* La dédicace
* Les remerciements
* Le sommaire
* La liste des tableaux et figures
* Le glossaire
* Résumé
* abstrac

**PHASE D’INSERTION**

**PHASE D’INSERTION**

**I - ACCUEIL EN ENTREPRISE**

1. **Accueil**

J'ai commencé mon stage à l'OIC le 11 juillet 2023. Les premières étapes ont été marquées par une phase d'intégration soigneusement planifiée afin d'assurer une transition en douceur vers l'environnement de travail.

1. **Intégration**

Au cours des premières semaines de mon stage au Centre d'innovation océaniques de l'OIC, j'ai été chargée de diverses tâches qui m'ont permis de me familiariser rapidement avec les objectifs stratégiques et les technologies importantes de l'entreprise. J'ai eu la chance de pouvoir collaborer étroitement avec des collègues plus expérimentés qui ont généreusement offert leurs connaissances et contribué activement à répondre à certaines des principales préoccupations actuelles en matière de développement.

Cette première période d'intégration m'a également donné une excellente occasion de faire connaissance avec les autres membres de mon équipe, d'en apprendre davantage sur leur position individuelle au sein de l'entreprise et de me familiariser avec les processus de travail et les normes professionnelles qui régissent les opérations quotidiennes.

Mon encadrant professionnel a conçu mon emploi du temps des premières semaines, du lundi au vendredi de 8h30 à 16h, de manière à maximiser ma participation aux formations techniques obligatoires, aux réunions régulières de l'équipe pour discuter de l'avancement du projet et aux plages horaires consacrées à la réalisation autonome des tâches qui m'étaient confiées. Grâce à cet emploi du temps équilibré, j'ai pu tirer le meilleur parti de ce temps d'intégration réellement pédagogique.

**II - PRÉSENTATION DE L'ENTREPRISE**

1. Situation Géographique

TODO Localisation

Stratégiquement basé à dans trois villes Kribi(siège), Bafoussam et à Douala - Bonamousadi ou j’ai effectué mon stage, OIC bénéficie des emplacements Stratégique au sein d'un écosystème technologique en plein essor.

2. Historique et Missions

a) Historique d’OIC

Depuis sa fondation en 2017, OIC a fait un parcours impressionnant. Initialement lancé avec une équipe restreinte, l'entreprise a surmonté les défis initiaux pour devenir un pilier de l'innovation numérique. Présentement classer deuxième incubateur le plus influent au Cameroun juste derrière son concurrent Activ’Space.

b) Les Missions d’Ocean Innovation Center

Les missions d'OIC sont de la plus haute importance, reflétant sa vocation à jouer un rôle pivot dans le soutien à l'incubation de startups innovantes est au cœur des missions d'OIC. En identifiant et en accompagnant des startups prometteuses, l'entreprise vise à créer un écosystème fertile pour la croissance entrepreneuriale. Grâce à des programmes d'incubation dédiés et à des partenariats stratégiques, OIC contribue à transformer des idées audacieuses en entreprises florissantes, stimulant ainsi le développement économique et technologique.

De plus, le développement numérique et l'innovation. L'entreprise s'est engagée à former et à éduquer la jeunesse, catalysant ainsi l'essor de compétences numériques cruciales. Grâce à des programmes de formation intensifs et des « bootcamps », OIC vise à transformer les jeunes talents en professionnels hautement qualifiés, prêts à relever les défis de l'industrie.

Enfin, OIC a embrassé le concept d'espaces de coworking, créant un environnement dynamique où les esprits créatifs peuvent fusionner et collaborer. Cette mission est motivée par la conviction qu'un espace de travail collaboratif favorise l'échange d'idées et la création de solutions novatrices. En fournissant des installations modernes et inspirantes, OIC encourage le partage de connaissances et l'émergence de projets collaboratifs fructueux.

**3. Organisation Interne de la Structure**

TODO organigramme

L'équipe de direction se compose de :

- **Jacques Bonjawo**, CEO et co-fondateur, possédant plus de 20 ans d'expérience dans le secteur de la technologie.

- **Anshumali Bhushan**, Senior Business Development Manager, ayant contribué à la création de partenariats stratégiques avec des entreprises de premier plan.

- **Dorian Nogneng**, Chief Technology Officer, ayant dirigé le développement d'un logiciel innovant de gestion de projet.

**4. Activités, Réalisations et Partenaires**

a) **Activités**

Les activités d'OIC témoignent de son engagement dynamique envers le développement technologique et l'encouragement de l'innovation. À travers une gamme variée d'activités, l'entreprise crée un environnement stimulant et propice à l'apprentissage, à la créativité et à la croissance professionnelle.

L'une des activités clés d'OIC est la mise en place de bootcamps et de formations spécialisées. Ces programmes intensifs immergent les participants dans des domaines tels que le développement web, l'intelligence artificielle et la cybersécurité. Les participants acquièrent des compétences pratiques et pertinentes, prêts à relever les défis du monde numérique en constante évolution.

De plus, OIC propose des séminaires et des conférences animés par des experts de l'industrie. Ces événements offrent aux participants un accès privilégié aux dernières tendances technologiques, aux bonnes pratiques et aux perspectives éclairées. Ils favorisent également le réseautage et l'échange de connaissances, enrichissant ainsi l'écosystème technologique local.

b) **Réalisations**

Les réalisations d'OIC reflètent son impact positif et sa contribution au développement numérique et à l'innovation. L'entreprise a joué un rôle crucial dans la création et l'essor de startups prometteuses, marquant le début d'un voyage entrepreneurial fructueux.

Parmi ses réalisations marquantes figure la startup "TechNovate", spécialisée dans la conception d'applications de santé numérique. Incubée au sein d'OIC, "TechNovate" a rapidement évolué pour devenir un acteur clé dans la fourniture de solutions de suivi de santé innovantes. Son application de suivi des maladies chroniques a atteint plus de 50 000 utilisateurs en seulement un an, témoignant de l'impact positif de l'incubation d'OIC sur l'innovation.

En termes de formation, OIC a formé plus de 1xx jeunes talents dans divers domaines technologiques au cours de la dernière année. Par exemple, le programme a doté des centaines de jeunes développeurs des compétences nécessaires pour prospérer dans l'industrie numérique en constante évolution. Ces jeunes talents ont ensuite trouvé des opportunités d'emploi dans des entreprises de renom, renforçant ainsi l'écosystème numérique local.

En somme, les réalisations d'OIC transcendent les frontières de l'innovation, englobant le succès de startups incubées, l'ascension de jeunes formés et le rayonnement de programmes lancés. En catalysant le potentiel des jeunes talents et en offrant un environnement propice à la création d'entreprises novatrices, OIC inscrit son empreinte dans le paysage technologique et contribue de manière significative à la transformation numérique et à la croissance économique.

c) Les Partenaires

5. Ressources de l'Entreprise

a) Ressources Matérielles

Les ressources matérielles d'OIC sont là pour offrir un environnement idéal aux participants. On y trouve :

- Ordinateurs : Pour un travail numérique efficace.

- Matériel de présentation : Comme des vidéoprojecteurs pour des présentations dynamiques.

Tous ces éléments aident les participants à apprendre et à créer dans un espace moderne et bien équipé, qu'il s'agisse de formations, de séminaires ou de sessions de travail collaboratif.

b) Ressources Logicielles

OIC met à disposition des logiciels essentiels pour faciliter le travail :

-Logiciels de Développement : Pour créer des applications et des sites web, comme la suite "JetBrains" et "Visual Studio Code".

- Logiciels de Gestion de Projet : Pour organiser et suivre les projets, comme "OpenProject".

En plus de cela, il y a des logiciels courants comme la suite bureautique de Microsoft pour la gestion de documents, la suite Adobe pour la création graphique, Google Chrome pour la navigation web, Antivirus pour la protection, et le Système Windows 10 comme système d'exploitation.

Ces logiciels aident les participants à développer, gérer et créer de manière efficace, couvrant une gamme variée d'activités au sein d'OIC.

OIC s'affirme comme un acteur clé de la formation et de l'innovation numérique, contribuant au développement technologique et à la croissance économique avec des initiatives concrètes et des résultats mesurables.

2ème partie : phase technique

Dossier 1 : L'Existant - Conception et Réalisation d'une Plateforme de Vente des Pagnes Camerounais

1. Présentation du Thème (ou du Projet)

Ce dossier se penche sur la création d'une plateforme en ligne dédiée à la vente de pagnes camerounais, des tissus traditionnels profondément ancrés dans la culture du Cameroun. L'objectif est de moderniser l'accès à ces pagnes tout en respectant leur signification culturelle.

2. Étude de l'Existant

Actuellement, l'achat de pagnes camerounais se fait principalement en personne, via des magasins physiques et des marchés locaux. Quelques vendeurs individuels opèrent en ligne, mais l'expérience globale d'achat reste limitée.

3. Critique de l'Existant

Les limitations de l'approche actuelle sont évidentes. Les options limitées en termes de motifs, la nécessité de se rendre physiquement dans des lieux de vente, et le manque d'informations en ligne rendent l'achat de pagnes moins pratique. De plus, la valeur culturelle de ces tissus est souvent difficile à appréhender en ligne.

4. Problématique

Face à ces problèmes, la question centrale est la suivante : Comment concevoir une plateforme de vente en ligne qui non seulement propose une variété de pagnes traditionnels, mais qui aussi parvienne à transmettre leur richesse culturelle tout en offrant une expérience d'achat simple et moderne ?

5. Proposition de solution

Proposition de Solution La proposition de solution est la création d'une plateforme en ligne conviviale et interactive appelée "FadhyStore". Cette plateforme offrira une gamme variée de pagnes authentiques tout en fournissant des informations détaillées sur chaque produit, ainsi qu'une histoire culturelle associée. Elle facilitera l'achat en ligne grâce à une interface intuitive et une option de livraison rapide. " FadhyStore " sera conçue pour valoriser la culture camerounaise tout en offrant une expérience d'achat moderne et engageante pour les clients, contribuant ainsi à la préservation et à la célébration des pagnes traditionnels.

- Dossier 2 : Le Cahier des Charges

I. Contexte et Justification de l'Étude / du Projet pour la Conception et Réalisation d'une Plateforme de Vente des Pagnes Camerounais

1- Contexte

La richesse culturelle des pagnes camerounais est un élément emblématique du patrimoine du Cameroun. Cependant, l'achat de ces pagnes est souvent limité par des contraintes géographiques et logistiques. Les marchés physiques ne permettent pas toujours un accès facile et varié à ces produits, entravant ainsi leur diffusion et leur appréciation.

1. Justification de l'Étude / du Projet

Face à cette situation, la création d'une plateforme de vente en ligne pour les pagnes camerounais devient essentielle. Cette plateforme permettra aux amoureux de la culture camerounaise, aussi bien locaux qu'internationaux, d'accéder facilement à une gamme étendue de pagnes traditionnels. Elle offrira également une opportunité de promouvoir et de préserver la signification culturelle des pagnes tout en répondant aux besoins modernes d'achat en ligne.

Le projet vise ainsi à éliminer les barrières géographiques et à faciliter l'achat des pagnes, contribuant ainsi à la diffusion et à la valorisation de cette riche tradition. La plateforme se propose de créer une expérience d'achat moderne, pratique et informative tout en célébrant le patrimoine culturel du Cameroun.

II. Les Objectifs de l'Étude / Projet pour la Conception et Réalisation d'une Plateforme de Vente des Pagnes Camerounais

Objectif Général :

L'objectif général de cette étude / projet est de concevoir et réaliser une plateforme de vente en ligne dédiée aux pagnes camerounais, visant à moderniser l'expérience d'achat, à faciliter l'accès à ces produits traditionnels et à promouvoir leur signification culturelle.

Objectifs Spécifiques :

1. Mettre en place une interface utilisateur conviviale et intuitive pour la plateforme, offrant une expérience d'achat en ligne fluide et agréable.

2. Proposer une gamme diversifiée de pagnes camerounais authentiques, avec des informations détaillées sur chaque produit.

3. Intégrer un système de paiement en ligne sécurisé, permettant aux clients de finaliser leurs achats en toute confiance.

4. Fournir des fonctionnalités de recherche avancées pour faciliter la découverte et la sélection de pagnes en fonction des préférences individuelles.

5. Créer des pages dédiées à chaque pagne, incluant des histoires culturelles et des significations, pour valoriser la richesse culturelle de ces produits.

6. Mettre en place un système de gestion des commandes et de suivi des livraisons pour assurer une expérience d'achat transparente.

7. Assurer la sécurité des données personnelles des clients et mettre en place des mesures de protection des informations sensibles.

8. Collaborer avec des artisans et des fournisseurs locaux pour garantir l'authenticité et la qualité des pagnes proposés.

En définissant ces objectifs spécifiques, le projet s'engage à créer une plateforme en ligne complète et fonctionnelle qui offre une expérience d'achat exceptionnelle et qui contribue à la préservation et à la célébration des pagnes camerounais.

III. Expressions des Besoins de l'Utilisateur pour la Conception et Réalisation d'une Plateforme de Vente des Pagnes Camerounais

A) Besoins Fonctionnels :

* Inscription et Connexion : Permettre aux utilisateurs de créer des comptes personnels et de se connecter facilement pour accéder à la plateforme.
* Catalogue de Produits : Afficher une liste complète de pagnes avec des filtres de recherche pour faciliter la découverte.
* Détails du Produit : Fournir des informations détaillées, y compris des images haute résolution et des descriptions culturelles, pour chaque pagne.
* Panier d'Achat : Permettre aux utilisateurs d'ajouter des produits à leur panier et de visualiser le contenu avant l'achat.
* Paiement en Ligne : Intégrer un système de paiement sécurisé pour finaliser les achats en ligne.
* Gestion des Commandes : Offrir un espace pour suivre l'état des commandes, les paiements et les livraisons.
* Fonctionnalités Sociales : Permettre aux utilisateurs de partager leurs choix de pagnes, de laisser des commentaires et des avis.
* Support Client : Fournir un moyen de contacter le support en cas de questions, de préoccupations ou de retours.
* Gestion de Compte : Permettre aux utilisateurs de gérer leurs informations personnelles et leurs préférences.
* Sécurité des Données : Garantir la protection des données personnelles et des informations sensibles des utilisateurs.

B) Besoins Non Fonctionnels :

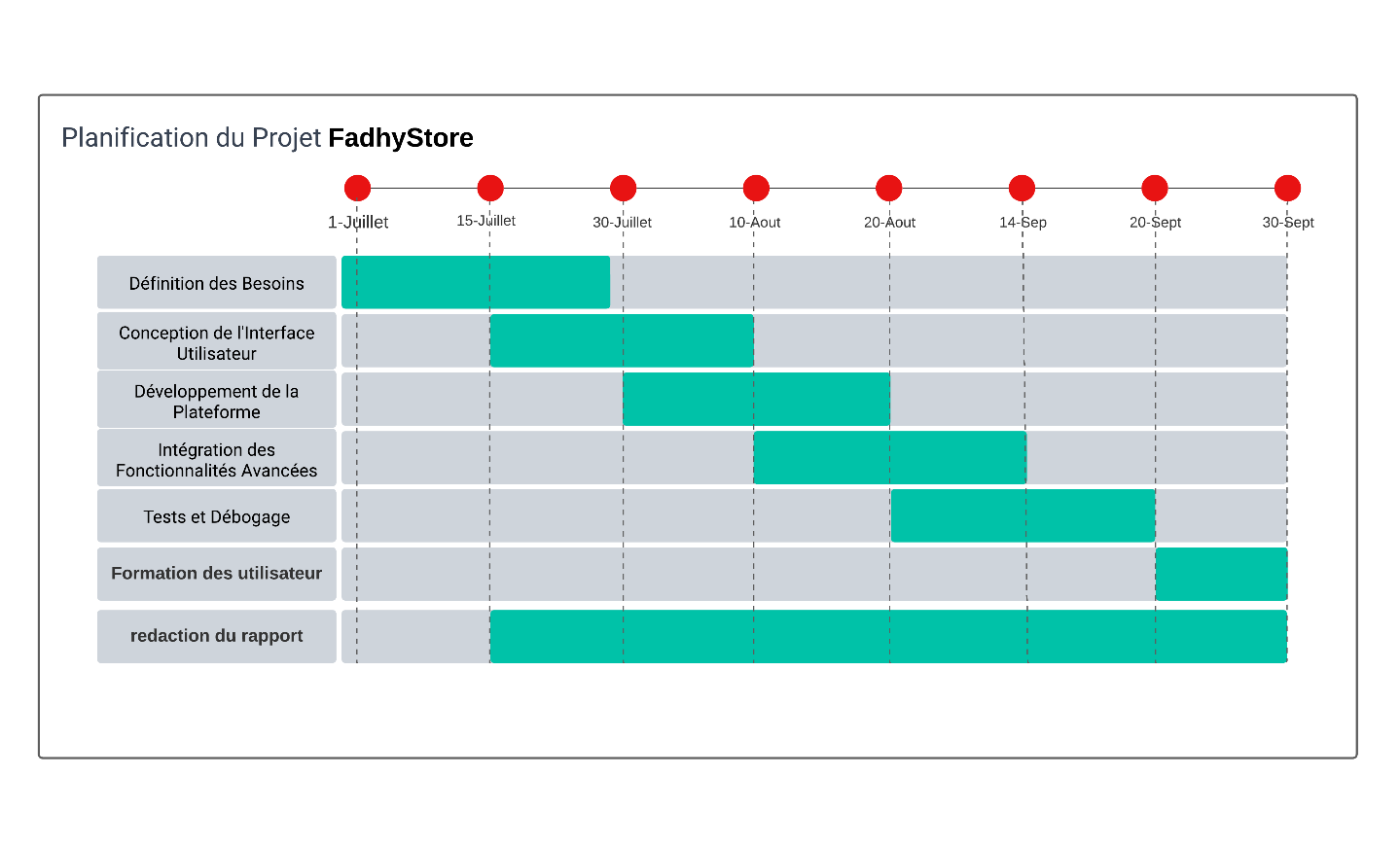
* Performance : Assurer une plateforme rapide et réactive, même lors de pics de trafic.
* Convivialité : Offrir une interface utilisateur intuitive et facile à utiliser pour une expérience agréable.
* Sécurité : Mettre en place des mesures de sécurité robustes pour protéger les transactions et les données des utilisateurs.
* Compatibilité : S'assurer que la plateforme fonctionne correctement sur différentes plateformes (ordinateurs, smartphones, tablettes) et navigateurs.
* Accessibilité : Garantir que la plateforme est accessible aux personnes ayant des besoins spécifiques.
* Stabilité : Éviter les erreurs et les pannes pour garantir une expérience ininterrompue.
* Évolutivité : Concevoir la plateforme pour qu'elle puisse évoluer avec de nouvelles fonctionnalités et une augmentation du nombre d'utilisateurs.
* Esthétique : Assurer un design visuellement attrayant et cohérent pour refléter la culture des pagnes.
* Performances en Temps Réel : Garantir que les mises à jour d'inventaire et de statut de commande sont en temps réel.

Confidentialité : Assurer que les informations personnelles des utilisateurs sont traitées de manière confidentielle et conforme aux réglementations en vigueur.

En identifiant ces besoins fonctionnels et non fonctionnels, le projet vise à créer une plateforme complète, sécurisée et conviviale qui répond aux attentes et aux exigences des utilisateurs, garantissant ainsi une expérience d'achat en ligne exceptionnelle pour les pagnes camerounais.

IV. Planification du projet/ étude Gant Project

La planification du projet est essentielle pour garantir une mise en œuvre efficace de la conception et de la réalisation de la plateforme de vente des pagnes camerounais. Pour ce faire, nous utilisons l'outil Gantt Project pour visualiser les différentes étapes du projet et les délais associés. Voici un aperçu du plan :



Planifications de taches.

V Estimation du coût du projet/étude

Pour la réalisation de ce projet, nous devons avoir à notre disposition de ressources matérielles, logiciels, humaines et également financières, que nous avons listés comme suite :

### V.1) Équipe de travail

Tableau 01 : équipe de travaille

|  |  |
| --- | --- |
| **Noms et Prénom** | **Fonction** |
|  | Modélisation, Conception, et Réalisation |
|  | Encadrant académique et chargé de communication à l’IAI Cameroun Centre technologique d'excellence de Paul Biya |
| M. LONTCHI LOIC STEVE | Chef de projet. Encadreur professionnel. Manageur du centre OIC Douala. |

### V.2) Resource matérielles

Tableau 02 : ressource matérielle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Matériels** | **Fonctions** | **Prix unitaire** | **Quantité** | **Prix totale (FCFA)** |
| **Ordinateur : i5 5e Gene 8Go Ram** | Équipement de développement | **250 000** | **03** | **750 000** |
| Box wifi Orange | Accès à internet | **39 900** | **01** | **39 900** |
| Total |  | | | |

### V.3) Ressources humaines

NB : La source de chaque prix est disponible et détaillée dans la webographie

Tableau 03 : ressources humaines

|  |  |  |
| --- | --- | --- |
| **Main d’œuvre** | **Nombres** | **Salaires** |
| **Analyste & concepteur** | 02 | 400 000 |
| **Designer** | 01 | 100 000 |
| **Programmeur** | 02 | 200 000 |
| **Tester** | 15 | 300 000 |
| **Total** | 15 | 1 000 000 |

### V.4) Grand total

Tableau 04 : Total

|  |  |
| --- | --- |
| **Types** | **Prix (F CFA)** |
| **Matérielles** |  |
| Humaines | **1 000 000** |
| Imprévues | **\* 10%** |
| Total de cout |  |

VI. Les contraintes du projet/ étude

VII. Les livrable

A la fin du délai fixé pour le développement de cette solution, les éléments qui constituent les livrables sont :

* Le code source de l’application
* Le dossier de conception
* Le dossier de réalisation
* Un manuel d’utilisation

**Conclusion**

L’élaboration de notre cahier de charges nous a ainsi permis d’avoir une idée précise des exigences nécessaires à la réalisation de notre plateforme. Nous allons aborder avec beaucoup de lucidité la prochaine partie qui est le dossier d’analyse et de conception.

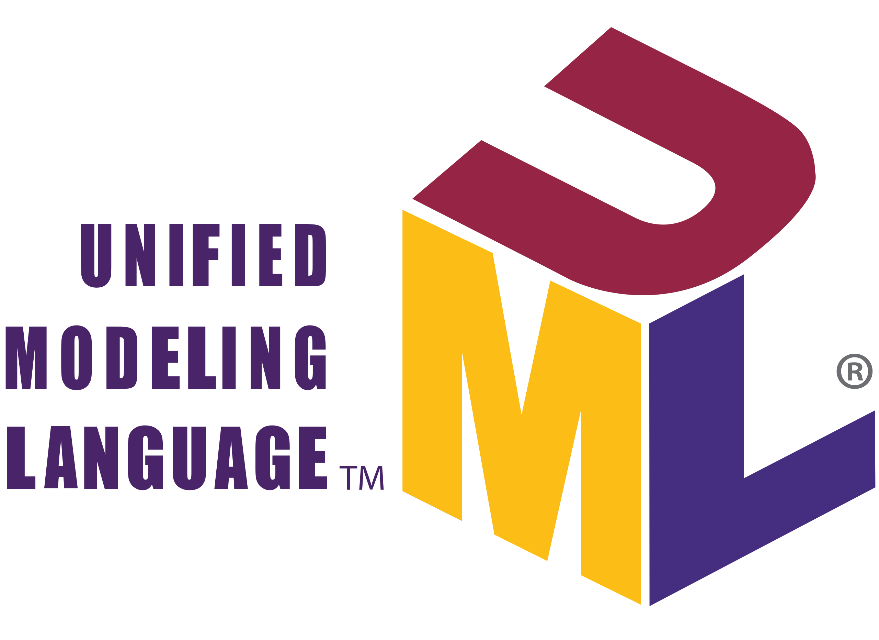
- Dossier 3 : le dossier d’analyse

1. Méthodologie

1.1. Etude comparative UML et MERISE

L'étude comparative entre UML (Unified Modeling Language) et MERISE permet de mieux comprendre les caractéristiques et les différences entre ces deux méthodes de modélisation en génie logiciel.

UML est une notation graphique standardisée permettant de représenter et concevoir des systèmes logiciels complexes. Les principaux diagrammes UML sont les diagrammes de classes, de cas d'utilisation, de séquences, d'états, d'activités, etc. UML permet une approche objet et une vision globale du système.



*Logo UML*

MERISE est une méthodologie de conception et d'analyse des systèmes d'information datant des années 1980. Elle repose sur la modélisation conceptuelle des données (MCD) avec le modèle entité-association et la modélisation organisationnelle des traitements (MOT). MERISE suit une approche structurée et descendante.

On peut donc dire qu'UML est plus récent, standardisé et adapté aux systèmes complexes orientés objet. MERISE est plus ancienne et suit une approche structurée autour des données et traitements. UML permet une vue d'ensemble avec ses divers diagrammes tandis que MERISE décompose le système en modules.

Les deux approches restent cependant complémentaires. UML permet une conception haut niveau que MERISE peut détailler de manière structurée. Le choix entre les deux dépendra du contexte et des besoins du projet de développement.

1.2. Etude comparative des processus unifiés

Le Rational Unified Process (RUP) est une méthodologie lourde et rigoureuse adaptée aux projets complexes et critiques. Elle découpe le cycle de développement en 4 phases successives pouvant chacune compter plusieurs mois :

- **Inception** : cadrage du projet, étude d'opportunité, analyse du contexte métier, premiers cas d'utilisation critiques.

- Elaboration : analyse détaillée des besoins, conception de l'architecture technique du système, planification des cycles de développement.

- **Construction** : conception et codage des composants logiciels, tests unitaires et d'intégration, recette technique. Répétition de cycles courts de 2 à 6 semaines.

- **Transition** : validation finale, formation des utilisateurs, déploiement, retours d'expérience.

Le RUP définit de nombreuses disciplines couvrant toutes les facettes du logiciel : besoins, analyse, conception, implémentation, test, gestion de projet, environnement, etc. De nombreux artefacts sont produits tout au long du processus.

Le Two Track Unified Process (2TUP) reprend les principes du RUP en distinguant l'ingénierie du système (aspects techniques) et l'ingénierie des cas d'utilisation (besoins métiers). Les cycles de développement sont plus courts (2 à 4 semaines) pour des livraisons plus fréquentes. Le 2TUP produit moins de documentation. Elle distingue deux axes :

- **L'ingénierie du système** : regroupe les activités techniques de conception architecturale, de développement logiciel et de tests. Des itérations de 2 à 4 semaines permettent de livrer rapidement des incréments fonctionnels.

- **L'ingénierie des cas d'utilisation** : analyse les besoins métiers, définit les cas d'utilisation et valide fréquemment les fonctionnalités avec le client. Recherche la satisfaction client.

Les phases du 2TUP sont allégées par rapport au RUP :

- **Inception** : cadrage du projet et des use cases prioritaires avec le client.

- **Elaboration** : analyse détaillée des besoins, architecture technique flexible, planning des premières livraisons.

- **Construction** : développement par itérations de 2 à 4 semaines et livraisons fréquentes de versions utilisables.

- **Transition** : déploiement final et retours d'expérience pour améliorations futures.

Les principes du 2TUP sont la satisfaction client, l'implication des utilisateurs, la réactivité au changement, et un souci constant de qualité et d'efficacité.

Le 2TUP est donc plus léger, pragmatique et adapté à l'agilité. Il convient à des projets de taille moyenne avec des équipes de 5 à 20 développeurs. Le RUP est plus structuré et documenté, requérant des équipes de taille importante pour des systèmes complexes à criticité élevée.

Le choix entre ces deux processus dépendra finalement de nombreux paramètres : envergure du projet, délais, budget, ressources, criticité, environnement technique...

1. Modélisation

Introduction

La modélisation d'un système logiciel est une étape essentielle dans le cycle de développement, qui permet de représenter graphiquement différents aspects du système avant sa réalisation technique. Le langage UML est un standard largement utilisé pour la modélisation objet.

Dans le cadre de notre projet de développement d'un système de vente en ligne de pagnes traditionnels camerounais, nous allons appliquer UML pour modéliser le système sous différents angles, avec plusieurs diagrammes complémentaires :

- Le diagramme de cas d'utilisation modélise les interactions entre les acteurs et le système en représentant les principales fonctionnalités.

- Le diagramme de communication montre les interactions entre objets pour une fonction donnée.

- Le diagramme de séquence décrit chronologiquement les échanges de messages entre objets pour réaliser un scénario.

- Le diagramme d'activité représente le déroulement des traitements sous forme d'enchaînement d'actions.

Ces différents diagrammes UML vont nous permettre de couvrir les besoins fonctionnels et les flux du système sous différents angles avant la phase de conception.

2.1. Diagramme des cas d’utilisation

Le DCU permet de décrire le système en faisant ressortir les acteurs et les cas d’utilisation (use case) qui sont les fonctionnalités du système.

1. Les composants du DCU

Les composants principaux d’un DCU sont représentés dans le tableau ci-après :

Tableau 9 : Les composants du DCU

|  |  |  |
| --- | --- | --- |
| Composant | Description | Représentation graphique |
| Acteur | Entité qui interagit avec le système. Il peut être une personne, une organisation ou un autre système. |  |
| Cas d’utilisation  (CU) | Une fonctionnalité du système ou alors un ensemble d’événements qui se produisent quand un acteur utilise un système pour achever un processus. |  |
| Association | Liaison entre les acteurs et les cas d’utilisation. |  |

1. Formalisme d’un DCU

L’aspect général d’un DCU est représenté sur le schéma suivant :

Figure 5 : Formalisme d'un DCU

e) Description textuelle de quelques cas d’utilisation

- Cas d’utilisation s’authentifier

|  |
| --- |
| Sommaire d’identification |
| Titre : S’authentifier ;  Objectif : Accéder aux fonctionnalités du système et à son interface de travail ;  Résumé : Permettre à chaque utilisateur de renseigner ses identifiants de connexion (login et mot de passe) ;  Acteur(s) concerné(s) : Tous les utilisateurs du système. |
| Description des enchainements |
| **Condition** : L’utilisateur doit au préalable posséder un compte dans la plateforme ;  **Présupposé** :  **Déclencher** : Au clic du bouton ‘Se connecter’ ;  **Scénario principal** :   * La page de connexion s’affiche avec le formulaire ; * L’utilisateur remplit les champs de login et de mot de passe, puis soumet le formulaire ; * Le système vérifie la validité des informations saisies et affiche d’accueil à l’utilisateur connecté.   **Scénario alternatif :**  Si le login ou le mot de passe entré par l’utilisateur est incorrect, un message d’erreur est renvoyé et le système retourne à l’étape 2 du scénario principal.  **Post condition succès** : l’authentification est réussie et l’utilisateur accède à la page d’accueil.  **Post condition échec** : l’opération d’authentification échoue, l’utilisateur reçoit un message d’erreur et le système lui renvoie le formulaire d’authentification. |

- Cas d’utilisation consulter les pagnes

Tableau 11 : Description textuelle du CU "Consulter les publications

|  |
| --- |
| Sommaire d’identification |
| Nom : Consulter les publications  Objectif : Prendre connaissance des objets perdus et trouvés publiés  Résumé : Naviguer sur la page des objets publiés et voir les photos des objets ainsi que leurs informations Acteurs concernés : Un utilisateur qui visite la plateforme |
| Description de l’enchainement |
| Précondition : Etre connecté et authentifié  Présupposé : L’utilisateur veut voir les publications des objets perdus et des objets trouvés  **Déclencher** : Clic sur le bouton ‘Consulter’  **Scénario principal** :   * Le système affiche la page des publications avec deux menus objets perdus et objets trouvés ; * L’utilisateur clique sur le choix des objets qu’il voudrait consulter * L’internaute fait défiler la page et lit les informations, il peut cliquer sur une publication pour voir plus de détails   **Scenario secondaire** : Si la connexion internet est interrompue, au clic du bouton consulter, l’accès à la page est impossible.  **Post-condition de succès** : Les pages s’affichent et des informations y figurent avec une barre de recherche pour pouvoir filtrer les post.  **Post-condition d’échec** : Les publications ne s’affichent pas. |

Cas d’utilisation publier un objet

Tableau : Description textuelle du CU "Publier un nouvelle article"

|  |
| --- |
| Sommaire d’identification |
| Titre : Publier un Produit ;  Objectif : enrichir la plateforme avec des nouveaux produits ;  Résumé : Permettre à l’administrateur du site d’ajouter de nouveaux produits ;  Acteur(s) concerné(s) : Les Adminitrateur. |
| Description des enchainements |
| Condition : L’utilisateur doit être s’authentifier avec un compte ayant les droits administrateurs ;  Présupposé : L’Administrateur a un nouveau produit a enregistrer sur la plateforme ;  Déclencher : Au clic du bouton ‘Ajouter’ ;  Scénario principal :   * Le menu s’affiche et l’utilisateur choisit une option ajouter un nouveau produit ; * La page d’ajout d’objet s’affiche avec le formulaire ; * L’utilisateur remplit les champs demandés, puis soumet le formulaire ; * Le système vérifie envoie les informations remplies à la BD pour l’enregistrement ; * Les informations sont enregistrées et un récapitulatif est envoyé à l’écran de l’utilisateur.   Scénario alternatif :  Si l’utilisateur n’a pas rempli tous les champs obligatoires, la soumission du formulaire échoue et on retourne à l’étape 2 du scénario principal.  **Post condition succès** : L’objet trouvé est publié avec succès et est visible par les utilisateurs de la plateforme.  **Post condition échec** : La publication échoue, l’utilisateur reçoit un message d’erreur et l’objet n’est pas publié dans la plateforme. |

2.2. Diagramme de communication

Appelé diagramme de collaboration en UML 1.x, il est devenu diagramme de communication à partir de UML 2.0. Il permet de montrer les interactions entre un ensemble d’objets pour réaliser une action. Dans le diagramme de communication, les objets sont disposés dans un espace lié par les liens et les messages circulent sur ces liens.

1. Les composants d’un diagramme de communication

Tableau : Les éléments d'un diagramme de communication

|  |  |  |
| --- | --- | --- |
| Elément | Description | Représentation |
| Objets | Les objets sont des instances des classes et sont rangés horizontalement |  |
| Messages | Elles indiquent les échanges entre les objets et les acteurs |  |
| Acteurs | Personne qui interagit et communique avec le système et les objets. |  |
| Lignes | Ce sont les éléments qui dirigent le flux de messages au sein du système dans le diagramme de communication |  |

1. Formalisme d’un diagramme de communication

Figure 9: Formalisme d'un diagramme de communication

2.3. Diagramme de séquence

Le diagramme des séquences (sequence diagram en anglais) permet de faire une représentation de façon séquentielle du déroulement des traitements et interactions entre les éléments du système et de ses acteurs. La représentation se concentre sur la séquence des interactions sur un point de vue temporel. De ce fait, le temps est représenté explicitement par une dimension verticale et celui-ci s’écoule de haut en bas en respectant la ligne de vie. Dans un diagramme de séquence, les principales informations sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique

1. Les composants d’un diagramme de séquences

Nous récapitulons dans le tableau ci-après les éléments principaux qui permettent de constituer le diagramme de séquences :

Tableau 15 : Les composants d'un diagramme de séquences

|  |  |  |
| --- | --- | --- |
| Composant | Description | Représentation |
| Objet | Les objets sont les instances de classe et sont rangés horizontalement. |  |
| Acteur | Personnes qui interagissent ou communiquent avec le système |  |
| Ligne de vie | La ligne de vie identifie l’existence de l’objet par rapport au temps. |  |
| Activation | Indique quand l’objet effectue une action |  |
| Message | Indique les communications entre les objets. Les messages qui vont de l’acteur vers l’objet sont dits « synchrones » et ceux qui vont de l’objet vers l’acteur sont dits « asynchrones ». |  |

1. Formalisme d’un diagramme de séquences
2. Diagrammes de séquences de quelques cas d’utilisation

2.4. Diagramme d’activité

Les diagrammes d'activité sont utilisés pour documenter le déroulement des opérations dans un système, du niveau commercial au niveau opérationnel (de haut en bas). L'usage général des diagrammes d'activité permet de faire apparaître les flots de traitements induits par les processus internes par rapport aux évènements externes.

1. Les composants d’un diagramme d’activités

Tableau 16 : Les composants du diagramme d'activités

|  |  |  |
| --- | --- | --- |
| Composant | Description | Représentation graphique |
| Activité | Représente une action faite par un objet |  |
| Transition | Les transitions sont utilisées pour marquer le passage d’une activité déjà accomplie vers une autre activité. |  |
| Etat initial | L’état initial marque le point d'entrée la première activité. Dans un diagramme d’activités, on trouve un seul état initial |  |
| Etat final | L'état final marque la fin du déroulement des opérations modélisées. On peut en trouver plusieurs dans un diagramme |  |
| Barre de synchronisation | lorsque plusieurs transition ce mêlent |  |
| Décision | Une décision est le résultat final d’une ou plusieurs activités |  |

1. Formalisme d’un diagramme d’activités

Figure 17 : Formalisme du diagramme d'activités

Dossier4 : dossier de conception

L'objectif de ce dossier est de fournir une vue d'ensemble de chaque aspect du projet ou du produit, en mettant l'accent sur les choix techniques, les fonctionnalités et les contraintes.

INTRODUCTION

Dans ce dossier, on retrouve les illustrations simplifiées d’architecture du système, lesquelles font apparaitre les différentes composantes de l’application, décrivent les procédures de ces composantes et font ressortir les attributs de l’application. Pour représenter cela, nous allons nous attarder sur les diagrammes de paquetage, le diagramme de classe, le diagramme de composants et le diagramme de déploiement.

1. LE DIAGRAMME DE CLASSES

Le diagramme de classe permet de représenter l’ensemble des informations finalisées qui sont gérées par le système. Ces informations sont structurées, c’est à dire qu’elles sont regroupées dans des classes. Le diagramme met en évidence d’éventuelles relations entre ces classes. Le diagramme de classes comporte quelques concepts : classe, attribut, identifiant, opération (méthode) et relation (association).

1. Les concepts du diagramme de classes

a) La classe

Une classe est un type de donnée abstrait caractérisé par des propriétés (attributs et méthodes) communes à un ensemble d'objets et permettant de créer des objets ayant ces propriétés. Une classe se présente avec UML sous forme d’un rectangle divisé en trois (03) sections comme le montre son formalisme ci-dessous :

|  |
| --- |
| NomDeLaClasse |
| -  -  -  Attribut\_1  ...  Attribut\_n  :  type\_  1  : ...  :  type\_n |
| + Operation\_1 ()  + ... ()  + Operation\_n () |

***Figure 19 : Formalisme d'une classe***

#### b) Les relations

Les relations sont les liens qui existent entre les classes dans un diagramme de classes. Il existe plusieurs types de relations entre classes : l’association, la généralisation/spécialisation et la dépendance.

 **L’association**

Une association représente une relation structurelle entre classes d’objets. La plupart des associations sont binaires, c’est à dire qu’elles connectent deux classes. On représente une association en traçant une ligne fléchée entre les classes associées comme on peut le voir sur le formalisme suivant :

|  |
| --- |
| Classe\_1 |
|  |

0..1 0..\*



Cl

asse\_2

***Figure 20 : Formalisme de l'association***

 **Les classes associations**

Les attributs d’une classe dépendent fonctionnellement de l’identifiant de la classe. Parfois, un attribut dépend fonctionnellement de 2 identifiants, appartenant à 2 classes différentes ou il peut arriver que l’on ait besoin de garder des informations (attributs ou opérations) propres à une association. Une classe de ce type est appelée classe association.

0..1

0..\*



Classe\_2



Classe\_1



Classe\_association

***Figure 21 : Formalisme de la classe association***

 **La composition**

La composition est un cas particulier de l’agrégation dans laquelle la vie des composants est liée à celle des agrégats. Elle fait souvent référence à une contenance physique. Dans la composition l’agrégat ne peut être multiple. La composition implique, en plus de l’agrégation, une coïncidence des durées de vie des composants : la destruction de l’agrégat (ou conteneur) implique automatiquement la destruction de tous les composants liés

0..1

0..\*



Classe\_2



Classe\_1

##### Figure 22 : Formalisme de la composition

 **L’agrégation**

Dans UML, l’agrégation n’est pas un type de relation mais une variante de l’association. Une agrégation représente une association non symétrique dans laquelle une des extrémités joue un rôle prédominant par rapport à l’autre extrémité. Les agrégations indiquent une relation de contenant contenu. Elle permet de modéliser des relations de type maître et esclaves

0..1

0..\*



Classe\_2



Classe\_1

##### Figure 23 : Formalisme de l'agrégation

 **La généralisation/spécialisation**

Le principe de généralisation / spécialisation permet d’identifier parmi les objets d’une classe (générique) des sous-ensembles d’objets (des classes spécialisées) ayant des définitions spécifiques. La classe plus spécifique (appelée aussi classe fille, classe dérivée, classe spécialisée, classe descendante …) est cohérente avec la classe plus générale (appelée aussi classe mère, classe générale …), c’est-à-dire qu’elle contient par héritage tous les attributs, les membres, les relations de la classe générale, et peut contenir d’autres



Classe\_mere



Classe\_fille3



Classe\_fille2



Classe\_fille1

***Figure 24 : Formalisme de la généralisation***

1. Diagramme de classe

2. Diagramme d’état transition

Un diagramme d’états transitions est un outil de modélisation utilisé pour représenter les différents états d’un système et les transitions entre ces états.

Le schéma ci-après représente le diagramme d’états transitions de notre système

Validation

commander()

Modifier()

Consulter()

Annulation

Archiver()



En attente



Validé



Modifié



Archivé



Consulté



***Figure 26 : Diagramme d'états transitions du système***

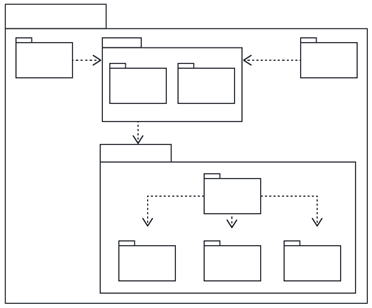
1. Diagramme de paquetage

Le diagramme de paquetage permet de représenter l’oragnisation des pakages et des dépendances entre eux dans un système logiciel. Un pakage est une unité d’organisation logicielle qui regroupe les classes, des interfaces, des ressources et même d’autres pakages. Le diagramme de paquetage UML aide à comprendre la structure globale du système et facilite la gestion des dépendances et des interactions entre composants.

### 1- Les éléments et le formalisme du diagramme de paquetage

Les éléments principaux d’un diagramme de paquetage sont les packages, représentés par des boites ainsi que les relations représentés par des flèches connectant les packages.

Un diagramme de paquetage se présente généralement comme suit :



***Figure 27 : formalisme du diagramme de paquetage***

***Figure 28 : Diagramme de paquetage de notre système***

### Le diagramme de packages de notre système

## CONCLUSION

Le dossier de conception ainsi élaboré nous a permis d’afficher l’image de notre système du point de vue statique. Cette partie a été consacrée à la modélisation de l’aspect statique du système, en se basant sur les spécifications détaillées des parties précédentes. Dans la prochaine partie, nous aborderons la réalisation ainsi que la description de quelques choix de techniques effectuées pendant le développement

- Dossier 5 : le dossier de réalisation ou de déploiement

Ce dossier représente la partie d’implémentation de la solution. C’est ici que sera présentée l’application et son architecture ainsi que les étapes de son déploiement.

1. OUTILS ET TECHNOLOGIES UTILISES
2. PRESENTATION DES L’ARCHITECTURE
3. DIAGRAMME DE COMPOSANT
4. DIAGRAMME DE DÉPLOIEMENT

INTRODUCTION

Le dossier de réalisation correspond à la phase de mise en œuvre de la solution. Il permet de décrire l’application en tant qu’instrument visible et manipulable. Il représente l’aboutissement de tous les dossiers précédents à savoir le dossier d’analyse et le dossier de conception. C’est donc une concrétisation de la solution proposée. Dans ce dossier nous allons donc présenter : les outils et technologies utilisées, l’architecture de notre système, le diagramme de composants suivi du diagramme de déploiement.

1. OUTILS ET TECHNOLOGIES UTILISES

Les outils que nous avons utilisés pour la réalisation de notre projet sont nombreux, pour différentes tâches et à des phases différentes. Il s’agit des langages et des logiciels de programmation.

1. Les logiciels utilisés

La réalisation d’une plateforme web requiert principalement les logiciels d’application suivants :

ϖ Un éditeur de texte qui est un logiciel d’édition des codes des programmes.

Exemples : Visual Studio Code, Sublime Text, Komodo, WebStorm, etc…

ϖ Un navigateur qui est un logiciel permettant de consulter les pages et les sites web.

Exemples : Google Chrome, Mozilla Firefox, Microsoft Edge, Brave, Safari, etc…

ϖ Un serveur local qui permet de faire tourner un site Internet sur son propre ordinateur et simule au mieux les conditions réelles d'un serveur d'hébergement web.

Exemples : Wamp Server, Xamp, EasyPHP, Caddy, etc…

ϖ Un logiciel de modélisation pour la conception des différents diagrammes de notre système.

Exemples : Star UML, Win’ Design, Power AMC, etc…

Nous avons récapitulé dans un tableau, tous les logiciels que nous avons utilisés avec les détails y afférents. Ce tableau se présente comme suit :

Tableau 17: Logiciels utilisés

|  |  |  |
| --- | --- | --- |
| Logiciel | Logo | Utilité |
|  |  |  |

1. Les langages utilisés

* Le HTML (Hyper Text Markup Language) Le html est le langage de base du web.

C’est un langage de balisage standard pour les documents conçus pour être affichés dans un navigateur Web. Ce langage nous a permis de structurer les contenus des pages de notre site web

* Le CSS (Cascading Style Sheet)
* Le JavaScript

1. PRESENTATION DES L’ARCHITECTURE

1- Architecture physique

Notre logiciel est une plateforme web structurée en trois (03) couches à savoir : la couche de présentation, la couche de traitement et la couche de donnée. Le noyau d’une application est constitué de la logique de présentation et la logique de traitement. C’est la répartition de ce noyau entre les différentes couchent qui détermine les architectures caractéristiques des systèmes client/serveurs, nous pouvons citer l’architecture 1-tiers, 2-tiers, 3-tiers. Notre plateforme sera déployée sur une architecture 3-tiers. Cette architecture peut être décrite par la figure ci-dessous :

1. Le client

Dans un réseau informatique, le client est le logiciel qui envoie des demandes à un serveur. Il peut s’agir d’un logiciel manipulé par une personne. Est appelé aussi client, l’ordinateur depuis lequel les demandes sont envoyées, équipé des logiciels relatifs aux différentes demandes par exemple, un navigateur web.

1. Le serveur d’application

Dans un réseau informatique, un serveur est à la fois un ensemble de logiciels et d’ordinateurs d’hébergement dont le rôle est de répondre de manière automatique à des demandes envoyées par des clients. Les serveurs sont l’usage courant des centres de traitements de données, les entreprises, les instructions, et le réseau internet, où ils sont souvent un point central et sont utilisés simultanément par de nombreux utilisateurs pour stocker, partager et échanger des informations.

1. Le serveur de base de données

C’est un Système de Gestion des Bases de Données qui contient les bases de données qui sont reliées à l’application. Le logiciel système client est remplacé par un logiciel système serveur dans une application ayant plusieurs données à traiter et plusieurs utilisateurs à se connecter.

1. Architecture logique

Pour notre système, il s’agit du MVC. Le MVC (Modèle Vue Contrôleur) est un motif d’architecture logicielle destiné aux interfaces graphiques et très populaires pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues, les contrôleurs.

1. Le modèle

Le modèle représente les données qui vont être utilisées dans l’application web. C’est ici que va être stockée la date, et tout ce qui permet de la modifier (getters, setters, etc..) que ce soit en local ou à distance.

1. La vue

La vue, c’est l’interface graphique de l’application. C’est via cet élément que vont se faire les interactions entre l’utilisateur et le code métier. Son but est de construire, à partir de ce que renvoie le serveur, une interface et de l’afficher à l’utilisateur.

1. le contrôleur

Le contrôleur est l’élément qui contient la logique métier. C’est ici que sont la plupart des algorithmes, calculs, etc. C’est aussi l’intermédiaire principal entre la vue et le modèle. Par exemple, la vue soumet un formulaire au contrôleur, qui gère sa validation via du code métier, et demande au modèle de faire des modifications dans la base de données.

Figure 36 : L'architecture MVC

1. DIAGRAMME DE COMPOSANT

Un diagramme de composants représente la structure physique et les dépendances entre composants d’un système logiciel. Il montre comment les composants sont assemblés pour former le système et comment ils interagissent entre eux et avec leur environnement. Le schéma ci-après représente notre diagramme de composants :

1. DIAGRAMME DE DÉPLOIEMENT

Le diagramme de déploiement représente la configuration matérielle et logicielle d’un système, ainsi que la manière dont les composants logiciels sont déployés sur cette configuration. 1- Formalisme d’un diagramme de déploiement

1. Le diagramme de déploiement de notre système

Pour notre système, le diagramme de déploiement se présente comme suit :

CONCLUSION

Le dossier de réalisation nous a permis de présenter le mode de déploiement de la solution. Il a donc été question dans cette partie de faire ressortir les technologies utilisées, montrer les différentes architectures, montrer les différents langages utilisés ainsi que le diagramme de déploiement qui modélise les composants matériels utilisés pour implémenter un système, et même également les associations entre composants. Cette section ainsi élaborée, nous allons procéder à la présentation du guide d’utilisation et d’installation de notre application.

- Dossier 6 : test de fonctionnalité

Il est conseillé de présenter ici les fonctionnalités essentielles de l’application développée.

- Dossier 7 : guide d’installation et guide d’utilisateur

Ce dossier représente la partie d’implémentation de la solution. C’est ici que sera présentée l’application et son architecture ainsi que les étapes de son déploiement.

APERÇU

I SPECIFICATIONS

II GUIDE D’INSTALLATION

III GUIDE D’UTILISATION

I SPECIFICATIONS

INTRODUCTION Le guide d’utilisation est un document que l’on élabore après avoir réalisé l’application. Ce manuel est important pour l’utilisateur car il renseigne les informations sur l’utilisation de l’application.

II GUIDE D’INSTALLATION

1- Ressources requises

1. Matériel requis

Un ordinateur portable, un ordinateur de bureau ou une tablette. Pour une meilleure expérience, votre appareil doit avoir de bonnes caractéristiques.

1. Système d’exploitation

Du fait que notre application soit une plateforme qui doit être hébergée sur un serveur en ligne, elle n’exige aucune compatibilité pour un système d’exploitation en particulier. La plateforme est donc consultable sur tout système d’exploitation au moyen d’un navigateur.

1. Navigateur web

Pour pouvoir jouir de façon optimale de toutes les fonctionnalités de la plateforme, nous vous recommandons fortement l’usage d’un navigateur internet stable et à jour

III GUIDE D’UTILISATION

Conclusion Générale

Annexes

- La bibliographie

- La webographie

- La lettre d’admission en stage signée par le chef d’entreprise

- Tout document ayant servi à la réalisation du projet

- Les documents annexes du cahier des charges dument (les fiches de stage)

- La table des matières