points faibles que chacun trouve au sein de la JIRAMA Centrale, ainsi que des suggestions pour éviter les problèmes ou améliorer l’existant.

La JIRAMA de Fianarantsoa, de par son projet de l’exploitation électrique, à gestion de la production électrique utile d’accélérer le travail de près les activités électriques de ce responsable de la production électrique au sein de la JIRAMA Centrale. Dans ce cas, il est important de disposer un outil qui permet la gestion de la synthèse de la production électrique au sein de l’Entreprise.

De ce fait, il est important de :

* Bien saisir les gens sur les procédures de chef d’exploitation.
* Réduire la perte de temps et aussi la réduction de calcul de la production à chaque jour.

Afin de prendre conscience de l’état actuel de la JIRAMA Centrale de Fianarantsoa, il est impératif de commencer par des entretiens ou des interviews auprès des différents services présents, déterminer ce qui fonctionne et ce qui ne fonctionne pas.

* + 1. **Besoin fonctionnel et non fonctionnel**

Le système dont l’entreprise veut se doter doit être opérationnel, évolutif, convivial, et offrant les informations nécessaires à temps réel.

Pour ceci, le système à réaliser doit satisfaire les besoins de la totalité des utilisateurs.

Nous présenterons dans ce qui suit tous les besoins fonctionnels ainsi que les besoins non fonctionnels du système :

**Besoins fonctionnels :**

* Gestion de la puissance courante ;
* Gestion Centre ;
* Gestion de la production ;
* Gestion Consommation gasoil ;
* Gestion Rapport

**Besoins non fonctionnels :**

À part les besoins fondamentaux, notre système doit répondre aux critères suivants :

* **La rapidité du traitement** : En effet, vu le nombre important d’opérations quotidiennes, il est impérativement nécessaire que la durée d’exécution des traitements s’approche le plus possible du temps réel.
* **La performance** : Un logiciel doit être avant tout performant, c’est-à-dire à travers ses fonctionnalités, répond à toutes les exigences des utilisateurs d’une manière optimale.
* **La convivialité** : Le futur logiciel doit être facile à utiliser. En effet, les interfaces utilisateurs doivent...

Être conviviale signifie être simplifier, ergonomiques et adaptées à l’utilisateur.

* 1. **Analyse de faisabilité**

Est-ce que le projet (la mise en place du logiciel) est faisable au niveau du financement, de compétence, organisation, moyens et techniques ? Dans ce cas, nous devons proposer plusieurs solutions pour la réalisation de ce projet, tout en gardant les différentes possibilités d’un meilleur projet, c’est-à-dire, ni trop coûteux, ni trop compliqué, mais facile à manipuler et rentable. Pour ce projet, nous aurons besoin de :

**Matériel :**

* Ordinateur de bureau ;
* RAM 4Go ;
* Processeur : Intel(R) Core i7, CPU @ 3.40GHZ 4.41GHZ ;
* Système d’exploitation Windows 10 (64 bits) ;
* Domaine : jirama.mg

**Logiciel :**

* Xampp, Visual Studio code, Pichon, WinDesign
* L’utilisation d’un navigateur non obsolète tel que Microsoft Edge ou Google Chrome ;

**Humaine :**

* **Maitre d’ouvrage** : personne physique ou morale propriétaire de l’ouvrage. Il est déterminé les objectifs, le budget et les délais de réalisation. C’est le Client ;
* **Maitre d’œuvre** : personne physique ou morale qui reçoit mission de la maitrise d’ouvrage pour assurer la conception et la réalisation de l’ouvrage. Les maitres d’œuvres sont :
  + Le chef du projet ;
  + Les experts ou les programmeurs
  + L’organisateurs
  + Le concepteur,
  + Le contrôleur,
  + L’investigateur,
  + Les utilisateurs

**Économique :**

La connaissance du coût du projet est très importante, afin de valoriser le budget estimé par rapport au coût défini. Pour éviter l’augmentation du coût de réalisation, on a opté pour l’utilisation des outils qui sont libres et gratuits.

**Orientation :**

Afin de démêler la limite d’opportunité précédente, il est nécessaire de mettre en place une application web ; combiner un langage de programmation web avec une base de données et les fonctionnalités essentielles pour accomplir l’objectif de l’application.

**NB** : Tous les outils de réalisation doivent être libres et gratuits.

* 1. **Analyse de l’existant**

Afin de pouvoir passer aux choix d’un outil de traitement en production de mise en œuvre des solutions proposées, il est nécessaire de passer par une analyse des dispositifs déjà présents au sein de la JIRAMA Centrale, comme matériels, logiciels et d’autres méthodes utilisées. Elle consiste à mettre à plat, de façon claire que possible l’analyse qualitative et quantitative du fonctionnement actuel de synthèse de la production électrique journalière au sein de la JIRAMA Centrale. Une analyse de l’existant comprend trois parties distinctes :

* La première consiste à recueillir l’exploitation ; elle est réalisée à partir d’entretiens ou des questionnaires, tableaux de bords, données statistiques, etc.
* La seconde consiste à analyser, classer et donner une vue synthétique de l’ensemble des rapports collectés par le domaine fonctionnel, en tenant compte des ressources humaines.
* La troisième consiste à esquisser une modélisation à grosse maille de données et des traitements.

L’état des lieux peut aboutir à une critique de l’existant de l’existant qui analyse des points positifs et négatif de l’organisation du travail déjà mise en place et dégage les améliorations à apporter : les rapports effectués et les rapports non effectuées, les services rendus et les services non rendues etc. Cette critique sera ainsi une transition vers l’analyse des besoins.

* + 1. **Existant Matériels**

Le JIRAMA est une société de produire électricité et l’eau qui supporte la vie de la population. Cette entreprise gère le rapport de la production électricité. Chaque zone dispose d’outil informatique pour les aider dans leurs rapports électriques la performance des ordinateurs varie selon le poste concerné. Au sein de la JIRAMA, le système d’exploitation les plus utilisé est bien évidement Windows, notamment Windows 10

Le tableau 5.1 ci-après contient l’inventaire des matériels informatiques existant à la JIRAMA

Tableau 5.1 : liste des matériels informatiques existants

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Postes** | **Nombres** | **Marque** | **SI** | **Caractéristique** |  |  |
| Directeur Générale |  | Ordinateur | Windows | AMD E1  1200 CPU | 500Go | 4Go |
| Responsable Communication | 2 | de  bureau HP | 10 (64 bits) | CPU  1.40 Ghz |  |  |
| Assistance du Directeur | 1 | Ordinateurs de bureau  ACER | Windows 10 (64 bits) | Intel ® core i7  CPU 2.40Ghz | 500Go | 8Go |
| Secrétaire |  |  |  |  |  |  |
| Comptable |  |  |  | Intel ® |  |  |
| Responsable d’appui au entreprises | 4 | Ordinateurs  bureau HP | Windows  10 (64bits) | Pentium®  CPU G640 T  @2.40 Ghz | 1TO | 4Go |
| Responsable de partenariat |  |  |  |  |  |  |
| Responsable de communication | 1 | Ordinateurs de bureau DELL | Windows10 (64bits) | Intel ® core i5 CUP @2.50Ghz | 320GO | 4Go |
| Responsable de rapport électricité | 2 | Ordinateurs Nec | Windows 10 (64bitys) | Intel® core i5 CUP @3.40Ghz | 750G0 | 8G0 |

* + 1. **Existant Logiciel**

Chaque ordinateur au sein de la JIRAMA a plusieurs logiciels d’application dont les plus courants sont :

* Antivirus : SMADAV 2023
* Nav web : Mozilla Firefox, Google Chrome, Microsoft Edge ;
* Logiciel Bureautique : Microsoft office 2016 (Word, Excel, Powerpoint, etc...)
* Communication : Email, Outlook

Tout d’abord, les responsables de rapport électricité qui veulent améliorer la méthode de ce rapport de la production remplir un formulaire dont le contenue sera ensuite transposé dans un fichier Excel, puis on leur fait passer un test visant à déterminer le niveau rapport pour le chef d’exploitation concerné, et enfin, pour pouvoir passer au niveau du rapport s’il doit passer ou non au niveau du rapport.

* 1. **La critique négative**

Dans cette partie nous allons parler des point fortes et points faibles de la JIRAMA, des solutions que nous proposons et de la solution que nous allons adopter.

Grace aux précédentes observations, nous sommes à présent en mesure d’identifier les problèmes avec la structure actuelle.

**L’utilisation de Microsoft office** : La plupart des sociétés existant à Madagascar utilisé des logiciel Microsoft office Excel pour faire des calcule complexe afin de gérer de gestion quoiqu’elle soit ; vu qu’il est reconnu dans le monde entiers de son efficacité. L’une de ce cas est de la société de la JIRAMA qui géré leur gestion de la production électrique avec leur logiciel. Or le traitement sous « Microsoft Office Excel » n’est pas sécurisé. Vu que cet outil est le plus utilisé actuellement dans les domaines, Microsoft office, bien que disposant d’une facilité à la portée de tous, présente des gros désavantages, notamment ;

* **Pour rapport** : il faut ré-saisir toutes les productions de toutes les centres à Madagascar comme s’il était une tout nouvelle production.

**Résultat**: il est impossible de déterminer la statistique de rapport qui saisit à chaque jour.

* **Pour la saisie** : plutôt que de puiser parmi le rapport électricité déjà collectées sur le centre lors de ses précédentes des productions, le service de la production électrique doit les ressaisir ou le meilleur des cas rechercher son non dans les archives.

**Résultat**: incorrecte des rapport, difficulté de gestion de rapport électricité donc cette méthode entraine lenteur du travail.

* 1. **Solution proposée**

Ayant prix conscient de la problématique « comment peut-on améliorer l’organisation du rapport électrique au niveau de la JIRAMA ? ». La JIRAMA à bien voulu informatiser la gestion de la synthèse de la production électrique à chaque jour.

Pour éviter les lenteurs Administratifs et avoir une bonne gestion de rapport.

**Solution 1** : la première solution proposée, est d’améliorer le mode de fonctionnement sur le traitement « Microsoft Office Excel » et de le renforcer pour mieux adapter à leurs besoins d’utilisations

**Avantages :**

* Pas de production d’utilisation ;
* Simple ;
* Habituels ;

**Inconvénients :**

* Non sécurisé ;
* Pas d’accès à distance ;
* Un utilisateur

**Solution 2** : Acheter un logiciel qui existe déjà ;

**Avantage :**

* Gains du temps et le logiciel est spécialisé à la saisie des données ;

**Inconvénients :**

* Le logiciel est couteux qui ne peut pas correspondre aux besoins attendus de l’organisation.

**Solution 3** : Concevoir et réaliser une application web de gestion de rapport d’une production électricité sein de JIRAMA.

**Avantage :**

* L’application peut répondre aux attentes du rapport.

**Inconvénients :**

* L’application demande de beaucoup du temps pour le réaliser.
  1. **Solution retenue**

Après une longue discussion et des échanges avec le responsable concerné, nous avons choisi comme décision finale, nous proposons d’informatiser le processus de suivis des productions. Nous avons décidé de réaliser un logiciel qui permettra de visualiser de la gestion de synthèse de la production électrique pour la JIRAMA. D’abord, nous allons crées une base de données pour tous les rapports de l’exploitation électrique, le service du rapport de la production, et les autres services technique. Pour cela, nous optons pour le développement d’une application objectif est la gestion de rapport d’une production électrique au sein de la JIRAMA.

# **Chapitre 6 : Conception du projet**

Apres analyse, il faut passer à la conception du système d’information pour aboutir à une base de données conforme aux attentes. Pour réussir un projet, la conception est la phase importante dans sa réalisation en informatique. Dans la conception on a choix un langage pour modélisation le sujet et d’utiliser ce langage pour représenter selon différents aspects dont chacun à l’aide d’un type de diagramme particulier. Pour créer le système d’exploitation électrique.

* 1. **. Présentation de l’outil de conception**

Il existe plusieurs outils de conception. Pour concevoir un système d’information, nous avons adopté un outil de conception tel que : Visual Paradigm, Win’Design, Paradigm Plus, Poseidon for UML, Visual UML. Nous allons présenter deux outils de conception.

* + 1. **WinDesign**

WinDesign est un outil de modélisation UML largement utilisé qui permet de créer des diagrammes complexes de manière intuitive. Il prend en charge divers types de diagrammes, notamment les diagrammes de cas d'utilisation, de classes, de séquences et d'activités. Dans le cadre de notre projet, WinDesign est utilisé pour modéliser les exigences, les fonctionnalités et l'architecture de l'application de gestion des tests et des échantillons. Il facilite également la documentation des processus métier et la communication des idées au sein de l'équipe.

* + 1. **Autres outils de conception**

Bien que WinDesign soit notre principal outil de conception, d'autres outils peuvent être envisagés pour des besoins spécifiques :

* **Visual Paradigm** : Pour des fonctionnalités avancées de collaboration et de gestion de projet.
* **Lucidchart** : Pour des diagrammes simples et une interface utilisateur conviviale.
  1. **Choix d’outils de conception**

Le choix de WinDesign pour la modélisation a été motivé par sa flexibilité, sa capacité à créer des diagrammes variés et son intégration avec les processus de développement agiles. Il permet une conception itérative, ce qui est crucial pour s'adapter aux besoins changeants du projet. WinDesign a été choisi pour sa robustesse et sa popularité dans le domaine de la modélisation UML.

* 1. **Règle de gestion**

En ingénierie, selon Wikipédia : les exigences doivent être écrient de telle maniéré qu’elles orientent la création et la modification d’un système selon les règles métier (ou règles de gestion) appropriées au contexte et au domaine et dans lequel le système doit être utilisé »

Avant de modélisation de système, il est donc important de saisir sur les exigences qui régissent les entités de la production électrique. Les règles de gestion sont la formulation de ces spécificités en phrases qui seront ensuite appliquées lors de la modélisation. Voici ceux du projet formulé à l’aide des phrases notamment composées autour d’un verbe sur lequel tourne un ou plusieurs entités, sera ensuite interprété pour aboutir aux premiers modèles. Afin de mieux structurer les données et d’éviter les redondances de la base des données, nous avons établi les règles de gestion suivante :

**RG1** : Un Centre doit être recherché plusieurs productions et la plusieurs productions doivent produire dans une Centre.

**RG2** : Un centre peut produire un ou plusieurs puissances et la puissance doit exister dans un centre.

**RG3** : Un groupe peut concerner un ou plusieurs productions.

**RG4** : Un groupe JIRAMA a de l’observation lorsqu’il obtient un évènement

**RG5** : Un centre fait entrer gasoil quand le gasoil est déjà dépensé.

**RG6** : Un centre peut exister délestage lorsque le groupe ne marche pas.

**RG7** : Centre doit avoir plusieurs pointent.

**RG8** : Un groupe est contrôle par l’heure de marche.

* 1. **Dictionnaire de données**

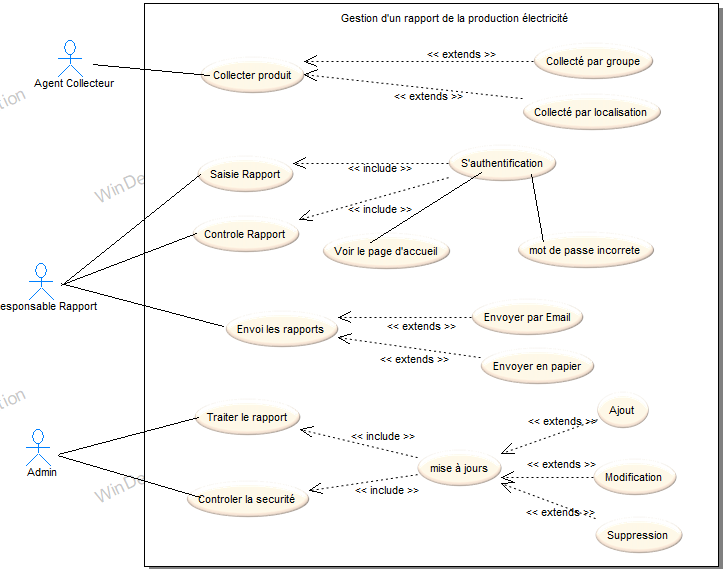
Le dictionnaire des données est l’inventaire des entités et de leurs attributs respectifs tirés de régles de gestion, il regroupe entre autres la structure de toutes les données que l’on va modéliser.

Tableau 6.1 : Dictionnaire de données de notre application

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom du champ | Description | Type | Taille | Format |
| Code | Code de date | N | 4 |  |
| CodeCentrale | Code du centrale | N | 4 |  |
| Codecons | Code de la consommation | N | 4 |  |
| Codecsp | Code de la consommation spécification | N | 4 |  |
| CodeDel | Code de delestage | N | 4 |  |
| CodeGosoil | Code de l’entrée Gasoil | N | 4 |  |
| CodeGroupe | Code du groupe | N | 4 |  |
| Codeheure | Code de l’heure | N | 4 |  |
| CodeObs | Code de l’observetion | N | 4 |  |
| CodePro | Code de la production | N | 4 |  |
| CodePuissance | Code de la puissance | N | 4 |  |
| CodeRapp | Code de rapport | N |  |  |
| CodetempEau | Code dela temperature de l’eau | N | 4 |  |
| DateDebut | Date de début | Date |  | jj-mm-aaaa |
| DateFin | Date fin | Date |  | jj-mm-aaaa |
| Datemaj | Date de mise à jour | Date |  | jj-mm-aaaa |
| HeureMarche | Heur de marche | Time | 08 | Hh :mm :ss |
| NombreCons | Nombre de consommation | AN | 10 |  |
| NombreCsp | Nombre de consomation spécifique | AN | 10 |  |
| NombrePro | Nombre de la production | AN | 20 |  |
| NombrePuissance | Nombre de la puissance | AN | 20 |  |
| NomCentrale | Nom du centrale | N | 20 |  |
| NomPuissance | Nom du puissance | N | 20 |  |
| TypeCons | Type de la consommation | AN | 30 |  |
| TupeGroupe | Type de Groupe | AN | 20 |  |
| TypePro | Type de la Production | AN | 30 |  |

* 1. **Modélisation** 
     1. **Schéma de diagramme de cas d’utilisation globale**

Le diagramme de cas d'utilisation global représente les interactions entre les différents acteurs (administrateur, technicien, médecin) et le système de gestion des tests et des échantillons. Ce diagramme sera créé dans WinDesign pour visualiser les fonctionnalités principales et les interactions.

Figure 6.1 : Diagramme global de cas d’utilisation

* + 1. **Spécification détaillée des cas d’utilisation d’un système**

Il modélise une interaction entre le système informatique à concevoir et l’utilisateur qui interagisse avec le système c’est-à-dire que les cas d’utilisation permettent d’exprimer le besoin des utilisateurs d’un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d’une vision informatique. Il capture le comportement d’un système, d’un sous-système, d’une classe ou d’un composant tel qu’un utilisateur extérieur le voit. Il partage la fonctionnalité du système en unités cohérentes, les cas d’utilisation : ayant un sens pour les acteurs. Il ne faut pas négliger cette première étape pour produire un logiciel conforme aux attentes des utilisateurs.

Pour élaborer les cas d’utilisation, il faut se fonder sur des entretiens avec les utilisateurs. Un diagramme de cas d’utilisation est composé par : l’acteur et le Cas d’utilisation.

L’acteur est l’idéalisation d’un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système.

Le cas d’utilisation est une unité cohérente d’une fonctionnalité visible de l’extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l’acteur qui l’initie.

Un cas d’utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service [Dr Hasina., 2024].

***Description du cas d’utilisation « S’Authentifier »***

Tableau 6.2 : Raffinement du cas d’utilisation « S’authentifier »

|  |  |
| --- | --- |
| 1. Sous cas d’utilisation : | 1. S’authentifier |
| 1. Acteur : | 1. Agents |
| 1. Pré-condition: | 1. Avoir login et mot de passe correct. |
| 1. Post-condition: | 1. Agents authentifié. |
| Description de scénario de base : | * L’utilisateur saisit son login et son mot de passe. * L’utilisateur clique sur «Login ». * Le système vérifie la combinaison login et mot de passe. * Login et mot de passe correcte, le système affiche l’accueil. |
| Exception : | S’il n’y a aucun Agent enregistré dans la base, un message ne s’affiche « Pas Agents enregistré ». |

En s’authentifiant pour avoir un accès au système, l’utilisateur parvient à son compte via le mot de passe et le nom d’utilisateur qu’il a choisi lors de la création de celui-ci. Ces données seront enregistrées dans la table intitulée utilisateur et c’est la raison pour laquelle le contrôleur effectue une opération de lecture sur cette table à la rechercher de l’utilisateur en question avant de permettre l’accès ou le refuser.

* 1. **Diagrammes de séquences** 
     1. **Diagramme de séquence de cas d’utilisation «S’Authentifier»**

Ce diagramme montre les interactions lors de la création d'un échantillon, y compris l'envoi des données et la réponse du serveur.

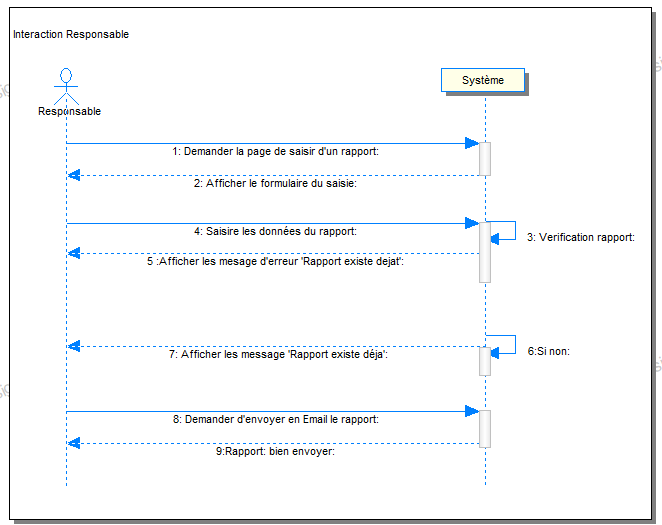


Figure 6.2 : Diagramme de séquence de création d'échantillon

* 1. **Diagramme d’activité**

Le diagramme d'activité est attaché à une catégorie de classe et décrit le déroulement des activités de cette catégorie. Le déroulement s'appelle "flot de contrôle". Il indique la part prise par chaque objet dans l'exécution d'un travail. Il sera enrichi par les conditions de séquencement. Il pourra comporter des synchronisations pour représenter les déroulements parallèles.

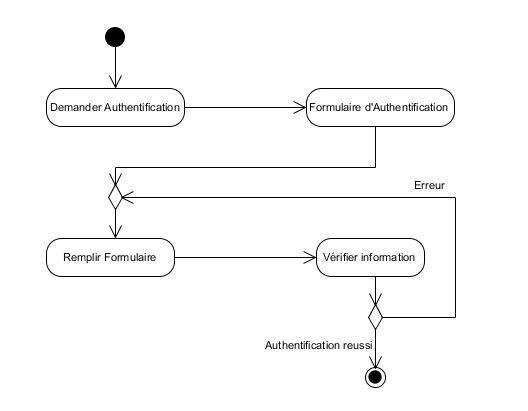
La notion de couloir d'activité va décrire les responsabilités en répartissant les activités entre les différents acteurs opérationnels. Chaque activité sera placée dans une colonne (couloir) qui correspond à l'acteur.

Figure 6.3: Diagramme d’activité de l’authentification

Et pour notre application, son diagramme d’activité en globale est représenté suivante :

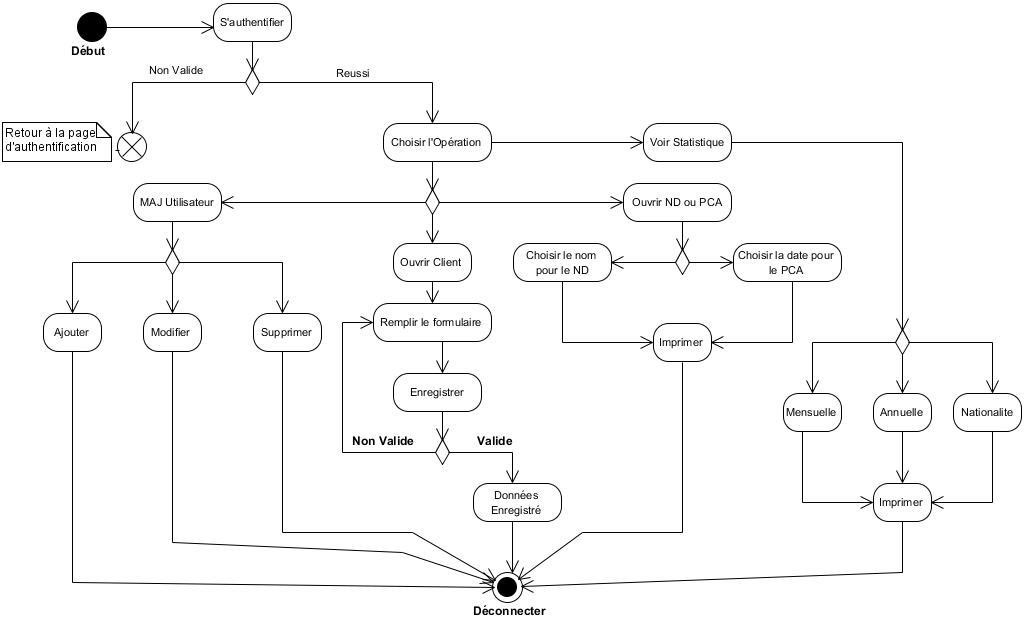


Figure 6.4 : Diagramme d’Activité Globale de l’application

* 1. **Diagramme de classe du projet**

Le diagramme de classe décrit les principales classes du système, leurs attributs, méthodes et relations. Il représente la structure des données de l'application de gestion des tests et des échantillons

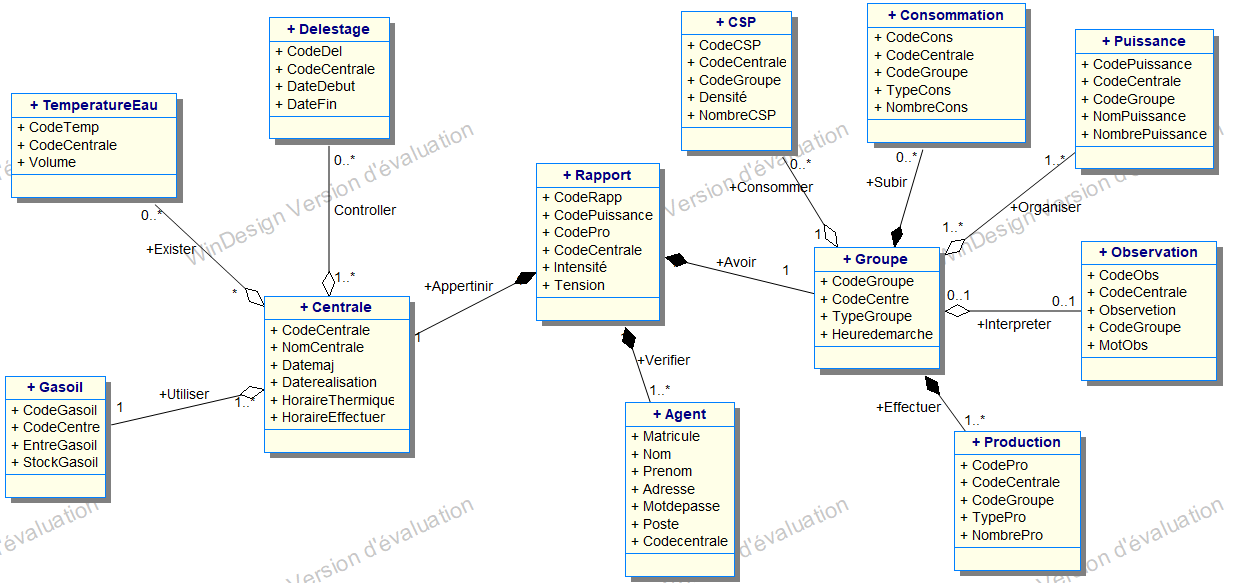
****

Figure 6.5 : Diagramme de classe de l’application

**PARTIE III – REALISATION DU PROJET**

# **Chapitre 7 : Spécification des Outil de Réalisation**

Pour mettre en œuvre le logiciel, il faut déterminer quelle technologie est la mieux adapter au vu des existants et des bus à atteindre, il faut également chois des outils adaptés à ces dernières afin de permettre une mise en œuvre réglementaire.

* 1. **Présentation des technologies utilisées**

D'après le titre, ce chapitre consiste à présenter les outils nécessaires et les technologies avec lesquelles le développer pour aboutir à l'objectif de créer une application web.

Pour ce projet, les outils suivants structureront la réalisation :

* Le langage de programmation PHP ;
* Les Framework web (optionnel) ;
* Le SGBD ;
* Le Serveur d’application ;
* L’environnement de développement.
  + 1. **Le langage de programmation PHP**

Il existe plusieurs types des langages courants pour développer une application web, l'objectif est de savoir quel langage est le plus adapté aux besoins.

PHP [7] : **HyperText Preprocessor**, plus connu sous son sigle **PHP** (acronyme récursif), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur http [h], mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

PHP appartient à la grande famille des descendants du C, dont la syntaxe est très proche. En particulier, sa syntaxe et sa construction ressemblent à celles des langages Java et Perl, à ceci près que du code PHP peut facilement être mélangé avec du code HTML au sein d'un fichier PHP.

Ses principaux atouts sont :

* La gratuité et la disponibilité du code source (PHP est distribué sous licence GNU GPL).
* Sa richesse fonctionnelle : PHP comporte plus de 1000 fonctions.
* La simplicité d’écriture des scripts ;
* La disponibilité sur le web de nombreux scripts PHP prêts à l’emploi ;
* La disponibilité d’inclure le script PHP au sein d’une page HTML
* La simplicité de liaison avec des bases de données. De nombreux système de base de données sont supportés, mais le plus utilisé avec le PHP et MySQL, le système de base de données gratuit et disponible sur la plateforme Unix, Lunix, et Windows.

Ce language de programmation permet essentiellement de construire des site web dynamique, particulièrement lorsqu’ils sont reliés à une base de données.

* + 1. **Framework PHP**

Il existe de nombreux outil de logiciels informatique permettant de développer rapidement et efficacement des application web enrichies (RIA, Rich Internet Application). Parmi eux, on retiendra deux grandes catégories (excepté les serveurs applicatifs) :

* Le CMS (Contentent Management System) en français, (Système de Gestion de Contenu) permettant la création de site disposant surtout d’un contenu journalistique si l’on peut dire ;
* Les Framework web ou plateformes applicatives

**a. *Définition***

Un Framework est un espace de travail modulaire, constitué d’un ensemble de bibliothèques, d’outils et de convention permettant de développement rapide d’application. Il fouit aussi une structure pour l’application. Par extension, un Framework web est un ensemble d’outil spécialisés pour le développement d’applications web.

***b. Avantages et Inconvénients***

Les avantages de l’utilisation de ces plateformes applicatives sont : liberté, rapidité, qualité, cohérence et pérennité (Cohérence et pérennité sont deux atouts essentiels).

L’apprentissage de la plateforme est plus ou moins long et fastidieux, tous ne sont pas aussi bien documentés, l’apprentissage de l’outil est un élément très important à intégrer dans le développement du projet, la rapidité de développement du projet va dépendre de la bonne connaissance de l’outil.

Le modèle-vue-contrôleur (en abrégé MVC, de l’anglais Model-View-Controller) est un patron d’architecture et une méthode de conception qui organise l’interface homme-machine (IHM) d’une application logicielle [La notion de MVC] [r]. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs

Le motif MVC a été créé par Trygve Reenskaug lors de sa visite du Palo Alto Research Center (abr. PARC) en 1978 [Stephen Walther]. Le nom original est **Thing Model View Editor pattern**, puis il a été rapidement renommé **Model-View-Controller pattern**. Le patron MVC a été utilisé la première fois pour créer des interfaces graphiques avec le langage de programmation Smalltalk en 1980.

**Avantages du MVC**

L’approche MVC apporte de réels avantages :

* Une conception claire et efficace grâce à la séparation des données de la vue et du contrôleur.
* Un gain de temps de maintenance et d’évolution du site.
* Une plus grande souplesse pour organiser le développement du site entre différents développeurs (indépendance des données, de l’affichage (webdesign) et des actions).

**Inconvénients**

L’inconvénient majeur du modèle MVC n’est visible que dans la réalisation de petits projets, de sites internet de faible envergure. En effet, la séparation des différentes couches nécessite la création de plus de fichiers (3 fois plus exactement) :

* Un fichier pour le modèle.
* Un fichier pour le contrôleur
* Un fichier pour la vue

Il n’est donc pas très intéressant de recourir à ce système dans ce cas-là.

**Modèles**

Le modèle correspond aux données, la plupart du temps stockées dans une base de données. Mais celles-ci peuvent également être contenues dans un fichier XML ou dans des fichiers texte. Les données peuvent être exploitées sous forme de classes dans un langage de programmation orientée objet (comme PHP5).

**Vues**

Le modèle correspond aux affichages ou bien rendu qu’on souhaite montrer aux utilisateurs.

**Contrôleurs**

Le contrôleur est l’élément qui va utiliser les données pour les envoyer à la vue. Son rôle est donc de récupérer les informations, de les traiter en fonction des paramètres demandés par la vue (par l’utilisateur, exemple : afficher les derniers articles), puis de renvoyer à la vueles données afin d’être affichées. Le contrôleur peut donc instancier différents objets (classe User, classe Articles, ...) qui enverront des requêtes vers la base de données ou récupéreront des données XML. La transmission à la vue des données récupérées se fait généralement à l’aide d’un \*\*Template\*\* : insertion des données dans un fichier HTML qui va être envoyé au navigateur.

* 1. **SGBD**
     1. **Définition d’une base de données**

Une base de données est une entité dans laquelle il est possible de stocker des données de façon structurer et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents. Ainsi, la notation de base de données est généralement couplée à celle de réseau, afin de pouvoir mettre en commun ces informations, d’où le nom de la base. On parle généralement le système d’information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données.

* + 1. **Utilité d’une base de données**

Une base de données permet de regrouper des données au sein d’un même enregistrement.

Cela est d’autant plus utile que les données informatiques sont de plus en plus nombreuses. Une base de données peut être locale, c’est-à-dire utilisable sur une machine par un utilisateur, ou bien repartie, c’est-à-dire que les informations sont stockées sur des machines distantes et accessibles par réseau. L’avantage majeur de l’utilisateur d’une base de données est la possibilité de pouvoir être accédées par plusieurs utilisateurs simultanément.

* + 1. **Choix du SGBD**

Pour créer une application, il est nécessaire d’utiliser un outil de stocker des données afin de stocker physiquement les données. En effet, il existe plusieurs type SGBD qui permettent de stocker des données mais quel est le plus adapté pour ce projet. D’âpres les choix de langage et Framework utiliser précédemment, les choix SGBD doit correctement aux besoins et être compatible avec ces choix. Pour cela, avant de présenter les différents SGBD, il est important d’évaluer les capacités requises pour mettre en œuvre le projet.

Pour ce projet, le stockage de données consiste à stocker des informations textuelles tous simplement et pas de données médiatisées, donc l’utilisation d’un capacité moyenne est largement suffisante. Il existe plusieurs SGBD et avant d’en choisir une, voici un tableau de comparaison de quelques SGBD.

Tableau 7.1: Comparaison du SGBD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SGBD** | **MySql** | **PostegresSql** | **Oracle** | **XAMPP** |
| Gratuit | OUI | OUI | OUI/NON | OUI |
| Simple à utiliser | OUI | NON | NON | OUI |
| Capacité | MOYENNE | MOYENNE | GRANDE | MINIMALE |
| Facile à installer | OUI | NON | NON | OUI |
| Facile à maintenir | OUI | OUI | NON | OUI |

A partir de ce tableau, nous pouvons déduire que MySQL est le meilleur choix pour le projet,  
PostgreSQL présentant quelques difficultés au niveau de son utilisation, et d’un autre côté, Oracle et SQLite présentent une trop grande incompatibilité avec les capacités requises.

* + - 1. **Serveur d’application**

Dans le tableau suivant la comparaison des différents serveurs

Tableau 7.2 : Comparaison des différents serveurs de développement

|  |  |  |
| --- | --- | --- |
| **Serveur** | **Plateforme** | **Contenue du paquetage** |
| WampServeur 2.5 | Windows | Apache, MySQL, PHP |
| Easy PHP 5.3.9 | Windows | Apache MySQL, PHP |
| XAMPP 7.0.13 | Linux, Windows, OS X | Apache MySQL, PHP |

* 1. **Caractéristique des SGBD**

L’architecture à trois niveaux définis par le standard permet d’avoir une indépendance entre les données et les traitements. D’une manière générale, un SGBD doit avoir les caractéristiques suivantes :

* **Indépendance physique** : le niveau physique peut être modifié indépendant du niveau conception. Cela signifie que tous les aspects matériels de la base de données n’apparaissent pas pour l’utilisateur, il s’agit simplement d’une structure transparente de représentation des informations.
* **Indépendance logique** : le niveau conceptuel doit pouvoir être modifié sans remettre en cause le niveau physique, c’est-à-dire que l’administrateur de la base doit pouvoir la faire évoluer sans que cela gêne les utilisateurs.
* **Rapidité d’accès** : le système doit pouvoir fournir les réponses à la requête le plus vite possible, cela implique des algorithmes de recherche rapides.
* **Administration centralisée** : le SGBD doit pouvoir éviter dans la mesure du possible des informations redondantes, afin d’éviter d’une part un gaspillage d’espace mémoire mais aussi des erreurs.
* **Sécurisation des données** : le SGBD doit représenter des mécanismes permettant de gérer les droits d’accès aux données selon les utilisateurs.
  1. **Points forts et faibles des SGBD**

Dans le cadre du développement de notre projet, le choix du Système de Gestion de Base de Données (SGBD) est crucial pour assurer une gestion optimale des données, notamment pour le stockage des échantillons, des résultats de tests, et des informations utilisateurs. Voici un aperçu des points forts et faibles des principaux SGBD considérés.

* + 1. **PostgreSQL (SGBD choisi) :**

**Points forts :**

**Performances élevées** : PostgreSQL est reconnu pour sa gestion efficace des transactions et son support des opérations complexes.

**Support des types de données complexes** : PostgreSQL permet de gérer des types de données avancés comme JSON, ce qui est essentiel pour stocker des objets complexes.

**Sécurité** : PostgreSQL offre de robustes mécanismes de contrôle d'accès et de sécurité, incluant des certificats SSL et des stratégies basées sur des rôles utilisateurs.

**Open-source** : Ce SGBD est libre d’utilisation, ce qui réduit les coûts du projet.

**Points faibles :**

**Configuration avancée** : Pour tirer le meilleur parti de PostgreSQL, une expertise technique est parfois nécessaire, notamment pour les grandes bases de données.

**Consommation de ressources** : PostgreSQL peut être plus gourmand en ressources comparé à d'autres SGBD plus légers comme SQLite.

* + 1. **MySQL :**

**Points forts :**

**Large communauté** : MySQL est largement utilisé dans l'industrie, ce qui garantit une vaste documentation et un support communautaire.

**Simplicité** : MySQL est relativement simple à installer et à configurer, même pour des projets à petite échelle.

**Points faibles** :

**Manque de support avancé pour certains types de données** : Contrairement à PostgreSQL, le support de JSON ou d'autres types de données complexes est limité.

**Transactions limitées** : MySQL peut avoir des limites dans la gestion des transactions complexes par rapport à PostgreSQL.

* + 1. **XAMPP :**

**Points forts :**

**Légèreté :** XAMPP est un SGBD très léger, ce qui en fait un excellent choix pour les petites applications ou des tests locaux.

**Sans configuration serveur** : Il fonctionne sans serveur, ce qui simplifie son utilisation dans des applications embarquées.

**Points faibles :**

**Pas conçu pour les grandes bases de données** : Pour des projets nécessitant une forte concurrence d'accès aux données, SQLite montre rapidement ses limites.

**Conclusion :** Pour ce projet, **PostgreSQL** a été choisi en raison de sa robustesse, de sa gestion avancée des données, et de ses performances pour les transactions complexes nécessaires à la gestion des échantillons et des résultats de tests dans un laboratoire.

* 1. **Présentation du Framework Laravel**
     1. **Introduction**

Laravel est un [framework](https://fr.wikipedia.org/wiki/Framework) [web](https://fr.wikipedia.org/wiki/Web_application) [open-source](https://fr.wikipedia.org/wiki/Open-source) écrit en [PHP](https://fr.wikipedia.org/wiki/PHP)[1](https://fr.wikipedia.org/wiki/Laravel#cite_note-WebDesignerArticle-1) respectant le principe [modèle-vue-contrôleur](https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur) et entièrement développé en [programmation orientée objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet). Laravel est distribué sous [licence MIT](https://fr.wikipedia.org/wiki/Licence_MIT), avec ses sources hébergées sur [GitHub](https://fr.wikipedia.org/wiki/GitHub).

* + 1. **Historique**

Laravel a été créé par Taylor Otwell en juin 2011.

Le référentiel Laravel/laravel présent sur le site [GitHub](https://fr.wikipedia.org/wiki/GitHub) contient le code source des premières versions de Laravel. À partir de la cinquième version, le framework est développé au sein du référentiel Laravel/framework.

En peu de temps, une communauté d'utilisateurs du framework s'est constituée[1](https://fr.wikipedia.org/wiki/Laravel#cite_note-WebDesignerArticle-1), et il est devenu en 2016 le projet PHP le mieux noté de GitHub.

Laravel reste pourtant basé sur son grand frère [Symfony](https://fr.wikipedia.org/wiki/Symfony), pour au moins 30 % de ses lignes (utilisation de "Symfony component").

* + 1. **Historique des versions**

Les versions [LTS](https://fr.wikipedia.org/wiki/Long-term_support) sont supportées pendant 2 ans pour les corrections de bogues, 3 ans pour les patchs de sécurité. Les autres versions sont supportées pendant 6 mois pour les corrections de bogues, et 1 an pour les patchs de sécurité

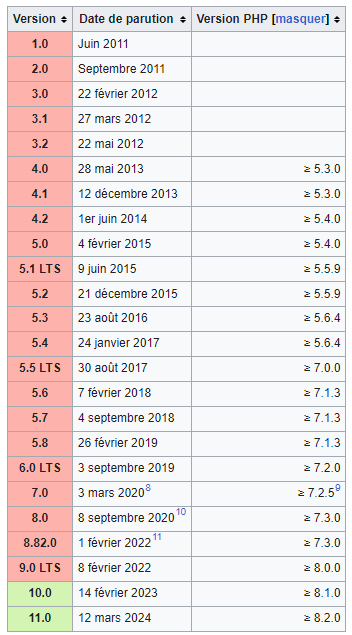


Figure 7.1 : historique de version laravel

* + 1. **Fonctionnalités**

La version 5.0 de Laravel nécessite au minimum PHP 5.4 et son installation est basée sur le gestionnaire de paquets [Composer](https://fr.wikipedia.org/wiki/Composer_(logiciel))[1](https://fr.wikipedia.org/wiki/Laravel#cite_note-WebDesignerArticle-1). Depuis la version 5.3, Laravel nécessite PHP 5.6 au minimum. La version 5.6 (février 2018) nécessite quant à elle PHP 7.1.3 au minimum.

Laravel fournit des fonctionnalités en termes de routage de requête, de [mapping objet-relationnel](https://fr.wikipedia.org/wiki/Object-relational_mapping) (un système baptisé Eloquent implémentant [Active Record](https://fr.wikipedia.org/wiki/Active_record_(patron_de_conception))), d'authentification, de vue (avec Blade), de [migration de base de données](https://fr.wikipedia.org/wiki/Migration_(informatique)), de [gestion des exceptions](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_d%27exceptions) et de [test unitaire](https://fr.wikipedia.org/wiki/Test_unitaire).

L'équipe Laravel propose également un micro-framework plus léger, Lumen.

# **Chapitre 8 : Mise en œuvre et implémentation**

Dans la mise en œuvre, nous allons parler de l‘architecture de l’application, qui est la manière dont l‘application interagit avec les composants matériels et logiciels, elle décrit d‘une manière symbolique et schématique les différents composants d‘un ou plusieurs systèmes informatiques, elle se subdivise en 2 parties : l‘architecture matérielle et l‘architecture logicielle. Ensuite, nous allons voir quelques extraits de code important de notre application.[Karl Eugen K & al, 2008].

* 1. **Architecture logicielle**

L'architecture logicielle décrit la structure globale du logiciel, notamment la façon dont les composants et les modules interagissent entre eux pour répondre aux besoins du système.

**Front-end** : L'interface utilisateur est développée en Next.js, un framework JavaScript basé sur React, qui permet de gérer le rendu côté serveur (SSR) ainsi que la génération de sites statiques (SSG). Cela permet une meilleure performance et un bon référencement SEO.

**Back-end** : Le serveur est implémenté avec Nest.js, un framework back-end basé sur Node.js qui utilise TypeScript. Nest.js offre une structure modulaire et une architecture basée sur des services, facilitant ainsi la maintenance, l’extension et l’intégration avec les bases de données et les autres API.

**Base de données** : PostgreSQL est utilisé comme système de gestion de base de données relationnelle. Ce choix est motivé par sa robustesse, ses performances élevées, et sa compatibilité avec les transactions complexes ainsi que les fortes contraintes de sécurité.

**Sécurité et authentification** : L'authentification des utilisateurs est gérée par des JSON Web Tokens (JWT), qui permettent d'assurer des sessions sécurisées tout en évitant de stocker des informations sensibles côté serveur.

**Communication API** : Le back-end expose des API RESTful qui permettent aux différents services de communiquer entre eux et avec le front-end. Chaque service est responsable de sa propre logique métier.

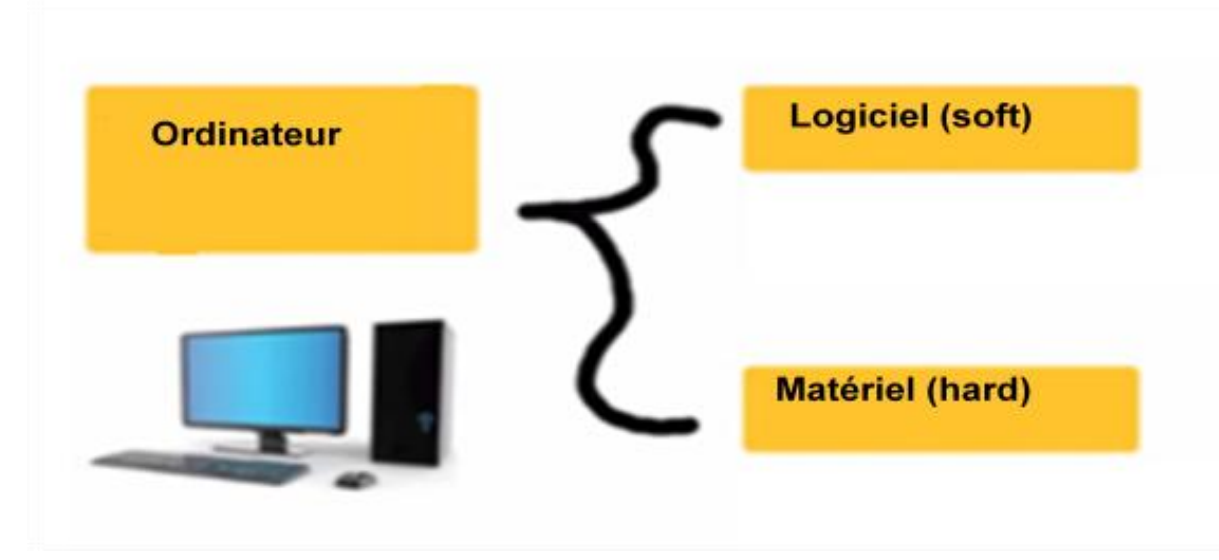


Figure 8.1 2: Les deux parties principales

La partie logicielle regroupe tous les programmes qu’on trouve dont un ordinateur, ces programmes peuvent être :

* Les systèmes d’exploitation qui permettent d’exploiter les ressources matérielles et d’installer les autres types de programmes (Windows, linux, MacOs, …) ;
* Les logiciels d’applications qui sont des programmes dédiés à  
  des applications bien spécifiées (Word, Excel, Paint …) ;

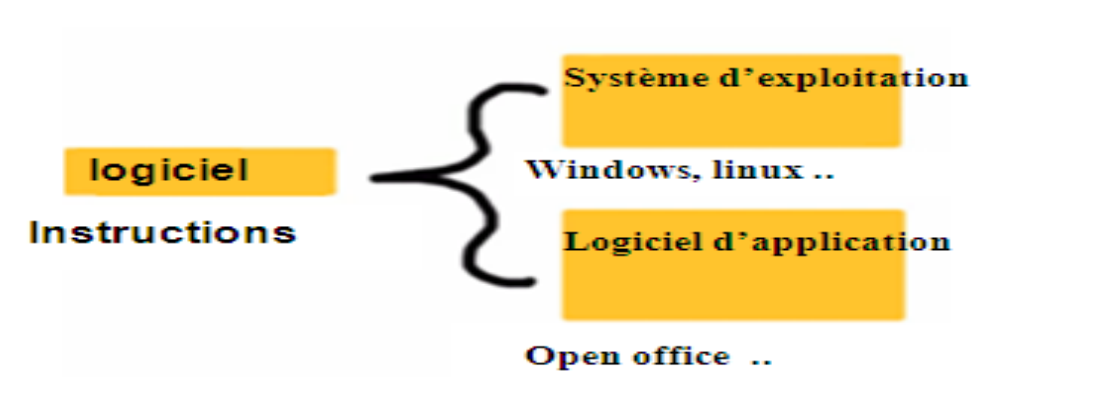


Figure 8.2 3: les instructions de logiciel

Donc l’utilisateur va utiliser l’ordinateur grâce aux deux parties matérielles et logicielles, il peut par exemple utiliser open office (partie logicielle) pour écrire un texte, mais il ne peut faire ça que si open office est installé dans le système d’exploitation (partie logicielle), et ce système d’exploitation permet à l’utilisateur d’utiliser les ressources matérielles tels que le clavier, la sourie et l’écran (partie matérielle).

Nous nous intéressons dans ce chapitre à la partie matérielle. Cette partie appelée aussi partie physique est composée de trois parties.

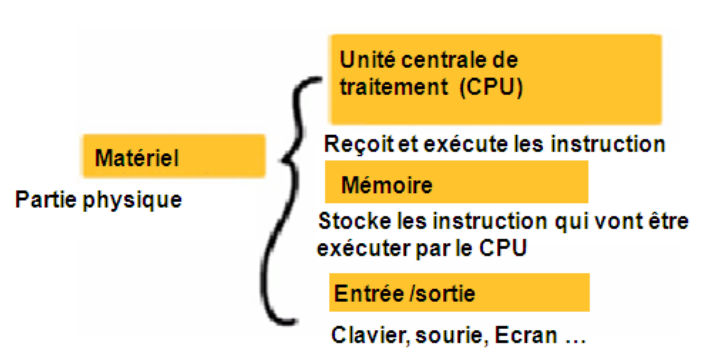


Figure 8.3: la partie matérielle des ordinateurs

Avant d’expliquer les différentes parties, on va analyser quelques définitions qui sont nécessaires pour comprendre la suite du chapitre.

* Notion de donnée : une donnée brute qui n’a pas subi de traitement ;
* Notion d’information : Une information est considérée comme une donnée traitée (la moyenne d’un étudiant est une information qui est calculée – traitée –) ;
* Notion d’instruction : Elle correspond à une action simple ;
* Notion de programme : un ensemble d’instruction qui ont un but précis et claire ;
* Notion de processus : un programme qui est en train d’être exécuter par l’ordinateur, c’est un programme en exécution ;
* Notion d’ordinateur : c’est une machine capable de recevoir des données, de les traitées suivant un programme pour obtenir des résultats.
  1. **Architecture matérielle**

L’architecture matérielle comprend toutes les caractéristiques générales, la conception, le choix et l’organisation des différents dispositifs électroniques des appareils  
informatiques.

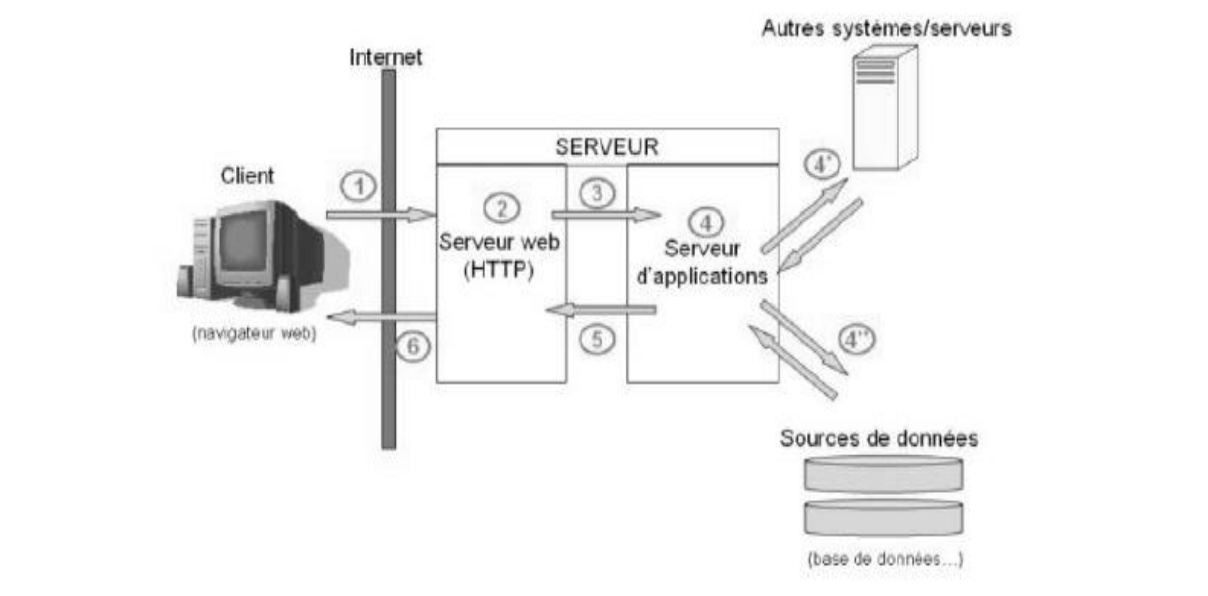


Figure 8.4 : Architecture matérielle

L’architecture matérielle décrire l’agencement ou la disposition des composants informatique et comment ils interagissent pour pouvoir fonctionnement correctement.

L’application, utilisant un SGBD client/serveurs est disposé de telle sorte que chaque PC disposant des applications puis communiquer avec la machine serveurs.

Pour utiliser correctement l’application en question, la figure nous montre la disposition de matériel requise pour que l’application soit utilisable [Rakotonirina V., 2014].

Ce schéma peut se résumer comme suit :

* Le client émet une requête (Appelle une URL) pour demander une ressource au  
  serveur. Exemple : http://leserveur.com/home. Il ne sait pas ici si la réponse qui va lui  
  parvenir est statique (page HTML simple) ou dynamique (générée par une application  
  WEB). Dans notre cas, il s'agit d'une application répondant à l'adresse home sur le  
  serveur leserveur.com.
* Côté serveur, c'est le serveur web (exemple : Apache) qui traite les requêtes HTTP  
  entrantes. Il traite donc toutes les requêtes, qu'elles demandent une ressource statique  
  ou dynamique. Seulement, un serveur HTTP ne sait répondre qu'aux requêtes visant  
  des ressources statiques. Il ne peut que renvoyer des pages HTML, des images ;
* Ainsi, si le serveur HTTP s'aperçoit que la requête reçue est destinée au serveur  
  d'applications, il la lui transmet. Les deux serveurs sont reliés par un canal, nommé  
  connecteur.
* Le serveur d'applications (Apache) reçoit la requête à son tour. Il est, lui, en mesure de la traiter. Il exécute donc le morceau d'application (le contrôleur) auquel est destinée la requête, en fonction de l'URL. Cette opération est effectuée à partir de la configuration du serveur.
* Une fois sa réponse générée, le serveur d'applications la renvoie, par le connecteur, au  
  serveur web. Celui-ci la récupère comme s'il était lui-même allé chercher une ressource statique. Il a simplement délégué la récupération de la réponse, et celle-ci a été générée, mais ce n'est plus le problème.
* La réponse est dorénavant du simple code HTML, compréhensible par un navigateur. Le serveur HTTP peut donc retourner la réponse au client.
  1. **Architecture microservice** 
     1. **Introduction**

L'architecture microservice est un style architectural où une application est divisée en plusieurs petits services indépendants, chacun étant responsable d'une fonctionnalité spécifique du système. Contrairement aux architectures monolithiques, chaque service est autonome et peut être déployé, testé, et mis à jour indépendamment des autres.

Les microservices communiquent entre eux par des protocoles légers, généralement des API REST ou des messages asynchrones, et chaque service dispose de sa propre base de données ou stockage de données.

* + 1. **La différence entre Architecture monolitiques et Architecture microservices.**

**Architecture monolithique :**

Une architecture monolithique regroupe l'ensemble des fonctionnalités d'une application en un seul bloc, où tous les modules sont interconnectés. Cela signifie que chaque modification, mise à jour ou débogage nécessite de manipuler l'intégralité de l'application. Bien que cette approche soit simple à mettre en œuvre pour de petites applications, elle devient complexe à maintenir et à déployer au fur et à mesure que le système s'étend.

**Architecture microservices :**

Contrairement à une architecture monolithique, l'architecture microservice divise les fonctionnalités en services indépendants. Chaque microservice est dédié à une fonctionnalité spécifique, et peut être développé, testé et déployé de manière autonome. Cela rend le système plus flexible, plus facile à maintenir, et plus adapté aux environnements à grande échelle.

* + 1. **Les avantages clés des microservices**

**Scalabilité** : Les microservices permettent de faire évoluer certaines parties de l'application indépendamment. Cela signifie que si une fonctionnalité spécifique connaît une augmentation de la demande (par exemple, la gestion des résultats de tests dans un laboratoire), elle peut être mise à l'échelle sans affecter les autres parties du système.

**Déploiement indépendant** : Chaque service peut être déployé indépendamment. Cela facilite la mise à jour continue des composants sans perturber l'ensemble du système.

**Tolérance aux pannes** : En cas de panne d’un service particulier, le reste de l’application continue de fonctionner normalement. Cela améliore la fiabilité globale du système.

**Flexibilité technologique** : Les développeurs peuvent utiliser différentes technologies pour chaque microservice, en fonction des besoins spécifiques (par exemple, un microservice peut être développé en Node.js, un autre en Python). 9

**Meilleure gestion des équipes** : Dans des projets de grande envergure, différentes équipes peuvent travailler sur des microservices distincts, ce qui améliore la productivité et facilite la gestion des tâches.

* + 1. **Les inconvénients des microservices**

**Complexité accrue** : Gérer une architecture microservices est plus complexe qu'une architecture monolithique, car il faut gérer les communications inter-services, la cohérence des données entre les différents services, et assurer la surveillance de l'ensemble du système.

**Problèmes de latence** : Étant donné que les microservices communiquent entre eux via des appels réseau, cela peut ajouter de la latence par rapport à un système monolithique où tout se passe en interne.

**Problèmes de sécurité** : Chaque microservice expose généralement une API, augmentant ainsi les points d'entrée possibles pour les attaques. Il est donc nécessaire de sécuriser chaque service de manière indépendante.

**Gestion des transactions complexes** : Assurer la cohérence des données à travers plusieurs microservices peut devenir difficile, notamment pour les transactions qui impliquent plusieurs services.

* + 1. **Conclusion d’architecture microservice**

L'architecture microservice est idéale pour les applications à grande échelle, en évolution rapide, ou nécessitant une disponibilité continue et un déploiement indépendant. Cependant, elle introduit également des défis techniques, notamment en termes de gestion des communications inter-services, de sécurité et de cohérence des données. Dans le cadre du projet, cette architecture a été choisie pour permettre la flexibilité, la modularité et l’évolutivité, tout en garantissant la fiabilité du système global.

# **Chapitre 9 : Présentation de l’application**

Après avoir choisi les outils nécessaires au développement du projet, il est à présent temps de passer à la pratique. La réalisation parlera notamment de l’interface de l’application web, et de son fonctionnement en général.

* 1. **Présentation des interfaces**

Dans cette partie, nous allons présenter les différentes de l’application.

* + 1. **Front office**

Selon Wikipédia : « une application de Front office ou logiciel de Front office est la partie du système informatique accessible aux utilisateurs finaux ou aux clients, par opposition au back office ».

* + 1. **Back office**

Selon Wikipédia : « une application de back office est la partie d’un système informatique qui n’est pas accessible aux utilisateurs finaux ou aux clients, par opposition à une application de Front office ».

En d’autres termes, Front Office (littéralement "boutique") et Back Office (littéralement "arrière-boutique") sont généralement utilisés pour décrire les parties de l’entreprise (ou de son système de la production électrique) dédiées respectivement à la relation directe avec le client et à la gestion propre de l’entreprise.

* 1. **Présentation de l’application web**

Par définition « une application web est un site web dont le contenu des pages est partiellement indéterminé. Le contenu final d’une page est déterminé uniquement lorsque l’utilisateur requiert une page depuis le serveur web. Le contenu final d’une page variant d’une requête à une autre en fonction des actions de l’utilisateur, ce type de page est appelé page dynamique ».

La présentation d’application sert à présenter l’interface de l’application, les gestions d’erreurs, le guide de l’utilisation, la fonctionnalité, les graphiques, etc.

* + 1. **Interface de l’application**

L’interface de l’application est très importante au niveau de l’ergonomie d’IHM, elle est la première visualisée par utilisateur : Dans ce cas, elle doit être attirante, simple et efficace afin d’éviter que l’utilisateur se plaigne du désagrément d’un coup d’œil, c’est-à-dire qu’il faut bien définir l’objectif de l’application en première vue.

En parlant d’ergonomie d’Interface Homme Machine (IHM), il existe de type de standardisation d’interface appelé « standards de facto » qui est les normes internationales de visualisation d’interface d’application ou site web comme suit :

* Dans le choix des couleurs : les couleurs utilisées devront correspondre aux logos de l’entreprise et ne doivent pas dépasser plus de 3 variantes dans une application ;
* Dans le choix des typographies : Il est fortement contre-indiqué d’utiliser plus de deux polices sur une application web, les polices conseillées sont : Arial, Verdana, Helvetia, etc. et les polices stylisées doivent être utilisées avec parcimonie.
* Dans la disposition des informations : l’application doit comporter des menus, des sous-menus et des utilitaires permettant de naviguer et d’exécuter les différentes tâches possibles sur l’application. Ces derniers auront été classés par ordre d’utilisation.

D’après la standardisation de facto sur le choix des couleurs et l’analyse du couleur logo de la JIRAMA, nous permettons de définir les couleurs utilisées pour l’application :

* Le font d’application sur l’entête est blanc gris.
* Couleur dans le font est blanc.
* Le grand texte est bleu ainsi que sur le paragraphe son noire.

En respectant, la standardisation de facto précédente, l’interface générale de l’application devrait se présenter comme la figure 9.1 qui représente la page au niveau de front office.

****

Figure 9.1 : interface générale de l’application

* + 1. **Liste des agents dans la centrale**

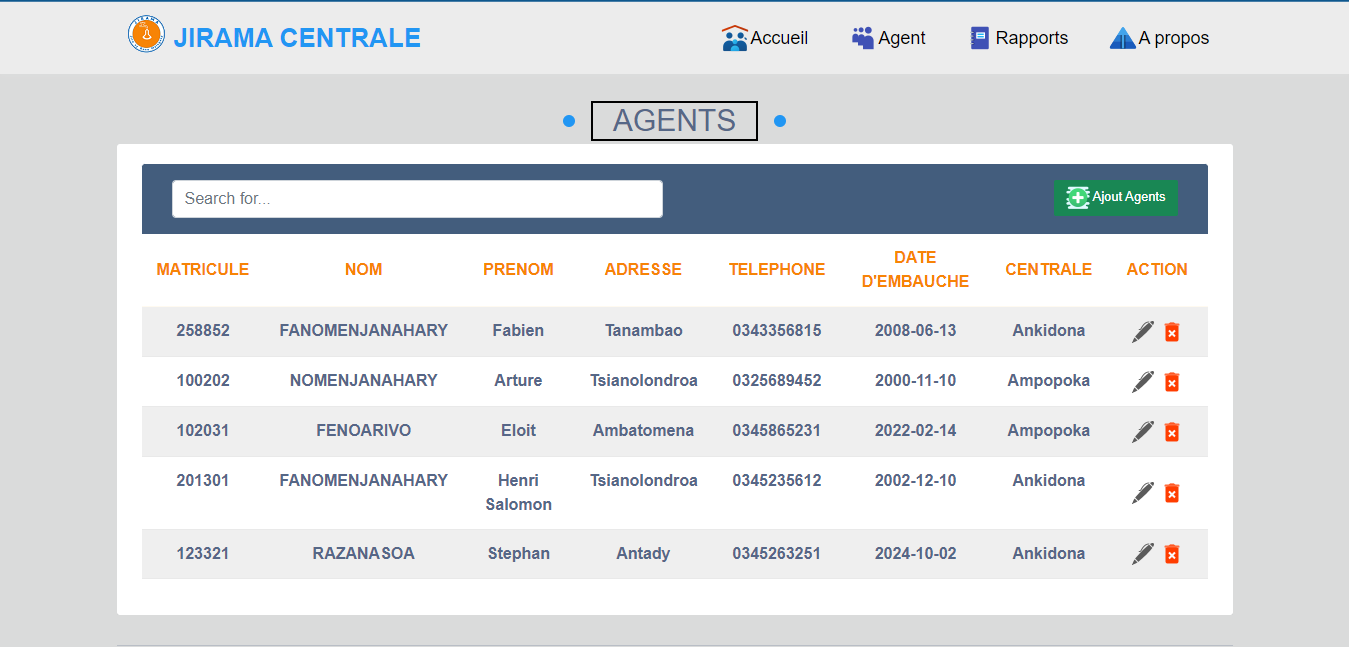
Voici les listes des agents qui inclut dans la base de donnée et l’insertion des agents dans les bases des données

Figure 9.2 : menu principale

* + 1. **Interface de choix de rapport**

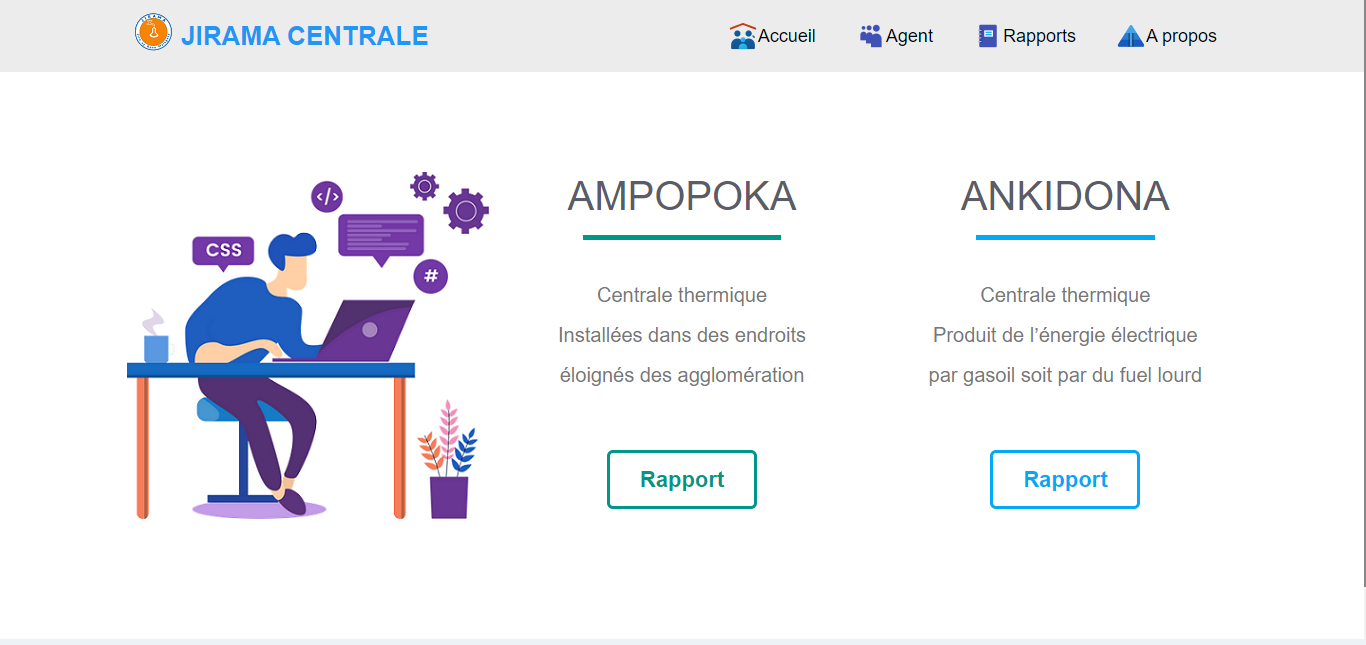
Voici les représentations des rapports qui remplis des agents dans le rapport de production électricité

Figure 9.3 : interface de choix de rapport qu’on remplir

* + 1. **Formulaire du rapport**

Dans la figure 9.4 ce formulaire il y a deux types dans l’exploitation électricité et dans la partie carburante

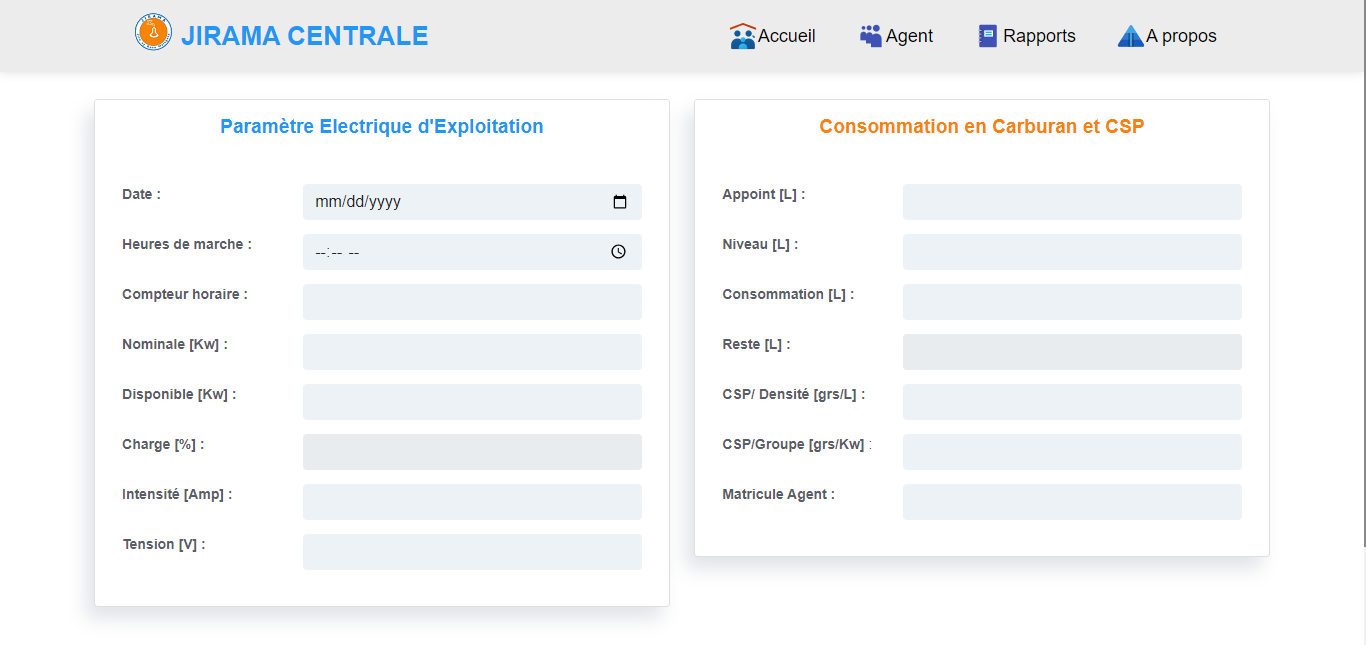
****

Figure 9.4 : formulaire du rapport de la production électricité

**Conclusion Générale et Perspective**

# **CONCLUSION**

En guise de conclusion, nous pouvons dire que cette conception pratique a consommé notre formation théorique, dans cet apprentissage que nous avons appris beaucoup de nouvelles expériences et ça pourrait nous accompagnées dans notre futur.

Ce projet a traité de l’analyse du système de gestion à la conception. En effet on a entamé l’étude structurelle et fonctionnelle utilisant une conception orientée objet. Après l’analyse on a tiré le modèle de domaine et exprimé les besoins du MERISE.

La création de cette application doit suivre des étapes bien précises avant d’être  
réalisée. L’expression des besoins se reporte à définir ce que l’on attend au système à mettre en place. Pour cela, le recueil de plus d’informations possibles auprès du demandeur est  
primordial, car ceci va nous permettre d’apprendre l’opportunité et la faisabilité du projet.

Le projet nous a permis en particulier d’être confrontés en pratique aux problématiques de modélisation qui est vraiment importante. Il nous a permis de s’adapter dans le monde professionnel. De plus, il nous a permis aussi d’approfondir les connaissances  
sur l’utilisation de langage JAVA, et de maitriser le Framework Spring.

Malgré cela, l’application que nous avons créée n’a pas atteint sa performance parfaite. Des améliorations doivent être envisagées afin d’augmenter cet exploit. Nous avons comme abords, de créer une application pour que les employés pour faire la demande d’absence.

Ce stage nous a été bénéfique soit au point de vue professionnelle que pédagogique. Elle nous a renforcé nos connaissances et aussi de découvrir de nouvel horizon dans la vie quotidienne.

# 

# **BIBLIOGRAPHIQUE**

1. [Richard Grin, 2000] « *Introduction aux bases de données* », Université de Nice Sophia Antipolis, version 2.1;
2. [Mr BAKARI, 2024] « *Cours PHP* » Support de cours de PHP, Troisième année en Développement d’Application d’Internet d’Intranet ;
3. [Dr Hasina, 2024] « *Cours UML* » Support de cours de UML, Troisième année en Développement d’Application d’Internet d’Intranet ;
4. [Ullman L., 2004,] « *MySQL Développement de site web* » ;
5. -[Bakari M., 2021]. *Cours de conduite de projet informatique. Université de Fianarantsoa*
6. [Md Hoby,2021-2022] « *Conception et réalisation d’une application pour la gestion d’assurance automobile au sein d’EMIT »,* Département Informatique, Centre Universitaire de Formation Professionnalisante / Université de Fianarantsoa EMIT;
7. [Sylvie V., 2010.] *Cycle de vie du logiciel et bonnes pratiques de développement* ;
8. [Sarra R.A, 2017] « *Conception et développement d'une application web de rapport journalier »*, Ecole supérieur des sciences économiques et commerciales Tunisie. Licence appliquée en informatique de gestion 2017 ;
9. [Rumbaugh J., Jacobson I. et Booch G., 1999]. *The Unified Modeling Language Reference Manual*, Addison- Wesley ;
10. [O’Reilly T., 2005,] *« Comment le web 2.0 améliorera les services aux citoyens ? »* ;
11. [Muller P.-A., 2008*.] Modélisation objet avec UML*, Eyrolles ;

# **WEBOGRAPHIE**

1. : [La nation de MVC,2024], disponible sur http//www.openstudiot.fr/lab/Comprender-la nation-MVC.html, consulté le 22 juillet 2024 ;
2. [Christophe C., 2017] *Apprendre Laravel de A à Z [en ligne] TUTORA*. Disponible sur : <http://www.tutora.fr/apprendre-JavaSwing > ;
3. > [hounwanou,1993] *Teacher du Net cours Laravel*. Disponible sur :< <http://www.youtube.com/hounwanou1993>>;
4. [Jean-François P., 2014,] *Ergonomie d’une application web [en ligne].* Comment Ça Marche. Disponible sur : < http://www.commentcamarche.net/contents/1255-ergonomied-une-application-web >;
5. [Jean-François P., 2010] *Introduction en PHP [en ligne].* Comment Ça Marche. Disponible sur : < http:// www.commentcamarche.net/contents/1351-php-introduction >;
6. [Jean-François P., 2017*] Base de données et systèmes de gestion de base de données* (ou SGBD) [en ligne]. Comment Ça Marche. Disponible sur : <http://www.commentcamarche.net/contents/104-bases-de-donnees-introduction >
7. : [laravel,2024], disponible sur <http//www.laravel.download>
8. : [Bootstrap], disponible sur http//fr.wikipedia.org/wiki/bootstrap\_(framework),
9. : [tableau comparatif de Framework], disponible sur, http/www.comparatifframework.com/tableau-comparatif,
10. : [Architecture logiciel], disponible sur http//fr.wikipedia.org/wiki/Architecture\_logicielle,
11. : [Architecture matérielle] disponible sur http/fr.wikipedia.org/wiki/Architecture\_materiel,

# **ANNEXE**

**Installation Laravel et WampServer**

1. **Installation Laravel**

L'installation de Laravel est assez simple, mais elle nécessite quelques prérequis. Voici un guide étape par étape pour installer Laravel sur votre machine locale :

* 1. **Prérequis :**
* PHP : Laravel nécessite PHP 8.0 ou une version ultérieure.
* Composer : Un gestionnaire de dépendances pour PHP.
* Serveur Web : Apache ou Nginx.
* Base de données : MySQL, SQLite, PostgreSQL, ou tout autre système de gestion de base de données pris en charge par Laravel.
  1. **Étapes pour installer Laravel :**

**1. Installer Composer**

Laravel utilise Composer pour gérer ses dépendances. Si vous n'avez pas encore installé Composer, vous pouvez le faire en suivant ces étapes :

* Téléchargez Composer via le site officiel [https://getcomposer.org/](https://getcomposer.org/).
* Suivez les instructions pour installer Composer globalement sur votre système.
* Vérifiez que Composer est bien installé en exécutant la commande « composer –version » dans votre terminal

**2.** **Installer Laravel via Composer**

Une fois que Composer est installé, vous pouvez installer Laravel en utilisant Composer.

* Ouvrez votre terminal et exécutez la commande suivante : « composer create-project --prefer-dist laravel/laravel nom\_du\_projet ».
* Remplacez `nom\_du\_projet` par le nom que vous voulez donner à votre application Laravel.

**3. Configurer le serveur local**

Laravel inclut son propre serveur web de développement, donc vous n'avez pas nécessairement besoin de configurer Apache ou Nginx pour développer localement. Utilisez la commande suivante pour lancer le serveur de développement :

« cd nom\_du\_projet »

« php artisan serve »

Cela va lancer un serveur web local, et vous pourrez accéder à votre application à l'URL suivante : `http://localhost:8000`.

**4. Configurer l’environnement**

Le fichier `.env` contient les configurations de votre application, comme les informations de la base de données, le cache, les files d'attente, etc.

* Modifiez le fichier ‘.env’ pour configurer votre base de données et autres paramètres

**5. Générer une clé d'application**

Une fois l’installation terminée, il est important de générer une clé d'application : « php artisan key:generate »

Cela générera une clé qui sera utilisée pour le chiffrement de l'application.

**6. Migrer la base de données**

Si vous avez configuré votre base de données dans le fichier `.env`, vous pouvez exécuter les migrations pour créer les tables par défaut de Laravel :

« php artisan migrate »

Une fois toutes ces étapes effectuées, vous devriez être en mesure d'accéder à votre application Laravel dans votre navigateur à l'adresse `http://localhost:8000`.

1. **Qu’est-ce que WampServer ?**

WampServer est un serveur composer des outils tel que : Apache, MySQL, PHP pour héberger des application web localement. WAMP est exclusivement disponible que pour Windows, on pourrait même penser que WAMP veut dire Windows, Apache, MySQL et PHP.

1. **Comment installer WampServer ?**

Pour commencer, il faut au préalable avoir pris soin de télécharger la dernière version stable de WAMP sur leur site officiel. Le Serveur est disponible pour plusieurs plateformes en 32 et 64 bits, donc veillez à choisir celle qui s’accord au système cible en suivant le lien <http://www.wampserver.>com/dowload.

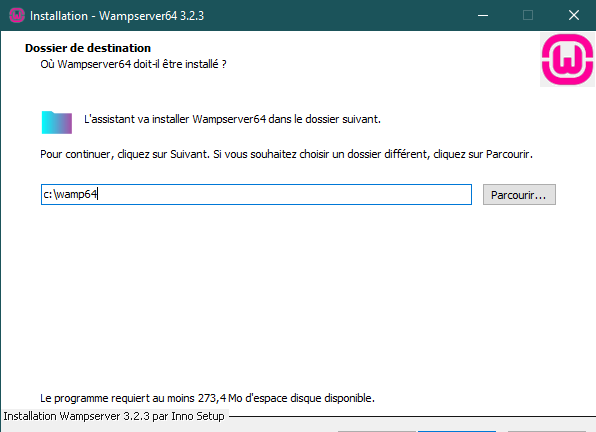


Figure 1 : installation de Wampserver

Une fois le fichier d’installation adéquat téléchargement, il suffit de le lancer et de suivre les directives. Il sera notamment demandé de choisir quels composant installer et ou le dossier de WAMP sera situé sur la machine.

1. **Comment Administrer une base de données dans WampServer ?**

WampServer est accompagné de plusieurs outil qui nous permet de facilement héberger notre site. PhpMyAdmin est l’un de ces outils. Il offre une interface intuitive pour mieux administrer notre base de données. Il est aussi facile à utiliser.

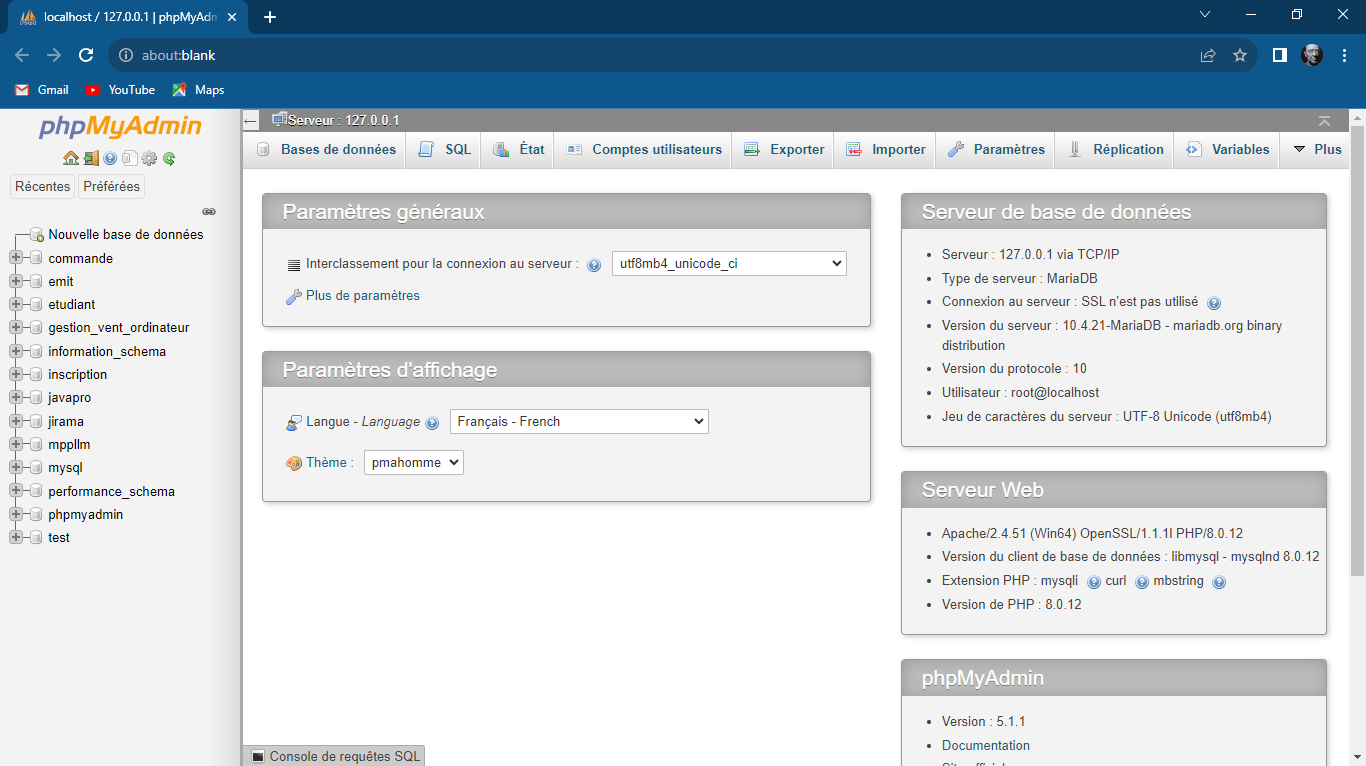


Figure 2: Interface de PhpMyAdmin

# **RESUME**

Nous avons accompli notre stage au sein du JIRAMA Tsianolondroa Fianarantsoa au service informatique dans lequel nous avons conçu et réalisé un projet de développement d’une application « Mise en place d’une application web de gestion d’un rapport de production électricité ». Durant ce stage, malgré les contraintes de temps, on a réussi à concevoir l’analyse et une partie de l’application et surtout on a eu des expériences professionnelles. Dans ce document se trouve tous les détails concernés, depuis l’étude de l’existant jusqu’à la réalisation de l’application, comme la présentation des outils utilisés et les modélisations du système. Nous avons comparé les outils utilisés par rapport à d’autres et étudiés l’architecture de l’application après avoir modélisé les besoins de l’utilisateur.  
L’application s’est aboutie avec une interface facile à manipuler et les fonctionnalités requis  
par la production que nous avons illustrée dans des captures d’écran.

Mot-Clé : Application Web, Gestion d’un rapport de production électricité, Laravel, UML.

# **ABSTRACT**

We accomplished our practicum with in JIRAMA Tsianolondroa Fianarantsoa to the service informatique which have a title " Mise en place d’une application web de gestion d’un rapport de production électricité ". During that workshop, in spite of the constraints of time, we succeeded in conceiving the analys is and a part of the application and especially we got professional experiences. In this document are all concerned details, since the survey of the existing until the realization of the application, as the presentation of the tools used and the modeling of the system. We compared the tools used in relation to other and studied the architecture of the application after having specify the user's needs. The application resulted to itself with an easy interface to manipulating and functionalities required by the operators that we illustrated in captures of screen.

Key-words : Application Web, Gestion d’un rapport de production électricité, UML, Laravel,