UNIVERSITE DE FIANARANTSOA

ECOLE NATIONALE D’INFORMATIQUE

MEMOIRE DE FIN D’ETUDE POUR L’OBTENTION DU DIPLOME

DE MASTER PROFESSIONNEL

Mention : Informatique

Parcours : Informatique générale

Intitulé

CONCEPTION ET REALISATION D4UNE PLATEFORM

DE LA CREATION DU CV EN LIGNE

Presenté le …

Par MAHITSILAZA Fabriola Davidson

Membre de jury :

President :

Examinateur :

Rapporteurs : Monsieur ANDRIAMBOLOLOMAHARAVAKA Bobarisoa, Ingenieur en Informatique

Année Universitaire 2022-2023

**CURRICULIUM VITAE**

**Etat civil**

**Nom :** MAHITSILAZA

**Prénom :** Fabriola Davidson

**Date et lieu de naissance :** 16 Décembre 1997 à Ambatofinandrahana

**Adresse :** Tsaramandroso Fianarantsoa

**Situation matrimoniale** **:** Célibataire

**Tel :**  0342279794

**Email :**  [mahitsilazafabrila@gmail.com](mailto:mahitsilazafabrila@gmail.com)

**Formation et diplôme :**

**2023 : Deuxieme année de formation en Master professionnel**

Parcours : Informatique Générale

ENI Fianarantsoa

**2022 : Premiere année de formation en Master professionnel**

Parcours : Informatique Générale

ENI Fianarantsoa

**2021 : Troisieme année de formation en Licence professionnel**

Parcours : Informatique Générale

ENI Fianarantsoa

**2020 : Deuxieme année de formation en Licence professionnel**

Parcours : Informatique Générale

ENI Fianarantsoa

**2019 : Premier année de formation en Licence professionnel**

Parcours : Informatique Générale

ENI Fianarantsoa

**2017 : Obtention du diplôme Baccalauréat Série D**

Lycée RAHERIVELORAMAMONJY Fianarantsoa

**Experience professionnelles et stage :**

**2023 :** Stage au sein de Hightao-mg

Theme : Conception et realisation d’une plateforme de creation du CV en ligne

Janvier mars 2023: “Application plateforme de recrutement des développeurs” Symfony 6 -> Maintenance de cette projet en cours, migration de php 7 a version 8 utilisation de docker, migration de symfony 5 vers symfony 6, Implémentation des fonctionnalités: de gestion des utilisateur authentification par réseaux sociaux comme google, LinkDin, Facebook, développement des fonctionnalités: utilisation de Cach Redis pendant le test QCM de candidature, traitement du résultat des réponses du candidature. Mise en place de l’API de communication avec le frontend. ReactJs ->Consommation de l’API REST; mise en page des affichages.Traitement du temps pendant le test

Theme : Application plateforme de recrutement des développeurs

Septembre 2022 - Desambre 2022 : “Application de gestion de Ressource Humaine en France RHPLUS Symfony 6 -> Implémentation des fonctionnalités: de gestion des utilisateur et de leurs authentification, Mise en place de l’API de communication avec le frontend. Sécurisation des données. Utilisation et adaptation de client du MAITRE DATA, Utilisation de pattern en structure hexagonale, assurer la maintenance du code par cette design pattern , API documentantion utilisation du bundle Nelmioapidoc, lancement sous docker. Déploiement et mise en ligne du projet sur l'infrastructure d'hébergement SCALINGO .

juin 2022 - Aout 2022 : “Application gestion de cours en ligne Symfony 5 -> Maintenance du code, développement des fonctionnalités: création des services pour faire une manipulation des images afficher en pdf et pdf en image, mise en forme des pdf avec l’image importer en automatique dans le CV des utilisateur dans cette application, Mise en place de l’API . Dans le partie admin utilisation de l’affichage en TWIG.

Mars 2022- Mai 2022 : “Application de gestion de bourse extérieure” Symfony 5 -> développement des fonctionnalités: de gestion des bourses; de gestion des utilisateurs et de leurs authentification; de notification par e-mail. Mise en place de l’API de communication avec le frontend. Sécurisation des données. Vue.js -> consommation des API pour la communication avec le backend, mise en page des pages de gestion des bourses, mise en place de la géolocalisation.

Janvier 2022 - Mars 2022 : “Application web de gestion d’une clinique” Symfony 4 -> Implémentation des fonctionnalités: de gestion des personnels; de gestion du planning des personnelles d’un service; de suivi des patients; de notification des patients; de gestion des stocks du clinique; de la facturation automatique; de transfert du patient vers d’autre clinique; d’exportation des données sur Excel. Déploiement et mise en ligne du projet sur l'infrastructure d'hébergement MADANET . Vue.js -> Consommation des API, mise en page: du calendrier; de la gestion des patients; de la gestion des stocks.

Octobre 2021- décembre 2021 : “Application de gestion de stock ” Node.JS et Express.JS -> Mise en place de l’API REST; conception et développement des fonctionnalités: de gestion des utilisateurs et des administrateurs, de gestion des mouvements de stock, du journal du vente, de facturation automatique, de notification; mise en place et utilisation du framework ORM SEQUELIZE pour la relation avec la base de donnée. Vue.JS ->Consommation de l’API REST; mise en page des affichages: de traitement du mouvement de stock, des histogrammes des bilans de sorties et d'entrées. Développement de la fonctionnalité d’exportation en pdf

Juillet 2021 - Septembre 2021 : “Application web de réservation en ligne Bus scolaire” PHP et Symfony 4 -> Mise en place d’un front-office de réservation en ligne. Développement d’un back-office de gestion des utilisateurs. Implémentation du module de notification par sms. Javascript -> Mise en page du frontoffice et du back-office ; exportation des données sous format Excel

Mars 2021 - mai 2021 : “Application de gestion de stock des Produits de Première Nécessité (PPN)” Symfony 4-> Implémentation des fonctionnalités: de gestion des utilisateurs; de gestion des stocks, de gestion des paiements. Mise en place d’un API Rest pour la communication avec le backend. Vue.js -> Consommation de l’API REST; mise en page des différentes pages de l’application

Mars 2020 à Juin 2020 : Stagiaire auprès de l’entreprise SNUT Madagascar Conception et développement d'une application web de gestion des dépistages du malaria. Technologies: Node.JS, Angular-cli9

Novembre 2018 à Janvier 2019 :Stagiaire auprès de l’entreprise CVB Madagascar Conception et développement d’une application interne de gestion de stock Technologies: Codeigniter 3, Jquery

**Connaissance en Informatique :**

**Langages de programmation :** C ,C++,JAVA ,PHP ,VB

**Technologies web :** HTML,CSS,PHP,Javascrpit,nodejs,JSP

**Framework :** Bootstrap,Angular ,Symfony,Vuejs,reactjs,expressjs,CI

**SBGD:**  Microsoft SQL , Serveur, Mysql,Oracle

**Installation et configuration des serveur** : CLUSTER ,

**Méthodes de conception** : Merise et 2TUP

**Langage de modélisation :** UML

**Outil bureautique :** Traitement de text(word,excel,Power Point)

**System d’exploitation :** Windows, Linux

**Connaissance linguistiques :**

Grille d’évaluation : Trés bonne(TB), Bonne(B), Assez Bonne(AB)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Comprendre à l’audition | Lire | Ecrire | Parler |
| Français | AB | B | B | AB |
| Anglais | AB | B | B | AB |

**Autre et divers :**

Sport : Basketball, Football

Loisirs : Lecture, Internet

**Chapitre 2** : **Présentation de l’entreprise Hightao-mg**

Cette partie présentera l’entreprise Hightao-mg pour avoir un aperçu de l’environnement au sein duquel s’est déroulé le stage

1. **Fiche d’identification**

Les renseignements généraux qui concernent Hightao-mg :

* Siège : VR 6 BG Ambohidreny Mahazoarivo Antananarivo 101 MADAGASCAR
* Dirigeant: Mme RAZAFINDRAFARA Gabrielle
* Tél: +261 34 27 344 54
* Site web : [www.hightao-mg.com](http://www.hightao-mg.com)
* Courriel: [rakoto@hightao-mg.com](mailto:rakoto@hightao-mg.com)

2.**Breve historique :**

Hightao-mg anciennement Hightao est une entreprise malgache de développement Web créer en juillet 2011.

Le nom « hightao » est issu de la contraction des deux clés de l’entreprise :

* « haitao » pour savoir – faire
* High du mot high tech

A sa création, l’entreprise se nommait Hightao et était composée de ces 3 membres fondation. Les membres fondateurs se connaissaient à travers une association d’informatique « Tanora mijoro ». Avant déjà travaillé ensemble, la création de l’entreprise à été alors assez rapide et le cadre du travail assez productif. Quelques mois plus tard, un développeur backend et un designer UX issus de la mémé association ont rejoint l’entreprise.

Dès ces débuts, l’entreprise à travailler en « assistance technique » pour des entreprises en France. Ce cœur de métier a permis à l’entreprise d’évoluer rapidement et d’être en quête de développeur : autonome, rigoureux et talentueux. La recherche de nouveau collaborateur avec ces caractéristiques s’est avéré difficile : d’où la décision d’ajouter une partie formation dans l’activité de l’entreprise. Ainsi l’entreprise, des ses débute à allouer un budget de formation pour les développeurs de son équipe.

En Janvier 2020, suite à la volonté des membre fondateurs de réaliser de nouveaux projets,l’entreprise à été reprise par notre directrice actuelle et a depuis changé de nom en hightao-mg. Le cœur de métier de l’entreprise est resté et ses valeur restent les mémes.

1. **Objectif**

**3.1 Missions**

L’entreprise hightao-mg est une entreprise de développement web. Notre mission principale est ainsi de produire des application web : fonctionnelle est orientées utilisateurs .

Etant une entreprise de service, nous avons comme besoin de nos développeurs. Ainsi une de nos missions est de mettre en place un environnement de travail propice au bien etre des développeurs.

**3.2 Culture d’entreprise et valeurs**

Au sein de l’entreprise, nous mettons en place un cadre de travail permettant a chaque collaborateur :

1.D’etre autonome : Chaque collaborateur est responsable de ses taches et de son planning. Il est aussi garant de sa communication interne et surtout vers le client. Le responsable d’exploitation a comme mission d’être facilitateur de cette communication entre le client et le collaborateur, mais ce dernier reste le principal responsable

2. D’être innovant : chaque collaborateur en tant « qu’assistant technique « de nos clients doit fournir des solutions répondant à leurs besoins. Ainsi, nos collaborateurs sont amenés à trouver des solutions innovantes tous les jours soit en utilisant leurs connaissances et leurs expériences, soit en « échangeant entre collaborateurs. La culture du dialogue est ainsi très importante au sein de l’entreprise.

3. D’avoir le droit à l’erreur : pour permettre à chaque collaborateur d’être autonome et d’être innovant, on préconise la prise d’initiative et donc le risque de faire des erreurs. Les erreurs sont ainsi acceptées et ne sont surtout pas sanctionnées. Il est par contre demande a chaque collaborateur, d’analyser son erreur et de ne pas la refaire.

4. De se former : pour mieux appréhender son travail, chaque collaborateur de l’entreprise est invité a se former sur différents sujets : techniques, organisationnelles. L’entreprise permet ainsi aux collaborateurs de consacrer du temps à se former mais aussi participe a l’abonnement sur les sites de formation en ligne. Nous mettons aussi en place de stages encadrés par nos développeurs.

**3.3. Activités**

Notre cœur de métier est le développement web pour des entreprises internationales ou offshore. Nous réalisons un certain nombre d’applications :

* D’application de vente e-commerce
* D’application d’intranet de gestion d’entreprise ou de comité d’entreprise
* D’application dédiée : de recherche d’emploi, de visioconférence, etc. …

Plus en détail , l’entreprise effectue :

* L’analyse des besoins ;
* Les spécifications et la conception de base de données ;
* La conception graphique orientée utilisateur ;
* Le développement de l’application ;
* Le déploiement ;
* La maintenance et le support ;

Le tableaux 4 suivant représente les différentes technologies utilisées par l’entreprise

|  |  |
| --- | --- |
| Web Back-end | Symfony , Laravel |
| Web Front-end | React,Angular,Vuejs |
| CMS et e-commerce | Prestashop, Wordpress |
| Technologies mobile | Android,IOS, et PWA, (Progressive Web Apps) |
| Base de donnée | Mysql, Sqlite,PostgressSQL,SQL,Server,MongoDB |

**2.4 Organigramme**

L’organisation au sein de notre entreprise est destinée à être le plus horizontale possible. Sur les 12 personne composant l’entreprise, 10 sont développeurs.

**DIRECTRICE**

Développeur backend

Stagiaire   
dev backend

Développeur frontend

Stagiaire   
dev frontend

Développeur  
fullstack CMS

Design UI :UX

Responsable

financière

Responsable exploitation

**Chapitre3 . Description du projet**

La description du projet est un ensemble d’actions à réaliser pour satisfaire un objectif défini, dans le cadre d’une mission précise, ayant un début et une fin nécessitant la mise en œuvre de ressource humaine et matérielles pour la réalisation .

Ce chapitre décrit de façon détaillée de ce projet : les formulation, les objectifs et besoins de l’utilisateur , les moyens nécessaires pour la réalisation du projet et les résultats attendus

**3.1. Formulation**

Depuis que les CV bien structurée avec templete adorée nous avons toujours eu une difficulté pour faire ca en manier rapide et mettre en place cette forme. Non seulement au niveau de déplacement, mais aussi au niveau de la sécurisation du stockage de cette CV. Face à ce problème nous avons décidé de mettre en place une application web pour le résoudre.

**3.2.Objectif et besoins d’utilisateur**

L’objectif de ce projet est de concevoir et mettre en place une plateforme pour la gestion de la CV pour rendre vos CV en meilleur vue et rapide.

Le besoin de l’utilisateur se résume comme suit :

* Gestion utilisateur
* Gestion client
* Gestion de payement
* Gestion de facturation

**3.3.Moyen nécessaire du projet**

Pour satisfaire le besoin attendus, l’application de ce projet dans le service incite la mise en place d’un nouveau système de traitement , surtout sur le plan matériel.

**3.3.1 Moyen humains**

Les moyen humain sont l’ensemble des équipes pour la conception et réalisation du projet

* Développeur : chargées du développement et du déploiement
* Encadreur professionnel

**3.3.2 Moyen matériel**

La configuration de machine que nous avons utilisé pour développer cette application est dréssée dans le tableau ci-dessous

|  |  |  |  |
| --- | --- | --- | --- |
| Marque | Processeur | Mémoire | Disque dur |
| ASUS ROG | Core™ i7-7700HQ | 16 Go | 1To(hdd) et 250Go(ssd) |

**3.3.3 Moyen logiciel**

Plusieurs logiciel ont été utilisée pour la conception et la réalisation de ce projet.

|  |  |  |
| --- | --- | --- |
| Designation | Caracteristique/Description | Commentaire |
| Visual paradigme | Version 8 | Outils de moderation |
| Visual studio code | Version 3.2 | Pour le developpement de l’application |
| Docker et docker composer | Version 20.10.21 et v2.13.0 | Deplyement dans le container |
| Chrome | 106.0.5249.103 | Navigateur |

**3.4 Résultats attendus**

En utilisant cette application, la division bénéficie des fonctionnelités suivantes

* Traitement de la mise a jour et enregistrement des iinformation dans la base sont rapide, fiables, et efficaces.
* Normalisation de la donnée
* Facilitée de mettre le theme de la couleur de la CV
* Facilite de mettre en choix de la templete de la cv de la forme
* Integration automatique de information déjà existe sur facebook ou google

A la fin de ce projet, l’application doit etre réalisée et que l’application soit directement fonctionnelle.

**3.5 Chronogramme de travaux**

Pour bien organiser les tache a realiser ,nous avons suivi le planning represente par le calendrier dans la figure ci-dessous

**Sary eto**

Parti II : ANALYSE ET CONCEPTION

**Chapitre 4 : Analyse préalable**

L’analyse préalable consiste à analyser la situation actuelle pour étudier la qualité et la faiblesse afin de pouvoir corriger les problemes et d’apporter une application pour le futur système en function des critiques formulée sur l’existant

**4.1.Analyse de l’éxistant**

L’analyse consiste à collecter des données indispensables à la création de l’application ,il consiste aussi à déterminer est faisable ou non , c’est-à-dire si la création de cette application apportera du benefice pour la societe

**4.1.1 Organisation actuelle**

* **Mode de traitement :**

Tous le traitement sont effectués manuellement en cas de modifications ou de nouvelle entrée

* **Mode de travail :**

Tous les CV sont telecharger automatique apres la payement correct vie stripe

* **Personne impliquée :**

Tous le personne faire une payement doit etre avoir une numero de facture enregistrer sur l’admin de ces application au cas ou le CV est disparus ou verification de la facturation.

**4.1.2.1 Inventaire de moyen materiel**

-Outil de communication : mail

**4.2 Critique de l’existant**

La critique de l’existant consiste à recenser tous les point forts et les points faibles du système actuel

1. **Point faible du système actuel**

* Le theme de la cv sont créer manuelement
* Les information sont placée manuellement
* Les couleur de la CV manuel
* Mise en place de la photo CV sont en manuel

1. **Point fort de la système actuel**

* Choix de la theme existe
* Choix de la couleur de cette theme
* Upload automatique de la photos de la CV
* Information bien structure a chaque etape

**4.3 Conception avant projet**

**4.3.1 Solution proposée**

Plusieur probleme persistent, sutout l’utilisation de l’office word pour faire enregistrer les information pour faire personnaliser les information sans template,mais jusqu’à present il n’y a aucun autre moyen de regler tout ca, nous avons proposé une solution pour le resoudre ; c’est de concevoir et réaliser une plateform pour la gestion de création de la CV personnalisée en ligne.

**4.3.1 Solution proposée**

**4.3.3.1 choix de la méthode de conception**

Dans cette partie nous allons comparer le methode Merise et 2TUP pour le method de conception

Le tableau

|  |  |
| --- | --- |
| MERISE | 2TUP |
| MERISE est une méthode systémique d'analyse et de conception de systèmes d'information. C'est-à-dire qu'elle utilise une approche systémique. | Le 2TUP (2 Track Unified Process) est un processus de développement logiciel qui implémente le Processus Unifié. Le 2TUP propose un cycle de développement en Y qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu’échangent les acteurs et le système, à produire le cahier de charges et à modéliser le contexte. |
| MERISE propose de considérer le système réel selon deux points de vue: - Une vue statique (données) - Une vue dynamique (traitements). | Le 2TUP utilise le langage UML qui 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 à 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. |

Nous allons choisir le 2TUP comme méthode pour les fonctionnalités qu’elle apporte qui sont vraiment avancé par rapport à Merise

**Présentation 2TUP et UML [2]**

Pour modéliser d’une manière claire et précise la structure et le comportement de notre système indépendamment de tout langage de programmation, nous allons adopter la démarche 2TUP et le langage de modélisation UML (Unified Modeling Language).

* Présentation de 2TUP

2TUP signifie «2 Track Unified Process». Le 2TUP est un processus proposé par Valtech. 2TUP est un processus de développement logiciel qui met en œuvre la méthode du processus Unifié. Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise.

Un processus unifié est un processus construit sur UML (Unified Modeling Language). Plus exactement, ce sont les meilleures pratiques du développement objet suivies pour la réalisation d’un système.

Un processus unifié se distingue par les caractéristiques suivantes :

* + Itératif : Le logiciel nécessite une compréhension progressive du problème à travers des raffinements successifs et développer une solution effective de façon incrémentale par des itérations multiples.
  + Piloté par les risques : les causes majeures d’échec d’un projet logiciel doivent être écartées en priorité.
  + Centré sur l’architecture : le choix de l’architecture logicielle est effectué lors des premières phases de développement du logiciel. La conception des composants du système est basée sur ce choix
  + Conduit par les cas d’utilisation : le processus est orienté par les besoins utilisateurs présentés par des cas d’utilisation.

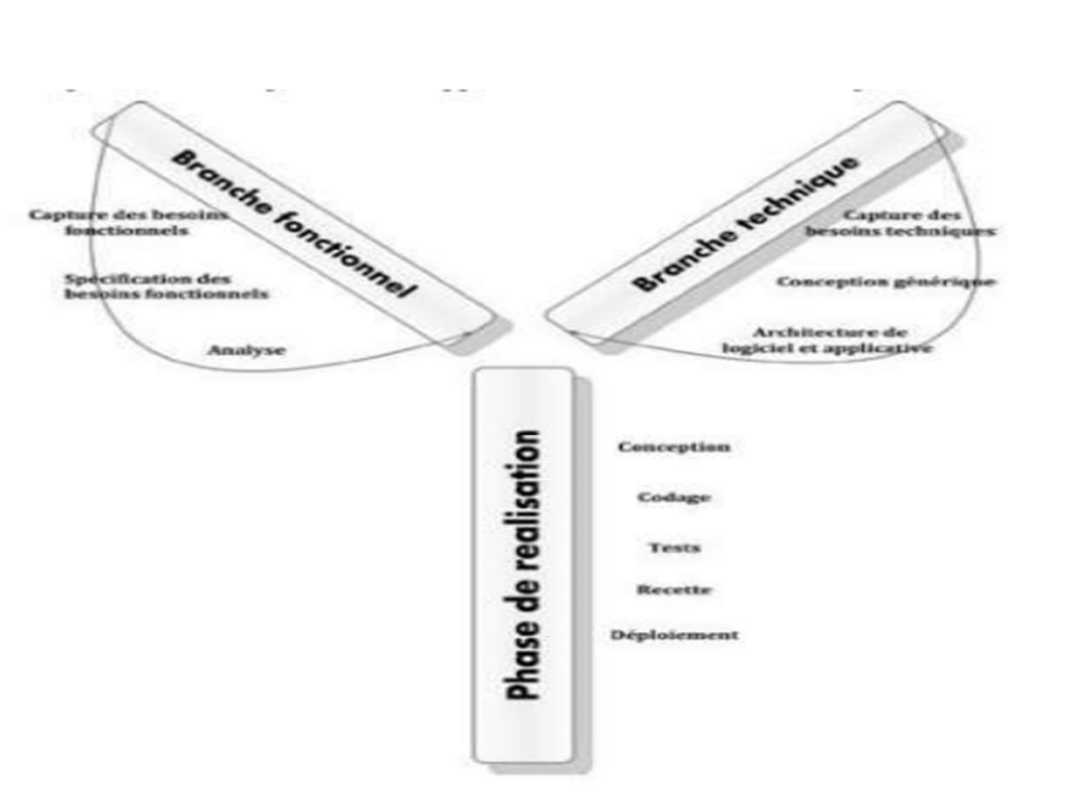
Dans la communauté objet et nouvelles technologies, il existe plusieurs processus unifiés en vogue comme eXtreme Programming (XP) et Rational Unified Process (RUP). Dans notre étude, on a choisi de travailler avec le processus 2TUP parce qu’il cible des projets de toute taille, et il a pu faire une large place dans le domaine de la technologie et les risques des projets.

Le 2TUP propose un cycle de développement qui sépare les aspects techniques des aspects fonctionnels et propose une étude parallèle de deux branches : celle fonctionnelle (étude de l’application) et celle technique (étude de l’implémentation).

Illustré dans Figure 2 le processus 2TUP s’articule autour de trois branches :

* Une branche technique
* Une branche fonctionnelle

Une branche fonctionnelle



Les principales étapes de la branche fonctionnelle se présentent comme suit :

* L’étape capture des besoins fonctionnels : Cette phase a pour objectif de définir la frontière fonctionnelle entre le système et son environnement. Les activités attendues des différents utilisateurs par rapport au système.
* L’étape d’analyse : consiste à étudier, précisément, les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en terme de métier.

La Branche technique : capitalise un savoir-faire technique et/ou des contraintes techniques. Elle constitue un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent l’être, en effet, indépendamment des fonctions à réaliser.

Les principales étapes de la branche technique se présentent comme suit :

* L’étape capture des besoins techniques : Cette étape recense toutes les contraintes sur les choix de technologies pour la conception du système. Les outils et le matériel sélectionnés ainsi que la prise en compte des contraintes d’intégration avec l’existant (pré requis d’architecture technique).
* L’étape conception générique : Définit les composants nécessaires à la construction de l’architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle permet de générer le modèle de conception technique qui définit le Framework.

La Phase conception et réalisation : consiste à réunir les deux branches, permettant démener une conception applicative et enfin la livraison d'une solution adaptée aux besoins.

Cette fusion conduit à l’obtention d’un processus en forme de Y.

Les principales étapes de cette branche se présentent comme suit :

* L’étape conception préliminaire: Cette étape permet de produire le modèle de conception système.

Ce dernier organise le système en composants, délivrant les services techniques et fonctionnels ; ce qui induit le regroupement des informations des branches technique et fonctionnelle.

* L’étape conception détaillée: permet d’étudier comment réaliser chaque composant. Le résultat fournit l’image prête {fabriquer du système complet
* L’étape de codage: permet d’effectuer la production des composants et les tests des unités de code au fur et à mesure de leur réalisation.
* L’étape de recette: consiste à valider les fonctionnalités du système développé.

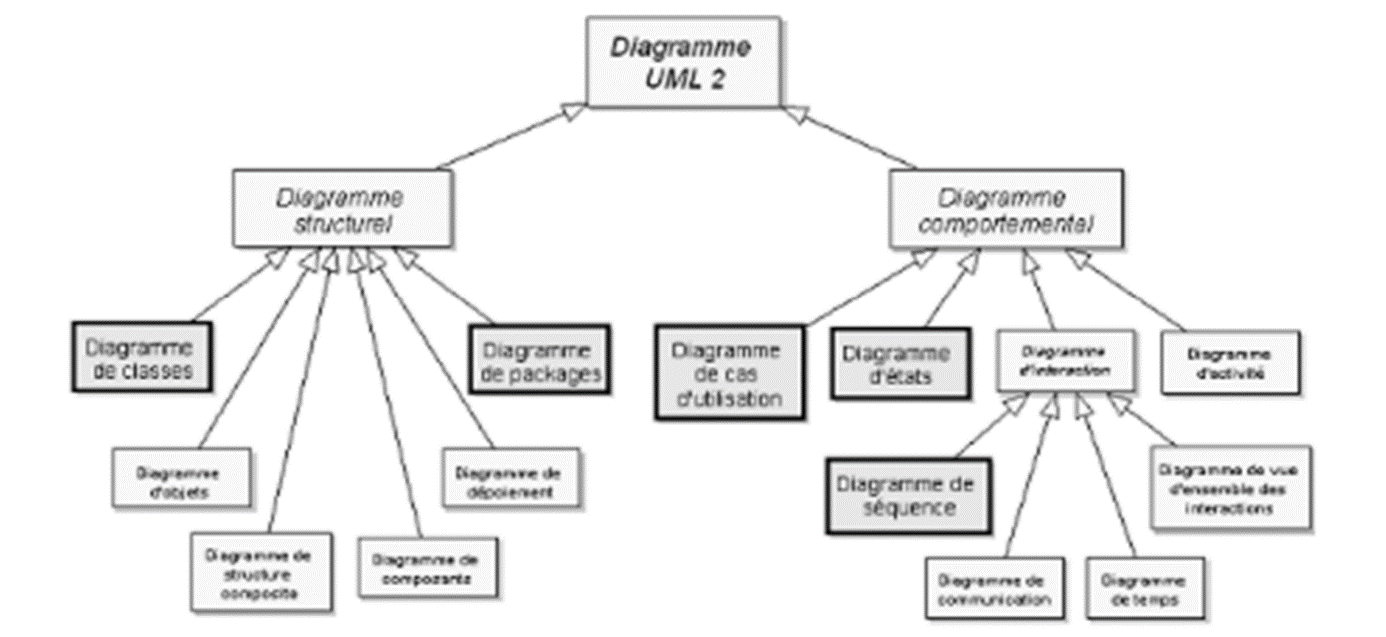
**Présentation UML**

**Définition**:

.UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire les besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML Représente le standard de modélisation objet le plus répandu et le plus utilisé aujourd’hui.

UML résulte de l’unification de techniques ayant fait leurs preuves pour l’analyse et la conception de grands logiciels et de systèmes complexes.

* UML est une norme Il est nécessaire qu'une méthode objet soit définie de manière rigoureuse et unique afin de lever les ambiguïtés. De nombreuses méthodes objet ont été définies, mais aucune n'a su s'imposer en raison du manque de standardisation. C'est pourquoi l'ensemble des acteurs du monde informatique a fondé en 1989 l’OMG (Object Management Group), une organisation à but non lucratif, dont le but est de mettre au point des standards garantissant la compatibilité entre des applications programmées à l'aide de langages objet et fonctionnant sur des réseaux hétérogènes (de différents types). A partir de 1997, UML est devenue une norme de l'OMG, ce qui lui a permis de s'imposer en tant que méthode de développement objet et être reconnue et utilisée par de nombreuses entreprises.
* UML est un langage de modélisation objet UML comble une lacune importante des technologies objet, il permet d’exprimer, d’élaborer et de modéliser au sens de la théorie des langages ; de ce fait, il contient les éléments constitutifs de ce derniers: concepts, une syntaxe et une sémantique
* UML décrit un méta modèle La puissance et l’intérêt d’UML est qu’il normalise la sémantique des concepts qu’il véhicule, il repose sur un méta modèle pour permettre à n’importe qui de déchiffrer son intention de manière non équivoque, il est donc primordiale de s’accorder sur la sémantique des éléments de modélisation, bien avant de s’intéresser à la manière de les présenter.
* Les diagrammes UML La notation UML s’articule autour de treize types de diagrammes la représentation des concepts particuliers d’un Ces types de diagrammes sont répartis en deux grands groupes et comportementaux



Diagrammes structurels

-Diagramme de classe : class diagram Les diagrammes de classes expriment de manière générale la structure statique d’un système, en termes de classes et de relations entre ses classes. Outre les classes, ils représentent un ensemble d’interfaces et de paquetages, ainsi que leurs relations.

Les diagrammes de classes contiennent généralement les éléments suivant :

Les classes : Une classe est la description d’un ensemble d’objet qui partage les mêmes attributs, les mêmes opérations, les mêmes relations et la même sémantique. Une classe est symbolisée par un rectangle.

La notation UML s’articule autour de treize types de diagrammes ; chacun d’eux étant dédié la représentation des concepts particuliers d’un Système logiciel.

Ces types de diagrammes sont répartis en deux grands groupes : les Diagrammes structurels

Attribut : Un attribut est une propriété nommée d’une classe qui décrit un ensemble de valeurs que les instances de cette propriété peuvent prendre. Une classe peut ne pas avoir, comme elle peut avoir un ou plusieurs attributs.

Opération : Une opération est une abstraction de ce que peut réaliser un objet et qui est réalisable par tous les objets de la classe. Une classe peut ne pas avoir comme elle peut avoir plusieurs opérations.

Les relations d’association d’agrégation et de composition Une association : représente une relation sémantique durable entre deux classes. Une agrégation est un particulier d’association non symétrique exprimant une relation de contenance. Une composition est une agrégation plus forte

-Diagramme d’objets : Object diagram Les diagrammes d’objets servent, d’une part à inventorier les objets composant une application à un instant donné ainsi que les relations, d’autre part à donner une image statique des relations entre ces objets. Ils peuvent également être mis en œuvre pour tester la pertinence d’un diagramme de classe.

-Diagramme de composant : component diagram Les diagrammes de composants servent à représenter la configuration logicielle ainsi que les relations d’un système, en permettant également de représenter les programmes, les sous programmes et les interrelations.

-Diagramme de déploiement : deployment diagram Les diagrammes de déploiement représentent un ensemble de nœuds ainsi que leurs relations. On les utilise pour illustrer la vue de déploiement statique d’une architecture. Les diagrammes de déploiement sont apparentés aux diagrammes de composant, car un nœud englobe généralement un ou plusieurs composants

Diagramme comportementaux

-Diagramme de cas d’utilisation : use case diagram Les diagrammes de cas d’utilisation représentent un ensemble de cas d’utilisation, d’acteurs et leurs relations. Ils représentent la vue statique des cas d’utilisation d’un système et sont particulièrement importants dans l’organisation et la modélisation des comportements d’u système.

Les cas d’utilisation :

-Diagramme d’états-transitions : State machine diagram Ils ont pour rôle de représenter les traitements (opération) qui vont gérer le domaine étudié. Ils définissent l’enchainement de l’état de classe et font donc apparaître l’ordonnancement des travaux. Le diagramme d’états-transition est associé à une classe pour laquelle on gère différents états : il permet de représenter tous les états possibles ainsi que les événements qui provoquent les changements d’états.

- Diagramme d’Interaction :

* Diagramme de communication: représentation spatiale des interactions entre objets ;
* Diagramme de temps : utilisé pour explorer le comportement des objets d'un système à travers une période de temps. Il est destiné à l’analyse et la conception de systèmes ayant des contraintes temps-réel. Il s’agit là de décrire les interactions entre objets avec des contraintes temporelles forte
* Diagramme de vue d’ensemble d’interaction : permet de donner une vue d’ensemble des interactions du système. Il est réalisé avec le même graphisme que le diagramme d’activité. Chaque élément du diagramme peut ensuite être détaillé à l’aide d’un diagramme de séquence ou d’un diagramme d’activité.
* Diagramme de séquence : Séquence diagram

Un diagramme de séquence met en évidence le classement des messages par ordre chronologique. En général, on place l’objet qui débute l’interaction à gauche, puis on continue en progressant vers la droite, les objets les plus subordonnés étant tout à fait à droite. On place ensuite les messages envoyés et reçus par ces objets le long de l’axe des ordonnées, par ordre chronologique, du haut vers le bas. Cela donne au lecteur une indication visuelle claire du flot de contrôle dans le temps.

-Diagramme d’activité : activity diagram

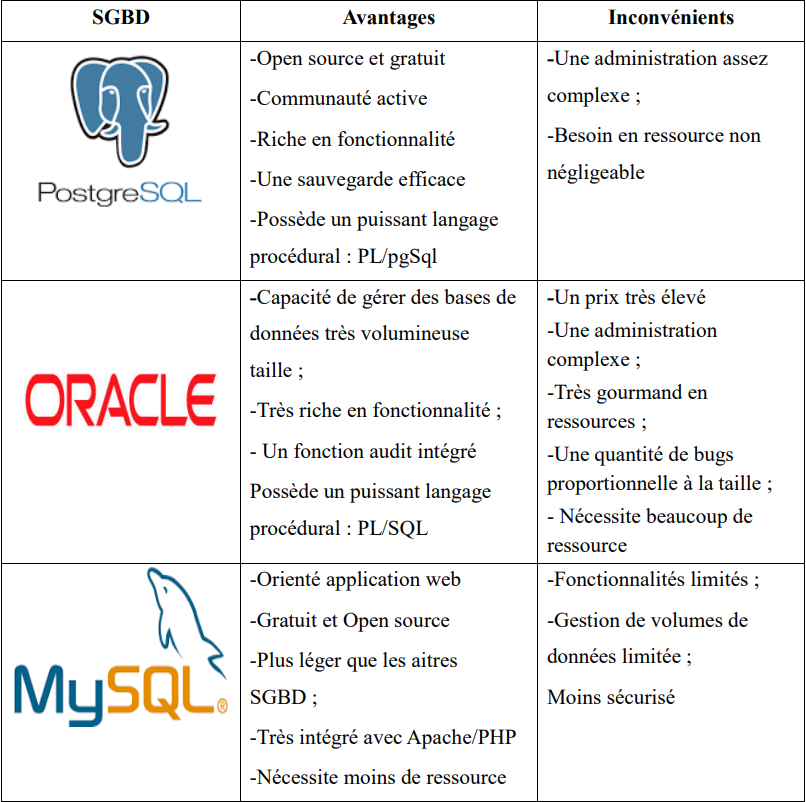
Le diagramme d'activité est attaché à une catégorie de classe et décrit le déroulement des activités de cette catégorie. Le déroulement s'appelle "flot de contrôle". Il indique la part prise par chaque objet dans l'exécution d'un travail. Il sera enrichi par les conditions de Séquence.

La modélisation objet consiste à créer une représentation graphique des concepts abstraits, en définissant des vues qui permettent de décrire les principaux aspects du système objet, dans le but de faciliter l’analyse et la conception. Nous avons adopté ce langage de notation pour la conception des diagrammes. De ce fait, elle permet de couvrir le cycle de vie d’un logiciel depuis l’analyse des besoins jusqu’au codage.

**4.3.3.1 choix de SGBD**

De nombreux SGBD sont disponibles sur le marché, partant des SGBD gratuits jusqu’aux destinés spécialement aux professionnels, comportant de plus nombreuses fonctionnalités, mais plus coûteux

Le tableau 7 montre la comparaison sur MySQL, SQL Server, Oracle [8] [9]



Selon l’étude comparative qu’on a vue précédemment, on a choisi MySQL par rapport aux avantages qu’elle apporte

-**Présentation de MYSQL**

MySQL est un système de gestion de base de données relationnelle (SGBDR) basé sur SQL (Structured Query Language).

MySQL fonctionne sur pratiquement toutes les plates-formes, y compris Linux, Unix et Windows. Il est entièrement multi-thread avec un noyau de threads, et fournit des API (Application Programming Interface) pour de nombreux langages de programmation, notamment C, C + +, Eiffel, Java, Perl, PHP, Python, et Tcl.

MySQL est utilisé dans une large gamme d'applications, Le commerce électronique, les bases de données Web

Selon MySQL AB, avec plus de dix millions de serveurs MySQL installés dans le monde entier, MySQL est devenu le leader mondial du marché des Bases de Données. MySQL compte des clients prestigieux comme Google, la NASA ou Suzuki..

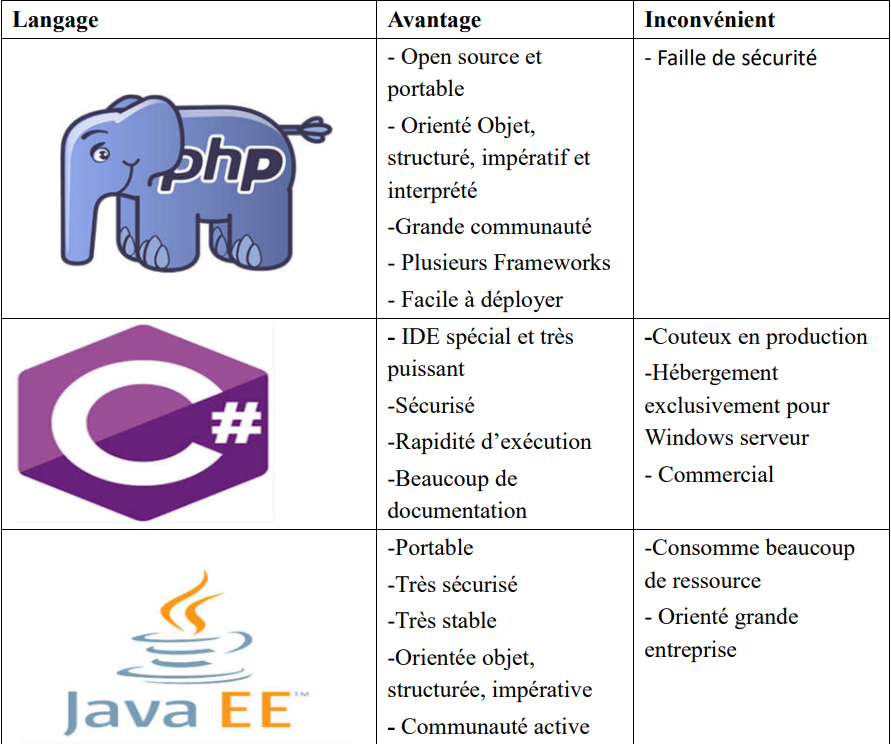
MySQL bénéficie d'un large public, car:

* Il est facile à comprendre:Sa syntaxe simple en fait un langage facile à comprendre pour les programmeurs et des débutants
* Le Langage est fonctionnel: MySQL fonctionne sur de nombreuses plates-formes différentes.
* Dispose d'une vaste bibliothèque de fonctions et d'API: API pour C, C + +, Eiffel, Java, Perl, PHP, Python, Ruby et Tcl sont disponibles. Les fonctions SQL sont mises en place en utilisant une bibliothèque de classes optimisées
* Multi Thread : Complètement multi-thread utilisant un noyau de threads.
* Haute capacité de storage: Pour vous donner une idée: De grosses entreprises actuelles utilisent le serveur MySQL avec plus de 100 000 tables et 1 000 000 000 d'enregistrements.

**4.3.3.3 Choix du langage de programmation**

Comme on avait choisi une méthode de développement orientée objet et SGBD, cette approche influe aussi sur le choix du langage à adopter. On peut rajouter quelques-uns à savoir la portabilité la facilité, la multidisciplinarité et pas mal d’autres comme la sécurité. Dans cette section, nous allons comparer de la manière la plus objective possible les quatre langages de programmations suivantes pour la réalisation de notre application : PHP, C#, JEE, JavaScript

Le tableau 8 représente la comparaison entre PHP, C#, JEE [7}



Les comparaisons faites ci-dessus nous permettent de choisir le langage de programmation PHP et on utilisons aussi JavaScript comme langage d’implémentation de notre application.

-**Présentation de langage PHP [4]**

PHP est un langage de script utilisé le plus souvent côté serveur : dans cette architecture, le serveur interprète le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS par exemple) et des données (JPEG, GIF, PNG par exemple) pouvant être interprétés et rendus par un navigateur web. PHP peut également générer d'autres formats comme le WML, le SVG et le PDF.

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le Web. PHP est le plus souvent couplé à un serveur Apache bien qu'il puisse être installé sur la plupart des serveurs HTTP tels que IIS ou nginx. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.

C'est un langage peu typé et souple et donc facile à apprendre par un débutant, mais, de ce fait, des failles de sécurité peuvent rapidement apparaître dans les applications. Pragmatique, PHP ne s'encombre pas de théorie et a tendance à choisir le chemin le plus direct. Néanmoins, le nom des fonctions (ainsi que le passage des arguments) ne respecte pas toujours une logique uniforme, ce qui peut être préjudiciable à l'apprentissage.

Son utilisation commence avec le traitement des formulaires puis par l'accès aux bases de données. L'accès aux bases de données est aisé une fois l'installation des modules correspondants effectuée sur le serveur. La force la plus évidente de ce langage est qu'il a permis au fil du temps la résolution aisée de problèmes autrefois compliqués et est devenu par conséquent un composant incontournable des offres d'hébergements.

Il est multi-plateforme : autant sur Linux qu'avec Windows il permet aisément de reconduire le même code sur un environnement à peu près semblable (quoiqu'il faille prendre en compte les règles d'arborescences de répertoires, qui peuvent changer).

Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite comme tout langage de programmation une bonne compréhension des principales fonctions usuelles ainsi qu'une connaissance aiguë des problèmes de sécurité liés à ce langage.

La version 5.3 a introduit de nombreuses fonctions nouvelles : les espaces de noms (Namespace) — un élément fondamental de l'élaboration d'extensions, de bibliothèques et de frameworks structurés, les fonctions anonymes, les fermetures, etc.

En 2018, près de 80 % des sites web utilisent le langage PHP sous ses différentes versions37.

Le langage PHP fait l'objet, depuis plusieurs années maintenant, de rassemblements nationaux organisés par l'AFUP38 (l'Association française des Utilisateurs de PHP), où experts de la programmation et du milieu se retrouvent pour échanger autour du PHP et de ses développeurs. L'association organise ainsi deux évènements39 majeurs : le « Forum PHP », habituellement en fin d'année, et les « AFUP Day », qui ont lieu au cours du premier semestre, simultanément dans plusieurs villes.

**-Présentation de JavaScript [10]**

JavaScript est un langage de script, multi-plateforme et orienté objet. C'est un langage léger qui doit faire partie d'un environnement hôte (un navigateur web par exemple) pour qu'il puisse être utilisé sur les objets de cet environnement

JavaScript contient une bibliothèque standard d'objets tels que Array, Date, et Math, ainsi qu'un ensemble d'éléments de langage tels que les opérateurs, les structures de contrôles et les instructions. Ces fonctionnalités centrales et natives de JavaScript peuvent être étendues de plusieurs façons en fournissant d'autres objets, par exemple :

* JavaScript côté client étend ces éléments de base en fournissant des objets pour contrôler le navigateur et le Document Object Model (DOM). Par exemple, les extensions du langage côté client permettent de placer des éléments dans un formulaire HTML, de réagir aux événements déclenchés par l'utilisateur (les clics, la saisie d'un formulaire, les actions de navigation, etc.).
* JavaScript côte serveur étend ces éléments de base avec des objets utiles pour le fonctionnement sur un serveur tels que la possibilité de communiquer avec une base de données, manipuler des fichiers, passer d'une application à une autre, etc.

**4.3.3.4 Choix de Framework**

En programmation informatique, un Framework désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d’une partie d'un logiciel (architecture). Nous allons voir dans une étude comparative les Framework PHP comme Symfony et Laravel.

Le tableau 9 représente l’étude comparative des Framework PHP.[6]



Pour ce projet nous avons choisi Symfony, de plus il est rapide à mettre en place, car sa documentation d’aide est prête à l’emploi. Et il possède aussi beaucoup de bundle de pour faciliter la tâche.

-**Présentation du Framework Symfony**

Symfony est un ensemble de composants PHP ainsi qu'un Framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d’accélérer le développement d'un site web

**Histoire**

L'agence web française SensioLabs est à l'origine du framework Sensio Framework. À force de toujours recréer les mêmes fonctionnalités de gestion d'utilisateurs, gestion ORM, etc., elle a développé ce Framework pour ses propres besoins3 . Comme ces problématiques étaient souvent les mêmes pour d'autres développeurs, le code a été par la suite partagé avec la communauté des développeurs PHP.

Le projet est alors devenu Symfony (conformément à la volonté du créateur de conserver les initiales S et F de Sensio Framework), puis Symfony2 à partir de la version 24 . La version 2 de Symfony casse la compatibilité avec la branche 1.x. À partir de la version 2, les ruptures de compatibilité entre les versions sont documentées pour faciliter les montées de version.

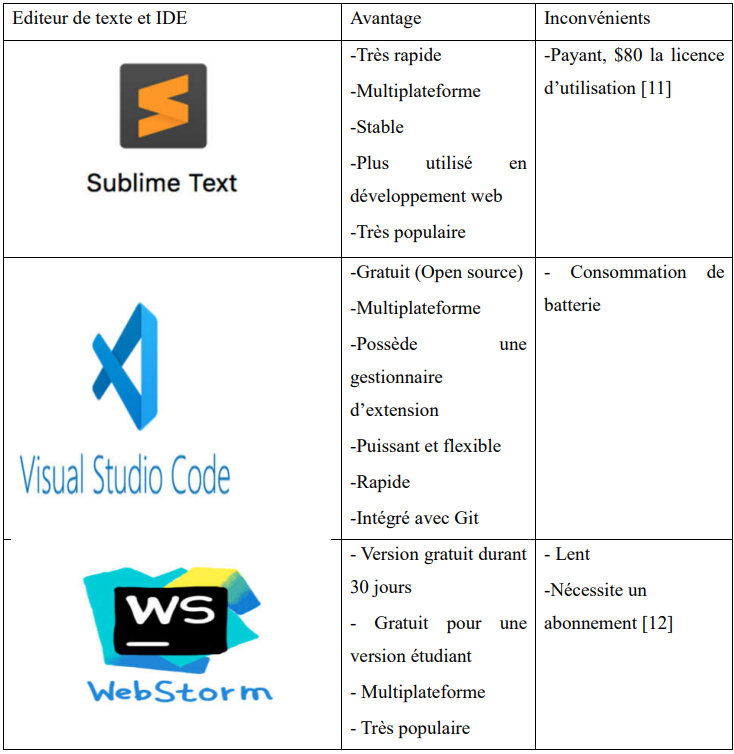
**Utilisations**

Symfony est utilisé par Askeet et beaucoup d'autres applications6 , dont Delicious7 et Yahoo! Bookmarks8 . En 2009, Dailymotion a porté son code sur Symfony et a continué la transition9 .

Les projets les plus connus utilisant Symfony sont eZ Publish (utilisation de Symfony 3 en full stack10), Drupal depuis la version 8 (utilisation de composants Symfony), phpBB, Laravel, Joomla!, Composer, Magento, Piwik11, PrestaShop depuis la version 1.7 (utilisation de composants Symfony).

**4.3.3.5 Choix de l’outil de développement**

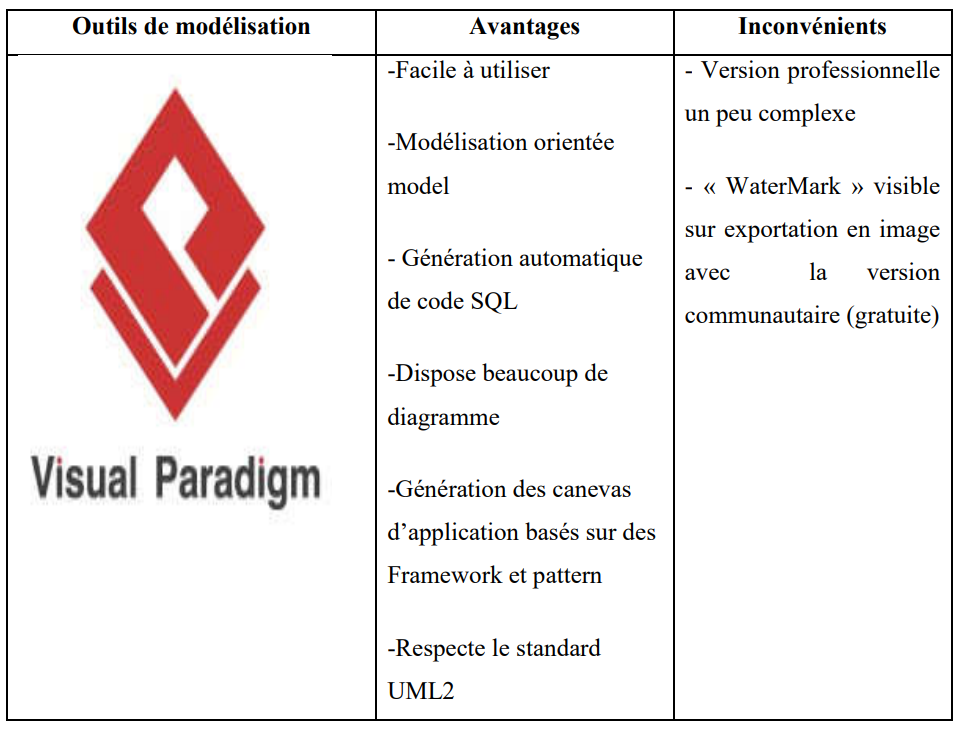
Les environnements de développements sont des programmes qui regroupent un ensemble d’outils pour le développement de logiciels. De façon générale, un environnement de développement contient un éditeur de texte, un compilateur, des outils automatiques de fabrication et dispose des différents outils pour faciliter et bien organiser ses codes. Le tableau 10 représente la comparaison entre d’éditeurs de texte et IDE pour le développement de notre projet

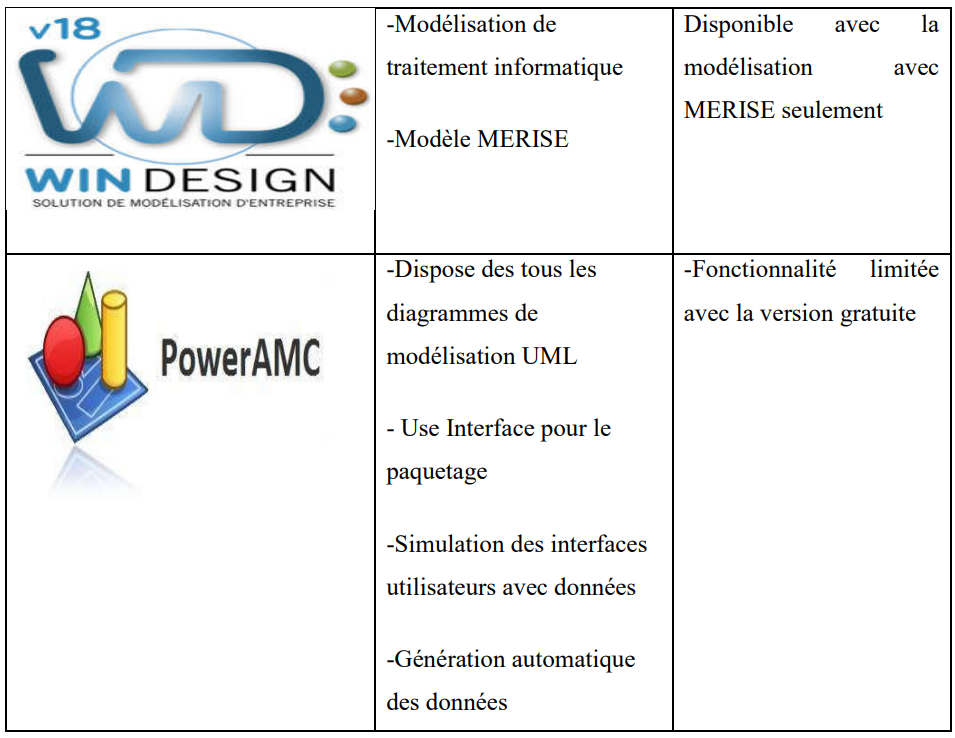


Après avoir fait une bonne comparaison, nous avons choisi Visual Studio Code pour ses avantages divers comme étant Environnement de Développement En résumé, Visual Studio Code (VSC) est rapide, beau et agréable à utiliser, sans parler de sa vaste communauté, car l’éditeur a séduit beaucoup de développeurs avec ses extensions et ses grandes fonctionnalités.

**4.3.3.6 Choix de l’outil de modélisation**

La modélisation des données est un processus consistant à conceptualiser les relations entre différents types d’informations au sein d’une organisation, indépendamment de la structure, des processus. Le tableau 10 illustre la comparaison entre Visual Paradigm, Power AMC, WIN DESIGN

****

****

D’après la comparaison sur le tableau ci-dessus, nous adoptons Visual paradigm comme Outils de modélisation.

**Présentation Visual paradigm**

Visual Paradigm (VP-UML) est un UML CASE outil SysML et Business Process Modeling Notation (BPMN) du groupe de gestion d'objets (OMG) . En plus de la prise en charge de la modélisation, il fournit des capacités de génération de rapports et d'ingénierie de code, y compris la génération de code. Il peut l'ingénierie inverse des diagrammes à partir du code et fournir une ingénierie aller-retour pour divers langages.

**Fonctionnalités de Visual Paradigme.**

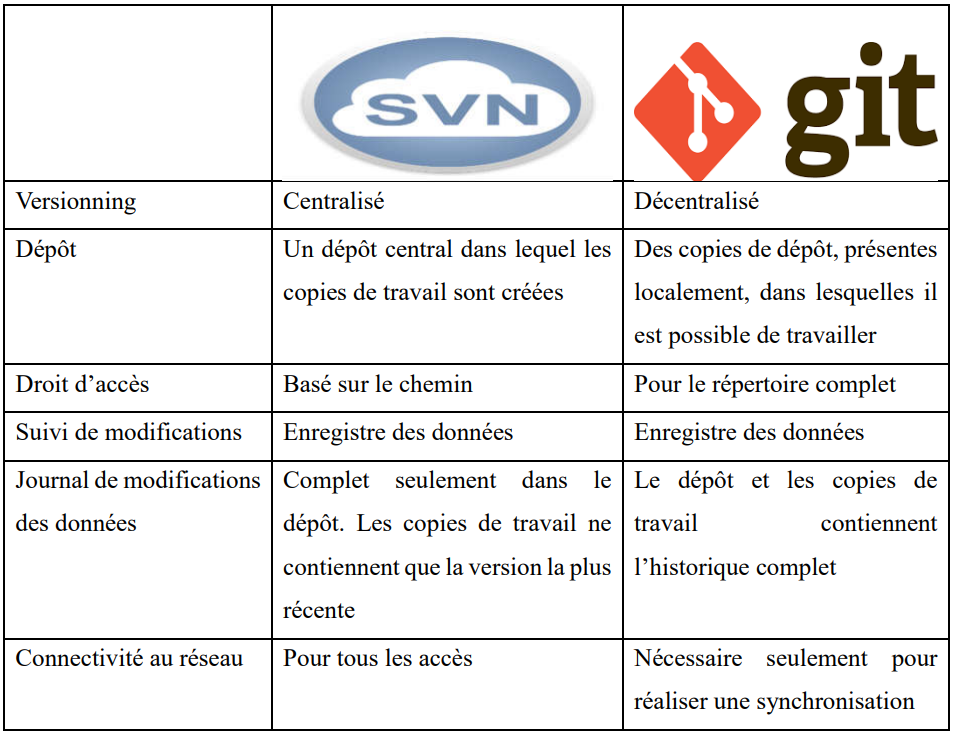
Fonctionnalités de Visual Paradigme.

* La modélisation UML 2.0 qui inclut ses 13 diagrammes ;
* La génération de code de programmation dans une bonne partie des langages communs, plus exactement : Java, C#, VB.NET, PHP, ODL, ActionScript, IDL, C++, Delphi, Perl, XML Schema, Python, Objective-C, Objective-C 2.0, Ada95 et Ruby ;
* La modélisation de bases de données relationnelles
* La génération de code SQL et le déploiement dans les principaux SGBDR, à savoir : MySQL, MS SQL Server, Oracle, HSQL, Sybase ASE, Sybase SQL Anywhere, PostgreSQL, Cloudscape-Derby, DB2, Ingres, OpenEdge, Informix, Firebird, FrontBase, Cache, SQLite et H2
* Le « reverse engineering », qui consiste en la création automatique de modèles depuis du code de programmation ou depuis une base de données existante
* La gestion des exigences ;
* L’analyse d’impacts, qui permet de connaître à l’avance les conséquences d’un changement
* La création de rapports automatisés

**4.3.3.7 Choix de logiciel de gestion de version**

Un logiciel de gestion de version est un logiciel qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées dessus. Ce système permet par exemple de mutualiser un développement. Un groupe de développeurs autour d'un même développement se sert de l'outil pour stocker toute évolution du code source. Le système gère les mises à jour des sources pour chaque développeur, conserve une trace de chaque changement

Le tableau 12 montre une étude comparative des logiciels de gestion de version



En conclusion, SVN repose sur un système de versionning centralisé. Cela signifie qu’un seul répertoire général existe et tous les utilisateurs y ont accès. Étant donné que les modifications ne peuvent être fusionnées, le système empêche que deux utilisateurs modifient un même fichier simultanément. Ce dernier est attribué au premier internaute qui l’ouvre et il reste protégé des autres utilisateurs tant qu’il n’a pas été fermé.

Alors que Git repose sur un système de versionning décentralisé. Il existe un dépôt dans lequel tous les changements sont incorporés et qui permet aux utilisateurs d’y télécharger leurs propres copies de travail. Ils disposent également d’un dépôt complet qui comprend l’historique local sans qu’ils n’aient besoin d’une connexion permanente au réseau. Par ailleurs, les modifications sont transférées rapidement dans le dépôt principal. En conséquence, Git ne propose pas de système de verrouillage, mais permet à tous les utilisateurs de créer leurs propres branches qui seront ensuite téléchargées dans le dépôt principal. Chaque utilisateur dispose par ailleurs des droits de lecture et d’écriture standards pour tout le répertoire.

* **Logiciel de version choisi**

Nous avons donc adopté l’utilisation de Git comme outils de versionning puisqu’il n’a pas besoin d’une connectivité au réseau de manière continue pour pouvoir travailler sur un projet de n’importe où ; n’a pas besoin en aucun cas de droits de lecture ou d’écriture pour des répertoires particuliers et attache beaucoup d’importance au transfert rapide des modifications..

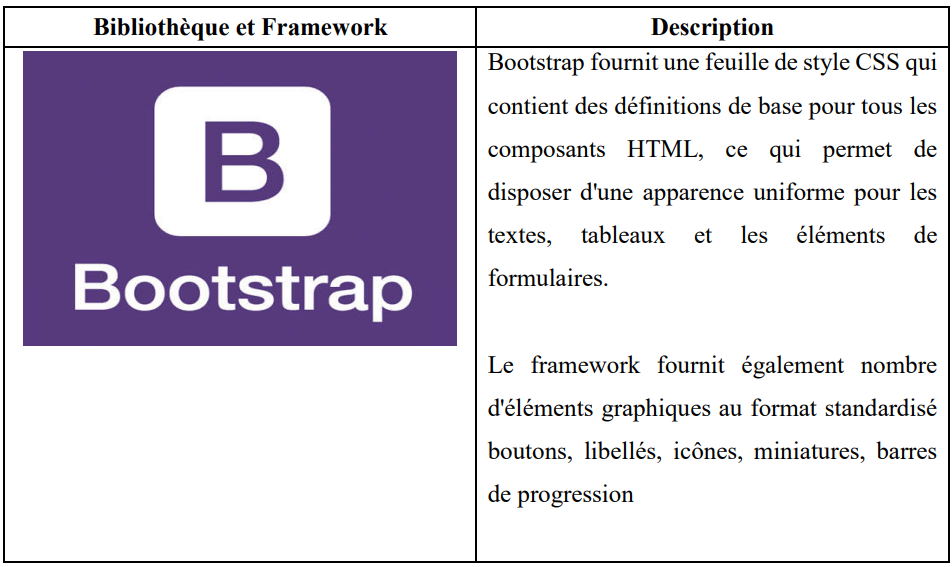
**4.3.3.7 Système d’exploitation**

Le choix du système de travail est sans aucun doute l’étape la plus importante lors du choix des outils. Pour le poste de développement, Nous avons adopté Linux Mint 20.1 comme système d’exploitation ; pour sa prise en main facile, l’occupation d’une faible place sur le disque dur par rapport ou autre système d’exploitation, Linux mint offre un gestionnaire d’applications avec de nombreuses applications gratuites.

**4.3.3.8 Autres outils supplémentaires**

Pour contribuer au développement ergonomique de l’application, nous avons intégré quelque bibliothèque et Framework sur ce projet

Le tableau 13 montre la bibliothèque, Framework de développement



A part de cette bibliothèque et Framework que nous avons mentionnée ci-dessus, nous utilisons aussi AJAX

**CHAPITRE 5 : ANALYSE CONCEPTUELLE**

Dans un projet informatique, l’analyse conceptuelle est une phase dans laquelle on étudie le système actuel en vue d’améliorer ce dernier, on proposant une modélisation du nouveau système à venir. L’analyse conceptuelle résulte de l’observation de la réalité que l’on va chercher à modéliser.

**5.1 Dictionnaire des donnée**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attributs | Description | Type | Taille | Observation |
| email | **Mail d’utilisateur** | **AN** | **20** |  |
| roles | **Type role d’utilisateur** | **AN** |  |  |
| password | **Mot de passe d’utilisateur** | **AN** |  |  |
| plainpassword | **Mot de passe non hacher d’utilisateur** | **AN** |  |  |
| firstname | **Nom d’utilisateur** | **AN** | **20** |  |
| lastname | **Prenom d’utilisateur** | **AN** | **20** |  |
| phone | **Numero phone** | **N** | **10** |  |
| adress | **Adresse d’utilisateur** | **AN** | **20** |  |
| googleid | **Identification du google d’utilisateur** | **AN** |  |  |
| facebookid | **Identification du facebook d’utilisateur** | **AN** |  |  |
| Title professionnal | **Nom du profession d’utilisateur** | **T** | **255** |  |
| statement |  | **T** | **255** |  |
| createAt | **Date creation du compte** | **D** |  | dd-mm-yyyy |
| lastLogin | **Le dernier authentification** | **D** |  | dd-mm-yyyy |
| education | **Education d’utilisateur** | **N** | **25** |  |
| experience | **experience d’utilisateur** | **N** | **25** |  |
| workreference | **Reference de la travail** | **N** | **25** |  |
| abilitie |  | **N** | **25** |  |
| interests | **interet** | **N** | **25** |  |
| userTemplate | **Tempelete de l’utilisateur** | **N** | **25** |  |
| achievement |  | **N** | **25** |  |
| userlanguage | **language** | **N** | **25** |  |
| username | **Mail d’utilisateur** | **AN** |  |  |
| photos | **Photos d’utilisateur** | **AN** |  |  |
| resetToken | **Reset de la jeton** | **AN** |  |  |
| siteweb | **url de la site web d’utilisateur** | **AN** |  |  |
| skype | **Skype d’utilisateur** | **AN** | **25** |  |
| youtube | **Youtube d’utilisateur** | **AN** | **25** |  |
| behance | **Behance d’utilisateur** | **AN** | **25** |  |
| twitter | **Twitter d’utilisateur** | **AN** | **25** |  |
| linkdin | **Linkdln d’utilisateur** | **AN** | **25** |  |
| facebookName | **Nom facebook d’utilisateur** | **AN** | **25** |  |
| Year experience | **Année d’experience d’utilicateur** | **D** |  | dd-mm-yyyy |
| time | **Temp d’experience** | **T** |  |  |
| text | **Description du travail** | **T** | **255** |  |
| Color | **Coleur du template** | **AN** | **10** |  |
| Font | **Font du template** | **AN** | **10** |  |
| Name | **Nom de template** | **AN** | **20** |  |
| Image | **Photos du template** | **AN** | **20** |  |
| File | **Emplacement du template** | **AN** | **10** |  |
| personName | **Nom de la personne** | **AN** | **20** |  |
| Position | **Position d’utilisateur** | **AN** | **10** |  |
| Description | **Description d’utilisateur** | **T** | **255** |  |
| level | **Level d’utilisateur** | **AN** | **10** |  |
| price | **Prix du template** | **N** | **10** |  |
| Typeprice | **Type du choix de prix** | **AN** | **10** |  |
| title | **Titre** | **AN** | **10** |  |
| payement | **Payement de la tarif** | **N** | **10** |  |
| nameCard | **Nom de la carte** | **AN** | **10** |  |
| Token | **Token de la payement** | **AN** | **10** |  |
| facture | **Facture de la template a payer** | **AN** | **25** |  |
| Description | **Description** | **T** | **255** |  |
| contract |  |  |  |  |
| City | **Ville d’utilisateur** | **AN** | **10** |  |
| startMonth | **Debut de la commencement du parcours** | **D** |  | dd-mm-yyyy |
| endMonth | **Fin de la commencement du parcours** | **D** |  | dd-mm-yyyy |
| userCurrent |  | **AN** | **10** |  |
| Degree | **Level d’utilisateur** | **AN** | **10** |  |
| startYear | **Debut de la commencement du parcours** | **D** |  | dd-mm-yyyy |
| startYear | **fin de la commencement du parcours** | **D** |  | dd-mm-yyyy |

**5.2 Règle de gestion**

Pour bien fonctionner, une application doit suivre des règles strictes sans lesquelles il est difficile de suivre son évolution. Notre application suit les règles suivantes :

RG 01 : Chaque utilisateur de l’application doit posséder un compte

RG 02 : Seul le Super Administrateur de l’application qui peut attribuer un compte pour voir le payement et le facturation

RG 03 : Seul le super Administrateur qui peut gérer les utilisateurs simples

RG 04 : Chaque utilisateur a un compte banquaire pour faire le payement

RG 05 : Chaque payement avoir une facturation

**5.3 Representation et specification des besoins.**

La spécification des besoins représente la première phase du cycle de développement d’une application. Elle doit décrire sans ambiguïté l’application à développer.

Nous allons spécifier l’ensemble des besoins fonctionnels et non fonctionnels liés à notre application. Ensuite, nous allons modéliser les spécifications semi-formelles des besoins à l’aide des diagrammes de cas d’utilisation et les diagrammes de séquences de systèmes.

**5.3.1 Specification des besoins fonctionnels**

Cette spécification a pour objet de présenter les besoins fonctionnels auxquels doit répondre notre application.

1. **Identification des acteurs**

Nous allons maintenant énumérer les acteurs susceptibles d’interagir avec le système. Tout d’abord, nous commençons par définir ce qu’on entend par le terme acteur.

Définition :

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

Dans notre cas , on n’a 3 acteurs : Visiteur, Utilisateur simple, administrateur

* Visiteur : C’est un acteur qui est en train de fouiller sur le net pour chercher le template du CV
* Utilisateur simple : cet acteur est un visiteur ayant déjà créé un compte sur notre site pour chercher pour faire ou créer le CV, voir le tarif du CV.
* Administrateur : cet acteur assure le dynanisme du site et veille sur les mise à jour des informations (template du CV, facturation des client déjà fait le payement),responsable de la payement par client.

1. **Identification de cas d’utilisation**

Un cas d’utilisation (use case) represente un ensemble de séquences d’actions réalisée par le système et produisant un résultat observable intéressant pour un acteur particulier. Un cas d’utilisation modélisé un service rendu par le systéme. Il exprime les interactions acteurs/systéme et apporte une valeur ajoutée « notable » à l’acteur concerné.

|  |  |  |
| --- | --- | --- |
| **Cas d’utilisation** | **Acteur** | **Message émis/recus** |
| Consulter | Visiteur | Chercher le template disponible sur cette site |
| Affiche la liste de la template |
| S’inscrire | Visiteur | Demande d’inscription |
| Remplissage d’information |
| S’authentifier | Utilisateur simple | Demande d’authenification |
| Resultat d’authenification |
| Voir le template | Utilisateur simple | Selection de la template prefere |
| Resultat de la template selectionnée |
| Voir les information remplir | Utilisateur simple | Validation de formulaire |
| Affichage d’information existe |
| Voir le forme de la template de cette CV | Utilisateur simple | Donwload de cette CV |
| Validation de cette confirmation |
| Voir le tarif de la CV | Utilisateur simple | Faire une selection de la tarif |
| Resultat de la selection |
| Voir le formulaire de la payement par stripe | Utilisateur simple | Faire une payement |
| Resultat de la payement |
| Telechargement de la CV automatique et manuel | Utilidateur simple | Telechargement de cette cette en pdf |
| Resultat de la telechargement |
| Historique de la CV de cette utilisateur | Utilisateur simple | Liste de la CV effectué |
| Resultat de la liste affiche en resume |
| Voir le tableu de bord | Administrateur | Demande de consultation de la tableu de bord |
| Affichage de la tableau de bord |
| Gérer un utilisateur | Administrateur | Demande de gérer l’utilisateur |
| Affichage des utilisateur existe |
| Gerer le payement | Administrateur | Demande de gerer le payement |
| Affichage des payement des client effectue le payement |

**5.3.1.1 Analyse de besoins fonctionnels**

Dans la suite, nous désignons, par centre d’intérêt , les différent service offerts par notre application :

Apres l’authentification ,les utilsateur pourront entrer dans l’application qui lui permet

**ADMINISTRATEUR**

* Voir le tableau de bord
* Consultet la liste de client inscrit dans l’application
* Gerer les utilisateur
* Gerer les payement

**CLIENT**

* Voir le page d’accueil
* Choisir le template du CV et le theme
* Remplir le information utilisée dans le CV
* Faire une payement par compte bancaire

**5.3.2 Spécification des besoin non fonctionnel :**

**a)** Les besoins non fonctionnels spécifient les propriétés du système telles que les contraintes d’environnement et d’implémentation, la performance, la maintenance, l’extensibilité et la flexibilité. Certains besoins non fonctionnels sont généraux et ne peuvent pas être rattachés à un cas d’utilisation particulier

b)Les contraintes ergonomiques

Les contraintes ergonomiques sont les contraintes liées à l’adaptation entre les fonctionnalités de l’application, leurs interfaces et leur utilisation.

Pour notre application, nous devons obéir aux contraintes ergonomiques suivantes :

* Permettre un accès rapide dans l’application
* Interface simple et compréhensible
* L’application doit être développée avec un langage compréhensible
* Présenter les informations d’une façon simple et claire
* Faire apparaître les choix ou les saisies de l’information

**b- Les contrainte techniques**

Les contraintes techniques remarquées durant l’analyse sont

* Il faut que toutes les interfaces de l’application soient homogènes. En effet, les différentes pages doivent suivre le même modèle de représentation (couleurs, images, textes défilant) de chaque acteur (Client, Administrateur)
* Le code doit être extensible et maintenable pour faciliter toute opération d’amélioration ou d’optimisation

**c- Les contrainte de déployement**

Pour les contraintes de déploiement, voici ce que nous pouvons citer :

* L’application doit être déployée sur un ordinateur avec un système d’exploitation Windows ou Linux (multiplateforme)
* La spécification des besoins fonctionnels et non fonctionnels procure une vision claire du sujet et une compréhension plus profonde des tâches à réaliser. Elle mène également à prévoir une spécification semi-formelle de l’application

**5.3.3 Spécification semi-formelle des besoins**

La phase de modélisation nécessite des méthodes permettant de mettre en place un modèle d’analyse. Dans ce cas, nous avons adopté le langage UML, « Unified Modeling Language », qui est un langage de modélisation graphique des données et des traitements. À travers le langage UML, nous pouvons donner plusieurs vues d’un système grâce aux différents diagrammes qu’on va concevoir.

UML n’est pas une méthode (i.e. Une description normative des étapes de la modélisation): ses auteurs ont, en effet, estimé qu’il n’était pas opportun de définir une méthode en raison de la diversité des cas particuliers. Ils ont préféré se borner à définir un langage graphique qui permet de représenter et de communiquer les divers aspects d’un système d’information UML est caractérisé par un travail d’expert qui utilise l’approche orientée objet et suivant une norme. Sa notation limite les ambiguïtés et les incompréhensions. Il est aussi indépendant du langage de programmation et peut être utilisé pour modéliser n’importe quel système.

1. **Diagramme de cas d’utilisation de l’application**

Les cas d’utilisation sont une technique de description du système étudié privilégiant le point de vue de l’utilisateur. Il s’agit de la solution UML pour représenter le modèle conceptuel. Les cas d’utilisation décrivent sous la forme d’actions et de réactions, le comportement d’un système du point de vue d’un 58 utilisateur. Les cas d’utilisation servent à structurer les besoins des utilisateurs et les objectifs correspondants du système.

Les éléments constitutifs des cas d’utilisation sont :

- Acteur : entité externe qui agit sur le système. Le terme acteur ne désigne pas seulement les utilisateurs humains, mais est des classificateurs qui représentent des rôles au travers d'une certaine utilisation (cas) et non pas des personnes physiques. Ce sont des acteurs types.

* Cas d’utilisation : ensemble d’actions réalisées par d’un acteur.
* Les cas d’utilisation peuvent être structurés
* Les cas d’utilisation peuvent être organisés en paquetages

L’ensemble des cas d’utilisation décrit les objectifs du système

SARY ETO

Avant de devenir client, un visiteur ne possède que la possibilité de voir la page d’accueil, et la possibilité de faire l’inscription.

SARY ETO

Diagramme de cas d’utilisation d’administrateur

SARY ETO

Cas d’utilisation global

SARY ETO

* Raffinement de cas d’utilisation

Le raffinement consiste à avoir une approche de conception où on affine à chaque étape le niveau de détails de chaque cas d’utilisation

* Cas d’utilisation « Gerer les information »
* SARY ETO

|  |  |
| --- | --- |
| Titre | Gerer information |
| But | Ajouter,lister, modifier les information |
| Résume | L’utilisateur elle meme peut modifier les information des client |
| Acteur | Utilisateur |