|  |  |  |
| --- | --- | --- |
| INSTITUT AFRICAIN D’INFORMATIQUE  AFRICAN INSTITUTE OF COMPUTER SCIENCES    Représentation du Cameroun  BP: 13719 Yaoundé (Cameroun)  Tél: (237) 22 21 11 43 / (237) 99 99 82 49  Site web: [www.iaicameroun.com](http://www.iaicameroun.com) |  | NOM ENTREPRISE (EN FRANÇAIS)  NOM ENTREPRISE (EN ANGLAIS)  Une image contenant texte, logo, Police, symbole  Description générée automatiquement  Adresse de l’entreprise  Ville  Tél:  Site web: |

RAPPORT DE FIN DE STAGE

**VOTRE THEME ICI**

Stage effectué du 1er juillet 2025 au 30 septembre 2025

En vue de l’obtention du **Diplôme de Technicien Supérieur (DTS) en Informatique option Génie Logiciel**

**Rédigé par :**

* **NOM & PRENOM**, étudiant au niveau II à l’Institut Africain d’Informatique

**Sous la supervision de :**

|  |  |  |
| --- | --- | --- |
| **ENCADRANT ACADEMIQUE** |  | **ENCADRANT PROFESSIONNEL** |
| **Nom de l’encadrant académique**  Niveau académique  Niveau professionnel | **&** | **Nom de l’encadrant professionnel**  Niveau académique  Niveau professionnel |

***Année académique 2024-2025***

# 

# DÉDICACE

À

La famille NDEME ASSEYI

# REMERCIEMENTS

Nous tenons à remercier toutes les personnes qui ont contribué de près ou de loin au bon déroulement de notre stage, à la réalisation de ce rapport jusqu’à la présentation de notre travail. Nos sincères remerciements s’adressent :

* Au représentant résident de l’Institut africain d’Informatique et représentation du Cameroun M. Armand Claude ABANDA ;
* À …
* À …
* À …
* À …
* À …
* Au comité des étudiants « IAI United » pour l’aide aux étudiants durant toute l’année ainsi qu’à la période allant du stage aux soutenances.

# SOMMAIRE

[DÉDICACE III](#_Toc202724684)

[REMERCIEMENTS IV](#_Toc202724685)

[SOMMAIRE V](#_Toc202724686)

[LISTE DES TABLEAUX VI](#_Toc202724687)

[LISTE DES FIGURES VII](#_Toc202724688)

[SIGLES ET ABBREVIATIONS VIII](#_Toc202724689)

[GLOSSAIRE IX](#_Toc202724690)

[RESUME X](#_Toc202724691)

[ABSTRACT XI](#_Toc202724692)

[INTRODUCTION GÉNÉRALE 1](#_Toc202724693)

[1ère PARTIE : PHASE D’INSERTION 2](#_Toc202724694)

[2ème PARTIE : PHASE TECHNIQUE 7](#_Toc202724695)

[CONCLUSION GÉNÉRALE 64](#_Toc202724696)

[BIBLIOGRAPHIE 1](#_Toc202724697)

[WEBOGRAPHIE 2](#_Toc202724698)

[ANNEXES 3](#_Toc202724699)

[TABLE DES MATIÈRES 4](#_Toc202724700)

# LISTE DES TABLEAUX

Générer votre liste des tableaux ici… (Références / Insérer une table des illustrations / Légende : Tableau / OK)

# LISTE DES FIGURES

[Figure 1 location\_scrum\_master 5](#_Toc203637710)

[Figure 2 organigramme\_scrum\_master 6](#_Toc203637711)

# SIGLES ET ABBREVIATIONS

* **UML:** Unified Modeling Language.
* **2TUP:** Two Tracks Unified Process.
* Entrez vos sigles et abréviations ici en forme de tirets…

# GLOSSAIRE

* **MERISE :** Méthode d’Etude et Réalisation Informatique pour les systèmes d’Entreprise.
* Entrez votre glossaire ici sous forme de tirets…

# RESUME

Insérez votre texte ici…

# ABSTRACT

Insérez votre texte (en anglais) ici…

# INTRODUCTION GÉNÉRALE

Insérez votre texte ici…

# 1ère PARTIE : PHASE D’INSERTION

Résumé :

|  |
| --- |
| Le dossier d’insertion est le tout premier document rédigé durant la période de stage académique. C’est la partie du stage qui présente la structure d’accueil, son fonctionnement, ainsi que l’accueil de l’étudiant en entreprise. Cette phase d’insertion dure généralement deux semaines et permet au stagiaire de se familiariser avec son nouvel environnement. |

Aperçu :

Introduction

1. Accueil et intégration en entreprise.
2. Présentation de l’entreprise.

Conclusion

**INTRODUCTION**

La phase d’insertion est une période (généralement de 02 semaines) réservée à l’étudiant pour découvrir et se familiariser avec son environnement de travail ou lieu de stage. Ici, il devra donc connaître son maître de stage communément appelé encadrant professionnel et apprendre à coexister avec ses collègues de stage. C’est également une période au cours de laquelle, l’entreprise attribue un thème au stagiaire afin qu’un travail puisse être mené par rapport à un problème constaté. Ainsi, cette phase s’est déroulée entre les murs de la société SCRUM MASTER, dont la présentation fera l’objet des pages qui suivent.

## ACCUEIL ET INTÉGRATION EN ENTREPRISE

#### Accueil en entreprise

Le 01er Juillet 2025 a été en effet la date marquant notre toute première entrée en tant que stagiaire au sein de la structure SCRUM MASTER située à carrière. Nous avons bénéficié d’un accueil chaleureux et remarquable de la part de M. KENE TILONG et de M. WANGBARA, tous les deux Ingénieurs Informaticiens. Ces derniers ont pris le temps et le soin de nous expliquer ce qu’est SCRUM MASTER, de nous présenter le but de la période d’imprégnation en entreprise, leurs attentes à l’égard des stagiaires pendant la période de stage et l’espace qui nous a été réservé pour effectuer notre stage.

#### Intégration en entreprise

Durant cette phase d’insertion de deux semaines, il a été question pour nous de nous adapter et de nous familiariser avec notre structure d’accueil, en côtoyant quotidiennement les départements de la structure devant intervenir de près ou de loin à la réalisation de notre projet. Cette période à SCRUM MASTER nous a permis de présenter la quintessence des projets que nous envisageons et de sélectionner le plus réaliste afin d’en faire notre thème de stage.

## PRÉSENTATION DE L’ENTREPRISE

#### Historique

SCRUM MASTER est une entreprise informatique pensée par deux ingénieurs des travaux en Génie Logiciel et Systèmes et Réseaux Informatiques. L’envie de fournir des solutions informatiques de qualité est le déclencheur de cette aventure constituée au départ de cinq (03) personnes en Janvier 2025. Par la suite, des difficultés se sont enchaînées et rien que deux (02) personnes ont cru au projet jusqu’à sa concrétisation en Février 2025 avec l’obtention du statut de SARL (Société Anonyme à Responsabilité Limité).

#### Situation géographique

Venant de l’hôpital des sœurs, la localisation de SCRUM MASTER est facilitée par le plan ci-dessous

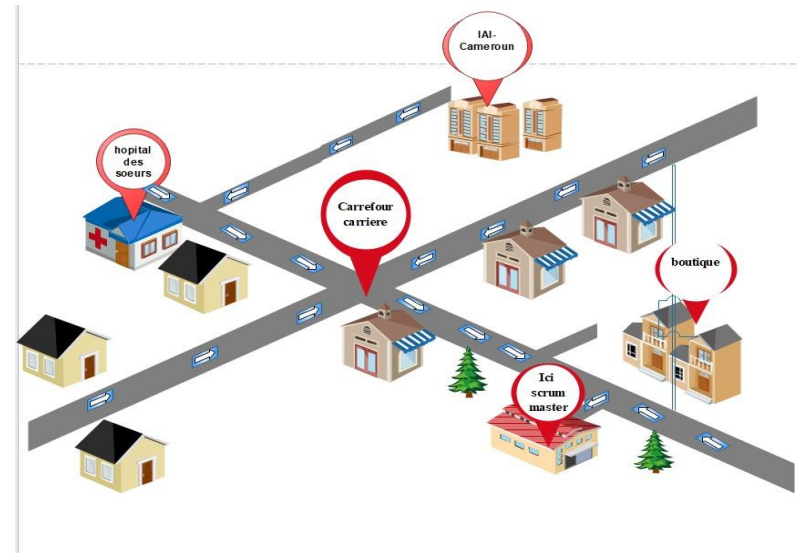


Figure 1 : plan de localisation de scrum master

#### Organigramme

SCRUM MASTER comporte en son sein deux (02) directions et plusieurs départements. En ce qui nous concerne, nous avons été affectés à la Direction Technique et plus précisément dans le département Génie logiciel, telle qu’illustré ci-dessous :

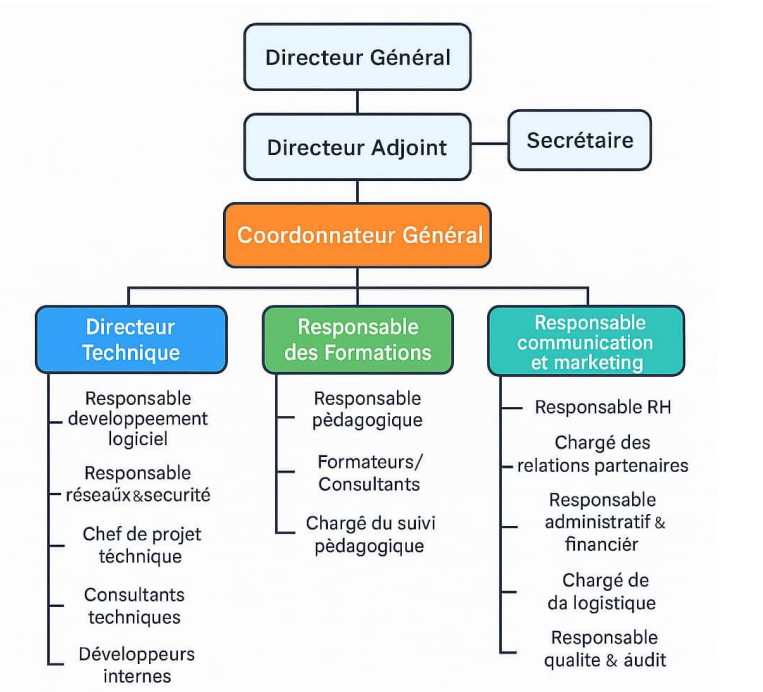


Figure  : organigramme de scrum master

#### Mission

SRUM MASTER est une société qui s’est donnée pour mission de résoudre au quotidien les problèmes que rencontrent les différentes entreprises liées à l’informatique ou quel que soit le domaine d’activité. SRUM MASTER, dans le but de vulgariser les TIC (Technologies de l’Information et de la Communication), délivre des attestations de fin de formation dans différents domaines de l’informatique.

#### Activités

La société a pour objet :

* Conception et réalisation des applications web, mobiles et de bureau ;
* Administration des réseaux informatiques ;
* Formation à la conception et développement logiciel ;
* Formation en administration réseaux ;
* Maintenance informatique ;
* Infographie ;
* Produit de services, commerce général.

Et, généralement, toutes opérations financières, commerciales, industrielles, mobilières et immobilières, pouvant se rattacher directement ou indirectement à l'objet ci-dessus ou à tous objets similaires ou connexes

#### Ressources matérielles et logicielles

##### Ressources matérielles

Insérer votre texte ici…

Insérez votre tableau ici…

##### Ressources logicielles

S’agissant des ressources logicielles, SCRUM MASTER en dispose de plusieurs lui permettant d’effectuer ses différentes tâches quotidiennes.

|  |  |  |
| --- | --- | --- |
| NOMS | TYPES | ROLES |
| Licence Windows 11 | Licence | Mettre à niveau un appareil Windows 10 Professionnel vers Windows 11 Professionnel |
| 300 giga sur Google Cloud | Stockage | Storage utilise des serveurs distants pour enregistrer des données telles que des fichiers, des données d'entreprise, des vidéos ou des images. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**CONCLUSION**

Rendu au terme de cette phase d’insertion qui portait non seulement sur la présentation générale de l’entreprise, mais aussi sur notre familiarisation avec l’environnement de travail ; il en découle que cette étape s’est déroulée dans un cadre convivial, ceci avec un esprit de confiance dû au professionnalisme et à l’expérience du maître de stage et de ses collaborateurs. En outre, cette phase nous a été bénéfique du fait de la perception de la nécessité et de l’importance du travail collaboratif, et surtout le plus important de se confronter aux réalités du milieu professionnel.

# 2ème PARTIE : PHASE TECHNIQUE

Résumé :

|  |
| --- |
| Il a été question pour nous dans la phase d’insertion de nous familiariser avec notre structure d’accueil. L’accent sera mis dans cette phase sur les différentes caractéristiques, spécificités et attentes du sujet soumis à notre étude. |

Aperçu :

Introduction

* Dossier 1 : L’Existant
* Dossier 2 : Le cahier des charges
* Dossier 3 : Dossier d’Analyse
* Dossier 4 : Dossier de Conception
* Dossier 5 : Dossier de Réalisation
* Dossier 6 : Test de fonctionnalités
* Dossier 7 : Guide d’installation et guide d’utilisation

Conclusion

## Introduction

Insérez votre texte ici… (Introduction de la 2e partie : PHASE TECHNIQUE).

## Dossier 1 : L’Existant

Résumé :

|  |
| --- |
| Le dossier de l'existant analyse en détail le système actuel, décrivant ses fonctionnalités, ses performances et ses limitations. Il identifie les besoins non satisfaits et les problèmes à résoudre, fournissant ainsi une base solide pour guider le développement de la nouvelle application. Ce dossier permet de comprendre les forces et faiblesses du système en place et d'établir des recommandations pour améliorer et optimiser la solution future. |

Aperçu :

Introduction

1. Présentation du thème/projet
2. Etude de l’existant
3. Critique de l’existant
4. Problématique
5. Proposition de solution

Conclusion

### Introduction

L’analyse de l’existant constitue une étape cruciale dans la conception de notre plateforme de

## PRÉSENTATION DU THÈME/PROJET

Notre solution SaaS est conçue pour révolutionner la formation et l'opérationnel des administrateurs réseaux en combinant trois modules complémentaires. Dans l'application, les utilisateurs pourront construire visuellement des topologies réseau via un éditeur drag-and-drop, générer automatiquement les configurations techniques grâce à notre moteur intelligent, et consolider leurs connaissances avec des parcours pédagogiques personnalisés. Elle permettra non seulement aux étudiants et professionnels de maîtriser les configurations complexes, mais aussi d'éviter les erreurs courantes grâce au système de validation en temps réel. La solution facilitera ainsi le travail des formateurs qui pourront suivre la progression individuelle et assigner des exercices adaptés au niveau de chaque apprenant.

L'administrateur académique quant à lui disposera d'un tableau de bord complet pour gérer les comptes utilisateurs, superviser les topologies créées, publier de nouveaux contenus pédagogiques et analyser les statistiques d'apprentissage. Les fonctionnalités clés incluent sont **Simulation réseau**: Construction de topologies avec équipements virtuels (routeurs, switchs, pare-feux), **Automatisation** : Génération de configs VLAN, OSPF, BGP et validation intelligente des erreurs, **Espace pédagogique** : Cours interactifs avec exercices intégrés à la sandbox, **Gestion administrative** : Suivi des étudiants, création de scénarios pratiques, reporting.

Cette plateforme unifiée élimine les allers-retours entre outils fragmentés et ressources théoriques déconnectées de la pratique, réduisant de 70% le temps nécessaire pour maîtriser les concepts réseaux tout en garantissant l'exactitude technique des configurations déployées.

## Etude de l’existant

#### Cas 1 : Étude de Cisco Packet Tracer

Cisco Packet Tracer constitue la solution de référence dans le domaine de la simulation réseau éducative. Développé par Cisco Systems, cet outil permet aux étudiants de concevoir, configurer et dépanner des réseaux virtuels. Les fonctionnalités principales incluent :

* Interface graphique pour construire des topologies réseau via glisser-déposer
* Bibliothèque d'équipements virtuels (routeurs, switchs, terminaux)
* Mode simulation pour visualiser le flux des paquets
* Environnement limité aux technologies Cisco

**Processus pédagogique observé** :

1. L'étudiant construit une topologie réseau
2. Configure manuellement chaque équipement via CLI
3. Teste la connectivité via des outils intégrés (ping, traceroute)
4. Valide le fonctionnement via le mode simulation

**Limites identifiées** :

* Absence de génération automatique de configurations
* Pas de détection d'erreurs en temps réel
* Contenu pédagogique non intégré à l'interface
* Compatibilité exclusive avec les équipements Cisco

#### Cas 2 : Étude des Plateformes d'Automatisation Professionnelle

Les solutions professionnelles comme Ansible, Terraform et Batfish offrent des capacités avancées d'automatisation réseau :

**Ansible (Red Hat)** :

* Automatisation des déploiements via playbooks YAML
* Bibliothèque de modules réseaux (ios\_config, junos\_config)
* Requiert des compétences en programmation
* Interface en ligne de commande uniquement

**Batfish (Open Source)** :

* Validation sémantique des configurations
* Détection des erreurs de sécurité et performances
* Analyse de l'impact des changements
* Non intégré à un environnement visuel

Les solutions professionnelles comme Ansible, Terraform et Batfish offrent des capacités avancées d'automatisation réseau :

**Ansible (Red Hat)** :

* Automatisation des déploiements via playbooks YAML
* Bibliothèque de modules réseaux (ios\_config, junos\_config)
* Requiert des compétences en programmation
* Interface en ligne de commande uniquement

**Batfish (Open Source)** :

* Validation sémantique des configurations
* Détection des erreurs de sécurité et performances
* Analyse de l'impact des changements
* Non intégré à un environnement visuel

## CRITIQUE DE L’EXISTANT

Par la présentation du fonctionnement actuel, nous avons relevé les limites présentées dans le tableau qui suit :

Tableau 1:critique de l'existant

|  |  |  |
| --- | --- | --- |
| CRITIQUES | LIMITES | SOLUTION |
| Absence de génération automatique de configurations. | Le système ne génère pas automatique des configurations et n’accompagne pas l’étudiant dans sa configuration. | Service de génération de configurations automatique et d’accompagnement dans la configuration de son réseau. |
| Pas de détection d'erreurs en temps réel. | Lors des configurations, le système ne détecte pas les erreurs en temps réel. | Service de vérification instantané des configurations apportées à un équipement du réseau. |
| Courbe d'apprentissage abrupte pour les débutants. | La difficulté à maîtriser les systèmes de déploiement de réseaux et leurs configurations. | Offre une expérience utilisateur intuitive et un environnement favorable visant à la simplification et l’hesence dans l’étude du fonctionnement du système. |
| Absence de parcours d'apprentissage guidé. | Absence d’espace d’apprentissage pour étudiant dans des sujets clés. | Service de micro-e-learning accompagnant les étudiants dans le parcours scolaire. |

## PROBLÉMATIQUE

Pour donner suite à l’existant présenté ci haut ainsi qu’aux différentes limites exposées, on peut se poser la question de savoir comment faciliter le processus d’apprentissage et de configuration des réseaux informatique par des ingénieurs réseaux ?

## PROPOSITION DE SOLUTION

Face aux différentes limites rencontrées, il sera question pour nous de mettre sur pieds une plateforme d’accompagnement des techniciens réseaux novices dans l’apprentissage et la configuration des leurs architectures réseaux en leurs offrant un espace de montage de topologie physique, un espace de micro-e-learning, et enfin un espace de déploiement sur site.

**CONCLUSION**

Au terme de cette étape, il a été question de faire une descente sur le terrain dans le but de recueillir le maximum d’informations possibles en rapport avec notre système d’étude. Par la suite, il a été question d’en faire une critique afin de ressortir les limites, les conséquences engendrées et enfin de proposer des solutions qui permettront de résoudre efficacement le problème général.

## Dossier 2 : Cahier des charges

Résumé :

|  |
| --- |
| Le cahier des charges est un document contractuel établi entre le maitre d’œuvre et le maitre d’ouvrage qui étale les besoins du client. Il étudie et présente avec exactitude les exigences formulées par les utilisateurs en ce qui concerne le projet, son déroulement et les résultats. |

Aperçu :

Introduction

1. Contexte et justification de l’étude /du projet
2. Les objectifs de l’étude/projet
3. Expressions des besoins de l’utilisateur
4. Planification du projet/ étude / Gant project
5. Estimation du coût du projet/étude
6. Les contraintes du projet/ étude
7. Les livrables

Conclusion

### Introduction

Insérer votre texte ici…

### Contexte et justification de l’étude /du projet

#### Contexte

Insérer votre texte ici…

#### J**ustification**

Insérer votre texte ici…

### Bénéficiaire et cible

#### Bénéficiaire

Insérer votre texte ici…

#### Cible

Insérer votre texte ici…

### Les objectifs de l’étude/projet

#### Objectif général

Insérer votre texte ici…

#### Objectifs spécifiques

Insérer votre texte ici…

### Expressions des besoins de l’utilisateur

#### Besoins fonctionnels

Insérer votre texte ici…

#### Besoins non fonctionnels

Insérer votre texte ici…

### Planification du projet/ étude / Gantt Project

#### Intervenants

Insérer votre texte ici…

#### Diagramme de Gantt

Insérer votre texte ici…

Insérez votre diagramme de Gant ici…

### Estimation du coût du projet/étude

#### Ressources Matérielles

|  |  |  |  |
| --- | --- | --- | --- |
| Ressource | Quantité | Cout unitaire(FCFA) | Cout total(FCFA) |
| Ressource 1 | Quantité 1 | Cout unitaire 1 | Cout total 1 |
| Ressource 2 | Quantité 2 | Cout unitaire 2 | Cout total 2 |
| Ressource 3 | Quantité 3 | Cout unitaire 3 | Cout total 3 |
| Ressource 4 | Quantité 4 | Cout unitaire 4 | Cout total 4 |

#### Ressources logicielles

|  |  |  |
| --- | --- | --- |
| Ressource | Logo/Description | Cout unitaire(FCFA) |
| Ressource 1 | Logo/Description1 | Cout unitaire 1 |
| Ressource 2 | Logo/Description1 | Cout unitaire 2 |
| Ressource 3 | Logo/Description3 | Cout unitaire 3 |
| Ressource 4 | Logo/Description4 | Cout unitaire 4 |

#### Ressources humaines

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fonction | Quantité | Durée  (jour/semaine) | Cout(jour/semaine) | Cout total(FCFA) |
| fonction 1 | Quantité 1 | Durée 1 | Cout 1 | Cout total 1 |
| fonction 2 | Quantité 2 | Durée 2 | Cout 2 | Cout total 2 |
| fonction 3 | Quantité 3 | Durée 3 | Cout 3 | Cout total 3 |
| fonction 4 | Quantité 4 | Durée 4 | Cout 4 | Cout total 4 |

#### Cout total

|  |  |
| --- | --- |
| Ressources | Total ( FCFA) |
| Ressources matérielles | Cout total |
| Ressources logiciels | Cout total |
| Ressources humaines | Cout total |
| Total provisoire | Cout total |
| Imprévus (10 ou 20%) | Cout total |
| Total | Cout total |

#### Les contraintes du projet/ étude

Insérer votre texte ici…

#### Les livrables

Insérer votre texte ici…

### Conclusion

Insérer votre texte ici…

## Dossier 3 : Dossier d’Analyse

Résumé :

|  |
| --- |
| Le dossier d’analyse a pour but d’affiner le choix de la méthode à utiliser tout au long du projet. Il permet dans la même lancée de faire comme son nom l’indique une analyse détaillée du projet à mettre en œuvre. |

Aperçu :

Introduction

1. Méthodologie
   1. Etude comparative UML et MERISE
   2. Etude comparative des processus unifiés
2. Modélisation
   1. Diagramme des cas d’utilisation
   2. Descriptions textuelles
   3. Diagramme de communication
   4. Diagramme de séquence
   5. Diagramme d’activité

Conclusion

### Introduction

Insérer votre texte ici…

### Méthode et langage d’analyse

L’analyse est une étape fondamentale dans la conception d’un logiciel. C’est la base de tous travaux de réalisation de système d’information. Un système d’information est un système organisé de ressources, de personnes, et de structures qui évoluent dans une organisation et dont le comportement coordonné vise à atteindre un but commun. Plusieurs méthodes de langages ont été développées pour faciliter et normaliser l’analyse et la conception des systèmes d’information parmi lesquels nous avons principalement UML et MERISE.

#### Etude comparative UML et MERISE

MERISE (Méthode d’Étude et de Réalisation Informatique pour les Systèmes d’Entreprise) est une méthode d’analyse, de conception structurelle et de réalisation des systèmes d’informations très utiles notamment dans les entreprises françaises. Elle est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. Son but principal est d’arriver à concevoir un système d’information (SI). C’est une méthode systémique d’analyse et de conception de SI qui propose de considérer le système réel selon deux points de vue : une vue statique (données) et une vue dynamique (traitement).

UML (Unifie Modelling Language) quant à lui, est un langage de modélisation des systèmes standards, qui utilise des diagrammes pour présenter chaque aspect d’un système en s’appuyant sur la notion d’orienté objet qui est un véritable atout pour ce langage. UML propose donc une approche différente de celle de MERISE en ce sens qu’il associe les données aux traitements. En effet, avec UML, centraliser les données d’un type de traitement associé permet de limiter les points de maintenance dans le

code et faciliter l’accès à l’information en cas d’évolution de logiciel. De plus, UML décrit la dynamique du système d’information comme un ensemble d’opérations attachées aux objets du système.

|  |  |
| --- | --- |
| MERISE | UML |
| Méthode d’étude et de réalisation informatique pour les systèmes d’entreprises. | Unified Modeling Language |
| Est une méthode systémique d’analyse et de conception de systèmes d’information. C’est-à-dire qu’elle utilise une approche systémique. | N’est cependant pas une méthode, mais plutôt un langage de modélisation objet qu’il faut associer une démarche pour en faire une méthode. C’est le cas de la méthode 2TUP ; RUP et XP. |
| Propose de considérer le système réel selon deux points de vue :   * Une vue statique (données). * Une vue dynamique (traitements).   C’est-à-dire qu’avec la méthode Merise, nous avons une étude séparée des données et des traitements. | Propose une approche différente de celle de Merise en ce sens qu’il associe les données et les traitements.  Car avec UML, centraliser les données d’un type et les traitements associés permet de limiter les points de maintenance dans le code et facilite l’accès a l’information en cas d’évolution du Logiciel. De plus ; UML décrit la dynamique du système d’information comme un ensemble d’opérations attachées aux objets du système. |

#### Choix et présentation de la démarche d’analyse

Après cette étude comparative entre les méthodes UML et MERISE, notre choix sera porté sur la méthode UML.

UML se définit comme étant un langage de modélisation graphique et textuel destiné à l’architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leurs structures aussi bien que leurs comportements, concevoir des solutions et communiquer des points de vue. L'UML utilise les points forts de ces trois approches pour présenter une méthodologie plus cohérente et plus facile à utiliser. Il représente les meilleures pratiques de création et de documentation des différents aspects de la modélisation des systèmes logiciels et d'entreprise. UML unifie à la fois les notions et les concepts orientés objets. Il ne s’agit pas d’une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d’un langage. C’est un langage standard de modélisation des systèmes d’informations à objet.

La version d’UML utilisée ici est la version 2.5.1 qui compte quatorze (14) diagrammes, et une division du système en deux (02) grandes vues :

* La vue statique composée des diagrammes UML structurels qui représente le système physiquement et comporte six (07) diagrammes
* Diagramme de classe
* Diagramme de composants
* Diagramme de structures composite
* Diagramme de déploiement
* Diagramme d’objets
* Diagramme de packages
* Diagramme de profil
* La vue dynamique constituée des diagrammes UML comportementaux qui représente les interactions effectuées dans le système, elle comporte sept (07) diagrammes :
* Diagramme d'activité
* Diagramme de communication
* Diagramme global d’interaction
* Diagramme de séquence
* Diagramme d'états-transition
* Diagramme de temps
* Diagramme de cas d’utilisation

UML étant un langage formel et normalisé, il est un bon support de communication dans l’élaboration d’une solution informatique, sa polyvalence et sa souplesse fond de lui un langage universel. Pour développer des solutions, on a besoin de lui associer une méthode générique qui s’attache à ses diagrammes, par exemple la méthode Agile. Il en existe plusieurs: Scrum, XP (Extreme Programming), RUP (Rational Unified Process), FDD (Feature Driven Development), etc…

#### Etude comparative des processus unifiés

La maîtrise des processus de développement implique une organisation et un suivi des activités : c’est ce à quoi s’attachent les différentes méthodes qui s’appuient sur l’utilisation du langage UML pour modéliser un système d’information. **UP** (**Unified Process**) est une méthode générique de développement de logiciel. Générique signifie qu'il est nécessaire d'adapter UP au contexte du projet, de l'équipe, du domaine et/ou de l'organisation (exemple : **R.U.P, U.P, X.P, 2 TUP**…).

Parmi les PU les plus populaires, on trouve :

* **Rational Unified Process (RUP)** : Développé par Rational Software, RUP est l'un des PU les plus complets et les plus structurés. Il est basé sur une approche en cascade avec des phases distinctes et des livrables bien définis.
* **Unified Process (UP)** : UP est une version plus légère et plus adaptable de RUP, créée par l'Object Management Group (OMG). Il offre plus de flexibilité aux équipes pour adapter le processus à leurs besoins spécifiques.
* **Extrême Programming (XP)** : XP est une méthodologie agile qui met l'accent sur la collaboration, les tests fréquents et les livraisons incrémentielles. Il est particulièrement adapté aux projets avec des exigences changeantes.
* **Scrum** : Scrum est une autre méthodologie agile qui utilise des cycles de développement courts appelés sprints. Il met l'accent sur la communication et la visibilité entre les membres de l'équipe.
* **Choisir le processus unifié adéquat**

Le choix du PU le plus adapté à un projet dépend de plusieurs facteurs, tels que la taille et la complexité du projet, les compétences de l'équipe, les exigences du client et la culture de l'entreprise.

* **RUP** est un bon choix pour les projets de grande envergure et critiques où une structure et une planification détaillées sont importantes.
* **UP** est un bon choix pour les projets qui nécessitent plus de flexibilité et d'adaptation.
* **XP** est un bon choix pour les projets avec des exigences changeantes et une équipe expérimentée.
* **Scrum** est un bon choix pour les projets qui nécessitent une livraison rapide et une communication efficace entre les membres de l'équipe.

#### Choix et présentation du processus unifié

Pour des raisons d'adaptation du processus unifié (UP), nous optons pour le processus **2TUP** **(Two Tracks Unified Process)** qui est construit sur le langage UML. Le processus unifié répète un certain nombre de fois une série de cycles. Tout cycle se conclut par la livraison d’une version du produit aux clients et s’articule en 4 phases :

**Création, élaboration, construction et transition**, chacune d’entre elles se subdivisant à son tour en itérations.

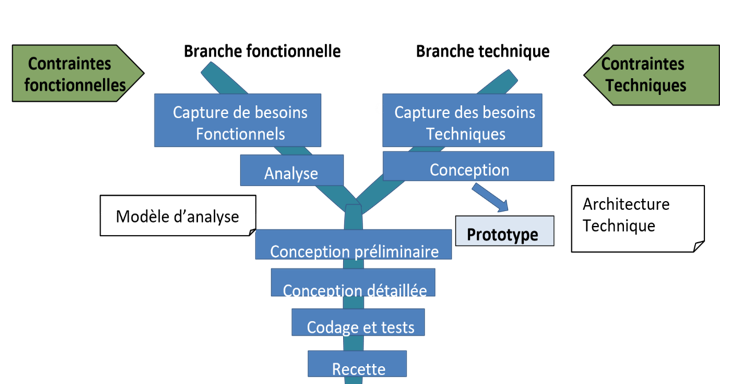
Dans le processus **2TUP**, les activités de développement sont organisées suivant 5 workflows qui décrivent :

* + - * La capture des besoins ;
      * L’analyse ;
      * La conception ;
      * L’implémentation ;
      * Et le test.

Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, par la suite de produire le cahier des charges et à modéliser le contexte.

Ce processus vise à améliorer l'efficacité du développement en dissociant les aspects techniques des aspects fonctionnels du système. Il s'articule autour de 3 phases essentielles à savoir : **une branche technique, une branche fonctionnelle et une phase de réalisation.**

Le schéma ci-dessous est l’illustration du processus **2TUP :**



### Modélisation

#### Diagramme des cas d’utilisation

Les diagrammes de cas d’utilisation identifient les fonctionnalités fournies par le système, ces fonctionnalités sont appelées ici cas d’utilisations, les utilisateurs qui interagissent avec le système (acteurs), et les interactions entre ces derniers. C’est une description de l’ensemble des opérations que l’utilisateur pourra effectuer dans le système.

Les cas d’utilisations sont représentés par une ellipse contenant le nom du cas d’utilisation. Un acteur et un cas d’utilisation sont mis en relation par une association représentée par **une ligne**.

|  |  |
| --- | --- |
| Composant du diagramme | Représentation Graphique |
| Acteur | D:\rapport\acteur.png |
| Cas d’utilisation | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Association | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |

Trois types de relations sont pris en charge par la norme UML. Nous pouvons les distinguer dans le tableau ci-dessous :

|  |  |  |
| --- | --- | --- |
| Type de relation | Description | Représentation graphique |
| Inclusion | Un cas 1 est un inclus dans un cas 2 si l'exécution du cas 1 passe obligatoirement par l’exécution du cas 2. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Extension | On dit qu’un cas 1 étend un autre cas 2 si la réalisation du cas 1 ajoute un ou plusieurs comportements à la réalisation du cas 2. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Généralisation  (Héritage) | Un cas 1 hérite d’un cas 2 si le cas 1 en plus de réaliser complètement les actions du cas 2 réalise ou non d’autres nouvelles actions. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |

Exemple de diagramme de cas d’utilisation :

Une image contenant diagramme, texte, cercle, ligne

Description générée automatiquement

##### Diagramme de cas d’utilisation globale

Votre diagramme de cas d’utilisation globale ici…

##### Diagrammes de cas d’utilisation spécifique

Vos diagrammes de cas spécifique ici…

#### Descriptions textuelles

Chaque cas d'utilisation, est associé à une série d'actions représentant la fonctionnalité voulue, ainsi que les stratégies à utiliser dans l'alternative où la validation échoue, ou des erreurs se produisent.

Le formalisme d’une description textuelle est le suivant :

|  |  |
| --- | --- |
| **Le titre** | |
| **Résumé** | Brève description de l'objectif du cas d'utilisation. |
| **Acteurs** | Entités (humaines ou systèmes) qui interagissent avec le système. |
| **Date de création** | Date à laquelle le cas d'utilisation a été rédigé. |
| **Responsable** | Personne en charge du cas d'utilisation. |
| **Version** | Numéro de version de l’application. |
| **Précondition** | Conditions à remplir avant le déclenchement du cas d'utilisation. |
| **Scénario nominal** | Séquence principale des étapes d'exécution. |
| **Scénarios alternatifs** | Variantes ou chemins alternatifs du scénario principal. |
| **Scénario d’exception** | Cas où une anomalie ou une erreur survient. |
| **Post condition de succès** | Résultat attendu si tout se passe bien. |
| **Post condition d’échec** | Conséquence en cas d'échec. |
| **Exigences non fonctionnelles** | Contraintes techniques (performance, sécurité, etc.) |

##### Cas 1

Votre tableau de description textuelle ici…

##### Cas 2

Votre tableau de description textuelle ici…

##### Cas 3

Votre tableau de description textuelle ici…

##### Cas 4

Votre tableau de description textuelle ici…

#### Diagramme de communication

Un diagramme de communication dans le langage de modélisation unifié fait référence à un graphique qui représente le flux de messages dans un système. En un mot, il montre comment les parties d’un système interagissent ou, dans ce cas, communiquent entre elles.

Comme tout autre diagramme, le diagramme de communication UML comporte également plusieurs composants qui constituent son intégralité. La bonne chose à propos de ce schéma est qu’il peut être réalisé avec seulement quelques composants.

Le formalisme est le suivant :

|  |  |  |
| --- | --- | --- |
| Composant | Description | Représentation |
| Acteur | C’est une entité qui interagit avec le système ou qui est extérieur à lui. Sa représentation est identique à celui d’un diagramme de cas d’utilisation. | D:\rapport\acteur.png |
| Objet | Il est représenté par un rectangle contenant le nom et le type de l'instance. | |  | | --- | | Objet | |
| Les connecteurs | Les relations entre les lignes de vie sont appelées connecteurs et se représentent par un trait plein reliant deux lignes de vie. |  |
| Les messages | Désignent une communication particulière entre les lignes de vie, et sont généralement ordonnés selon un ordre de numéro croissant. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |

##### 3.1. Cas 1

Votre diagramme ici…

##### 3.2. Cas 2

Votre diagramme ici…

##### 3.3. Cas 3

Votre diagramme ici…

##### 3.4. Cas 4

Votre diagramme ici…

#### Diagramme de séquence

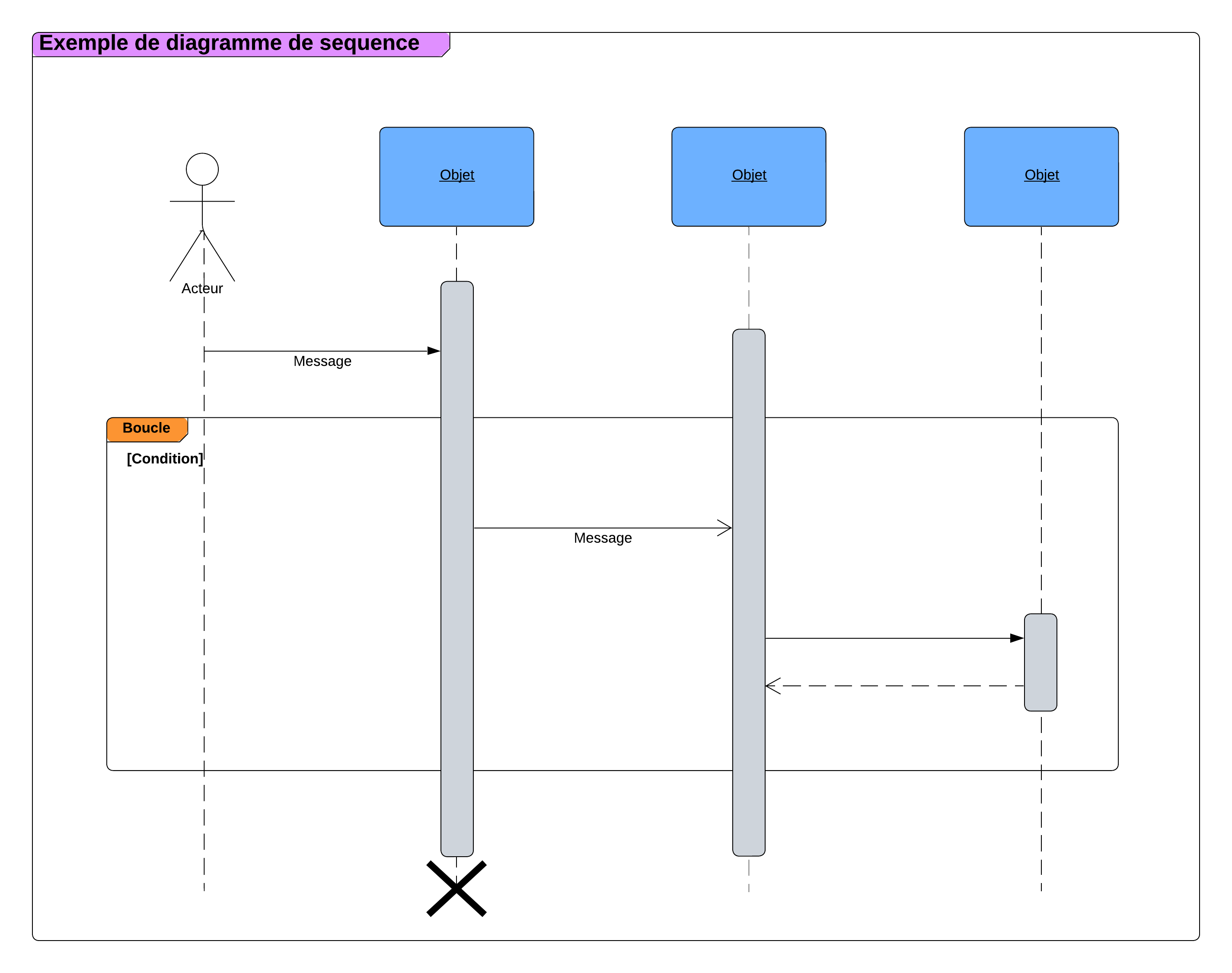
Les diagrammes de séquences sont des représentations graphiques des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. Le diagramme de séquences permet de décrire les interactions entre les objets au sein d’un diagramme des cas d’utilisation.

Le diagramme de séquence énumère les objets horizontalement et le temps verticalement. Il modélise l’exécution des différents modèles en fonction du temps. Dans ce diagramme, les objets et les acteurs sont énumérés en colonnes avec leur ligne de vie verticale indiquant la durée de vie de l’objet.

Le formalisme est le suivant :

|  |  |  |  |
| --- | --- | --- | --- |
| Composant Description Représentation | | | |
| Objet | | Se décrit comme étant l’instance d’une classe | |  | | --- | | Objet | |
|
| Acteur | | Il s’agit d’une personne qui interagit et communique avec le système. | D:\rapport\acteur.png |
| Ligne de vie | | La ligne de vie, représente le déroulement  temporel d'un processus. |  |
| Activation | | Il s’agit ici du temps nécessaire pour qu’un objet ou un acteur accomplisse une tâche, elle indique quand l’objet effectue une action. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Message synchrone | On utilise ce symbole lorsqu’un expéditeur doit attendre une réponse à un message avant de continuer. Le diagramme doit montrer à la fois l’appel et la réponse | | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Messages de retour asynchrones | Les messages asynchrones ne nécessitent pas de réponse avant que l’expéditeur ne continue. Seul l’appel doit être inclus dans le diagramme. | | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Message de retour | ces messages sont des réponses aux appels. | | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |

**Exemple de diagramme de séquence :**



##### Cas 1

Votre diagramme ici…

##### Cas 2

Votre diagramme ici…

##### Cas 3

Votre diagramme ici…

##### Cas 4

Votre diagramme ici…

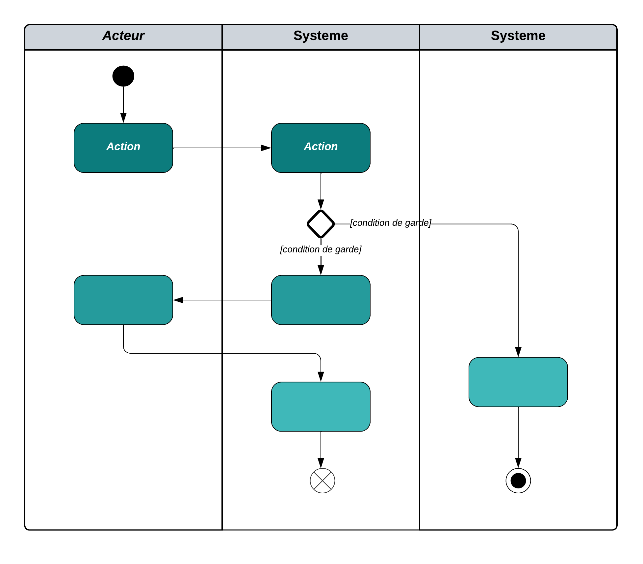
#### Diagramme d’activité

Les diagrammes d'activités sont particulièrement adaptés à la description des cas d'utilisation. Plus précisément, ils viennent illustrer et consolider la description textuelle des cas d'utilisation.

L’usage général du diagramme d'activité permet de mettre l'accent sur les traitements et de formaliser graphiquement la séquence d’actions réalisées dans un cas d'utilisation.

|  |  |  |
| --- | --- | --- |
| Composant Description | | Représentation |
| Une action | Rectangle aux coins arrondis qui représente une opération élémentaire et instantanée | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Une transition | Ligne qui représente le passage d’une activité à une autre | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Nœud initial | Cercle plein à partir duquel un flot débute dans un diagramme d’activité |  |
| Nœud final | **Un nœud de fin d'activité** : un petit cercle plein marquant l’arrêt l’exécution de l'activité  enveloppante s'arrête |  |
| **Un nœud de fin de flot :** cercle vide barré d'une croix marquant l’arrêt d’un flot à  l'intérieur du diagramme |  |
| Couloir | Les diagrammes d’activités font intervenir les acteurs de chaque activité. Chaque activité sera placée dans une colonne (couloir) qui correspond à l’acteur qui l’effectue. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Barre de synchronisation | Barre horizontale sur laquelle convergent les activités qui sont exécutées en parallèle. |  |

Exemple de diagramme d'activité :



##### Cas 1

Votre diagramme ici…

##### Cas 2

Votre diagramme ici…

##### Cas 3

Votre diagramme ici…

##### Cas 4

Votre diagramme ici…

### Conclusion

Insérez votre texte ici…

## Dossier 4 : Dossier de Conception

Résumé :

|  |
| --- |
| Le dossier de conception permet de modéliser dans son ensemble la solution proposée et de recueillir les informations nécessaires à la mise sur pied d’une base de données complexes et efficaces. De ce fait il prévoit le système futur. |

Aperçu :

Introduction

1. Diagramme de classe
2. Diagramme d’état transition
3. Diagramme de paquetage

Conclusion

### Introduction

Insérer votre texte ici…

### Diagramme de classe

Le diagramme de classe met en évidence d’éventuelles relations entre ces classes. Le diagramme de classes comporte 6 concepts : **classe, attribut, identifiant, relation, opération, généralisation/spécialisation**. Ce diagramme est constitué des éléments suivants :

* **Classe :**il s’agit de la description abstraite à un ensemble d’objet : elle définit leurs structures, leurs comportements et leurs relations.
* **L’attribut :** Il représente la modélisation d’une information élémentaire représentée par son nom et format.

**UML** définit 3(trois) niveaux de visibilité pour les attributs :

* + **Public (+) :** l’élément est visible par tous les clients de la classe.
  + **Protégé (#) :** l’élément est visible que par les sous-classes ou les classes filles.
  + **Privé (-) :** l’élément n’est visible que par les objets de la classe auxquels il appartient.
* **L’identifiant :** c’est un attribut particulier, qui permet de repérer de façon unique chaque objet, instance de la classe.
* **Multiplicité :** elle définit le nombre d’instance de la classe, elle est définie par un nombre entier ou un intervalle de valeur (1...1, 0...1, 1...\*, 0...\*).
* **Associations :** Une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances.

Il existe plusieurs types de relations dont les plus connues sont les suivants :

* + **Généralisation : :** Relation d'héritage, dans laquelle les objets de l'élément spécialisé (Classe enfant) peuvent remplacer les objets de l'élément général

(Classe parent)

* + **Agrégation :** utilisé pour les relations de types « contenant / contenue »
  + **La composition :** La composition est un cas particulier de l’agrégation dans laquelle la vie des composants est liée à celle des agrégats.

Le tableau suivant nous donne une liste assez détaillée du formalisme du diagramme de classe en un tableau :

|  |  |
| --- | --- |
| Elément | Représentation |
| Classe | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Association | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Composition | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Agrégation | 0.\*1.\* |
| Généralisation | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Multiplicité | 0.. \*, 0...1, 1...1, 1...\* |
| Modificateur d’accès | +, -, #, ~, / |

Exemple de diagramme de classe :

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

Le diagramme de classe de notre projet est le suivant :

Insérez votre diagramme de classe ici…

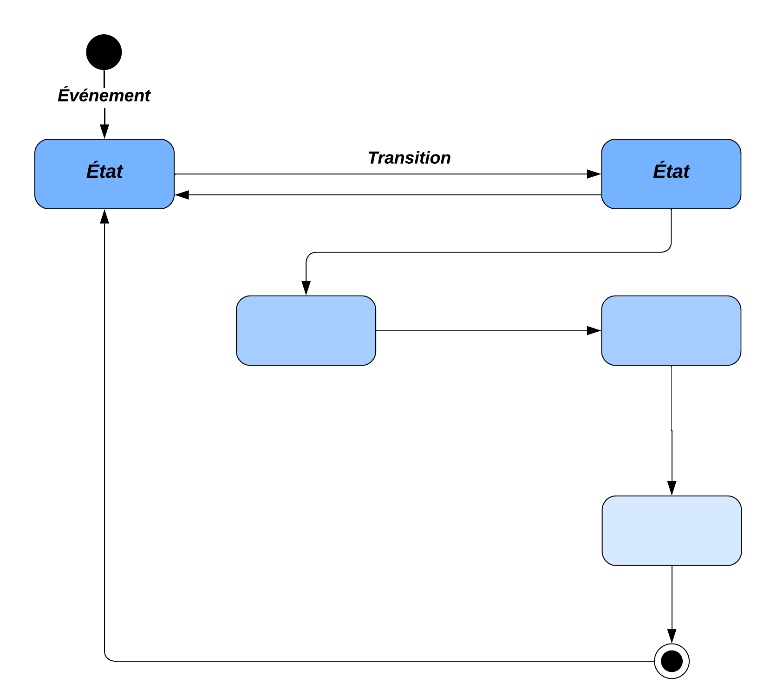
### Diagramme d’état transition

Le diagramme d'états-transition est comme son nom l’indique décrit les transitions entre les états et les actions que le système ou ses parties réalisent en réponse à un événement. Les diagrammes d'états-transitions permettent justement de spécifier les réactions d'une partie du système à des événements discrets. Un événement se produit à un instant précis et est dépourvu de durée.

Le diagramme d'états-transition décrit l'état ou le comportement d’un objet à un événement précis du système. A cet effet il est composé des éléments suivants :

|  |  |  |
| --- | --- | --- |
| Elément | Description | Représentation |
| Etat initial | Il s'agit de l'état dans lequel il se trouve lors de sa création. Est représenté par un point noir. |  |
| Etat final | Correspond à une étape où l'objet n'est plus nécessaire dans le système et où il est détruit. Il se représente par un point entouré d’un cercle. | Une image contenant cercle  Description générée automatiquement |
| Etat intermédiaire | Est une situation stable qui possède une certaine durée pendant laquelle un objet exécute une activité ou attend un  Événement. Il est représenté par un rectangle arrondi contenant son nom. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Transition | Une transition définit la réponse d'un objet à l'arrivée d'un événement. Elle est représentée par une flèche. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |

Exemple de diagramme d’état transition :



Quelques diagrammes d’état-transition de notre projet ci-dessous :

Insérez vos diagrammes ici…

### Diagramme de paquetage

Les diagrammes de package (ou diagramme de paquetages) permettent de décomposer le système en catégories ou parties plus facilement observable s’appelés « packages ». L'utilisation la plus courante pour les diagrammes de paquetages est d'organiser des diagrammes de Cas d'Utilisation et des diagrammes de Classes. Bien que l'utilisation des Diagrammes de Paquetages ne se limite pas à ces éléments UML.

Le formalisme est le suivant :

|  |  |  |
| --- | --- | --- |
| Composant | Description | Représentation |
| Package | Regroupement des cas d’utilisation en catégorie | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Merge | Représente la fusion de deux packages. Le contenu du package cible est intégré au package source. |  |
| Access | Indique que les éléments d'un package ont accès aux éléments publics d’un autre package. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Import | Permet à un package d’importer les éléments publics d’un autre package. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |

Le diagramme paquetage de notre projet est le suivant :

Insérez vos diagrammes ici…

### Conclusion

Insérez votre texte ici…

## Dossier 5 : Dossier de Réalisation :

Résumé :

|  |
| --- |
| La réalisation consiste à la mise en œuvre de projets dans un langage de programmation conformément aux spécifications définies dans le dossier précédent. Elle renferme en son sein les diagrammes de composant et de déploiement. A la sortie de cette partie, il sera produit une documentation de programmation expliquant l’architecture de la base de données et l’architecture de notre code. |

Aperçu :

Introduction

1. Choix des technologies
2. Architecture physique et logique du système
3. Diagramme de déploiement
4. Diagramme de composant

Conclusion

### Introduction

Insérez votre texte ici…

### Choix des technologies

Insérez votre texte ici…

### Architecture physique et logique du système

#### Architecture physique

Insérez votre texte ici…

#### Architecture logique

Insérez votre texte ici…

### Diagramme de déploiement

Le diagramme de déploiement est une vue statique qui sert à représenter l’utilisation de l’infrastructure physique par le système. Et la manière dont les composants du système sont répartis ainsi que les relations entre eux .

Le formalisme est le suivant :

|  |  |  |
| --- | --- | --- |
| Elément Description Représentation | | |
| Les composants | Il représente une entité logicielle du système (Fichier de code source, programmes, documents, fichiers de ressource, etc.). Un composant est représenté par une boîte rectangulaire avec deux rectangles dépassant du côté gauche portant le nom du composant. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Les nœuds | Un nœud représente un ensemble d'éléments matériels du système. Cette entité est représentée par un cube tridimensionnel. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Les Associations | Une association, représentée par une ligne pleine entre deux nœuds, indique une ligne de communication entre les éléments matériels. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Les dépendances | Une dépendance est utilisée pour mobiliser la relation entre deux composants. La notation utilisée pour cette relation de dépendance est une flèche de pointillés. |  |

Exemple de diagramme de déploiement :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Le diagramme suivant est une architecture utilisée pour le déploiement de notre application :

Insérez votre diagramme ici…

### Diagramme de composant

Le diagramme de composants est principalement employé pour décrire les dépendances entre divers composants (les dépendances entre fichiers exécutables et les fichiers sources). Le diagramme de composants modélise les composants logiciels utilisés pour implémenter un système et l’association entre ces composants

Le diagramme de composant décrit les composants et leurs dépendances dans leurs environnements de réalisation. A cet effet il est composé des éléments suivants :

|  |  |  |
| --- | --- | --- |
| Elément Description Représentation | | |
| Les Composants | Un composant représente une entité logicielle d'un système. Un composant est représenté par une boîte rectangulaire, avec deux rectangles dépassant du côté gauche. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Dépendances | Elles sont utilisées pour modéliser la relation entre deux composants. La notation utilisée pour cette relation de dépendance est une flèche pointillée, se dirigeant d'un composant donné au composant dont il dépend. | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |
| Interface (Nœud) | Lien entre deux composants logiciels. Il en existe 2 :   * Interface fournie * Interface requise | C:\Users\MANGOUBA BRICELLE\Desktop\Diagramme Bricelle\Diagramme sans nom.png |

Exemple de diagramme de composants :

Une image contenant texte, diagramme, capture d’écran, Police

Description générée automatiquement

Le diagramme suivant montre la structure des composants logiciels du système :

Insérez votre diagramme ici…

### Conclusion

Insérez votre texte ici…

## Dossier 6 : Test de fonctionnalités

Résumé :

|  |
| --- |
| Un test désigne une procédure de vérification partielle d'un système. Son objectif principal est d'identifier un nombre maximal de comportements problématiques du logiciel. Il permet ainsi, dès lors que les problèmes identifiés seront corrigés, d'en augmenter la qualité. |

Aperçu :

Introduction

1. Fonctionnalités
2. Tests

Conclusion

### Introduction

Insérez votre texte ici…

### Fonctionnalités

Insérez votre texte ici…

### Tests

Insérez votre texte ici…

### Conclusion

Insérez votre texte ici…

## Dossier 7 : Guide d’installation et guide d’utilisateur

Résumé :

|  |
| --- |
| Le manuel d’installation et d’utilisation est le document établi après réalisation d’une application, d’un logiciel ou d’une plateforme. Il renseigne sur la question « comment installer et utiliser le logiciel, l’application ou la plateforme que l’on a devant soi ? » |

Aperçu :

Introduction

1. Guide d’installation
2. Guide utilisateur

Conclusion

### Introduction

Insérez votre texte ici…

### Guide d’installation

Insérez votre texte ici…

### Guide utilisateur

Insérez votre texte ici…

### Conclusion

Insérez votre texte ici…

## Conclusion

Insérez votre texte ici… (Conclusion de la 2e partie : PHASE TECHNIQUE).

# CONCLUSION GÉNÉRALE

Insérez votre texte ici…

# BIBLIOGRAPHIE

Insérez votre bibliographie ici…

# WEBOGRAPHIE

Insérez votre webographie ici…

# ANNEXES

# TABLE DES MATIÈRES

[DÉDICACE III](#_Toc202724561)

[REMERCIEMENTS IV](#_Toc202724562)

[SOMMAIRE V](#_Toc202724563)

[LISTE DES TABLEAUX VI](#_Toc202724564)

[LISTE DES FIGURES VII](#_Toc202724565)

[SIGLES ET ABBREVIATIONS VIII](#_Toc202724566)

[GLOSSAIRE IX](#_Toc202724567)

[RESUME X](#_Toc202724568)

[ABSTRACT XI](#_Toc202724569)

[INTRODUCTION GÉNÉRALE 1](#_Toc202724570)

[1ère PARTIE : PHASE D’INSERTION 2](#_Toc202724571)

[Introduction 3](#_Toc202724572)

[I. Accueil et intégration en entreprise 4](#_Toc202724573)

[1. Accueil en entreprise 4](#_Toc202724574)

[2. Intégration en entreprise 4](#_Toc202724575)

[II. Présentation de l’entreprise 4](#_Toc202724576)

[1. Historique 4](#_Toc202724577)

[2. Situation géographique 4](#_Toc202724578)

[3. Organigramme 4](#_Toc202724579)

[4. Mission 5](#_Toc202724580)

[5. Activités 5](#_Toc202724581)

[6. Ressources matérielles et logicielles 5](#_Toc202724582)

[6.1. Ressources matérielles 5](#_Toc202724583)

[6.2. Ressources logicielles 5](#_Toc202724584)

[Conclusion 6](#_Toc202724585)

[2ème PARTIE : PHASE TECHNIQUE 7](#_Toc202724586)

[Introduction 8](#_Toc202724587)

[Dossier 1 : L’Existant 9](#_Toc202724588)

[Introduction 10](#_Toc202724589)

[I. Présentation du thème/projet 11](#_Toc202724590)

[II. Etude de l’existant 11](#_Toc202724591)

[III. Critique de l’existant 11](#_Toc202724592)

[IV. Problématique 11](#_Toc202724593)

[V. Proposition de solution 12](#_Toc202724594)

[Conclusion 12](#_Toc202724595)

[Dossier 2 : Cahier des charges 13](#_Toc202724596)

[Introduction 14](#_Toc202724597)

[I. Contexte et justification de l’étude /du projet 15](#_Toc202724598)

[1. Contexte 15](#_Toc202724599)

[2. Justification 15](#_Toc202724600)

[II. Bénéficiaire et cible 15](#_Toc202724601)

[1. Bénéficiaire 15](#_Toc202724602)

[2. Cible 15](#_Toc202724603)

[III. Les objectifs de l’étude/projet 15](#_Toc202724604)

[1. Objectif général 15](#_Toc202724605)

[2. Objectifs spécifiques 16](#_Toc202724606)

[IV. Expressions des besoins de l’utilisateur 16](#_Toc202724607)

[1. Besoins fonctionnels 16](#_Toc202724608)

[2. Besoins non fonctionnels 16](#_Toc202724609)

[V. Planification du projet/ étude / Gantt Project 16](#_Toc202724610)

[1. Intervenants 16](#_Toc202724611)

[2. Diagramme de Gantt 16](#_Toc202724612)

[VI. Estimation du coût du projet/étude 17](#_Toc202724613)

[1. Ressources Matérielles 17](#_Toc202724614)

[2. Ressources logicielles 17](#_Toc202724615)

[3. Ressources humaines 18](#_Toc202724616)

[4. Cout total 18](#_Toc202724617)

[5. Les contraintes du projet/ étude 18](#_Toc202724618)

[6. Les livrables 19](#_Toc202724619)

[Conclusion 20](#_Toc202724620)

[Dossier 3 : Dossier d’Analyse 21](#_Toc202724621)

[Introduction 22](#_Toc202724622)

[I. Méthode et langage d’analyse 23](#_Toc202724623)

[1. Etude comparative UML et MERISE 23](#_Toc202724624)

[2. Choix et présentation de la démarche d’analyse 24](#_Toc202724625)

[3. Etude comparative des processus unifiés 26](#_Toc202724626)

[4. Choix et présentation du processus unifié 27](#_Toc202724627)

[II. Modélisation 28](#_Toc202724628)

[1. Diagramme des cas d’utilisation 28](#_Toc202724629)

[1.1. Diagramme de cas d’utilisation globale 30](#_Toc202724630)

[1.2. Diagrammes de cas d’utilisation spécifique 30](#_Toc202724631)

[2. Descriptions textuelles 30](#_Toc202724632)

[2.1. Cas 1 31](#_Toc202724633)

[2.2. Cas 2 31](#_Toc202724634)

[2.3. Cas 3 31](#_Toc202724635)

[2.4. Cas 4 31](#_Toc202724636)

[3. Diagramme de communication 32](#_Toc202724637)

[3.1. Cas 1 32](#_Toc202724638)

[3.2. Cas 2 33](#_Toc202724639)

[3.3. Cas 3 33](#_Toc202724640)

[3.4. Cas 4 33](#_Toc202724641)

[5. Diagramme de séquence 34](#_Toc202724642)

[5.1. Cas 1 35](#_Toc202724643)

[5.2. Cas 2 36](#_Toc202724644)

[5.3. Cas 3 36](#_Toc202724645)

[5.4. Cas 4 36](#_Toc202724646)

[6. Diagramme d’activité 36](#_Toc202724647)

[6.1. Cas 1 37](#_Toc202724648)

[6.2. Cas 2 38](#_Toc202724649)

[6.3. Cas 3 38](#_Toc202724650)

[6.4. Cas 4 38](#_Toc202724651)

[Conclusion 39](#_Toc202724652)

[Dossier 4 : Dossier de Conception 40](#_Toc202724653)

[Introduction 41](#_Toc202724654)

[I. Diagramme de classe 42](#_Toc202724655)

[II. Diagramme d’état transition 44](#_Toc202724656)

[III. Diagramme de paquetage 46](#_Toc202724657)

[Conclusion 47](#_Toc202724658)

[Dossier 5 : Dossier de Réalisation : 48](#_Toc202724659)

[Introduction 49](#_Toc202724660)

[I. Choix des technologies 50](#_Toc202724661)

[II. Architecture physique et logique du système 50](#_Toc202724662)

[1. Architecture physique 50](#_Toc202724663)

[2. Architecture logique 50](#_Toc202724664)

[III. Diagramme de déploiement 50](#_Toc202724665)

[IV. Diagramme de composant 52](#_Toc202724666)

[Conclusion 54](#_Toc202724667)

[Dossier 6 : Test de fonctionnalités 55](#_Toc202724668)

[Introduction 56](#_Toc202724669)

[I. Fonctionnalités 57](#_Toc202724670)

[II. Tests 57](#_Toc202724671)

[Conclusion 58](#_Toc202724672)

[Dossier 7 : Guide d’installation et guide d’utilisateur 59](#_Toc202724673)

[Introduction 60](#_Toc202724674)

[I. Guide d’installation 61](#_Toc202724675)

[II. Guide utilisateur 61](#_Toc202724676)

[Conclusion 62](#_Toc202724677)

[Conclusion 63](#_Toc202724678)

[CONCLUSION GÉNÉRALE 64](#_Toc202724679)

[BIBLIOGRAPHIE 1](#_Toc202724680)

[WEBOGRAPHIE 2](#_Toc202724681)

[ANNEXES 3](#_Toc202724682)

[TABLE DES MATIÈRES 4](#_Toc202724683)