

Introduction générale

Il ne fait désormais plus aucun doute que l'informatique est la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité moderne. En effet, les logiciels informatiques proposent maintenant des solutions à tous les problèmes de la vie, aussi bien dans des domaines professionnels que pour des applications personnelles. Et leurs méthodes de conception et de développement ont vu l'avènement d'autant de technologies qui facilitent leur mise en place et leurs donnent des possibilités et des fonctionnalités de plus en plus étendues.

Une société représente une des organisations ayant des ressources et des activités dont la gestion nécessite une application informatique. Ainsi, l'objectif de notre projet est de réaliser une application informatique interactive, fiable, conviviale et facile à intégrer dans l'environnement de travail, assurant la gestion des systèmes d'informations compte tenu des besoins exprimés. Cette application vise essentiellement à diminuer la complexité des traitements ainsi que le temps perdu lors de la gestion des projets, en particulier.

Dans le présent rapport, nous présentons en détail les étapes que nous avons suivies pour réaliser notre application. Ce rapport comporte quatre chapitres qui sont organisés comme suit :

Chapitre 1 : Intitulé Cadre générale du projet qui consiste à souligner le contexte de projet, présenter l'organisme d'accueil ainsi que la thématique de projet dans laquelle on va étudier les solutions existantes et la solution proposée

Nous abordons les principes des méthodologies agiles, en nous concentrant particulièrement sur la choix de méthodologie SCRUM. Ensuite nous finissons par le choix de langage de modélisation .

Chapitre 2 : Présente une étape primordiale, les spécifications des besoins, la modélisation et l'étude conceptuelle. C'est à ce niveau que nous avons évoqué l'aspect conceptuel de notre application en commençant par spécifier les besoins et les acteurs de systèmes et en passant par la précision des éléments du "Product Backlog", ainsi que la présentation de l'environnement matériel et logiciel utilisé pour la réalisation de notre application tout en précisant les choix techniques adoptés.

Chapitre 3 :

Chapitre 4 :

Chapitre 1

Cadre générale du projet

1.1 Introduction

Ce premier chapitre est dédié à présenter le cadre de notre projet au sein de la société DEV IT.

Nous commençons par une brève introduction de l'entreprise, mettant en lumière son domaine.

Ensuite, nous nous penchons sur une étude de l'existant, émettons des critiques constructives et proposons une solution pour répondre à ces défis.

1.2 Présentation de l'organisme d'accueil

« Dev-IT » est une société qui fait partie du groupe DevIT Group qui opère en Afrique du Nord, en partenariat avec des agences européennes. Le groupe compte 2 filières pour le moment, une en Tunisie, à Sousse plus particulièrement et une autre à Genève en Suisse.

Fondée officiellement en Novembre 2016, Dev-IT se spécialise dans les audits informatiques, développement web et mobile. Leur mission est l'accompagnement des entreprises à intégrer les nouvelles technologies de l'informatique et ce à travers leur offre polyvalente de services et de solutions qui ont pour objectifs de rendre les applications informatiques disponibles, performantes, sécurisées et évolutives. Leur objectif ultime est de mettre en place des solutions et des applications sur mesure qui sont adaptées aux be-

soins de leurs clients et qui sont adéquates à l'évolution des marchés, à l'essor technologique ainsi qu'à la digitalisation.



FIGURE 1.1 – Logo de l'entreprise DEV IT

1.3 Cadre de projet

1.3.1 Contexte de projet

Ce stage a été réalisé dans le cadre d'un projet de fin d'études pour l'obtention de la licence technologies de l'informatique de l'institut supérieur des études technologiques de Kairouan (ISETKR). Notre stage a été effectué au sein de l'entreprise DEV IT.

Le sujet est intitulé «Création d'une plateforme de gestion des projets»

1.3.2 Analyse de l'existant

1.3.3 Etude de l'existant

Sur le marché, il existe plusieurs plateformes de gestion des projets, et parmi les outils les plus connus on trouve ASANA et JIRA.

1.3.4 Asana

Il s'agit d'une plate-forme robuste qui sert à vos équipes de rester concentrées sur les objectifs, les projets et les tâches quotidiennes. Voici les fonctionnalités les plus importantes qu'elle offre :

- Tableau de bord permet de visualiser facilement votre travail.
- Voir la grande image. Clouez votre timing en visualisant les travaux sur un calendrier. Repérez facilement les trous et les chevauchements dans votre horaire et effectuez rapidement les ajustements nécessaires.
- Gardez une trace de ce qui est le plus important.
- Pas besoin de réinventer la roue. Transformez les processus courants en modèles que votre équipe entière peut utiliser pour que les projets se déroulent sans heurts à chaque fois.
- Partagez des informations avec les bonnes personnes. Rendez les équipes et les projets privés afin de créer un espace sécurisé pour les travaux sensibles.
- Partagez des informations avec les bonnes personnes. Rendez les équipes et les projets privés afin de créer un espace sécurisé pour les travaux sensibles.

1.3.5 Jira

Qui est l'un des meilleurs outils de gestion de produit qui comprend :

- Le suivi et la gestion des problèmes
- La gestion des produits
- Un Tableau de bord configurable
- Des Rapport sur l'avancement du projet
- Support des méthodologies Scrum & Kanban

1.3.6 Critique de l'existant

Suite à une analyse approfondie du système existant pour la gestion de projet, plusieurs limitations ont été identifiées :

- **Fragmentation de l'information** : Les informations essentielles relatives aux projets étaient dispersées à travers diverses sources et plateformes, ce qui rendait difficile leur consolidation et leur accès rapide.
- **Complexité de l'information décisionnelle** : Les outils actuels pour la surveillance et l'analyse des projets étaient complexes et nécessitaient

une expertise approfondie pour être pleinement exploités, ce qui compliquait la prise de décision rapide.

- **Incohérence des données et des interfaces** : Des erreurs et des incohérences étaient observées dans les données et les interfaces utilisées, ce qui pouvait entraîner des malentendus et des retards dans les activités de gestion de projet.
- **Risque de satisfaction client insatisfaisante** : En raison des lacunes mentionnées ci-dessus, il existait un risque accru de ne pas répondre efficacement aux attentes des clients, ce qui aurait pu compromettre leur satisfaction globale.

Ces défis soulignent l'importance cruciale de mettre en place une nouvelle solution de gestion de projet, avec pour objectif de simplifier et de centraliser les informations, d'améliorer l'efficacité des processus décisionnels, de réduire les pertes de temps et d'assurer une satisfaction client optimale.

1.3.7 Solution proposée

Pour résoudre les défis identifiés et améliorer la gestion de projet, nous proposons la mise en place d'un nouvel outil de gestion basé sur les principes suivants :

- **Centralisation des Informations** : Développer une plateforme intégrée qui centralise toutes les informations relatives aux projets, y compris les tâches, les ressources, les calendriers et les rapports, pour assurer une accessibilité facile et rapide à toutes les parties prenantes.
- **Simplicité d'utilisation** : Concevoir une interface conviviale et intuitive qui simplifie l'interaction avec le système, réduisant ainsi la complexité et facilitant l'adoption par les utilisateurs.
- **Contrôle de Qualité des Données** : Mettre en place des mécanismes de contrôle de qualité des données pour garantir la fiabilité et la cohérence des informations, réduisant ainsi les risques d'erreurs et d'incohérences.
- **Personnalisation** : Offrir des fonctionnalités modulables et personnalisables en fonction des besoins spécifiques de chaque équipe et projet, permettant une adaptation optimale aux exigences variées des clients.

Cette solution vise à améliorer la gestion globale des projets en réduisant les inefficacités et en augmentant la transparence et la collaboration entre les équipes. En adoptant cette approche, nous visons à répondre aux attentes

des clients et à garantir le succès et la satisfaction dans la réalisation des projets.

1.4 Méthodologie de travail

Dans le contexte évolutif du développement logiciel, la méthodologie adaptée a une grande importance. Pour notre projet, nous avons choisi la méthodologie Agile, en particulier Scrum. Cette décision découle d'une évaluation attentive des besoins du projet et de notre désir de livrer une solution performante et alignée sur les attentes du client, tout en optimisant les ressources. La figure 1.2 illustre la méthodologie SCRUM.

Scrum Process

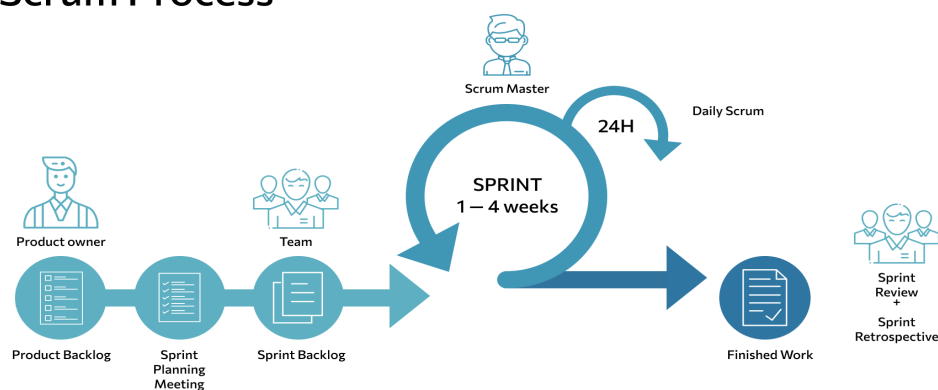


FIGURE 1.2 – Méthodologie Scrum

1.4.1 Méthode Agile

La méthodologie agile est une manière de gérer les projets qui met l'accent sur la flexibilité et la collaboration, ce qui permet d'adapter de manière efficace les objectifs et les réalisations face aux évolutions constantes. Avec les méthodes agiles, le client occupe une position centrale dans la gestion de son projet et voit rapidement la mise en service de son logiciel.

1.4.2 Présentation de la méthodologie SCRUM

Scrum représente une méthode Agile qui simplifie la coopération au sein des équipes et les guide dans l'accomplissement de tâches de grande valeur. Elle offre un modèle de principes, de rôles et de règles pour les orienter à chaque itération et favoriser leur amélioration constante.

Scrum utilise des "sprints" de deux semaines où l'équipe doit produire un résultat concret. Il y a aussi deux autres événements importants : les réunions quotidiennes de coordination, qui durent 15 minutes et les bilans de sprint orchestrés par le Scrum Master. Ces évaluations des sprints permettent à l'équipe de faire le point et de planifier des améliorations pour les prochaines étapes.

1.4.3 Langage de modélisation

Pour assurer une description précise du fonctionnement du système, plusieurs langages et approches de conception sont disponibles. Dans notre projet, nous avons utilisé le Langage de Modélisation Unifié UML (Unified Modeling Language) développé par l'Object Management Group(OMG), pour guider notre processus de conception.

1.5 Conclusion

Dans ce chapitre, nous avons présenté le contexte générale de notre projet ainsi que la critique de l'existant ,la solution proposée et l'étude méthodologique

Chapitre 2

Préparation De Projet

2.1 Introduction

Cette partie consiste en une étape analytique dans laquelle nous allons recenser et factoriser les besoins des utilisateurs de l'application. Ceci est fortement lié à l'étude du projet menée au Cours du premier chapitre. Pour ce faire cette phase doit répondre aux questions suivantes : Quels sont les besoins fonctionnels de l'application ? Quelles sont les contraintes qui doivent être prises en considération ?

2.2 Spécifications des besoins

2.2.1 Besoins fonctionnels

Les besoins primordiaux de notre application incluent la gestion efficace des utilisateurs, la traçabilité des interactions avec les clients, la planification et le suivi des projets, ainsi que la gestion collaborative des tâches.

L'application CRM que nous développons vise à fournir des fonctionnalités spécifiques à différents acteurs clés, à savoir :

Administrateur

- Création, modification et suppression des utilisateurs (Développeurs, Responsables Commerciaux, Chefs de Projet).
- Gestion des portfolios et des projets.

- Suivi des statistiques .

Responsable Commercial

- Gestion des clients : création, modification et suppression.
- Suivi des interactions clients.

Chef de Projet

- Gestion détaillée des projets : création, modification et suppression.
- Affectation des membres à chaque tâche.
- Suivi du déroulement des projets .

Développeur

- Suivi des tâches assignées : consultation, mise à jour.
- Collaboration sur les projets avec les autres membres.

2.2.2 Besoins non fonctionnels

Les besoins non fonctionnels spécifient les propriétés du système afin de garantir la Cohérence, la confidentialité et l'intégrité des données. Le système doit être fiable : la validité de l'application. Réutilisabilité : aptitude de site à être utilisé en tout ou en partie dans de nouvelles applications.

La performance d'exécution

Le temps d'exécution du système doit être minimal pour ne pas gêner l'utilisateur. Ce temps dépend de la complexité du code implémenté, du serveur d'application utilisé, du débit de la ligne de connexion et de la conception de la base de données.

La sécurité

Le système doit respecter un niveau de sécurité élevé afin de garantir la confidentialité de l'accès des membres

L'ergonomie

L'interface de cette application doit être ergonomique, conviviale et voire même apte à aider l'utilisateur à mieux gérer son espace de travail.

2.2.3 Identification des Acteurs

Dans le cadre de notre application CRM, nous identifions les acteurs suivants, qui représentent les utilisateurs principaux de notre système :

TABLE 2.1 – Identification des Acteurs

Acteur	Description du Rôle
Administrateur	L'administrateur dispose d'un accès complet à l'ensemble de l'application CRM. Il est chargé de la gestion des utilisateurs, des projets, des clients, et de la supervision globale de l'activité.
Responsable Commercial	Ce membre est responsable de la gestion des clients, de leur suivi, ainsi que des interactions commerciales au sein de l'application.
Chef de Projet	Le chef de projet supervise la planification, l'exécution et le suivi des projets. Il affecte les membres aux tâches, gère les ressources et assure le bon déroulement des activités.
Développeur	Les développeurs sont chargés de l'exécution des tâches assignées dans les projets. Ils interagissent avec les autres membres, mettent à jour les statuts des tâches et collaborent sur les fonctionnalités.

2.3 Diagramme de cas d'utilisation générale

Le diagramme de cas d'utilisation permet de décrire l'interaction entre l'acteur et le système. Le cas d'utilisation est une description des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système. Un acteur et un cas d'utilisation sont mis en relation par une association représentée par une ligne. Le but principal du diagramme du cas d'utilisation est de définir le système du point de vue des utilisateurs et de définir les limites précises du système en utilisant une notation très simple

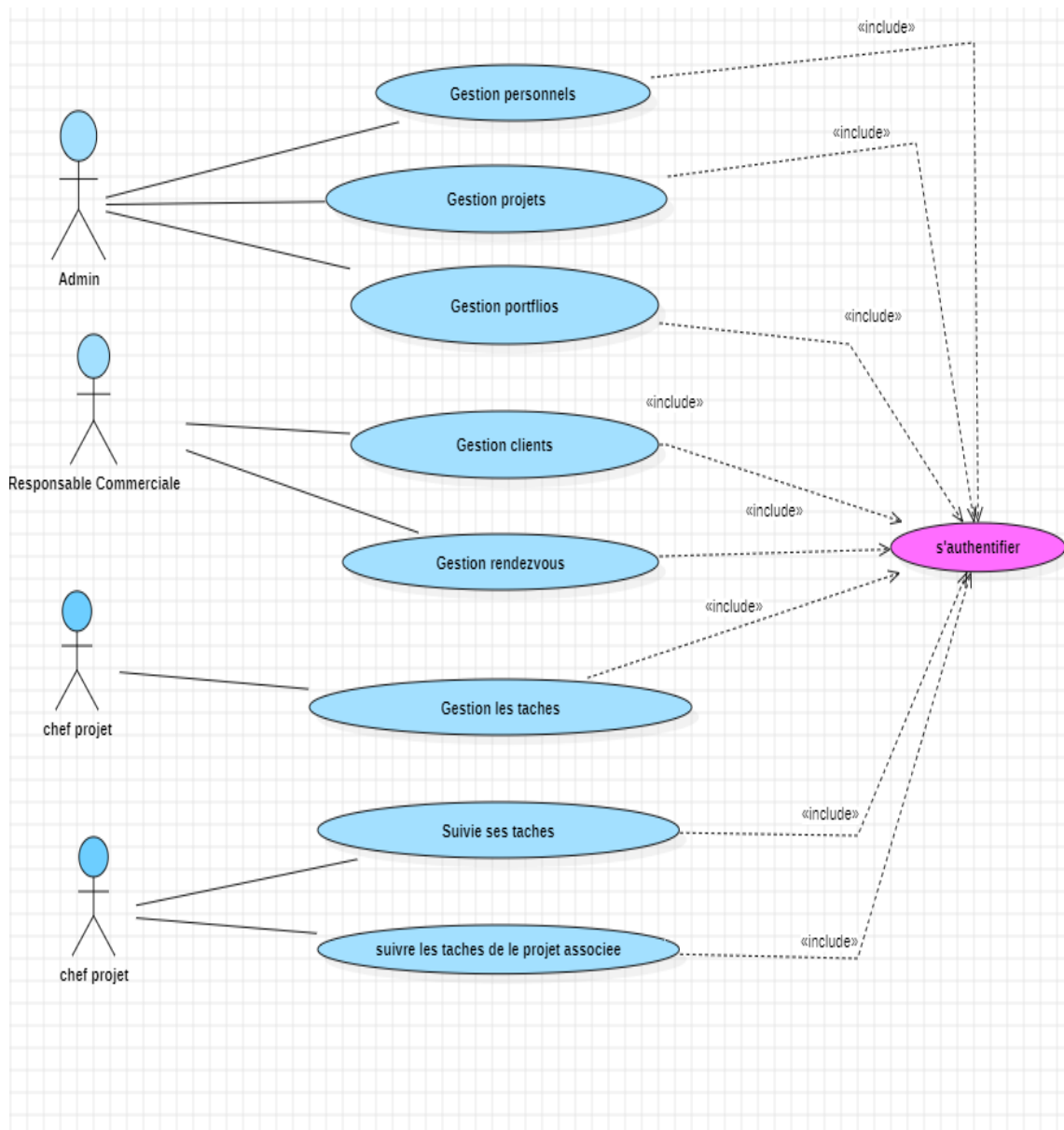


FIGURE 2.1 – Diagramme de cas d'utilisation générale.

2.4 Pilotage du projet avec SCRUM

Nous avons réalisé une étude en vue de classer les stories par ordre de priorité. Les références sur lesquelles nous nous sommes appuyés sont :

- **ID (Id.Story)** : Identifiant exclusif de la fonctionnalité.
- **Thème** : Cela désigne une fonction ou un service cohérent au sein du produit.
- **User Story** : Description directe et claire d'une caractéristique ou d'une attente formulée par un utilisateur.
- **Priorité** : C'est le niveau d'importance d'une fonctionnalité (élevée, moyenne, faible).
- **Estimation(j)** : C'est le nombre de jours approximatifs nécessaires pour finaliser une fonctionnalité.

2.4.1 Equipe et rôle

* Le Product Owner :

Nous avons une vision du produit à réaliser. Il travaille en dialogue avec l'équipe de développement qui doit suivre ses instructions. Il priorise les fonctionnalités qu'il développe ou modifie et valide les fonctionnalités finies. Il est en charge de la gestion du backlog produit.

* Le Scrum Master :

Il est responsable de la compréhension, du respect et de la mise en œuvre des méthodes Scrum qu'il maîtrise parfaitement. Un facilitateur qui aide les équipes à améliorer et maximiser la communication, la productivité et le savoir-faire. Il est considéré comme l'entraîneur de l'équipe de développement.

* Team Members :

L'équipe de développeurs chargée de transformer les exigences définis par le Product Owner en fonctionnalités utilisables.

2.4.2 backlog du produit

ID	Thème	User Story	Priorité	Estimation(j)
1	S'authentifier	En tant qu' utilisateur/administrateur , je peux me connecter sur l'application en saisissant mon e-mail et mon mot de passe	Élevée	6j
2	gérer personnels	En tant qu' administrateur , je peux ajouter ou modifier ou supprimer un utilisateur (développeur/chef projet/commercial)	Élevée	6j
3	gérer portfolio	En tant qu'administrateur, je peux gérer le portfolio de projets, y compris leur création, modification	Moyenne	5j
4	Gestion des statistiques	En tant qu'administrateur, je peux visualiser les statistiques globales de l'activité du projet, y compris le nombre total de projets et de clients	Moyenne	6j
5	Gestion des clients	En tant que responsable commercial, je peux gérer les informations des clients (ajout, modification, suppression)	Moyenne	5j
6	Gestion des projets	En tant que chef de projet, je peux gérer les projets en les associant aux clients	Élevée	5j
7	Affectation des tâches aux développeurs	En tant que chef de projet, je peux affecter les tâches aux développeurs pour les différents projets.	Élevée	5j
8	Suivi des tâches par les développeurs	En tant que développeur, je peux suivre mes tâches assignées dans les projets	Moyenne	3j
9	Consultation des détails des tâches assignées	En tant que développeur, je peux consulter les détails et l'état des tâches qui me sont assignées dans les projets.	Moyenne	3j

TABLE 2.2 – Backlog produit

2.4.3 Plannification de Sprint

TABLE 2.3 – Planification des sprints

Sprint	Fonctionnalité	Durée (jours)
Sprint 1 (Gestion des Utilisateurs)	Création, modification et suppression des utilisateurs	14
	Gestion des portfolios et des projets	10
	Suivi des statistiques	5
Sprint 2 (Gestion des Clients)	Gestion des clients : création, modification et suppression	8
	Suivi des interactions clients	5
Sprint 3 (Gestion des Projets)	Gestion détaillée des projets	15
	Affectation des membres à chaque tâche	10
	Suivi du déroulement des projets	12
Sprint 4 (Fonctionnalités des Développeurs)	Suivi des tâches assignées	7
	Collaboration sur les projets	5

2.5 Environnement de développement

2.5.1 Environnement matériel

Pour la réalisation du projet, nous avons utilisé un ordinateur portable pour le développement ayant les caractéristiques suivantes :

- Modèle : DESKTOP-IRIK9P1
- Processeur : 12th Gen Intel(R) Core(TM) i5-1235U 2.50 GHz
- Systèmes d'exploitation : 64 bits, processeur x64
- Mémoire : 8Go

2.5.2 Environnement logiciel

L'environnement logiciel utilisé pour réaliser notre projet est comme suit :

2.5.3 Outils de développement et modélisation

- **MongoDB :**

MongoDB est un programme de base de données multiplateforme orienté document disponible en source. Classé comme programme de base de données NoSQL, MongoDB utilise des documents de type JSON avec des schémas facultatifs.



FIGURE 2.2 – Logo MongoDB

- **Visual Studio Code** [1]

est un éditeur de code extensible développé par Microsoft pour Windows, Linux et OS X. Nous avons utilisé visuel code pour écrire le code de l'application.



FIGURE 2.3 – Visual Studio Code.

- **StarUML** [14]

StarUML est un logiciel de modélisation UML, cédé comme open source par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU GPL, Nous avons fait les diagrammes avec cette technologie.



FIGURE 2.4 – Staruml.

— **Postman :**

Est le seul environnement de développement d'API complet utilisé par plus de 7 millions de développeurs et 200 000 entreprises dans le monde cela rend le développement d'API plus rapide, plus facile et plus performant.



FIGURE 2.5 – Postman.

2.5.4 Langages de programmation

— **Node js**

Framework javascript ,nous l'avons utilisé pour créer le serveur web . Il offre la rapidité de ,la performance etla modularité.



FIGURE 2.6 – Node js.

- **React js** React (aussi appelé React.js ou ReactJS) est une bibliothèque JavaScript libre. Elle est maintenue par Meta (anciennement Facebook) ainsi que par une communauté de développeurs individuels et d'entreprises depuis 2013..

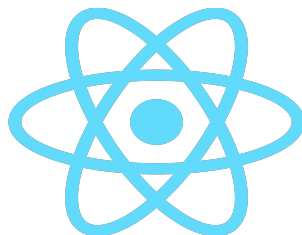


FIGURE 2.7 – React js.

- **Css 3** Un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.



FIGURE 2.8 – Css.

- **Html 5** HTML5 (HyperText Markup Language 5) est la dernière révision majeure du HTML (format de données conçu pour représenter les pages web). Cette version a été finalisée le 28 octobre 2014.



FIGURE 2.9 – HTML 5.

- **Java script** Un langage informatique utilisé sur les pages web. Ce langage a la particularité de s'activer sur le poste client. En d'autres termes c'est votre ordinateur qui va recevoir le code et qui devra l'exécuter. Ceci est en opposition par rapport à d'autres langages qui sont activés côté serveur.



FIGURE 2.10 – java script .

2.5.5 Framework utilisé

- **Bootstrap** Est une collection d'outils utile à la création de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

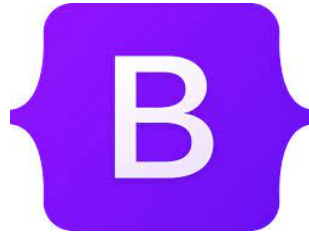


FIGURE 2.11 – Bootstrap

2.6 Architecture adoptée

Dans le domaine informatique, l'architecture logique décrit les composants abstraits (souvent les services et leurs clients) et les interactions entre eux. Dans ce contexte nous allons commencer par présenter le modèle MVC.

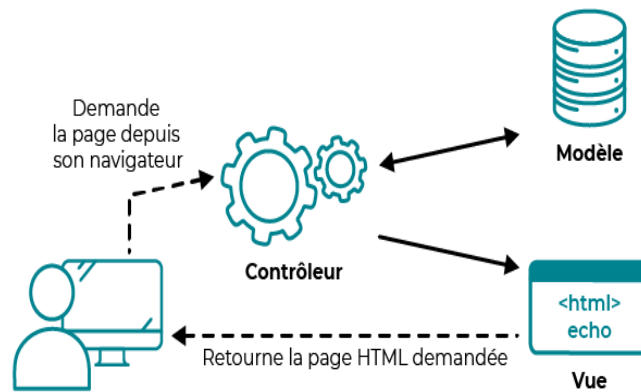


FIGURE 2.12 – Architecture MVC

Architecture MVC(Model View Controller)

- **Model :**

Le modèle représente la forme des données et la logique métier.

Il maintient les données de l'application. Les objets de modèle récupèrent et stockent l'état du modèle dans une base de données. Le modèle est une donnée et une logique métier.

- **View :**

View ou Vue est une interface utilisateur. Afficher les données d'affichage en utilisant le modèle pour l'utilisateur et leur permet également de modifier les données. La vue est une interface utilisateur.

- **Controller :**

Le contrôleur gère la demande de l'utilisateur. En règle générale, l'utilisateur interagit avec la Vue, ce qui déclenche à son tour la demande d'URL appropriée. Cette demande sera gérée par un contrôleur. Le contrôleur rend la vue appropriée avec les données du modèle en réponse. Le contrôleur est un gestionnaire de demandes.

2.7 Architecture du système :

En effet, nous choisissons l'architecture en un tiers pour la mise en place de notre plateforme. Cette architecture, comme le montre la figure ci-dessous, consiste à faire communiquer un poste client avec le serveur qui va accéder à la base de données. C'est pour cette raison, qu'il s'agit d'une architecture partagée entre :

- Le client : c'est le poste demandé de ressource équipée par un navigateur web pour la présentation.
- Un système de gestion de base de données : utilisé pour le stockage des données. Il permet aussi de fournir les données au serveur d'application.
- Un serveur d'application : son rôle consiste à répondre de manière automatique aux demandes envoyées par le client, il est considéré comme un intermédiaire entre le client et le serveur de base de données.

—

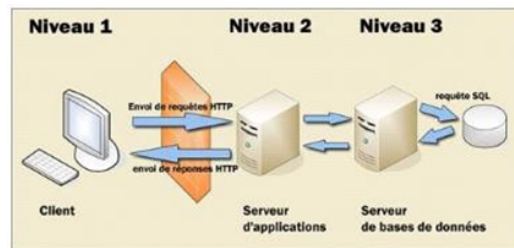


FIGURE 2.13 – architecture physique