



UNIVERSITE DE FIANARANTSOA

ECOLE NATIONALE D'INFORMATIQUE

MEMOIRE DE FIN D'ETUDES POUR L'OBTENTION DU
DIPLOME DE MASTER PROFESSIONNEL

Mention : Informatique

Parcours : Informatique Générale

Intitulé

CONCEPTION ET REALISATION D'UNE APPLICATION
WEB POUR LA GESTION D'INVESTISSEMENT DES
PROJETS EN LIGNE

Présenté le 2 Decembre2022

Par RAKOTOMANDIMBY Menye Marion

Membres du Jury :

Président : Monsieur MAHATODY Thomas, Professeur

Examineur : Monsieur RAMAMONJISOA Bertin Olivier, Professeur Titulaire

Rapporteurs : Monsieur RABARIVONY Zo, lead développeur

Monsieur GILANTE Gesazafy, Assistants d'Enseignement Supérieur et
de Recherche

Année universitaire 2020 - 2021

CURRICULUM VITAE

RAKOTOMANDIMBY

Menye Marion

Né le : 08 Novembre 1997 à Manakara

Adresse : 1M5 Andranovato Manakara (316)

Tél : 034 73 519 27

E-mail : marionmenye@gmail.com



FORMATIONS ET DIPLÔMES

- **2020-2021:** Deuxième année de formation en Master Professionnel à l'Ecole Nationale d'Informatique, Université de Fianarantsoa, Parcours : Informatique Généraliste.
- **2019-2020:** Première année de formation en Master Professionnel à l'Ecole Nationale d'Informatique, Université de Fianarantsoa, Parcours : Informatique Généraliste.
- **2018-2019:** Troisième année de formation en Licence Professionnelle à l'Ecole Nationale d'Informatique, Université de Fianarantsoa, Parcours : Informatique Généraliste.
- **2017-2018:** Deuxième année de formation en Licence Professionnelle à l'Ecole Nationale d'Informatique, Université de Fianarantsoa, Parcours : Informatique Généraliste.
- **2016-2017 :** Première année de formation en Licence Professionnelle à l'Ecole Nationale d'Informatique, Université de Fianarantsoa, Parcours : Informatique Généraliste.
- **2015-2016 :** Terminal au Lycée Présentation de Marie Manakara et obtention du BACC série C.

STAGES ET EXPERIENCES INFORMATIQUES

- **May 2022 – Octobre 2022** : Stage au sein du Société **MYAGENCY** Antananarivo, thème : « Conception et réalisation d'une application web pour la gestion d'investissement en ligne »
- **2021** : Réalisation des projets à l'Ecole National d'Informatique :
 - ❖ Développement d'une application mobile
 - ❖ React et Node Js en utilisant Docker
 - ❖ VueJS et Symfony
 - ❖ Conception avec UML 2
- **Novembre 2019 – Janvier 2020** : Stage au sein de la Direction de Système d'information du ministère de l'intérieur et de la décentralisation Antananarivo
- **2019** : Réalisation des projets à l'Ecole National d'Informatique :
 - ❖ Java
 - ❖ Jsp et Asp.net
 - ❖ Développement d'une application mobile avec le framework Ionic
 - ❖ RPC et NFS
 - ❖ Conception avec UML 2
 - ❖ VOIP avec Asterisk
- **Novembre 2018 – Janvier 2019** : Stage au sein de la Direction Régionale du Budget de Vatovavy Fitovinany.
Sujet : Conception et réalisation d'une application pour la Gestion du Personnel
- **2018** : Réalisation des projets à l'Ecole National d'Informatique :
 - ❖ PHP/MYSQL
 - ❖ Supervision et gestion d'un réseau avec **Cacti** sous **Debian**.
- **2017** : Réalisation des projets à l'Ecole National d'Informatique :
 - ❖ HTML + CSS
 - ❖ JavaScript
 - ❖ Caméra IP sous Android

<u>CONNAISSANCES INFORMATIQUES</u>

- ❖ **Méthode d'analyse et de conception** : MERISE.
- ❖ **Gestion de projet** : AGILE, SCRUM
- ❖ **Langage d'analyse et conception** : UML 2.
- ❖ **Framwork** : ionic , Symfony, Angular 5, laravel , Vue JS , React

- ❖ **Système de gestion de base de données** : MySQL, Oracle, PostgreSql , MongoDB
- ❖ **Développement web** : HTML, CSS, PHP, JavaScript, NodeJS
- ❖ **Langages de développement d'application** : Langage C, C++, Visual Basic ...
- ❖ **Systèmes d'exploitation** : Windows XP, 7, 8 et **Linux** : Kali, Debian,Centos...
- ❖ **Langages de modélisation:** UML
- ❖ **Outils de développement** : Codebloks, Sublime text, NetBeans, Visual Studio
- ❖ **Gestionnaire de version** : GIT
- ❖ **Serveur web** : WAMP, XAMP

CONNAISSANCES LINGUISTIQUES

Domaine Langues	COMPREHENSION A L'AUDITION	LECTURE	REDACTION ECRITE	COMMUNICATION ORALE
Anglais	B	B	B	B
Français	B	TB	B	B

Grille d'évaluation : TB : Très Bonne, B : Bonne, AB : Assez Bonne, P : Passable, NS : Niveau Scolaire.

LOISIRS ET SPORTS

Sports: Football, Natation, Basket-ball.

Loisirs:

Surf sur Internet, suivi des nouvelles actualités technologiques, Guitare.

SOMMAIRE GENERAL

CURRICULUM VITAE.....	I
SOMMAIRE GENERAL	IV
REMERCIEMENTS	VI
LISTE DES FIGURES	VII
LISTE DES TABLEAUX	VIII
LISTE DES ABREVIATIONS.....	IX
INTRODUCTION GENERALE.....	1
PARTIE I : PRESENTATIONS	2
Chapitre 1. Présentation de l'Ecole Nationale d'Informatique	3
1.1. Information d'ordre générale.....	3
1.2. Missions et historiques.....	3
1.3. Organigramme institutionnel de l'ENI	6
1.4. Domaines et spécialisation	7
1.5. Architecture des formations pédagogiques	8
1.6. Relation de l'ENI avec les entreprises et organismes.....	10
1.7. Partenariat au niveau international	11
1.8. Débouchés professionnels des diplômés	13
1.9. Ressources humaines	16
Chapitre 2. Présentation de MyAgency	17
2.1. Historiques et missions	17
2.1. Situation juridique	17
2.2. Organigramme.....	17
2.3. Services.....	18
2.4. Contacts	18
Chapitre 3. Description du projet.....	19
3.1. Formulation.....	19
3.1. Objectif et besoin d'utilisateur.....	19
3.2. Moyen nécessaire à la réalisation du projet	20
PARTIE II : ANALYSE ET CONCEPTION	24
Chapitre 4. Analyse préalable.....	25
4.1. Analyse de l'existant	25
4.2. Critique de l'existant	26
4.3. Conception avant projet.....	26
Chapitre 5. Analyse Conceptuelle	51
5.1. Etapes de l'élaboration du backlog de produit.....	51

5.2. Dictionnaire de données	64
5.3. Règle de gestion	69
5.4. Capture des besoins fonctionnels	70
5.5. Analyse	70
5.6. Capture des besoins techniques.....	77
5.7. Conception préliminaire	77
Chapitre 6. Conception détaillée.....	85
6.1. Architecture du système	85
6.2. Diagramme de séquence de conception.....	87
6.3. Diagramme de classe de conception	88
6.4. Diagramme de classe de conception global	89
6.5. Diagramme de paquetage	90
6.6. Diagramme de déploiements.....	91
PARTIE III : REALISATION.....	92
Chapitre 7. Mise en place de l'environnement de développement	93
7.1. Installation et configuration des outils	93
7.2. Architecture de l'application.....	98
Chapitre 8. Développement de l'application.....	99
8.1. Création de la base de données	99
8.2. Codage de d'application	100
8.3. Présentation de l'application.....	107
CONCLUSION	XI
BIBLIOGRAPHIE	XII
WEBOGRAPHIE.....	XIII
GLOSSAIRE.....	XIV
TABLE DES MATIERES	XV
RESUME	XVIII
ABSTRACT.....	XVIII

REMERCIEMENTS

Avant tout, je tiens à remercier Dieu tout puissant de m'avoir donné la force, le courage et surtout la santé, afin de mener à bien tout ce travail. J'adresse également mes vifs remerciements à :

- Monsieur **HAJALALAINA Aimé Richard**, Docteur HDR, et Président de l'Université de Fianarantsoa de m'avoir accueilli au sein de l'Université de Fianarantsoa ;
- Monsieur **MAHATODY Thomas**, Directeur de l'Ecole Nationale de l'Informatique qui m'a donné la permission d'effectuer mon stage, pour avoir accepté mon inscription au sein de son établissement et de m'a donné la permission d'effectuer mon stage.
- Monsieur **TOKY RABESON**, fondateur de la société MYAGENCY, de m'avoir donné l'opportunité d'intégrer à son entreprise.
- Monsieur **ENCADREUR**, Assistants d'Enseignement Supérieur et de Recherche, mon encadreur pédagogique, pour son organisation et son soutien au bon déroulement ;
- Monsieur **ANJARANOTOAVINA Mahery Harentsoa**, Lead développeur chez MYAGENCY, mon encadreur professionnel, pour son soutien et ses conseils ainsi que l'aide précieux qu'il a donné qui a abouti à l'accomplissement de ce mémoire ;
- Tous les membres de jury ici présent d'avoir pu consacrer leur temps pour l'examen de ce mémoire malgré leur lourde responsabilité ;
- Tout le personnel de l'entreprise MYAGENCY
- Tous les professeurs et enseignants de l'Ecole Nationale de l'Informatique pour leurs cours et les connaissances illimitées qu'ils nous ont transmises.

Je tiens pareillement à exprimer mes remerciements à toute ma famille plus particulièrement mes parents et mes proches qui m'ont soutenu moralement ainsi que financièrement durant le déroulement du stage.

LISTE DES FIGURES

Figure 1. Organigramme de l'ENI	6
Figure 2:La présente l'architecture des études correspondant au système LMD.....	9
Figure 3:Organigramme de la société MyAgency	17
Figure 4: Chronogramme des travaux:	23
Figure 5:Tableau de bord de Trello	46
Figure 6:Espace de travail de JIRA.....	47
Figure 7:Manifeste agile	48
Figure 8: Étapes de SCRUM.....	49
Figure 9: Diagrammes de cas d'utilisation.....	72
Figure 10: PU centré sur l'architecture	29
Figure 11: itératif et incrémental.....	Erreur ! Signet non défini.
Figure 12: Systèmes d'information soumis à deux natures de contraintes	31
Figure 13: Processus de développement en Y	32
Figure 14: Séquence de système pour le cas d'utilisation s'authentifier	78
Figure 15: Diagramme de séquence système du cas d'utilisation gérer utilisateur	79
Figure 16: Diagramme de séquence système du cas d'utilisation gérer catégorie de projet.....	79
Figure 17:Diagramme de séquence s'inscrire	80

LISTE DES TABLEAUX

Tableau 1. Organisation du système de formation pédagogique de l'école	8
Tableau 2 Listes des formations existantes à l'ENI	9
Tableau 3:Débouchés professionnels éventuels des diplômés	14
Tableau 4 Moyens humains	20
Tableau 5: Inventaire des moyens matériels	21
Tableau 6: Inventaire des moyens logiciels	22
Tableau 7:Inventaire des matériels de l'existant.....	25
Tableau 8: Comparaison des solutions proposées.....	26
Tableau 9 Comparaison entre MERISE et Processus unifié.....	28
Tableau 10 présente la Comparaison des outils de conception.	34
Tableau 10 Comparaison des outils de conception	34
Tableau 11 Tableau comparatif entre MySQL et PostgreSQL.....	35
Tableau 12 Comparatif des IDE.....	40
Tableau 13 Comparaison entre Git et SVN	41
Tableau 14 Comparatif des langages de programmation	42
Tableau 15 Comparaison entre les frameworks PHP	45
Tableau 16: backlog de produit.....	63

LISTE DES ABREVIATIONS

2TUP	: Two Track Unified Process
AUF	: Agence Universitaire de la Francophonie
B2B	: Business to Business
BPMN	: Business Process Model and Notation
CARI	: Colloque Africain sur la Recherche en Informatique
CCNA	: Cisco Certified Network Associate Routing and Switching
CHU	: Centre Hospitalier Universitaire
CNH	: Commission Nationale d’Habilitation
CNRE	: Centre National de Recherche sur l’Environnement
CNRICT	: Centre National de Recherches Industrielle et Technologique
CPU	: Central Processing Unit
CSS	: Cascading Style Sheets
CUR	: Centre Universitaire Régional
DD	: Disque Dur
ESPA	: Ecole Supérieur Polytechnique d’Antananarivo
FID	: Fonds d’Intervention pour le développement
FTM	: Foibe Tao-Tsaritanin’i Madagasikara
HDD	: Hard Disk Drive
HP	: Hewlett Packard
HTML	: Hypertext Markup Language
INPG	: Institut National Polytechnique de Grenoble
INSA	: Institut National des Sciences Appliquées de Lyon
IRD	: Institut de Recherche pour le développement
JS	: JavaScript
JSP	: Java Server Page
LMD	: Licence Master Doctorat
MEF	: Ministère de l’Economie et des Finances
MEN	: Ministère de l’Education Nationale
MeSupRES	: Ministère de l’Enseignement supérieur et de recherche scientifique
MFB	: Ministère des Finances et du Budget
MIC	: Madagascar Informatique Corporation
MVC	: Modèle Vue Contrôleur

OS	: Operating System
PHP	: Hypertext Preprocessor
PRESUP	: Programme de renforcement de l'Enseignement Supérieur
PU	: Processus Unifié
RAM	: Random Access Memory
SGBD	: Système de Gestion de Base de Donnée
SSII	: Société de Services en Ingénierie Informatique
UML	: Unified Modeling Language
UP	: Unified Process
UPST	: Université Paul Sabatier de Toulouse
VB	: Visual Basic
VOIP	: Voice Over IP
VP	: Visual Paradigm

INTRODUCTION GENERALE

L'identification et la sélection de bons projets d'investissement est un élément clé de développer un avenir prospère et durable. La décision d'aller de l'avant avec une bonne ou de mauvais projets, plus qu'impactant le profil économique de l'entreprise à court terme, aura tendance à avoir un impact durable sur la rentabilité à long terme. En effet on a une idée d'inventer une plateforme en ligne pour que l'utilisateur peut apporter son propre idée à un projet et aussi y investir facilement et rapide sans déplacement.

C'est à l'aide de cette idée que 'est traduite la création de projet Invest, qui consiste à concevoir et à réaliser une application web pour la gestion d'investissement en ligne

La réussite de ce projet comme n'importe quel autre dépend de la méthode adoptée. En général, il faut commencer par étudier le système existant sur lequel travailler et chercher ses avantages et ses éventuels inconvénients, ensuite explorer diverses solutions afin de retenir la meilleure et celle adaptée au système, et enfin mettre en place des améliorations ou un nouveau système après avoir procédé par des tests justifiant son bon fonctionnement.

Pour mener à bien ce projet, une méthode sera adoptée. On va également devoir choisir un langage de programmation, un système de gestion de base de données, un système de gestion de version et un outil de gestion de projet.

Nous allons donc développer ces principes en commençant par les présentations générales suivies de la partie analyse et conception, et enfin la réalisation

PARTIE I : PRESENTATIONS

Chapitre 1. Présentation de l'Ecole Nationale d'Informatique

On va entrer en détails au présentation de l'école nationale de l'informatique à cette chapitre.

1.1. Information d'ordre générale

L'Ecole Nationale d'Informatique, en abrégé ENI, est un établissement d'enseignement supérieur rattaché académiquement et administrativement à l'Université de Fianarantsoa.

Le siège de l'Ecole se trouve à Tanambao-Antaninarenina à Fianarantsoa.

L'adresse pour la prise de contact avec l'Ecole est la suivante : Ecole Nationale d'Informatique (ENI) Tanambao, Fianarantsoa. Le numéro de sa boîte postale est 1487 avec le code postal 301. Téléphone : 0340573336. Son adresse électronique est la suivante : eni@univ-fianar.mg. Site Web : www.univ-fianar.mg/eni

1.2. Missions et historiques

L'ENI se positionne sur l'échiquier socio-éducatif malgache comme étant le plus puissant secteur de diffusion et de vulgarisation des connaissances et des technologies informatiques.

Cette Ecole Supérieure peut être considérée aujourd'hui comme la vitrine et la pépinière des élites informaticiennes du pays.

L'Ecole s'est constituée de façon progressive au sein du Centre Universitaire Régional (CUR) de Fianarantsoa.

De façon formelle, l'ENI était constituée et créé au sein du (CUR) par le décret N° 83185 du 24 Mai 1983, comme étant le seul établissement Universitaire Professionnalisé au niveau national, destiné à former des techniciens et des Ingénieurs de haut niveau, aptes à répondre aux besoins et exigences d'Informatisation des entreprises, des sociétés et des organes implantés à Madagascar.

L'ENI a par conséquent pour mission de former des spécialistes informaticiens compétents et opérationnels de différents niveaux notamment :

- En fournissant à des étudiants des connaissances de base en informatique ;
- En leur transmettant le savoir-faire requis, à travers la professionnalisation des formations dispensées et en essayant une meilleure adéquation des formations par rapport aux besoins évolutifs des sociétés et des entreprises.
- En initiant les étudiants aux activités de recherche dans les différents domaines des
 - Technologies de l'information et de la communication (TIC).

L'implantation de cette Ecole Supérieure de technologie de pointe dans un pays en développement et dans une Province (ou Faritany) à tissu économique et industriel faiblement développé ne l'a pourtant pas défavorisée, ni empêchée de former des spécialistes informaticiens de bon niveau, qui sont recherchés par les entreprises, les sociétés et les organismes publics et privés sur le marché de l'emploi.

La filière de formation d'Analystes Programmeurs a été mise en place à l'Ecole en 1983, et a été gelée par la suite en 1996, tandis que la filière de formation d'ingénieurs a été ouverte à l'Ecole en 1986.

Dans le cadre du Programme de renforcement en l'Enseignement Supérieur (PRESUP), la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes des informatiques a été mise en place en 1986 grâce à l'appui matériel et financier de la Mission Française de coopération auprès de l'Ambassade de France à Madagascar.

Une formation pour l'obtention de la certification CCNA et / ou NETWORK + appelée « CISCO Networking Academy » a été créée à l'Ecole en 2002-2003 grâce au partenariat avec CISCO SYSTEM et l'Ecole Supérieure Polytechnique d'Antananarivo (ESPA). Cependant, cette formation n'avait pas duré longtemps.

Une formation de troisième cycle a été ouverte à l'Ecole depuis l'année 2003 – 2004 grâce à la coopération académique et scientifique entre l'Université de Fianarantsoa pour le compte de l'ENI et l'Université Paul Sabatier de Toulouse (UPST). Cette filière avait pour objectif de former certains étudiants à la recherche dans les différents domaines de l'Informatique, et notamment pour préparer la relève des Enseignants-Chercheurs qui étaient en poste.

Pendant l'année 2007-2008, la formation en vue de l'obtention du diplôme de Licence Professionnelle en Informatique a été mise en place à l'ENI avec les deux options suivantes de formation :

- Génie Logiciel et base de Données ;
- Administration des Système et réseaux.

La mise en place à l'Ecole de ces deux options de formation devait répondre au besoin de basculement vers le système Licence – Master – Doctorat (LMD). Mais la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes Informatiques a été gelée en 2009.

En vue de surmonter les difficultés de limitation de l'effectif des étudiants accueillis

à L'Ecole, notamment à cause du manque d'infrastructures, un système de « Formation Hybride » a été mise en place à partir de l'année 2010. Il s'agit en effet d'un système de formation semi-présentielle et à distance avec l'utilisation de la visioconférence pour la formation à distance. Le système de formation hybride a été ainsi créé à Fianarantsoa ainsi qu'Université de Toliara.

1.3. Organigramme institutionnel de l'ENI

Cet organigramme de l'Ecole est inspiré des dispositions du décret N° 83-185 du 23 Mai 1983.

L'ENI est administrée par un conseil d'Ecole, et dirigée par un directeur nommé par un décret adopté en conseil des Ministres. Le Collège des enseignants regroupant tous les enseignants-chercheurs de l'Ecole est chargé de résoudre les problèmes liés à l'organisation pédagogique des enseignements ainsi que à l'élaboration des emplois du temps. Le Conseil Scientifique propose les orientations pédagogiques et scientifiques de l'établissement, en tenant compte notamment de l'évolution du marché de travail et de l'adéquation des formations dispensées par rapport aux besoins des entreprises.

La figure 1 présente l'organigramme actuel de l'Ecole.

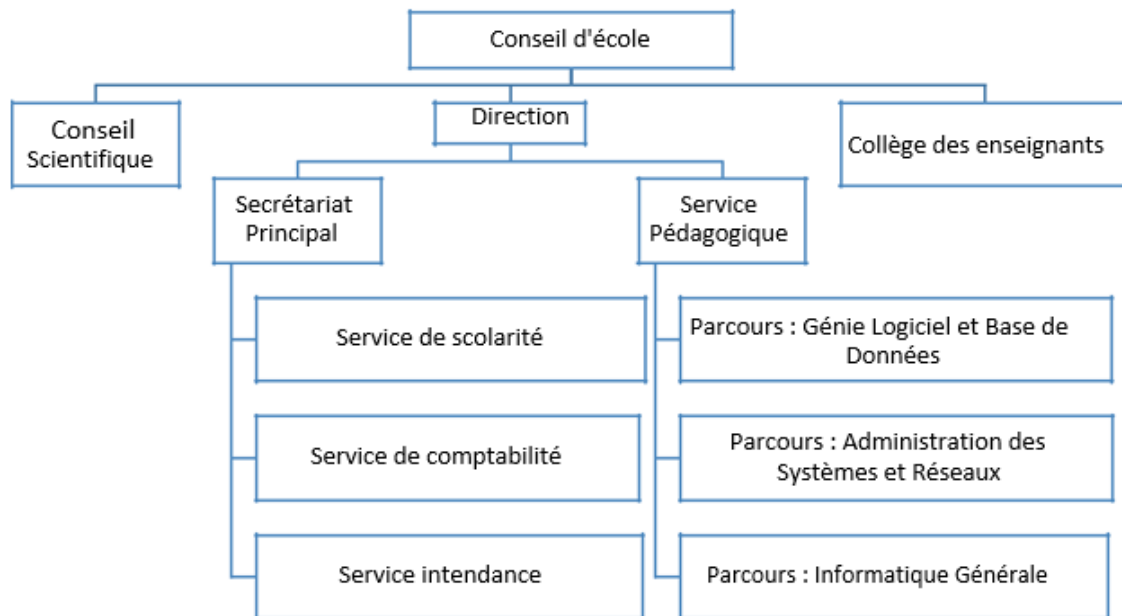


Figure 1. Organigramme de l'ENI

Sur cet organigramme, l'Ecole est placée sous la tutelle académique et administrative de l'Université de Fianarantsoa, et dirigée par un Directeur élu par les Enseignants – Chercheurs permanents de l'Etablissement et nommé par un décret pris en Conseil des ministres pour un mandat de 3 ans.

- Le Conseil de l'Ecole est l'organe délibérant de l'Ecole ;
- Le Collège des Enseignants propose et coordonne les programmes d'activités pédagogiques ;

- Le Conseil scientifique coordonne les programmes de recherche à mettre en œuvre à l'Ecole ;
- Le Secrétariat principal coordonne les activités des services administratifs (Scolarité, Comptabilité, et Intendance).

Conformément aux textes en vigueur régissant les Etablissements malgaches d'Enseignement Supérieur, qui sont barrés sur le système LMD, les Départements de Formation pédagogique ont été ainsi remplacés par des Mentions et des parcours. Et les chefs des Départements ont été ainsi remplacés par des responsables des mentions et les responsables des parcours.

- Un administrateur des Réseaux et Systèmes gère le système d'information de l'Ecole et celui de l'Université.

1.4. Domaines et spécialisation

Les activités de formation et de recherche organisées à l'ENI portent sur les domaines suivants :

- Génie logiciel et Base de Données ;
- Administration des Systèmes et Réseaux ;
- Informatique Générale ;
- Modélisation informatique et mathématique des Systèmes complexes.

D'une manière plus générale, les programmes des formations sont basés sur l'informatique de gestion et sur l'informatique des Systèmes et Réseaux. Et les modules de formation intègrent aussi bien des éléments d'Informatique fondamentale que des éléments d'Informatique appliquée.

Le tableau 1 décrit l'organisation du système de formation pédagogique de l'Ecole.

Tableau 1. Organisation du système de formation pédagogique de l'école

Formation Théorique	Formation pratique
<ul style="list-style-type: none"> - Enseignement théorique - Travaux dirigés - Travaux pratiques - Conférences 	<ul style="list-style-type: none"> - Etude de cas - Travaux de réalisation - Projets/ Projets tutoriels - Voyages d'Etudes - Stages en entreprise

1.5. Architecture des formations pédagogiques

Le recrutement des étudiants à l'ENI se fait uniquement par voie de concours d'envergure nationale en première année.

Les offres de formation organisées à l'Ecole ont été validées par la Commission Nationale d'Habilitation (CNH) auprès du Ministère de l'Enseignement Supérieur et de la Recherche Scientifique selon les dispositions de l'Arrêté N°31.174/2012-MENS en date du 05 Décembre 2012.

Au sein de l'ENI, il existe une seule mention (INFORMATIQUE) et trois parcours :

- Génie logiciel et Base de données ;
- Administration des Systèmes et Réseaux ;
- Informatique Générale.

L'architecture des études à trois niveaux conformément au système Licence- Master-Doctorat (LMD) permet les comparaisons et les équivalences académiques des diplômes au niveau international.

- L = Licence (Bac + 3) = L1, L2, L3 = 6 semestres S1 à S6 ;
- M = Master (Bac + 5) = M1, M2 = 4 semestres S7 à S10;

Le diplôme de licence est obtenu en 3 années des études après Baccalauréat. Et le diplôme de Master est obtenu en 2 ans après obtention du diplôme de LICENCE. Le MASTER PROFESSIONNEL est un diplôme destiné à la recherche emploi au terme des études.

Le MASTER RECHERCHE est un diplôme qui remplace l'ancien Diplôme d'Etudes Approfondies (DEA), et qui permet de s'inscrire directement dans une Ecole Doctoral en termes des études.

- D = Doctorat (Bac +8).

Le Doctorat est un diplôme qu'on peut obtenir en 3 ans après l'obtention du diplôme de Master recherche.

La figure 2 présente l'architecture des études correspondant au système LMD.

Figure 2: La présente l'architecture des études correspondant au système LMD.



La licence peut avoir une vocation générale ou professionnelle.

Le master peut avoir une vocation professionnelle ou de recherche.

Le Tableau 2 décrit Les Listes des formations en existantes à l'ENI.

Tableau 2 Listes des formations existantes à l'ENI

	FORMATION	
	LICENCE PROFESSIONNELLE	MASTER
Condition d'admission	Par voie de concours Formation Professionnelle : 100 candidats Formation hybride : 150 candidats	
Condition d'Accès	Bac de série C, D ou Technique	Etre titulaire de licence professionnelle
Durée de Formation	3 ans	2 ans
Diplôme délivré	Diplôme de Licence Professionnelle	Diplôme de Master Professionnel Diplôme de Master Recherche

L'accès en première année de MASTER se fait automatiquement pour les étudiants de l'Ecole qui ont obtenu le diplôme de Licence Professionnelle.

Le Master Recherche permet à son titulaire de poursuivre directement des études en doctorat et de s'inscrire directement dans une Ecole Doctorale.

Les Ecoles Doctorales jouissent d'une autonomie de gestion par rapport aux Etablissements de formation universitaire.

Il convient de signaler que par arrêté ministériel N° 21.626/2012 – MESupRES publié le 9 Août 2012 par la Commission National d'habilitation (CNH), l'Ecole Doctorale « Modélisation – Informatique » a été habilitée pour l'Université de Fianarantsoa.

Depuis l'année universitaire 2010-2011, l'ENI s'est mise à organiser des formations hybrides en informatique dans les différentes régions (Fianarantsoa, Toliara) en raison de l'insuffisance de la capacité d'accueil des infrastructures logistiques. En effet, le système de formation hybride semi - présentielle utilise la visioconférence pour la formation à distance. Bien qu'il n'existe pas encore au niveau international de reconnaissance écrite et formelle des diplômes délivrés par l'ENI, les étudiants diplômés de l'Ecole sont plutôt bien accueillis dans les instituts universitaires étrangères (CANADA, Suisse, France...)

1.6. Relation de l'ENI avec les entreprises et organismes

Les stages effectués chaque année par les étudiants mettent l'Ecole en rapport permanent avec plus de 300 entreprises et organismes publics, semi-publics et privés, nationaux et internationaux.

L'Ecole dispose ainsi d'un réseau d'entreprises, de sociétés et d'organismes publics et privés qui sont des partenaires par l'accueil en stage de ses étudiants, et éventuellement pour le recrutement après l'obtention des diplômes par ces derniers.

Les compétences que l'Ecole cherche à développer chez ses étudiants sont l'adaptabilité, le sens de la responsabilité, du travail en équipe, le goût de l'expérimentation et l'innovation.

En effet, la vocation de l'ENI est de former des techniciens supérieurs de niveau LICENCE et des ingénieurs de type généraliste de niveau MASTER avec des qualités scientifiques, techniques et humaines reconnues, capables d'évoluer professionnellement dans des secteurs d'activité variés intégrant l'informatique.

Les stages en milieu professionnel permettent de favoriser une meilleure adéquation entre les formations à l'Ecole et les besoins évolutifs du marché de l'emploi.

Les principaux débouchés professionnels des diplômés de l'Ecole concernent les domaines suivants :

- L'informatique de gestion d'entreprise
- Les technologies de l'information et de la communication (TIC)
- La sécurité informatique des réseaux
- L'administration des réseaux et des systèmes
- Les services bancaires et financiers, notamment le Mobile Banking
- Les télécommunications et la téléphonie mobile
- Les Big Data
- Le commerce, la vente et l'achat, le Marketing
- L'ingénierie informatique appliquée
- L'écologie et le développement durable

Parmi les sociétés, entreprises et organismes partenaires de l'Ecole, on peut citer : ACCENTURE Mauritius, Air Madagascar, Ambre Associates, Airtel, Agence Universitaire de la Francophonie (AUF) , B2B, Banque Centrale, BFG-SG, BIANCO, BLUELINE, CNaPS, Bureau National de Gestion des Risques et des Catastrophes (BNGRC), CEDII-Fianarantsoa, Data Consulting, Central Test, Centre National Antiacridien, CNRE, CHU, CNRIT, COLAS, Direction Générale des Douanes, DLC, DTS/Moov, FID, FTM, GNOSYS, IBONIA, INGENOSIA, INSTAT, IOGA, JIRAMA, JOUVE, MADADEV, MAEP, MEF, MEN, MESupRES, MFB, MIC, MNINTER, Min des postes/Télécommunications et du Développement Numérique, NEOV MAD, Ny Havana, Madagascar National Parks, OMNITEC, ORANGE, OTME, PRACCESS, QMM Fort-Dauphin, SMMC, SNEDADRS Antsirabe, Sénat, Société d'Exploitation du Port de

Toamasina (SEPT), SOFTWELL, Strategy Consulting, TELMA, VIVETEC, Société LAZAN'I BETSILEO, WWF ...

L'organisation de stage en entreprise continue non seulement à renforcer la professionnalisation des formations dispensées, mais elle continue surtout à accroître de façon exceptionnelle les opportunités d'embauche pour les diplômés de l'Ecole.

1.7. Partenariat au niveau international

Entre 1996 et 1999, l'ENI avait bénéficié de l'assistance technique et financière de la Mission Française de Coopération et d'action culturelle dans le cadre du Programme de

Renforcement de l'Enseignement Supérieur (PRESUP) consacré à l'Ecole a notamment porté sur :

- Une dotation en logiciels, micro-ordinateurs, équipements de laboratoire de maintenance et de matériels didactiques
- La réactualisation des programmes de formation assortie du renouvellement du fonds de la bibliothèque
- L'appui à la formation des formateurs
- L'affectation à l'Ecole d'Assistants techniques français

De 2000 à 2004, l'ENI avait fait partie des membres du bureau de la Conférence Internationale des Ecoles de formation d'Ingénieurs et Technicien d'Expression Française (CITEF).

Les Enseignants-Chercheurs de l'Ecole participent régulièrement aux activités organisées dans le cadre du Colloque Africain sur la Recherche en Informatique (CARI).

L'ENI avait également signé un accord de coopération inter-universitaire avec l'Institut de Recherche en Mathématiques et Informatique Appliquées (IREMIA) de l'Université de la Réunion, l'Université de Rennes 1, l'INSA de Rennes, l'Institut National Polytechnique de Grenoble (INPG).

A partir du mois de Juillet 2001, l'ENI avait abrité le Centre de Réseau Opérationnel (Network Operating Center) du point d'accès à Internet de l'Ecole ainsi que de l'Université de Fianarantsoa. Grâce à ce projet américain qui a été financé par l'USAID Madagascar, l'ENI de l'Université de Fianarantsoa avait été dotées d'une ligne spécialisée d'accès permanent au réseau Internet.

L'ENI avait de même noué des relations de coopération avec l'Institut de Recherche pour le Développement (IRD).

L'objet du projet de coopération avait porté sur la modélisation environnementale du Corridor forestier de Fandriana jusqu'à Vondrozo (COFAV). Dans ce cadre, un atelier scientifique international avait été organisé à l'ENI en Septembre 2008. Cet atelier scientifique avait eu pour thème de modélisation des paysages.

Et dans le cadre du programme scientifique PARRUR, l'IRD avait financé depuis 2010 le projet intitulé « Forêts, Parcs et Pauvreté dans le Sud de Madagascar (FPPSM). Des étudiants en DEA et des Doctorants issus de l'ENI avaient participé à ce Programme.

Par ailleurs, depuis toujours la même année 2010, l'ENI de Fianarantsoa avait été sélectionnée pour faire partie des organismes partenaires de l'Université de Savoie dans le cadre du projet TICEVAL relatif à la certification des compétences en TIC ;

Le projet TICEVAL avait été financé par le Fonds Francophone des Inforoutes pour la période allant de 2010 à 2012, et il avait eu pour objectif de généraliser la certification des compétences en Informatique et Internet du type C2i2e et C2imi.

Dans le cadre du projet TICEVAL, une convention de coopération avec l'Université de Savoie avait été signée par les deux parties concernées. La mise en œuvre de la Convention de Coopération avait permis d'envoyer des étudiants de l'ENI à Chambéry pour poursuivre des études supérieures en Informatique.

Enfin et non des moindres, l'ENI avait signé en Septembre 2009 un protocole de collaboration scientifique avec l'ESIROI – STIM de l'Université de la Réunion.

Comme l'ENI constitue une pépinière incubatrice de technologie de pointe, d'emplois et d'entreprises, elle peut très bien servir d'instrument efficace pour renforcer la croissance économique du pays, et pour lutter contre la Pauvreté.

De même que le statut de l'Ecole devrait permettre de renforcer la position concurrentielle de la Grande Ile sur l'orbite de la modélisation grâce au développement des nouvelles technologies.

1.8. Débouchés professionnels des diplômés

Le chômage des jeunes diplômés universitaires fait partie des maux qui gangrènent Madagascar. L'environnement socio-politique du pays depuis 2008 jusqu'à ce jour a fait que le chômage des diplômés est devenu massif par rapport aux établissements de formation supérieure existants.

Cependant, les formations proposées par l'Ecole permettent aux diplômés d'être immédiatement opérationnels sur le marché du travail avec la connaissance d'un métier complet lié à l'informatique aux TIC.

L'Ecole apporte à ses étudiants un savoir-faire et un savoir-être qui les accompagnent tout au long de leur vie professionnelle. Elle a une vocation professionnalisante.

Les diplômés en LICENCE et en MASTER issus de l'ENI peuvent faire carrière dans différents secteurs.

L'Ecole bénéficie aujourd'hui de 34 années d'expériences pédagogiques et de reconnaissance auprès des sociétés, des entreprises et des organismes. C'est une Ecole Supérieure de référence en matière informatique.

Par conséquent, en raison de fait que l'équipe pédagogique de l'Ecole est expérimentée, les enseignants-chercheurs et les autres formateurs de l'Ecole sont dotés d'une grande expérience dans l'enseignement et dans le milieu professionnel.

L'Ecole est fière de collaborer de façon régulière avec un nombre croissant d'entreprises, de sociétés et d'organismes publics et privés à travers les stages des étudiants. Les formations dispensées à l'Ecole sont ainsi orientées vers le besoin et les attentes des entreprises et des sociétés.

L'Ecole fournit à ses étudiants de niveau LICENCE et MASTER des compétences professionnelles et métiers indispensables pour les intégrer sur le marché du travail.

L'Ecole s'efforce de proposer à ses étudiants une double compétence à la fois technologique et managériale combinant l'informatique de gestion ainsi que l'administration des réseaux et systèmes.

D'une manière générale, les diplômés de l'ENI n'éprouvent pas de difficultés particulières à être recrutés au terme de leurs études. Cependant, l'ENI recommande à ses diplômés de promouvoir l'entrepreneuriat en TIC et de créer des cybercafés, des SSII ou des bureaux d'études.

Tableau 3: Débouchés professionnels éventuels des diplômés

LICENCE	<ul style="list-style-type: none"> - Analyste - Programmeur - Administrateur de Site Web/ de portail Web - Assistant Informatique et Internet - Chef de projet Web ou MultiMedia - Développeur Informatique ou MultiMedia - Intégrateur Web ou Web Designer - Hot Liner/ Hébergeur Internet - Agent de référencement - Technicien/Supérieur de help desk sur Informatique - Responsable de sécurité web - Administrateur de réseau - Administrateur de cybercafé
---------	---

MASTER	<ul style="list-style-type: none"> - Administrateur de cyber café - Administrateur de réseau et système - Architecture de système d'information - Développeur d'application /web /java/Python/ IOS /Android - Ingénieur réseau - Webmaster /web designer - Concepteur Réalisateur d'applications - Directeur du système de formation - Directeur de projet informatique - Chef de projet informatique - Responsable de sécurité informatique - Consultant fonctionnel ou freelance
--------	--

1.9. Ressources humaines

Les ressources humaines affectées à l'Ecole Nationale d'Informatique sont composées de :

- Directeur de l'Ecole : Professeur MAHATODY Thomas
- Responsable de Mention : Docteur RABETAFIKA Louis Haja
- Responsable de Parcours « Génie Logiciel et Base de Données » : Monsieur RALAIVAO Jean Christian
- Responsable de Parcours « Administration Systèmes et Réseaux » : Monsieur SIAKA
- Responsable de Parcours « Informatique Générale » : Monsieur GILANTE Gesazafy
- Nombre d'Enseignants permanents : 12 dont un (01) Professeur Titulaire, un (02) Professeur, Cinq (04) Maîtres de Conférences et cinq (05) Assistants d'Enseignement Supérieur et de Recherche.
- Nombre d'Enseignants vacataires : 10
- Personnel Administratif : 23

Chapitre 2. Présentation de MyAgency

On va voir maintenant la Présentation de la société MyAgency à cette chapitre.

2.1. Historiques et missions

La société MyAgency est un Agence web, forte d'une dizaine (13 ans) d'années d'expériences,

basé à Antananarivo Madagascar (GMT +3). <https://www.myagency.mg>.

C'est une entreprise au service du numérique spécialisé dans la transformation numérique à Madagascar.

2.1. Situation juridique

La situation Juridique de la société MyAgency est énuméré ci-dessous :

Raison Sociale	:	MyAgency
Statut	:	MyAgency
Numéro d'identification Fiscale	:	4 0004 256 23
Numéro Statistique	:	62011 11 202012100
Siège	:	Anjanahary Antananarivo

2.2. Organigramme

L'organigramme de la société MyAgency est représenté su la figure 3.

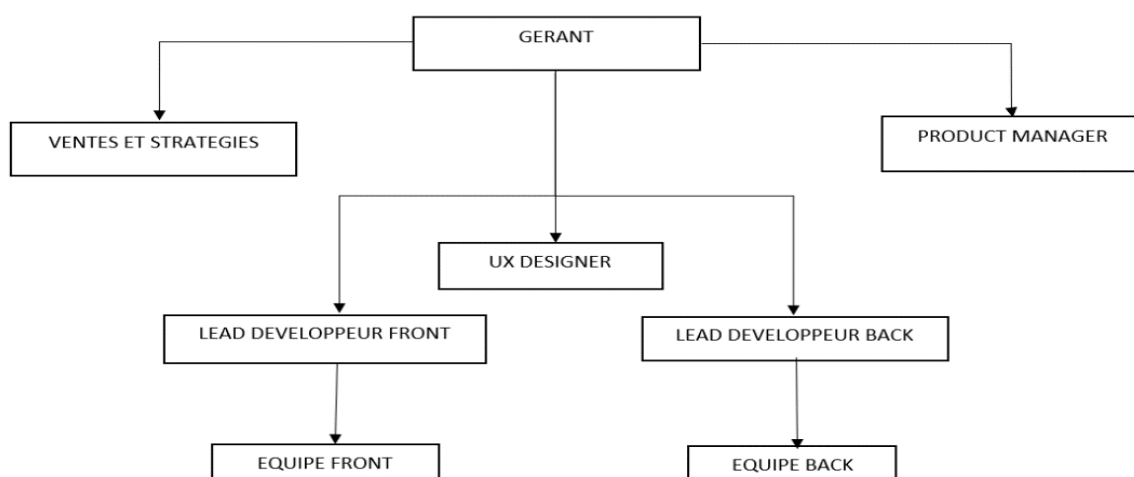


Figure 3: Organigramme de la société MyAgency

2.3. Services

Elle propose les services suivants :

- Elaboration de terme de référence ;
- Rédaction des spécifications techniques / cahiers des charges ;
- Création de Site vitrine, Site catalogue, Site e-commerce ;
- Hébergement de site ;
- Développement d'Application Web, Application Desktop, Application mobile ;
- Externalisation de données.

2.4. Contacts

Personne habilité à représenter la société :

Nom	:	Toky Rabeson ;
Numéro de téléphone	:	+2613486834 ;
Adresse électronique	:	contact@myagency.mg;
Site web	:	https://www.myagency.mg .

Chapitre 3. Description du projet

La description du projet est primordiale avant d'entamer l'analyse et conception. Ce chapitre nous permettra de décrire le projet, spécifier les objectifs du projet, organiser le projet.

3.1. Formulation

Pour avoir des idées à un projet et aussi investir à des projets intéressant. On a créé une application web pour la gestion d'investissement à un projet en ligne.

Pour ce faire, l'Analyse et la conception sont entamées par l'équipe de MyAgency

Ainsi le travail consiste à donner vie à ce projet et de satisfaire le client d'où le projet de conception et réalisation d'une application web pour la gestion d'investissement d'un projet en ligne.

3.1. Objectif et besoin d'utilisateur

On va illustrer maintenant quelle est l'objectif et le besoin d'utilisateur en fin d'avoir une résultat que ne serai pas hors sujet.

3.1.1. Objectif du projet

L'objectif est de concevoir et réaliser une application web pour la création et investissement d'un projet en ligne

3.1.2. Besoin de l'utilisateur

- S'authentifier ;
- Gestion des Projets ;
- Paiement en ligne pour postuler un projet ;
- Génération des factures après chaque paiement ;
- Gestion des Investissements ;
- Gestion de détail de projet et contrat par l'investisseur ;
- Création de fichier PDF contenant le paiement ;
- Notification Par Email ;

3.2. Moyen nécessaire à la réalisation du projet

Plusieurs Moyens est indispensable pour nous aider à la réalisation de ce projet . Ce partie nous permettra de présenter ces moyens nécessaire.

3.2.1. Moyens humains

Vue que la société travail en agile, l'équipe doit contenir un Product owner , un scrum master ,une équipe de développement et de tests qui sont représentés sur le tableau.

Tableau 4 décrit Les Moyens humains.

Tableau 4 Moyens humains

Fonction	Nombre	Commentaire
Product Owner	1	Représente le client
Chef de projet	1	Dirige le projet
Développeur	5	2 développeurs back end 1 développeur front end 2 intégrateurs
Designer	1	S'occupe de la conception visuel
Testeur	1	Tester l'application

3.2.2. Moyens Matériels

Tableau 5 décrit Les Inventaire des moyens matériels

Tableau 5: Inventaire des moyens matériels

Désignation	Caractéristique	Quantité	Utilité lors du projet
Ordinateur portable	Marque : LAPTOP ONKYO CPU : intel core i7-3820QM RAM : 32GB DD : 500 GB SSD + 128 GB SSD,	2	Réalisation du back
Ordinateur portable	Marque : Asus ROG CPU : corei7 RAM : 8 GB DD : 500 Go HDD	1	Réalisation du front
Ordinateur portable	Marque : Asus ROG CPU : corei7 DD : 500 Go HDD	1	Intégration
Ordinateur portable	MSI corei7 1To HDD	1	Design
Serveur VPS	CPU : Intel (R) Xeon ES-2650 V3 RAM : 4GB	1	Serveur pré prod
Imprimante	Imprimante HP ink Tank 410 series,	1	Impression des documents utilisé pour la mise en œuvre du projet
Internet	WIFIBER PRO prémium ORANGE	1	La connexion internet
Internet	FIBRE OPTIQUE TELMA	1	La connexion internet

3.2.3. Moyens logiciels

Le tableau n°6 représente les logiciels utilisés pour la mise en œuvre du projet

Tableau 6: Inventaire des moyens logiciels

Designation	Version	Utilité durant le stage
Robo mongo 3T	Neant	Administration de la base de données
MongoDB	4.4.4	Server de base de données
Visual studio Code	2020.1	IDE
Visual Paradigm	2017	Conception
Yarn	2	Gestionnaire de paquet Javascript
postman	10.8.1	Client http
Figma	10.0.12.14	Conception visuel de l'application
Just Color picker	5.5	Intégration palette couleur
skype	4.19.3	Communication
gitHub	Néant	Version
JIRA	néant	Gestion de projet
Google meet	Néant	Daily meeting
Gant Project	3.1.3102	Réalisation chronogramme de travaux

3.2.4. Résultats attendus

Les résultats attendus sont :

- Une application avec une interface intuitive ;
- Les données doivent être structurées ;
- L'accès à l'application doit être rapide et sécuriser ;
- L'application doit pouvoir évoluer c'est-à-dire que l'application doit être facile à maintenir ;
- L'application doit avoir une authentification ;
- L'application doit gérer de catégorie de projet ;
- L'application doit permettre à l'utilisateur de créer, postuler, investir à un projet ;
- L'application doit implémenter un formulaire de paiement en ligne ;
- L'application doit générer une facture en PDF pour chaque paiement .

3.2.5. Chronogrammes des travaux

Pour bien organiser les tâches à réaliser, nous avons suivi le planning représenté par le calendrier présenté dans la figure 3.

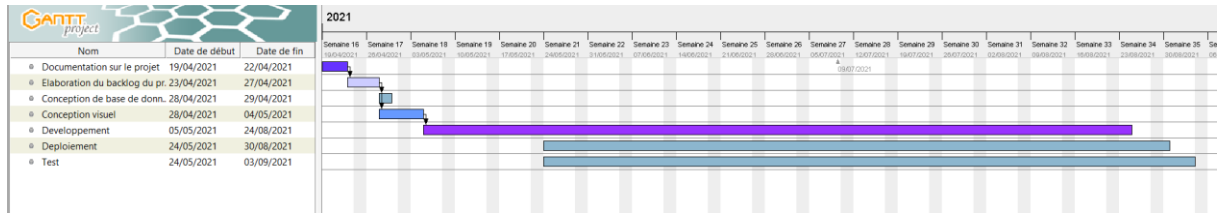


Figure 4: Chronogramme des travaux:

PARTIE II : ANALYSE ET CONCEPTION

Chapitre 4. Analyse préalable

La conception et la réalisation de ce projet nécessitent une analyse de la situation existante et une analyse conceptuelle afin de recueillir les informations nécessaires sur l'existant et les attentes de l'utilisateur.

4.1. Analyse de l'existant

Cette partie est d'une importance capitale. Elle va nous permettre de décrire toutes les facettes des fonctionnalités de notre système, tout en tenant compte d'une question pionnière à laquelle on répondra subséquemment : que fera notre système ainsi découvrir les critiques qu'on peut fournir à l'ancien système.

4.1.1. Organisation Actuelle

A l'heure actuelle il n'y a pas encore un logiciel de gestion d'un investissement d'un projet ligne chez MyAgency.

4.1.2. Inventaire des moyens matériels et logiciels

Le tableau 7 récapitule les moyens matériels et logiciels existant actuellement

Tableau 7: Inventaire des matériels de l'existant

Désignation	Caractéristique	Nombres	Usages	Type
Ordinateur	Core i5 RAM: 16Go HDD: 500Go	20	Pour les étudiants	Matériel
Site web	Néant	1	Diffusion d'information	Logiciel
Google meet	Néant	néant	Meeting	Logiciel

4.2. Critique de l'existant

Actuellement, il n'y a encore pas un logiciel de gestion d'Investissement de projet en ligne dans la société MyAgency.

4.3. Conception avant projet

Avant de faire la conception De comparaison de solution est indispensable enfin de résoudre les problème et aussi d'aider de faire de choix et de justification.

4.3.1. Comparaison de solutions

Afin de remédier aux problèmes, nous avons proposé 2 solutions :

Solution 1 : Commander un logiciel d'investissement auprès d'une entreprise concepteur de logiciel ;

Solution 2 : Développer et concevoir un logiciel sur mesure ;

Solution 3 : Télécharger un logiciel open-Source.

Tableau 8 Représente la comparaison des solutions proposées.

Tableau 8: Comparaison des solutions proposées

Solutions	Avantages	Inconvénients
Solution 1	Gains de temps	Cout élevé (des milliers d'euros) Perte de temps dans la recherche d'une entreprise
Solution 2	Sur mesure Adapté aux besoins Maintenable Donnée sécurisée	Temps de développement peut être élevé
Solution 3	Gains de temps	Une application qui ne peut pas s'adapter correctement à notre besoin.

La solution qui convient le plus à notre situation est la solution 2 qui est le développement d'un logiciel sur mesure, car les avantages cités sont meilleurs que les 2 autres solutions.

4.3.2. Choix et justification

À l'issue de la formulation des besoins, on termine par l'approfondissement des solutions. Et dans cette phase on procède aux choix des outils (moyen logiciel).

La phase consiste à étudier parmi un ensemble de solutions potentielles, la solution la plus ajustée sur des aspects fondamentaux.

Afin de bien développer notre projet avec rapidité et efficacité, nous devons d'abord choisir des outils pour nous aider, comme un serveur, un IDE, et même un outil de modélisation.

a) Méthode

Le tableau 9 montre la comparaison entre MERISE et le Processus unifié.

Tableau 9 Comparaison entre MERISE et Processus unifié

Méthodes	Avantages	Inconvénients
MERISE	<p>Le modèle en nombre limité (06) est progressivement élaboré, enrichi, et constitue des supports de communication et de participation pour les utilisateurs.</p> <p>MERISE permet de modéliser le métier de l'entreprise indépendamment des techniques, aux niveaux conceptuel et organisationnel.</p> <p>Cycle de vie séquentielle</p>	<p>Méthode non orientée objet</p> <p>MERISE n'exprime pas certaines notions (agrégation et composition)</p> <p>Merise s'appuie sur la décomposition des modules pour la règle de gestion alors que ceux-là sont difficilement extensibles et exploitables par de nouveaux systèmes.</p>
Processus unifié	<p>Piloté par le cas d'utilisation.</p> <p>Démarche centrée sur l'architecture.</p> <p>Démarche itérative et incrémentale.</p> <p>Cycle de vie incrémentale et itérative.</p> <p>Utilise le langage de modélisation UML.</p> <p>Exprime certaines notions (agrégation, composition)</p> <p>Ne préconise aucune démarche spécifique.</p>	<p>Les niveaux d'abstraction ne sont pas nettement distingués dans UP</p> <p>UP ne se préoccupe pas de normalisation (2^e ou 3^e forme normale).</p> <p>La mise en pratique de l'UP nécessite un apprentissage et passe par une période d'adaptation.</p>

On a choisi la méthode PU parce qu'elle utilise le langage de modélisation UML et aussi parce qu'elle permet d'avoir un cycle de vie itérative et incrémentale.

➤ Notion de processus unifié

Pour définir le processus unifié, on va simplement définir les 2 termes qui le composent :

- **Processus** : Suite continue d'opérations constituant la manière de fabriquer. En d'autres termes, c'est une succession de tâches dans le but d'accomplir un travail, un projet.
- **Unifié** : participe passé du verbe « unifier » être amené à l'unité, se fondre en un tout. En fait, les méthodes d'analyse et de conception orientée objet étaient variées jusqu'à ce que Rumbaugh, Jacobson et Booch aient eu l'idée de les unifier.

Le processus unifié s'appuie sur les principes suivants :

- **Piloté par les cas d'utilisation** : un cas d'utilisation représente une fonctionnalité qui satisfait un besoin d'un utilisateur. Le processus suit une voie spécifique, en procédant par une série d'enchaînement d'activités, dérivées d'un cas d'utilisation.
- **Centré sur l'architecture** : l'architecture logicielle représente les aspects statiques et dynamiques du système. L'architecture émerge des besoins de l'entreprise, tel qu'ils sont exprimés par les utilisateurs et reflétés par les cas d'utilisation. L'architecture propose une vue d'ensemble de la conception faisant ressortir les caractéristiques essentielles en laissant de côté les détails secondaires. Il faut noter que tout article est à la fois forme et fonction. L'une ou l'autre isolément ne saurait suffire. Les cas d'utilisation et l'architecture doivent s'équilibrer pour créer un article réussi. La figure suivante va représenter les différentes perspectives qui permettent de définir un modèle d'architecture appelée vue (4+1). Cette dernière a beaucoup inspiré UML.

Figure 5 : Illustre le PU centré sur l'architecture.

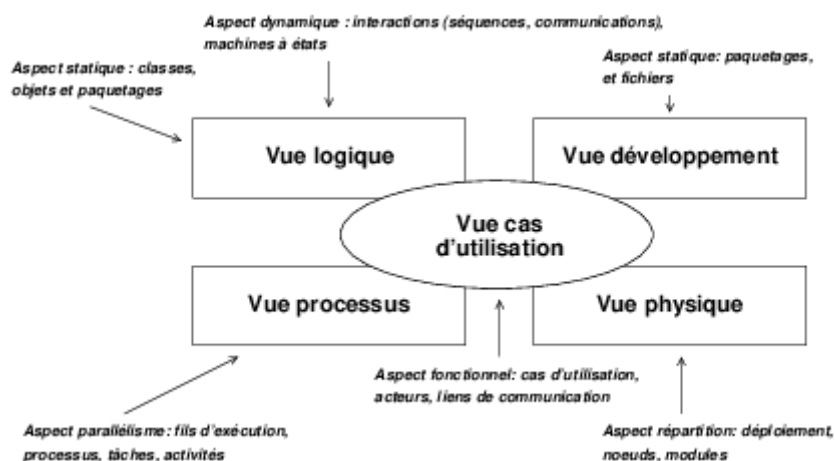


Figure 5: PU centré sur l'architecture

- **Itératif et incrémental** : vu que les projets à réaliser sont de plus en plus complexes et grands, l'idée est de découper le travail en mini projets. Chacun d'entre eux représente une itération qui donne lieu à un incrément. Les itérations désignent des étapes de l'enchaînement d'activités tandis que les incréments correspondent à des stades de développement du produit.

La figure 6 représente le concept d'itération et incrémentation.

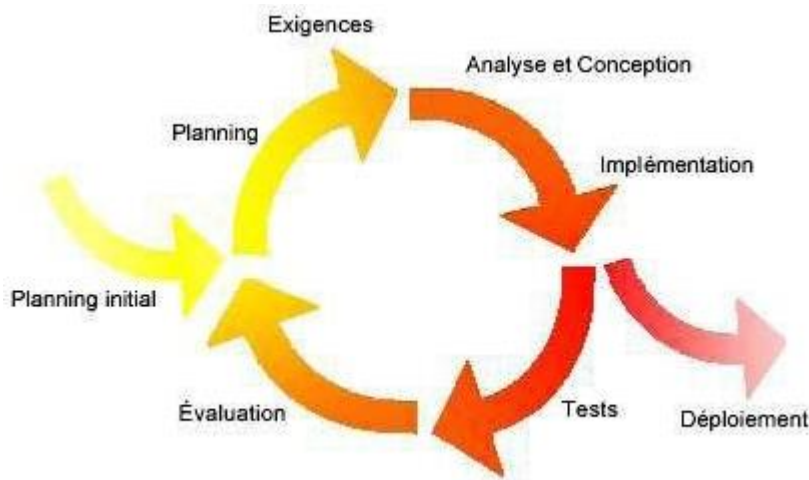


Figure 6: itératif et incrémental

Les phases du processus unifié

Le processus unifié se déroule en 4 phases : création, élaboration, construction et transition. Chaque phase répète un nombre de fois une série d'itérations. Et chaque itération est composée de 5 activités : capture des besoins, analyse, conception, implémentation et test.

- **Création** : c'est la première phase du processus unifié. Il s'agit de tracer ce qui doit figurer à l'intérieur du système et ce qui doit rester à l'extérieur, identifier les acteurs, lever les ambiguïtés sur les besoins et les exigences nécessaires dans cette phase.
- **Elaboration** : c'est la deuxième phase du processus. Il s'agit de purifier le modèle initial de cas d'utilisation, voire capturer de nouveaux besoins, analyser et concevoir la majorité des cas d'utilisation formulés et si possible implémenter et tester les cas d'utilisation initiaux.
- **Construction** : dans cette phase, il faut essayer de capturer tous les besoins. Ensuite, continuer l'analyse, la conception et surtout l'implémentation de tous les cas d'utilisation. À la fin de cette phase, les développeurs doivent fournir une version exécutable du système.
- **Transition** : il s'agit au cours de cette phase de vérifier si le système offre véritablement les services exigés par les utilisateurs, détecter les défaillances, combler les manques dans la documentation du logiciel et adapter le produit à l'environnement.

➤ La Méthode 2TUP

2TUP signifie « 2 Track Unified Process ». C'est un processus UP qui répond aux caractéristiques que nous venons de citer. Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes.

«2 Track » signifie littéralement que le processus suit deux chemins. Il s'agit des chemins « fonctionnels » et « d'architecture technique », qui correspondent aux deux axes des changements imposés au système informatique.

La figure 7 représente le système d'information soumis à deux natures de contraintes

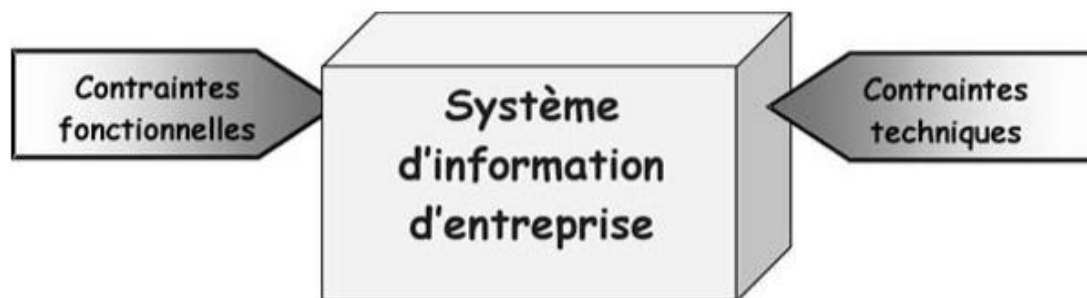


Figure 7: Systèmes d'information soumis à deux natures de contraintes

L'axiome fondateur du 2TUP consiste à constater que toute évolution imposée au système d'information peut se décomposer et se traiter parallèlement, suivant un axe fonctionnel et un axe technique.

À l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des deux branches. Cette fusion conduit à l'obtention d'un processus de développement en forme Y, comme illustré par la figure 8.

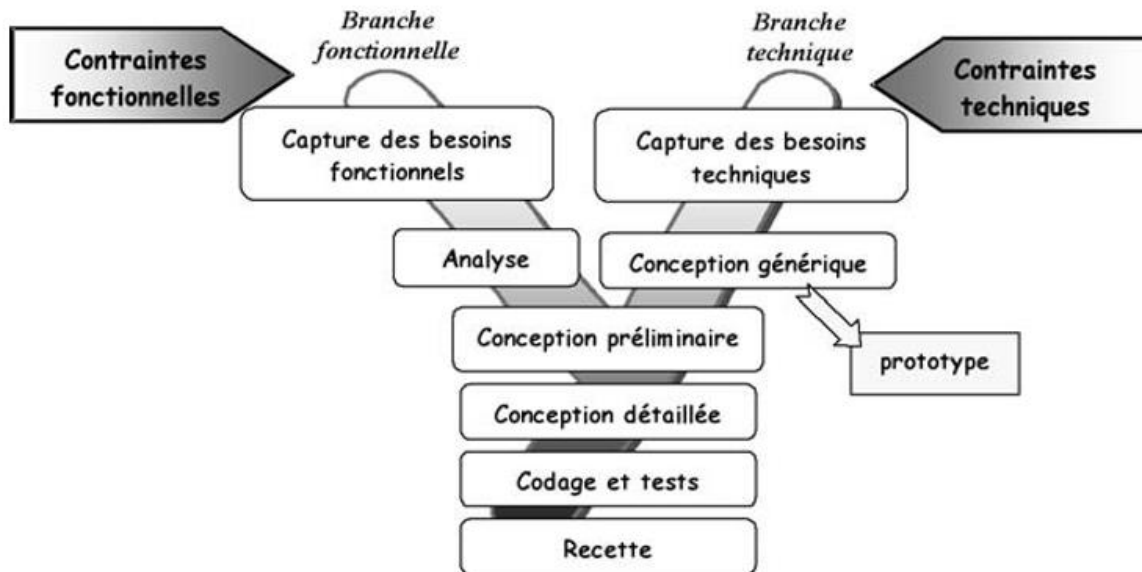


Figure 8: Processus de développement en Y

La branche gauche (fonctionnelle) comporte :

- La capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs. De son côté, la maîtrise d'œuvre consolide les spécifications et en vérifie la cohérence et l'exhaustivité l'analyse, qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier. Les résultats de l'analyse ne dépendent d'aucune technologie particulière.

La branche droite (architecture technique) comporte :

- La capture des besoins techniques, qui recense toutes les contraintes et les choix dimensionnant la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte de contraintes d'intégration avec l'existant conditionnent généralement des prérequis d'architecture technique.
- La conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est la moins dépendante possible des aspects fonctionnels. Elle a pour objectif d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système informatique et écarte la plupart des risques de niveau technique. L'importance de sa réussite est telle qu'il est conseillé de réaliser un prototype pour assurer sa validité.

La branche du milieu comporte :

- La conception préliminaire, qui représente une étape délicate, car elle intègre le modèle d'analyse dans l'architecture technique de manière à tracer la cartographie des composants du système à développer ;
- La conception détaillée, qui étudie ensuite comment réaliser chaque composant ;
- L'étape de codage, qui produit ces composants et teste au fur et à mesure les unités de code réalisées ;
- L'étape de recette, qui consiste enfin à valider les fonctions du système développé.

La méthode est par ailleurs incrémentale c'est-à-dire qu'à partir de la capture des besoins fonctionnels, on définit plusieurs cas d'utilisation représentant chacun un incrément du cycle de développement.

b) Outils de modélisation

Pour mener à bien ce projet, on aura besoin d'un outil de conception pour faire la modélisation du système ainsi que pour anticiper les scénarios des cas d'utilisation à travers des diagrammes.

Le tableau 11 illustre la comparaison des 3 outils de conception tels que : ArgoUML, Modelio et Visual Paradigm

Tableau 10 présente la Comparaison des outils de conception.

Tableau 11 Comparaison des outils de conception

Outils	Avantages	Inconvénients
ArgoUML	Multi plateforme (java) Génération de code en langage Java	Facile d'utilisation pour débutant mais pas pour les professionnels N'intègre que les diagrammes : Cas d'utilisation, Classe, Etat, Activité, Séquence, Collaboration, Déploiement
Modelio	Supporte tout l'UML2 Intègre BPMN Génération de code HTML, MS WORD, Open document Modélisation et application de design Patterns	Seulement pour Windows, Linux. N'intègre que les diagrammes : Cas d'utilisation, Classe, Objet, Etat, Activité, Séquence, Composant, Déploiement, Package, BPMN, Communication en UML
Visual Paradigm for UML (VP)	Dispose de différentes fonctions facilitant la conception des diagrammes aux débutants Multi plateforme(Java) Génération de code source à partir des diagrammes Réalisation du mapping objet relationnel automatiquement à partir du diagramme de classe	-Consomme Beaucoup de ressource

D'après le tableau comparatif, l'outil qui nous convient le mieux est visual paradigm en raison qu'il est multiplateforme et qu'il intègre tous les diagrammes de l'UML.

c) Langage de modélisation utilisé

Pour modéliser notre application, nous avons choisi la notation UML.

La modélisation est le pilier de toute activité qui conduit au déploiement de logiciels de qualité. Les modèles sont construits pour spécifier la structure et le comportement attendus d'un système, pour visualiser et contrôler son architecture, pour mieux comprendre son organisation et ainsi déceler les possibilités de simplification et de réutilisation et, enfin, pour gérer les risques encourus.

Unified Modeling Language ou UML est un langage de modélisation objet officiellement approuvé en 1997 par l'OMG (Objet Management Group). UML « unifie » des méthodes de conception logicielle orientées objet telles que Booch, OMT et OOSE qui coexistaient jusque-là sans beaucoup de compatibilité entre elles. UML les rassemble non pas en proposant une synthèse, mais en créant un « langage de modélisation », c'est-à-dire une notation unique pour faciliter la conception de programmes. C'est un langage graphique de modélisation des données et des traitements. UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

d) Système de gestion de base de données

La plupart des SGBD actuels sont construits autour du modèle relationnel pour simplifier la description des données. Dans le modèle relationnel, la structure de données est formée par un système de relation. Les classes d'objets et les liens sont représentés sous forme de relation. Nombreux de ces SGBD sont disponibles sur le marché, partant des plus petits jusqu'aux plus grands destinés spécialement aux professionnels. Chacun de ces SGBD possède des avantages et des inconvénients.

Le tableau 12 résume le comparatif des SGBD :

Tableau 12 Tableau comparatif entre MySQL et PostgreSQL

SGBD	Avantages	Inconvénients
MySQL	<ul style="list-style-type: none">• Plus léger que les autres SGBD• Multiplateforme, OpenSource et gratuit• Possède une plus grande base d'utilisateurs	<ul style="list-style-type: none">• Pas de vue matérialisée,• Support incomplet des triggers et procédures stockées,

	<ul style="list-style-type: none"> • Solution très courante en hébergement public • Très bonne intégration dans l'environnement Apache/PHP • Version cluster depuis la version 4 • Facilité de déploiement et de prise en main. • Plusieurs moteurs de stockage adaptés aux différentes problématiques, configurable au niveau table. 	<ul style="list-style-type: none"> • Ne supporte pas une énorme quantité
PostgreSQL	<ul style="list-style-type: none"> • OpenSource et gratuit • Fiable et relativement performant, tout en restant simple d'utilisation • Très riche fonctionnellement, notions d'héritage de tables, multitude de modules • Simple d'utilisation et d'administration • Héritage de tables 	<ul style="list-style-type: none"> • Les tables sont obligatoirement transactionnelles • La modification du fichier de sécurité pg_hba.conf nécessite un reboot pour être prise en compte • Pas possible de requêter sur plusieurs bases à la fois : chaque base nécessite sa connexion • Sauvegardes peu évoluées • Supporte les bases de moyenne importance • Pas de services Web • Pas de requêtes récursives • Solutions de réplication pas encore totalement packagées • Solution en cluster pas finalisée

MongoDB	<ul style="list-style-type: none"> • OpenSource et gratuit • Multiplateforme, OpenSource et gratuit • Agile et Flexible • Performance élevée et cohérence forte • Haute disponibilité(jeux de replicas) • Scalabilité horizontale • Analyse en temps réel • Indexation de textes • Compression 	<ul style="list-style-type: none"> • Gestion des ressources • Analyse de collection pendant le chargement • Recherche par facettes • Jointures entre les collections • Transactions entre plusieurs documents
----------------	--	--

On a choisi MongoDB en raison de son agilité , flexibilité et son performance élevée qu'il possède.

MongoDB :

MongoDB est un puissant système de base de données non relationnelle open source et gratuit, populaire pour le stockage de gros volumes de données. Il a été publié il y a 12 ans, en 2009, par 10gen (maintenant MongoDB Inc.) avec une Server Side Public License

Passons maintenant à l'examen de certaines des principales fonctionnalités de MongoDB :

- **Réplication** : MongoDB vous permet de faire plusieurs copies de vos données en utilisant la réplication (replica sets) et de les déployer sur différents serveurs. Cette fonctionnalité est utile pour la sauvegarde des données et la gestion des catastrophes. En cas de panne d'un serveur, vous pouvez toujours récupérer vos données à partir des copies stockées sur d'autres serveurs.
- **Indexation** : Vous pouvez indexer les champs des documents MongoDB comme primaires ou secondaires. Cela contribue à améliorer les performances de recherche dans la base de données et vous permet d'effectuer la recherche sur les index plutôt que sur le document entier, ce qui contribue automatiquement à augmenter la vitesse de recherche.
- **Requêtes ad-hoc** : Les requêtes ad-hoc (ad-hoc query) sont des commandes temporaires qui fournissent des retours différents pour l'exécution des requêtes. Cela

dit, MongoDB prend en charge les recherches par plage, par expression régulière (Regex) et par champ. En outre, les requêtes incluent des fonctions JavaScript définies par les utilisateurs et peuvent renvoyer des champs de document spécifiques. Vous pouvez mettre à jour ces requêtes ad hoc en temps réel à l'aide du langage de requête MongoDB (MQL), ce qui est utile pour les analyses en temps réel pour les entreprises.

- **Sharding** : MongoDB est capable d'évoluer horizontalement à l'aide du sharding, une méthode de distribution de grands ensembles de données sur différentes collections de données. Ici, les utilisateurs peuvent choisir une clé de shard (une clé principale avec une ou plusieurs répliques) pour déterminer la distribution des données dans une collection et diviser les données en différentes plages à travers les shards. Chaque shard fonctionne comme une base de données individuelle. La formation d'une base de données unique à l'aide d'autres shards contribue à l'équilibrage de la charge et à l'exécution de requêtes complexes.
- **Équilibrage de la charge** : MongoDB facilite le contrôle de la concurrence pour traiter simultanément plusieurs requêtes de clients vers différents serveurs. Cela permet de réduire la charge sur chaque serveur, d'assurer la disponibilité et la cohérence des données et de permettre des applications évolutives.
- **Stockage de fichiers** : Vous pouvez utiliser MongoDB comme système de fichiers, connu sous le nom de GridFS, qui s'accompagne de fonctions de réplication de données et d'équilibrage de charge pour plusieurs ordinateurs afin de stocker des fichiers. GridFS, ou système de fichiers en grille, comprend des pilotes MongoDB, et vous pouvez y accéder avec les plugins Nginx et lighttpd ou l'utilitaire mongofiles. MongoDB permet également aux développeurs de manipuler des fichiers et du contenu.
- **Agrégation** : L'agrégation de données permet aux utilisateurs d'obtenir des résultats similaires à ceux obtenus avec la clause SQL - GROUP BY. MongoDB offre trois moyens d'agrégation :
 1. **Pipeline d'agrégation** : Cette méthode offre de meilleures performances d'agrégation que les autres pour la plupart des opérations.
 2. **Agrégation à usage unique** : Elle est utilisée pour agréger les documents d'une collection.

3. **Fonction Map-reduce** : Elle est utilisée pour le traitement des données par lots afin d'obtenir des résultats agrégés. La fonction map regroupe les données par valeur-clé, tandis que la fonction reduce effectue des opérations sur celles-ci.
- **Collections plafonnées** : MongoDB prend en charge les collections plafonnées (capped collections), le type de collection qui gère l'ordre d'insertion des données.

e) Serveur

Pour notre environnement de développement local, nous avons installé séparément les serveurs sur Windows : NodeJS, MongoDB, Robot 4T

Le fait de les séparer nous fait bénéficier en performance pique sous windows, l'environnement de développement est très léger et plus rapide. Et on n'aura pas besoin d'apache.

f) IDE

Le choix d'un bon environnement de développement intégré est primordial avant de commencer à développer une application. Les principaux IDE pour NodeJS sont Visual Studio Code, Atom et Sublime Text.

Le tableau 12 montre la comparaison entre Visual Studio Code.

Tableau 13 Comparatif des IDE

IDE	Avantages	Inconvénients
Visual Studio Code	<ul style="list-style-type: none"> ➤ Supporte NodeJs ➤ Edition de code et refactoring ➤ Intégration des frameworks NodeJS ➤ Modélisation ➤ Excellent Fonctionnalité 	<ul style="list-style-type: none"> ➤ Gourmand en ressources ➤ Interface Complexe
Sublime Text	<ul style="list-style-type: none"> ➤ Supporte NodeJs ➤ Edition de code et refactoring ➤ Modélisation ➤ Moins Gourmand en ressources 	<ul style="list-style-type: none"> ➤ N'intègre aucun Intégration des frameworks NodeJS ➤ Fonctionnalité Moins Excellent
Atom	<ul style="list-style-type: none"> ➤ Supporte NodeJs ➤ Edition de code et refactoring ➤ Intégration des frameworks NodeJS ➤ Modélisation 	<ul style="list-style-type: none"> ➤ Gourmand en ressources ➤ Interface Complexe ➤ Moins Excellent en Fonctionnalité

D'après ce tableau, le choix se penche vers Visual Studio Code en raison de son excellence en matière d'intégration des Framework NodeJS.

g) Système de gestion de version

Un logiciel de gestion de versions est un logiciel qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées dessus. Il permet notamment de retrouver les différentes versions d'un lot de fichiers connexes [23]. Il est indispensable lorsqu'on travaille à plusieurs sur un projet afin d'en assurer l'efficacité. Il existe plusieurs outils de gestion de version tels que CVS, SVN, Mercurial, Git.... Mais nous allons faire un petit zoom sur SVN et Git.

Le tableau 13 va montrer leurs différences.

Tableau 14 Comparaison entre Git et SVN

Outils	Avantages	Inconvénients
Git	<ul style="list-style-type: none"> ➤ Logiciel Libre ➤ Multiplateforme ➤ Des copies de dépôt, Présentes localement, dans lesquelles il est possible de travailler ➤ Enregistre des données ➤ Gestion des branches. ➤ Le dépôt et les copies de travail contiennent l'historique complet ➤ La rapidité : 	<ul style="list-style-type: none"> ➤ Droit d'accès Pour le répertoire complet ➤ Le très grand nombre de commandes ➤ La complexité ➤ Le portage sous Window ➤ Les retours à la ligne
SVN	<ul style="list-style-type: none"> ➤ Logiciel Libre ➤ Multiplateforme ➤ Des copies de dépôt, Présentes localement, dans lesquelles il est possible de travailler ➤ Enregistre des données ➤ Basée sur le chemin ➤ Le dépôt et les copies de travail contiennent l'historique complet 	<ul style="list-style-type: none"> ➤ Temps d'operation Faible ➤ Complexité ➤ Le très grand nombre de commandes

Nous avons choisi Git comme système de gestion de version parce qu'il est performant et est décentralisé, très pratique pour des développements en équipe.

h) Choix de langage de programmation

Un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes [24]. On va voir dans le tableau 15 un comparatif entre les différents langages de programmation destiné au Web.

Comparaison avec les concurrents :

Il existe plusieurs langages de programmation qui pourraient être choisis pour réaliser cette application ... Cependant, une comparaison doit être établie afin de savoir le langage le mieux adapté à notre projet. Le tableau 15 montre l'étude comparative sur ces trois langages. Il est possible de développer une application web utilisant des Frameworks sur beaucoup de langage de programmation.

En général, quatre (3) solutions dominent le marché du web en ce moment :

- PHP
- NodeJS
- C#

Tableau 14 montre la comparaison des langages de programmation

Tableau 15 Comparatif des langages de programmation

	Avantages	Inconvénients
PHP	<ul style="list-style-type: none">➤ Gratuit➤ Variété de Framework Open Source➤ Facilité de déploiement➤ ORM :➤ Courbe d'apprentissage➤ L'hébergement est supporté presque partout➤ Portabilité	<ul style="list-style-type: none">➤ N'est pas adapté pour des traitements asynchrones
NodeJS	<ul style="list-style-type: none">➤ Rapide➤ Scalable➤ Plus grande stabilité de code➤ Beaucoup de flexibilité➤ Rendre Modulable et évolutif➤ Excellente performances	<ul style="list-style-type: none">➤ Pas bon pour le calcul lourd➤ Pas bon pour les bases de données relationnelles

C#	<ul style="list-style-type: none"> ➤ Il est compilé ? rapide d'exécution ● Le langage est soutenu par Microsoft 	<ul style="list-style-type: none"> ➤ L'hébergement se fait exclusivement sous Windows server. ● À la merci des choix technologiques de Microsoft.
----	---	---

En tenant compte du langage de programmation pour le développement de l'application et ses avantages sur tous les langages comparés tel qu'elle est gratuite, compatible sur presque tout l'hébergement, cela réduit considérablement la marge de choix.

Node JS :

NodeJS est une plateforme construite sur la base de l'exécuteur JavaScript de Google Chrome afin de pouvoir développer des applications évolutives et modulables en réseau. Ce développement pourra se faire facilement, rapidement et de façon efficace.

NodeJS utilise la programmation événementielle pour favoriser les échanges entre des clients et un serveur. De plus, les entrées-sorties se font de façon non-bloquante ce qui fait de NodeJS un outil léger et efficace. Cela avantage les applications faisant de l'échange de données intensif en temps réel.

Le fait que NodeJS soit sur une base JavaScript, tous les systèmes ayant des navigateurs supportant le JavaScript sont compatibles côté client. Côté serveur, il suffit d'installer le programme **node** qui sera ensuite disponible à l'exécution en ligne de commandes. Pour l'installation, il vous suffit de télécharger le fichier sur ce lien : NodeJS Installer. Ensuite, suivez les instructions à l'écran.

Justification :

On constate à travers ce tableau que ces langages sont à peu près similaires, mais on observe également que Node JS a des avantages par rapport à ces trois langages.

- Rapide, Plus grande stabilité de code , flexible
- Scalable : NodeJS est dit scalable. Ce terme signifie qu'un système a pour capacité d'évoluer facilement en fonction de la charge d'utilisation ou du besoin. Le mot anglais scalability désigne la capacité d'un produit à s'adapter à un changement d'ordre de grandeur de la demande (montée en charge). En particulier sa capacité à maintenir ses fonctionnalités et ses performances en cas de forte demande. Certains disent "mytique" la scalability de NodeJS, d'autres y croient. On vous laissera vous faire une idée sur

celle-ci après cette petite présentation. Il existe certains modules qui sont disponibles pour effectuer de la distribution de traitements à plusieurs machines différentes.

- **Rendre Modulable et évolutif** : Comme cela a été dit précédemment, l'objectif premier de NodeJS est d'exécuter les scripts JavaScript côté serveur. En second, NodeJS a pour but de rendre l'application développée modulable et évolutive. Une application doit être capable d'évoluer facilement en fonction de la charge qui lui est imposée (Par exemple, fort nombre de visites), un nouveau besoin ou encore un changement d'emplacement ou de type de base de données. Ce genre d'évènement doit être facilement gérable. C'est pourquoi NodeJS utilise des bibliothèques que l'on appelle **modules**. Il sera présenté, dans l'onglet Modules, quelques exemples de modules que j'ai jugé indispensables ou intéressants à connaître.

Il est donc tout à fait naturel que notre choix soit fixé sur NodeJS.

Pour remédier au fait que NodeJS n'ait pas une structure prédéfinie, nous allons lui assigner un Framework.

i) Framework

L'utilisation de Framework présente plusieurs avantages : l'accélération du temps de développement permet de mieux structurer le code et d'avoir une bonne architecture de codage, la conservation d'un code homogène et une pérennité certaine des projets. Enfin, les applications sont plus facilement sécurisées.

Les Frameworks NodeJS contiennent nativement une innombrable quantité de fonctions et de classes censées faciliter le travail du développeur.

Parmi la multitude d'excellents Framework NodeJS nous avons choisi de comparer trois d'entre eux, adoptant tous la structure MVC (Model-View-Controller ou Modèle-Vue-Contrôleur) :

Modèle : c'est le comportement de l'application.

Vue : c'est l'interface manipulée par l'utilisateur.

Contrôleur : c'est le gestionnaire des évènements de mise à jour de la vue et de synchronisation des informations par le modèle.

Le tableau 15 nous montre cinq Framework les plus utilisés et renommés : Express, KOA et METEOR.

Tableau 16 Comparaison entre les frameworks PHP

Framework	Avantages	Inconvénients
Express	<ul style="list-style-type: none"> - Simple pour le Codage - Architecture rapide et robuste - Développement asynchrone - Tonnes de fonctionnalités prêtes à l'emploi - Paiement intégration avec stripe paquets de cryptage très forts 	<ul style="list-style-type: none"> - Pas bon pour le calcul lourd - Pas bon pour les bases de donnée relationnelles
KAO	<ul style="list-style-type: none"> - Prise en charge des générateurs à partir de zéro L'utilisation de générateurs (une fonctionnalité de pointe, même pour Node.js) permettrait de nettoyer votre code du désordre causé par tous ces rappels ; rendre votre code plus facile à gérer. Stable - Extrêmement léger - Koa est très léger avec seulement 550 lignes de code. - les mots-clés sync/wait sont pris en charge et ont transcendé les fonctions du générateur - Intègre l'architecture MVC et la méthode ORM, 	<ul style="list-style-type: none"> - La communauté est relativement petite. Beaucoup de différences - Non compatible avec le middleware de style express. - Nous utilisons donc des générateurs qui ne sont compatibles avec aucun autre type de middleware de framework Node.js.
Meteor	<ul style="list-style-type: none"> ➤ Il est extrêmement rapide de mettre en place un MVP. ➤ Aucun middleware ou ORM requis ➤ Fonctionne bien si vous le réglez correctement ➤ Open source Il fait beaucoup de choses sympas automatiquement tout en réussissant à être assez flexible. ➤ Accepte les modules de Node.js . 	<ul style="list-style-type: none"> ➤ Le Front Blaze est obsolète et ce sera un gros investissement pour nous de passer à React. ➤ Le déploiement et l'intégration d'outils ne sont pas aussi simples que certaines autres plates-formes.

j) Choix des outils de gestion de projet

Le terme logiciel de gestion de projet peut désigner différents types de logiciels ayant pour objectif de faciliter le travail de gestion de projet. Le travail des logiciels de gestion de projet est généralement d'automatiser des tâches de sauvegarde et/ou de la gestion du temps. Nous allons voir un à un les deux (02) principaux logiciels de gestion de projet : TRELLO, et JIRA.

- TRELLO :

Trello permet de créer des projets qui prennent la forme de tableaux en plusieurs colonnes personnalisables. Dans chaque colonne se trouvent les tâches sous forme de cartes que les collaborateurs font avancer de gauche à droite au fur et à mesure de leur avancement [23]. Il est gratuit. Il s'apparente plus à un gestionnaire de tâches qu'un outil de gestion de projet dans le sens où il fait l'impasse sur la gestion des coûts, des temps, des budgets et les fonctions d'administration. La figure 9 représente le tableau de bord de Trello.

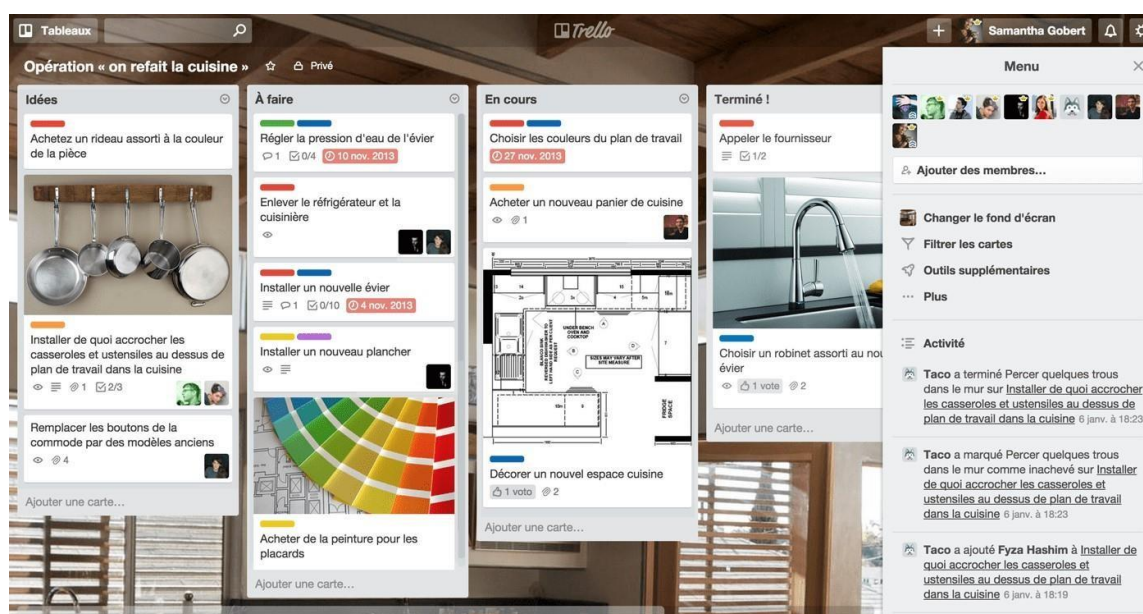


Figure 9:Tableau de bord de Trello

- JIRA :

Jira Software est conçu de sorte que chaque membre de votre équipe de développement puisse planifier, suivre et livrer d'excellents logiciels. Ce logiciel permet de planifier et de suivre le bon déroulement du projet.

La figure 10 présente l'espace de travail de JIRA.

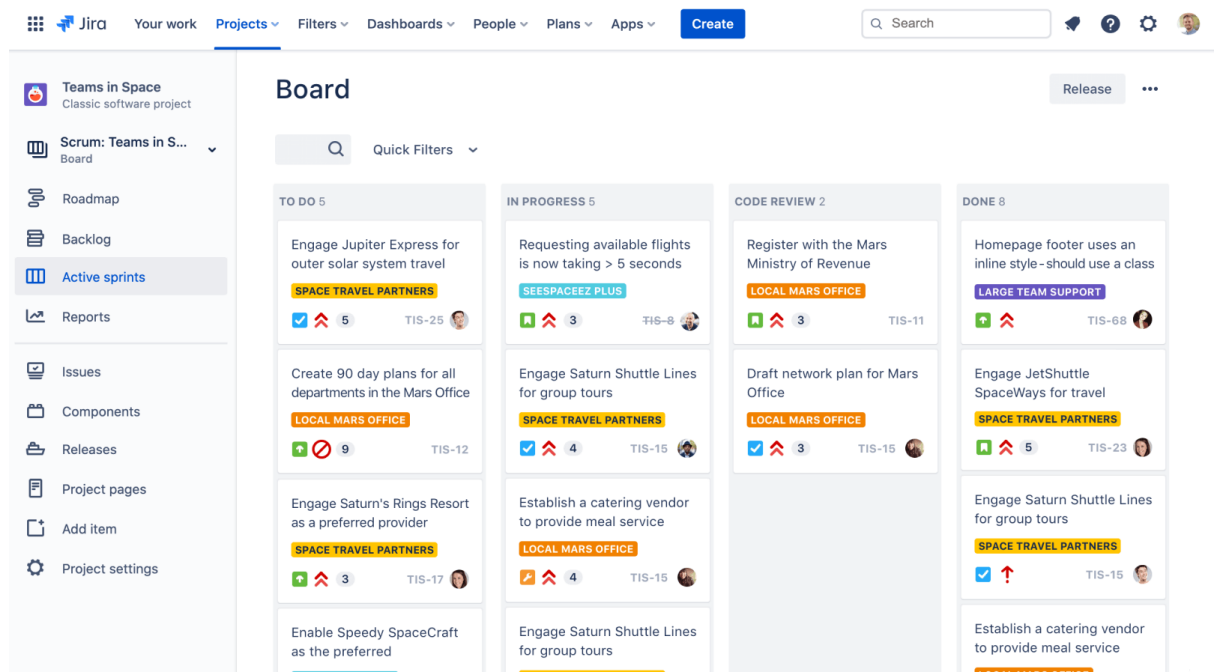


Figure 10: Espace de travail de JIRA

En fin du compte, après avoir considéré tous ces détails, nous avons choisi de combiner les 2 outils.

k) Méthode de gestion de projet

Présentation de la méthode AGILE :

Les méthodes agiles sont des méthodologies essentiellement dédiées à la gestion de projets informatiques. Elles reposent sur des cycles de développement itératifs et adaptatifs en fonction des besoins évolutifs du client. Elles permettent notamment d'impliquer l'ensemble des collaborateurs ainsi que le client dans le développement du projet [26].

Ces méthodes permettent généralement de mieux répondre aux attentes du client en un temps limité. Ces méthodes constituent donc un gain en productivité ainsi qu'un avantage compétitif tant du côté client que du côté du fournisseur.

Les méthodes agiles se reconnaissent toutes dans les valeurs suivantes.

- **L'équipe et la communication avant les outils et processus** : dans la vision agile, l'équipe est bien plus importante que les outils ou les procédures de fonctionnement. Il est préférable d'avoir une équipe soudée et dont les membres communiquent entre eux, composée de développeurs de niveaux différents, plutôt qu'une équipe composée d'experts qui travaillent de manière isolée. La communication est donc une notion fondamentale dans un contexte de développement agile.

- **La collaboration avant la négociation** : le client doit être impliqué dans le développement. Le fournisseur ne doit pas se contenter de négocier un contrat au début du projet, puis de refuser l'évolution des besoins du client. Le client doit collaborer avec l'équipe et fournir des comptes rendus réguliers sur l'adaptation du logiciel à ses attentes.
- **L'acceptation du changement et la flexibilité avant la planification** : la planification initiale et la structure du projet doivent être flexibles afin de permettre les évolutions attendues par le client. En effet, les premières livraisons du projet donnent très souvent suite à des demandes d'évolution.

La raison pour laquelle, on a préféré le manifeste agile qui met en premier plan la collaboration avec le client. Il consiste également à raccourcir les cycles de développement et à se focaliser sur le produit final.

Comparé avec les méthodes traditionnelles, le manifeste agile se marque par les quatre (4) points dans la figure 11 qui résume le manifeste agile.

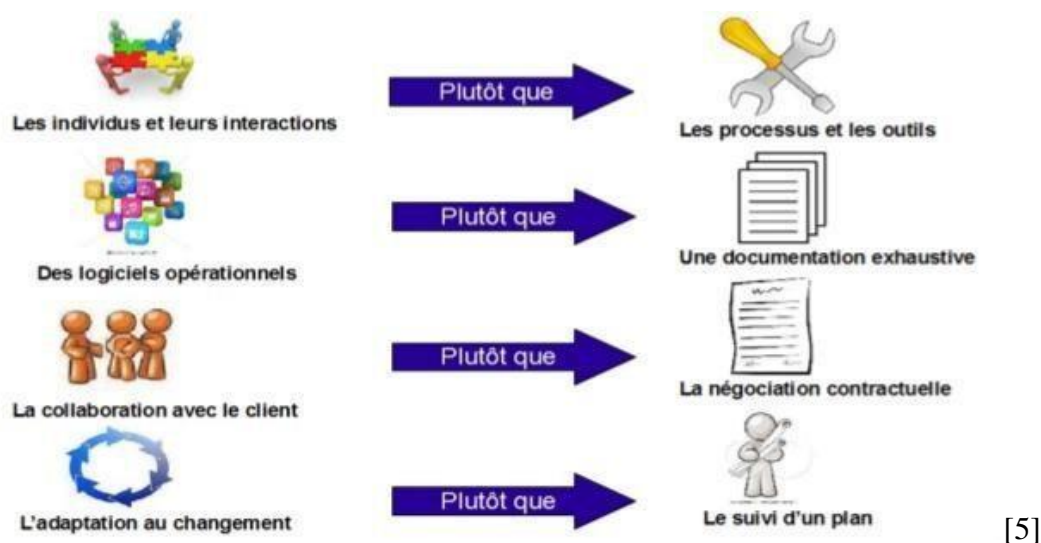


Figure 11:Manifeste agile

- La méthode scrum

La méthode Scrum est un cadre méthodologique de gestion de projet en Agile. Elle vise à accélérer les travaux et répondre plus vite et efficacement aux besoins du client .elle est marquée par une succession d'étapes représentée par la figure 12[27].

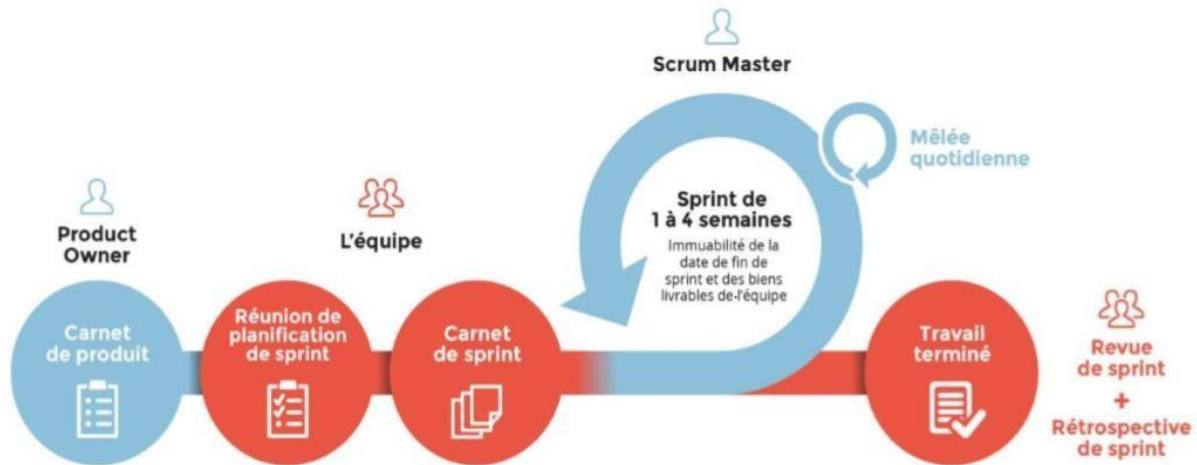


Figure 12: Étapes de SCRUM

D'après la figure, SCRUM est une méthode qui s'appuie sur le découpage des projets en itérations encore nommées « sprints ». Un sprint peut avoir une durée qui varie généralement entre une à quatre semaines. Avant chaque sprint, les tâches sont estimées en temps et en complexité à l'aide de certaines pratiques. Ces estimations permettent à la fois de planifier les livraisons, mais aussi d'estimer le coût de ces tâches auprès du client. Les fonctionnalités (encore appelées « user stories ») qui font l'objet d'un sprint constituent ce que l'on appelle un « sprint backlog » du produit éventuellement livrable à la fin du sprint.

Il est nécessaire de distinguer le sprint backlog du « product backlog » qui lui correspond à l'ensemble des fonctionnalités attendues pour le produit sur l'ensemble des sprints.

La méthode Scrum est aussi caractérisée par une « mêlée » quotidienne, dans laquelle les collaborateurs (chefs de projets, développeurs et responsables fonctionnels) indiquent tour à tour les tâches qu'ils ont effectuées la veille, les difficultés rencontrées et enfin ce sur quoi ils vont poursuivre leur travail le jour suivant. Cela permet d'évaluer l'avancement du projet, de mobiliser des ressources là où cela est le plus nécessaire, mais aussi de venir en aide aux collaborateurs rencontrant des difficultés lorsque celles-ci ont déjà été rencontrées auparavant par d'autres membres de l'équipe.

✓ Rôles

L'équipe Scrum est répartie en 3 rôles :

- **Le Product Owner** : c'est le responsable du produit, il représente les clients et les utilisateurs en transcrivant leurs besoins, définit et priorise les demandes produit.
- **Le Scrum Master**, qui n'est pas le chef de projet, mais il a pour charge de faciliter l'application de Scrum, sa mission est de tout mettre en œuvre pour que l'équipe travaille dans de bonnes conditions et se concentre sur l'objectif du projet. Il porte également une attention particulière au respect des différentes phases de Scrum.
- **L'équipe Scrum** : L'équipe se gère en toute autonomie et est en charge du développement du produit. Il n'y a pas de notion de hiérarchie, toutes les décisions sont prises ensemble. Elle regroupe les rôles habituellement nécessaires à un projet (architecte, concepteur, développeur, etc).

Réunions ou meeting

Les cérémonies Scrum sont donc:

- « **Sprint planning** » une fois par sprint : les équipes sélectionnent les tâches présentes dans le product backlog réalisable en un seul sprint (création du « sprint backlog »).
- « **Daily Scrum** » à chaque début de journée : chaque membre de l'équipe décrit ce qu'il a fait la veille, ce qu'il fera durant la journée et enfin si il a rencontré des problèmes.
- « **Sprint review** » une fois par sprint terminé : L'équipe présente ce qui a été accompli lors du sprint.

« **Sprint retrospective** » une fois par sprint terminé : prise de recul sur ce qui a fonctionné et ce qui n'a pas bien fonctionné, afin d'optimiser les prochains sprints

Chapitre 5. Analyse Conceptuelle

Après le recueil des besoins des utilisateurs et les différentes propriétés de l'application, on va entamer à une phase importante et indispensable dans le cycle de vie d'une application. C'est la phase de conception, elle a pour but d'expliquer toutes les étapes de l'élaboration et le déroulement de l'application

5.1. Etapes de l'élaboration du backlog de produit

Il y a cinq étapes qui faciliteront d'aboutir à l'élaboration du backlog de produit et qui permettent de répondre aux trois (03) questions suivantes : A qui sont-elles destinées ? Quelles seront les fonctionnalités à créer ? Dans quel ordre devraient-elles être livrées ?

Ces étapes sont :

Etape 1 : Vision globale du projet

Etape 2 : Liste détaillée des acteurs

Etape 3 : Thème ou regroupements de fonctionnalités

Etape 4 : Vision du premier release

Etape 5 : User stories

5.1.1. Vision globale de projets

« Offrir une application de gestion d'investissement des projets en ligne »

5.1.2. Liste détaillée des acteurs avec fiche

On entend par acteur humain , une machine ou une système qui ne fait pas partie de la solution à réaliser , mais qui participe au fonctionnement général de la solution par une interaction.

Il est nécessaire d'identifier puis détailler tous les intervenants du système dans tous ses aspects.

On peut citer , dans notre cas :

- ❖ Administrateur
- ❖ Utilisateur

Rôle : Administrateur
Description : Connexion au plateforme Voir détails profil Sélectionner utilisateur Supprimer utilisateur Annuler suppression utilisateur Valider suppression Créer de Catégorie de projet Supprimer le Catégorie de projet

<p>Lister et Editer la catégorie de projet</p> <p>Lister les projet</p> <p>Voir les détails du projet</p> <p>Lister les chiffres d'affaire reçu pour chaque transaction</p> <p>Gestion de transfert d'argent vers l'apporteur du projet</p> <p>Satisfaction : Facilite la gestion des utilisateurs de l'application</p> <p>Instance : 1 ou plusieurs</p> <p>Fréquence d'utilisation : Journalière</p> <p>Niveau de connaissance technique : Bon</p> <p>Niveau de connaissance métier : Bon</p>

Description : S'inscrire

Modifier le profil

Créer un nouveau projet

Lister des projet

Editer et supprimer des projet

Voir le détails de projet

Postuler de projet

Lister les Projets Postulées

Lister les factures sur l'utilisateur qui a postulé sur des projets

Lire le Détail de la facture de l'utilisateur qui a postuler sur des projets

Lister les projets pour affichage public

Voir le détail du projet avec résumé des transactions fait par les investisseurs

Voir le détail de projet et contrat vers le compte

Exporter de fichier PDF contenant le paiement

Investir à un projet

Envoyer un mail à l'apporteur de projet pour lui dire qu'il y a une demande de devenir investisseur

Liste / Détail des demandes de devenir investisseur sur un projet

Validation demande de devenir investisseur avec envoie mail contenant l'url de paiement

Création de page de paiement

Création de fichier PDF contenant le contrat

Système de paiement vers le compte de la plateforme

Satisfaction : Facilite la gestion des utilisateurs de l'application

Instance : 1 ou plusieurs

Fréquence d'utilisation : Journalière

Niveau de connaissance technique : Bon

Niveau de connaissance métier : Bon

5.1.3. Thème et regroupement de fonctionnalités

La collision des thèmes forme la base spécifiant l'objectif central du produit. C'est de la vision du produit qu'on extrait les thèmes principaux.

Notre vision est d'offrir une plateforme complète de mise en relation des candidats à la recherche d'emploi et des recruteurs proposant des offres. Nous allons donc énoncer ci-après les fonctionnalités classées selon une notion de couleur afin de prioriser les tâches à faire.

Notion de couleur :

- Rouge :  Très important
- Jaune :  Important
- Vert :  Utile

Fonctionnalités :

- **Connexion au plateforme**
 - *Description* : Faire une connexion au Platform
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : L'Administrateur connecté
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateurs
- **Voir le détail du profil**
 - *Description* : Voir la page détails du profil utilisateur
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Détails aperçu
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateurs

- **Suppression d'un utilisateur**
 - Description : suppression d'un utilisateur dans la plateforme
 - Valeur commerciale : élevée
 - Critère de satisfaction : utilisateur supprimé
 - Couleur de postit : Rouge
 - Utilisateur : Administrateur
- **Annuler suppression d'utilisateur**
 - *Description* : Annuler la suppression pendant une demande de confirmation de supprimer
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : la suppression est annulée
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateurs
- **Valider suppression**
 - *Description* : valider la suppression pendant une demande de confirmation de supprimer
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : la suppression avec succès
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateurs
- **Créer la catégorie du projet**
 - *Description* : Faire un formulaire de création de catégorie projet dans un page
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Catégorie du projet créée
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateurs
- **Supprimer de catégorie du projet**
 - *Description* : Faire un formulaire de création de catégorie projet dans un page
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Catégorie du projet supprimée
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateurs

- **Lister et éditer de catégorie du projet**
 - *Description* : Une page de listage et édition de catégorie de projet
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : catégorie de projet listé et édité
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateurs
- **Lister les projets**
 - *Description* : Une page pour la liste des projets créée par l'utilisateur
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : La liste des projets aperçu
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateurs
- **Voir les détails d'un projet**
 - *Description* : Une page pour voir le détails d'un projet
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Détails de projet aperçu
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateurs
- **Lister les chiffres d'affaire reçu pour chaque transaction**
 - *Description* : une page pour voir la liste d'affaire reçu pour chaque transaction
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : On peut voir la liste des chiffre d'affaire
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateurs

- **Gestion de transfert d'argent vers l'apporteur du projet**
 - *Description* : une page pour gérer la transfert d'argent vers l'apporteur du projet
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Transfert reçu avec succès
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateurs
- **Inscrire**
 - *Description* : une page d'inscription
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Inscription réussi
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateur , Apporteur , Investisseur
- **Créer un nouveau projet**
 - *Description* : une page de création de projet
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Création réussi
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Apporteur
- **Lister les Projets Postulés**
 - *Description* : une page de liste des projets postulés
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : résultat de la liste du projet
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Apporteur, Investisseur
- **Lister les factures sur l'utilisateur qui a postulé sur des projets**
 - *Description* : une page de liste des factures sur l'utilisateur qui a postulé sur des projets
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : résultat de la liste des factures
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateur, Apporteur, Investisseur
- **Lister les Projets Postulés**

- *Description* : une page de liste des projets postulés
- *Valeur commerciale* : élevée
- *Critères de satisfaction* : résultat de la liste du projet
- *Couleur de postit* : Rouge
- *Utilisateurs* : Administrateur, Apporteur, Investisseur
- **Lister le détail de la facture de l'utilisateur qui a postulé sur des projets**
 - *Description* : une page de détail de facture de l'utilisateur qui a postulé sur des projets
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Afficher le détail de la facture
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateur, Apporteur, Investisseur
- **Lister les Projets pour affichage public**
 - *Description* : une page de la liste des projets pour affichage public
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Affichage de la liste des projets public
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateur, Apporteur, Investisseur
- **Voir le détail du projet avec résumé des transactions fait par les investisseurs**
 - *Description* : une page pour voir le détail du projet avec résumé des transactions fait par les investisseurs
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Affichage du détail et du résumé des transactions fait par les investisseurs
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateur, Apporteur, Investisseur
- **Lister les Projets Postulés**
 - *Description* : une page de liste des projets postulés
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : résultat de la liste du projet
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateur, Apporteur, Investisseur

- **Voir le détail du projet et contrat vers le compte**
 - *Description* : Une page pour voir le détail du projet et la page de contrat
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Affichage de détail du projet
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateur, Apporteur, Investisseur
- **Création du fichier PDF contenant le paiement**
 - *Description* : Génération un fichier PDF contenant le paiement pour l'utilisateur
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Fichier PDF généré
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateur, Apporteur, Investisseur
- **Envoyer un email de confirmation de demande d'investissement**
 - *Description* : Envoyer à l'apporteur de projet un email de confirmation de demande d'investissement
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : email envoyé avec succès
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateur, Apporteur, Investisseur
- **Envoyer un mail à l'apporteur de projet pour lui dire qu'il y a une demande de devenir investisseur :**
 - *Description* : Notifier l'apporteur de projet s'il y a une demande de devenir investisseur
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Notification envoyé avec succès
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateur, Apporteur, Investisseur
- **Lister les détails de demandes de devenir investisseur sur un projet**
 - *Description* : une page pour voir la liste de demandes de devenir investisseur sur un projet
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Affichage les détails de demandes
 - *Couleur de postit* : Rouge

- *Utilisateurs* : Administrateur, Apporteur, Investisseur
- **Validation de demande de devenir investisseur avec envoi de mail contenant l'url de paiement**
 - *Description* : une page pour voir la liste de demandes de devenir investisseur sur un projet
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : résultat de la liste du projet
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateur, Apporteur, Investisseur
- **Créer la page de paiement**
 - *Description* : une page pour voir la page paiement
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Affichage du page de paiement
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Administrateur, Apporteur, Investisseur
- **Créer le fichier PDF contenant le contrat**
 - *Description* : Génération d'un fichier PDF contenant le contrat
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Fichier PDF créer avec succès
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Utilisateur
- **Système de paiement vers le compte de la plateforme**
 - *Description* : Une page pour gérer le paiement vers compte de la plateforme
 - *Valeur commerciale* : élevée
 - *Critères de satisfaction* : Paiement fait avec succès
 - *Couleur de postit* : Rouge
 - *Utilisateurs* : Utilisateur

5.1.4. Vision du premier release

La vision du premier release est « investir à des projets »

5.1.5. User stories

Un user story est un point de vue utilisateur guidé par les réponses aux trois (03) questions suivantes : Qui a fait la demande ou qui bénéficie de la demande ? Quelle est la demande ? Quelle valeur métier découle de la réalisation de ce besoin [6] ?

Voici donc la présentation de la vision des utilisateurs basée autour de trois (03) axes (le rôle , besoin et valeur métier) :

User story 1 : En tant qu'utilisateur déjà inscrit je veux me connecter afin d'accéder à la plateforme.

User story 2 : En tant qu'utilisateur déjà inscrit je veux récupérer un mot de passe afin d'accéder à la plateforme.

User story 3 : En tant que nouvel utilisateur je veux m'inscrire afin d'accéder à la plateforme

User story 4 : En tant qu'utilisateur non inscrit je veux remplir mon profil afin d'accéder à la plateforme

User story 5 : En tant qu'administrateur je veux connecter plateforme.

User story 6 : En tant qu'administrateur je veux accéder naviguer dans la liste afin de chercher des utilisateurs

User story 7 : En tant qu'administrateur je veux voir les détails de profils utilisateurs.

User story 8 : En tant qu'administrateur je veux ajouter de catégorie de projet

User story 9: En tant qu'administrateur je veux supprimer, éditer de catégorie de projet .

User story 10 : En tant qu'apporteur de projet je veux créer de nouveau projet au plateforme.

User story 11 : En tant qu'apporteur de projet je veux éditer et supprimer de projet crée.

User story 12 : En tant qu'apporteur de projet je veux rechercher de projet sur la liste.

User story 13 : En tant qu'apporteur de projet je veux trier la lister de projet par catégorie.

User story 14 : En tant qu'apporteur de projet je veux postuler de projet.

User story 15 : En tant qu'apporteur de projet je veux faire un paiement au plateforme pour que mon projet soit postulé et visible en public.

User story 16 : En tant qu'administrateur je veux valider une demande de postulation à un projet venant d'un apporteur de projet pour affichage public.

User story 17 : En tant qu'administrateur je veux envoyer un email à l'apporteur de projet après validation.

User story 18 : En tant qu'apporteur de projet je veux recevoir une notification de validation de mon projet avec résumé de transaction de paiement !

User story 18 : En tant qu'apporteur de projet je veux voir la liste des projets postulée.

User story 19 : En tant qu'utilisateur je veux voir le détail de chaque projet.

User story 20 : En tant qu'investisseur je veux demander à investir à un projet.

User story 21 : En tant qu'investisseur je veux remplir le contrat d'investissement et envoyer la demande.

User story 22 : En tant qu'apporteur de projet je veux valider un demande d'investissement.

User story 23 : En tant qu'investisseur je veux faire un paiement à l'apporteur de projet pour l'investir.

User story 24 : En tant qu'investisseur je veux je veux recevoir le résumé de transaction après chaque paiement d'investissement.

User story 25: En tant qu'Apporteur de projet je veux voir la liste de mes investisseurs

5.1.6. Backlog de produit

Un backlog produit est une liste hiérarchisée de tâches destinées à l'équipe de développement. Il est créé à partir de la feuille de route et de ses exigences. Les éléments les plus importants figurent en tête du backlog produit. Ainsi, l'équipe sait ce qu'elle doit livrer en priorité.

Tableau 16 illustre le backlog du produit de notre application.

Tableau 17: backlog de produit

Backlog de produit – JIRA				
User story	Titre	Importance	Estimation (jours)	Commentaire
1	Connexion	1	2	Design + Back + front
2	Récupération mot de passe	2	1	Design + Back + front
3	Inscription	4	3	Design + Back + front
4	Formulaire	3	2	Design + Back + front
5	Connexion Page Admin	2	2	Design + Back + front
6	Recherche	2	3	Design + Back + front
7	Détails de profils	3	4	Design + Back + front
8	Ajout de catégorie	4	5	Design + Back + front

9	Editer et supprimer catégorie	5	6	Design + Back + front
10	Création nouveau projet	10	7	Design + front + Back
11	Editer et supprimer de projet	5	5	Design + Back + front
12	Recherche de projet	5	3	Design + Back + front
13	Triage de la liste de projet	2	2	Design + Back + front
14	Postulation de projet	12	7	Design + Back + front
15	Païement pour postulation	10	5	Design + Back + front
16	Validation demande	2	2	Design + Back + front
17	Envoie d'email	5	5	Design + Back + front
18	Page de notification	6	5	Design + Back + front
19	Détail	8	6	Design + Back + front
20	Demande d'investissement	2	2	Design + Back + front
21	Formulaire d'investissement	12	8	Design + Back + front
22	Validation demande d'investissement	10	4	Design + Back + front
23	Païement pour chaque investissement	12	10	Design + Back + front
24	Résumé de transaction	10	5	Design + Back + front
25	Liste des investisseurs	5	5	Design + Back + front

5.2. Dictionnaire de données

Un dictionnaire des données est une collection de métadonnées ou des de données de référence nécessaire à la conception d'une base de données relationnelle. Il revêt une importance stratégique particulière, car il est le vocabulaire commun de l'organisation [15].

Le dictionnaire des données est un document qui regroupe toutes les données que vous aurez à conserver dans votre base (et qui figureront donc dans le MCD).

Il doit respecter les contraintes suivantes [2] :

- Tous les noms doivent être mono values et non décomposables ;
- Il ne doit pas y avoir d'homonyme ni de synonyme ;
- Il ne doit pas y avoir d'homonyme ni de synonyme ;
- Les données y sont regroupées par entité ;
- Les identifiants sont complètement précisés ;
- Les commentaires doivent être pertinents.

Le but de cette partie est d'établir une spécification des besoins fonctionnels qui découlent de l'étude de l'existant précédent. Le tableau 18 présente le dictionnaire des données.

Nom	Description	Domaine de valeur		forma t
		Type	Taille	
addressUt	Adresse de l'utilisateur	AN	30	
contenuPr	Contenu du projet	AN	250	
cvv	Code du compte	N	3	
dateCreatCat	Date de création du catégorie	Date	10	Yyyy -mm- dd
dateCreatInvst	Date de paiement de l'investissement	Date	10	Yyyy -mm- dd
dateCreatPr	Date de création du Projet	date	10	Yyyy -mm- dd

dateExpCp	Date d'expiration du compte	date	10	Yyyy -mm- dd
dateNaissUt	Date du mise à jour du projet	AN	10	Yyyy -mm- dd
dateNaissUt	Date de naissance utilisateur	AN	255	Yyyy -mm- dd
dateNot	Date de notification	date	10	Yyyy -mm- dd
datePost	Date de postulation	date	10	Yyyy -mm- dd
descriptPr	Description du projet	AN	100	
emailUt	Adresse email de l'utilisateur	AN	30	
idCat	Identifiant du catégorie	N	10	
idCp	Identifiant du compte bancaire	N	10	
idInvst	Identifiant de l'investissement	N	10	
idNot	Identifiant du notification	N	8	

IdRole	Identifiant du rôle de l'utilisateur	N	8	
idPos	Identifiant du postulation	N	8	
idPr	Identifiant du projet	N	8	
idUt	Identifiant de l'utilisateur	N	8	
imageCat	Image du catégorie	AN	250	
imagePr	Image du projet	AN	250	
imageUt	Image de l'utilisateur	AN	250	
isPostPr	statuts du projet créer en terme de postulation	Booléen	1	
montant	Montant à payer pour chaque paiement	N	10	
montantInvst	Montant de l'investisse ment	N	12	
montantMinPr	Montant minimale du projet	N	12	
montantPr	Montant du projet	N	12	

motifNot	Motif du notification	AN	30	
nomCat	Nom du catégorie	AN	30	
nomUt	Nom de l'utilisateur	AN	30	
numCartCp	Numéro du carte bancaire	N	12	
passUt	Code secret de l'utilisateur	AN	10	
prenomUt	Prenom de l'utilisateur	AN	30	
propCp	Propriétaire du compte bancaire Compte bancaire	AN	30	
signatureInvst	Signature de l'investissement	AN	100	
statusInvst	Status de l'investissement	AN	100	
statusPost	Statuts du postulation	AN	100	
statusPr	statuts du projet postulée	AN	10	
telUt	Numéro téléphone	N	12	

	de l'utilisateur			
titrePr	Titre du projet	AN	30	
typeUt	Type de l'Utilisateur	AN	30	

5.3. Règle de gestion

Une règle de gestion est une règle suivie par une société. Il peut s'agir d'une disposition légale, d'une exigence formulée par un client ou d'un article et règlement interne. Une règle de gestion est la traduction conceptuelle des objectifs choisis et des contraintes acceptées par l'entreprise. Elle est plus particulièrement liée aux traitements (règle d'action) ou aux données (règle de calcul)

- Une règle d'action décrira les actions que doit accomplir l'entreprise.
- Les règles de calcul décrivent la façon dont doivent d'accomplir les actions.

RG1 : Un utilisateur a qu'un seul rôle

RG2 : Un rôle peut être associé à plusieurs utilisateurs

RG3 : Un utilisateur administrateur peut ajouter un ou plusieurs catégorie .

RG4 : Un utilisateur peut créer un ou plusieurs projet

RG5 : Un projet est créé par un seul utilisateur

RG6 : Un projet crée peut être postulée pour affichage public

RG7 : Un projet crée est visible seulement par l'utilisateur qui le crée s'il n'était postulé

RG8 : L'utilisateur doit faire paiement au plateforme pour postuler un projet

RG9 : L'utilisateur doit recevoir une notification par email après chaque paiement

RG10 : L'utilisateur doit faire un paiement pour voir le détail de projet qui ne l'appartient pas

RG11 : L'utilisateur peut exporter en PDF son résumé de transaction

RG12 : L'utilisateur peut investir à un projet que n'appartient pas à lui

RG13 : l'utilisateur doit faire un contrat pour investir à un projet

RG14 : L'utilisateur doit faire un paiement pour investir à un projet

RG15 : L'utilisateur peut investir à un ou plusieurs projet

5.4. Capture des besoins fonctionnels

Pour répondre aux besoins des utilisateurs et pour réaliser une application de qualité, nous avons retenu tous les besoins des utilisateurs à travers les spécifications des besoins fonctionnels et techniques. Les besoins fonctionnels ou besoins métiers représentent les actions que le système doit exécuter, il ne devient opérationnel que s'il les satisfait. Après une étude détaillée de système, cette partie est réservée à la description des exigences fonctionnelles des différents acteurs de l'application. Ces besoins se regroupent dans les diagrammes des cas d'utilisation. Cette application doit couvrir principalement les besoins fonctionnels suivants :

- La gestion des utilisateurs
- Gestion de catégorie du projet
- La gestion de nouveau projet
- Postulation de projet pour affichage public
- Paiement au plateforme
- Notification par email
- Gestion d'investissement
- Paiement au propriétaire du projet
- Exportation en fichier PDF

5.5. Analyse

L'analyse serait la division , la dissolution , la fragmentation des éléments qui composent une situation complexe.

5.5.1. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation a pour but de donner une vision globale sur les interfaces du futur de l'application. C'est le premier diagramme UML constitué d'un ensemble d'acteurs qui agit sur des cas d'utilisation et qui décrit sous la forme d'actions et des réactions, le comportement du système du point de vue utilisateur.

Acteur : un acteur est un utilisateur qui communique et interagit avec les cas d'utilisation du système. C'est une unité ayant un comportement comme une personne, le système ou une entreprise.

Système : cet élément fixe les limites du système en relation avec l'acteur qui utilisent (en dehors du système) et les fonctions qu'il doit fournir (à l'intérieur du système).

Cas d'utilisation : un cas d'utilisation représente un ensemble de séquence d'actions à réaliser par le système et produisant un résultat observable intéressant pour un acteur particulier représenté par des ellipses et limité par un rectangle pour représenter le système.

Identification des acteurs

Administrateur : c'est la personne qui gère les utilisateurs et la catégorie.

Apporteur : c'est la personne qui s'occupe de la gestion de nouveau projet.

Investisseur : ce sont les personnes qui font un investissement à des projets.

La figure 13 montre le diagramme de cas d'utilisation

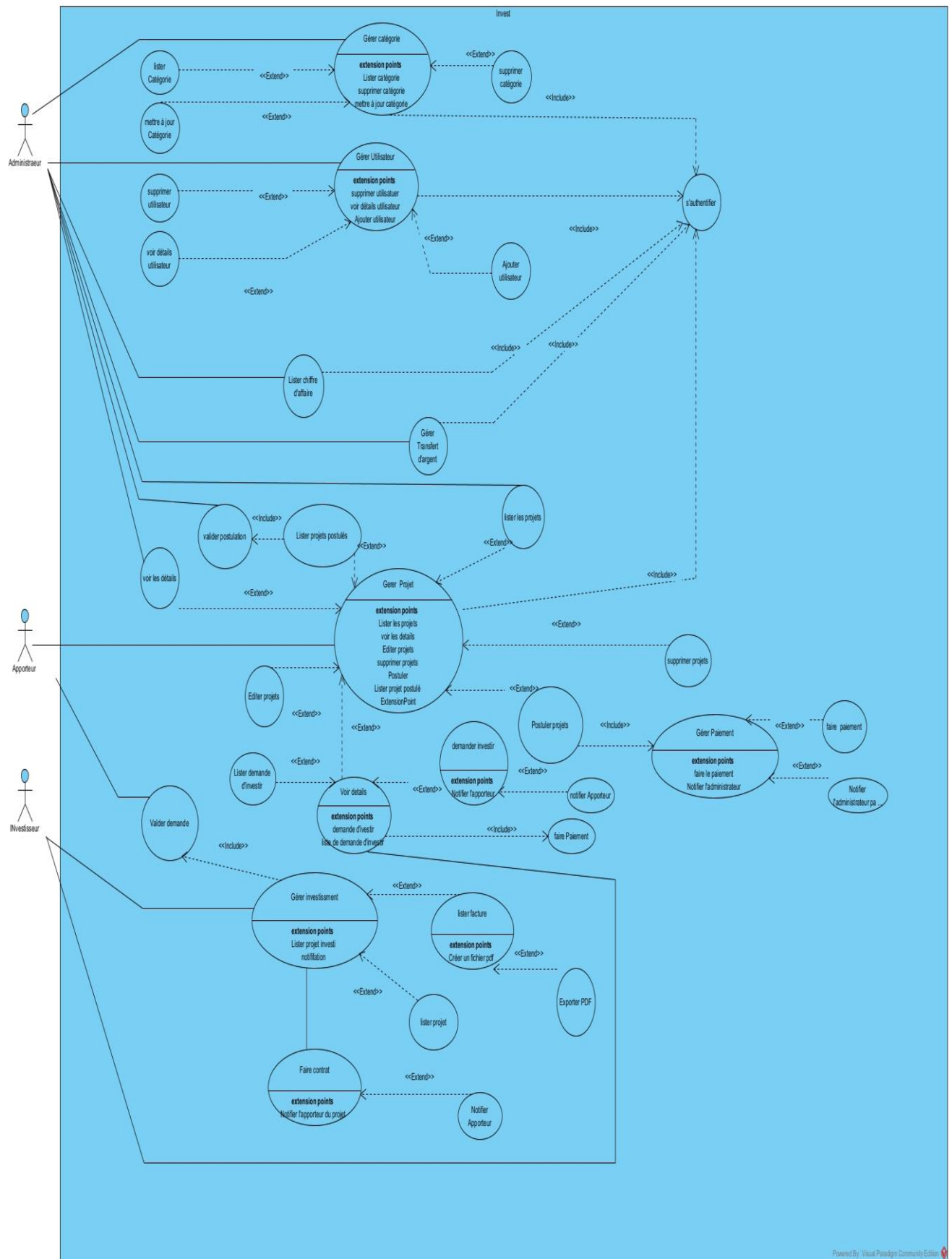


Figure 13: Diagrammes de cas d'utilisation

5.5.2. Priorisation et description des cas d'utilisation

a) Priorisation de cas d'utilisation

Priorité	Cas d'utilisation
1	S'authentifier
1	Gérer les utilisateurs
1	S'inscrire
2	La gestion de catégorie de projet
2	Création de projet
2	Postuler un projet
3	Notifier par email
3	Gérer un investissement
2	Exporter en PDF le résumé du transaction

b) Détails de cas d'utilisation

Description du cas d'utilisation « s'authentifier »

- Nom du cas d'utilisation : s'authentifier
- Acteur : administrateur , apporteur , investisseur
- Précondition : le responsable doit avoir un login/mot de passe
- Post condition : l'utilisateur arrive au page d'accueil
- Scénario nominal :
 - Taper URL au barre de recherche
 - Le système affiche le formulaire d'authentification
 - L'utilisateur le remplit
 - Le système vérifie les informations
 - Le système redirige l'utilisateur vers la page d'accueil
- Scénario d'exception :
 - Un message d'erreur s'affiche en quoi l'email ou mot de passe incorrect

Description du cas d'utilisation « Gérer les utilisateurs »

- Nom du cas d'utilisation : Gérer les utilisateurs
- Acteur : administrateur
- Précondition : l'utilisateur s'est authentifié en tant qu'administrateur
- Post condition : l'utilisateur accède à la liste des utilisateurs

- Scénario nominal :
 - L'administrateur doit être authentifié
 - Il doit appuyer sur l'item Utilisateurs
 - Le système affiche la liste des utilisateurs
 - L'utilisateur demande une mise à jour à un ou plusieurs utilisateurs
 - Le système envoie la modification et met à jour la liste
- Scénario d'exception :
 - L'utilisateur saisit des informations incorrectes, afficher le message d'erreur et retour aux premières étapes du scénario nominal

Description du cas d'utilisation « S'inscrire »

- Nom du cas d'utilisation : s'inscrire
- Acteur : Apporteur , Investisseur
- Précondition : l'email n'a pas encore de compte sur le système
- Post condition : il doit faire entrer un email valide et mot de passe plus ou égal à 6 caractère.
- Scénario nominal :
 - Appuyer sur le menu Inscription
 - Le système affiche les formulaires à remplir
 - L'utilisateur joint son dossier pour l'inscription au système
 - L'utilisateur envoie les informations au système
 - Le système l'enregistre et dirige vers page d'accueil de l'application
- Scénario d'exception :
 - L'utilisateur saisit des informations incorrectes, afficher le message d'erreur et retour aux premières étapes du scénario nominal

Description du cas d'utilisation « Gestion de catégorie »

- Nom du cas d'utilisation : Gestion de catégorie
- Acteur : Administrateur
- Précondition : L'administrateur déjà authentifié au plateforme
- Post condition : L'administrateur doit remplir le formulaire d'ajout de catégorie
- Scénario nominal :
 - Appuyer sur l'item catégorie
 - Le système affiche les formulaires à remplir
 - L'utilisateur remplit les formulaires

- Le nom de catégorie est unique
- L'utilisateur envoie les informations au système
- Le système l'enregistre et affiche la liste de catégorie
- Scénario d'exception :
 - L'administrateur saisit des catégories déjà existant, afficher le message d'erreur et retour aux formulaire d'ajout

Description du cas d'utilisation « Création de projet »

- Nom du cas d'utilisation : Création de projet
- Acteur : Apporteur
- Précondition : L'utilisateur doit être authentifié
- Post condition : L'utilisateur doit remplir le formulaire d'ajout de nouveau.
- Scénario nominal :
 - Appuyer sur le bouton nouveau projet
 - L'utilisateur remplit les formulaires
 - L'utilisateur envoie les informations au système
 - Le système l'enregistre et affiche la liste de projet crée
- Scénario d'exception :
 - L'utilisateur ne remplit pas tout le formulaire d'ajout, afficher le message d'erreur

Description du cas d'utilisation « Postuler un projet »

- Nom du cas d'utilisation : Postuler un projet
- Acteur : Apporteur
- Précondition : Un projet déjà crée .
- Post condition : L'Apporteur postule le projet qu'il veut postuler
- Scénario nominal :
 - Appuyer sur le bouton mes projets
 - Le système affiche la liste des projets ;
 - L'Apporteur clique sur le bouton postuler d'un projet ;
 - Le système dirige vers formulaire de paiement au Platform ;
 - Le système l'enregistre et dirige vers l'affichage des projets public ;
- Scénario d'exception :
 - L'utilisateur saisit des informations incorrectes sur le formulaire de paiement, afficher le message d'erreur .

Description du cas d'utilisation « Notifier par email »

- Nom du cas d'utilisation : Notifier par email
- Acteur : Utilisateur
- Précondition : L'utilisateur a fait un paiement.
- Post condition : L'utilisateur doit avoir un compte email fonctionnel.
- Scénario nominal :
 - Le paiement se fait avec succès
 - Le système envoie un email contenant le résumé de son transaction
- Scénario d'exception :
 - Paiement échoué , afficher le message d'erreur

Description du cas d'utilisation « Gérer un investissement »

- Nom du cas d'utilisation : Gérer un investissement
- Acteur : Investisseur
- Précondition : Il y a de projet projet déjà postulé dans l'affichage public.
- Post condition : L'investisseur doit voir le détails du projet.
- Scénario nominal :
 - L'investisseur doit cliquer sur le bouton s'investir .
 - Le système redirige sur un formulaire de demande d'investissement avec le contrat.
 - L'investisseur envoie le formulaire au propriétaire du projet .
 - L'Apporteur du projet valide ce contrat et envoie un email à l'investisseur.
 - L'investisseur va procéder au paiement après validation de l'apporteur.
 - L'investisseur doit recevoir un email après le paiement.
- Scénario d'exception :
 - L'investisseur remplit des informations incomplète sur le formulaire de demande d'investissement, afficher le message d'erreur .

Description du cas d'utilisation « Exporter en PDF le résumé du transaction»

Nom du cas d'utilisation : Exporter en PDF le résumé de la transaction

- Acteur : Administrateur , Apporteur , Investisseur
- Précondition : L'utilisateur à été déjà fait un paiement sur le plateforme.
- Post condition : Il doit voir le détails d'un résumé de transaction sur la liste factures.
- Scénario nominal :
 - Le système affiche des listes de transaction pour chaque paiement
 - L'Utilisateur clique sur une item et dirige vers le détail

- L'utilisateur cliquer sur le bouton export pour exporter en PDF

Scénario d'exception :

- **Aucun exception**

5.6. Capture des besoins techniques

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système, mais plutôt identifient des contraintes internes et externes du système. Et qui concerne notre application, nous avons dégagé les besoins suivants : le code doit être clair pour permettre des futures évolutions ou améliorations ; la disponibilité : l'application doit être disponible pour être utilisée à n'importe quel moment ; la sécurité : l'application doit respecter la confidentialité des données ; garantir l'intégrité et la cohérence des données à chaque mise à jour et à chaque insertion ; la convivialité de l'interface graphique : l'application doit fournir une interface conviviale et simple pour tout type d'utilisateur, car elle présente le premier contact de l'utilisateur avec l'application et par le biais de celle-ci, on découvrira se fonctionnalités.

5.7. Conception préliminaire

5.7.1. Diagrammes des séquences système

Le diagramme de séquences permet de représenter la succession chronologique des opérations et des collaborations entre objets selon un point de vue temporel. Il peut servir à illustrer un cas d'utilisation. On y met l'accent sur la chronologie des envois de messages qui est déterminée par sa position sur l'axe vertical du diagramme, le temps s'écoule de haut en bas de cet axe. La disposition des objets sur l'axe horizontal n'a pas de séquence pour la sémantique du diagramme.

- Cas d'utilisation s'authentifier

La figure 14 montre le diagramme de séquence système du cas d'utilisation s'authentifier

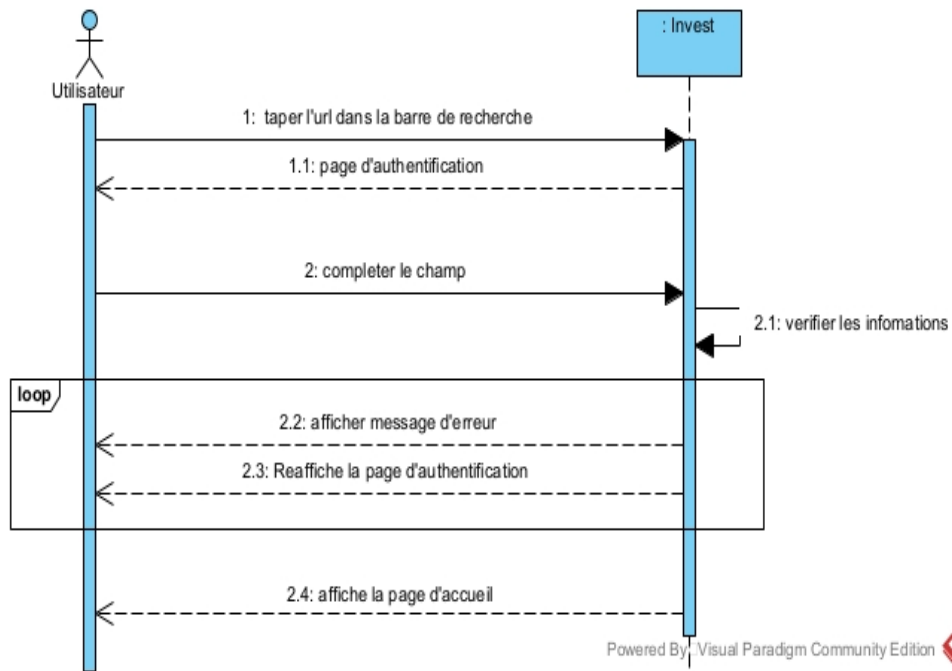


Figure 14: Séquence de système pour le cas d'utilisation s'authentifier

○ Cas d'utilisation « Gérer utilisateurs »

La figure 15 montre le diagramme de séquence système du cas d'utilisation gérer utilisateur

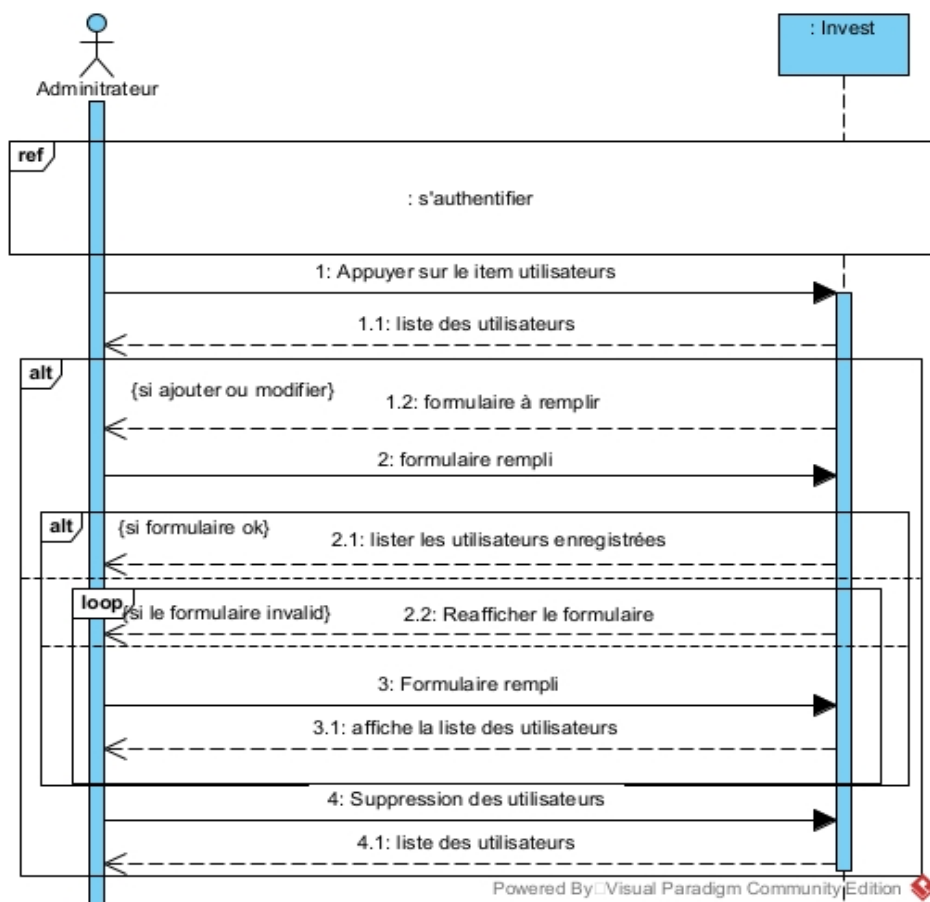


Figure 15: Diagramme de séquence système du cas d'utilisation gérer utilisateur

- Cas d'utilisation « Gérer la catégorie de projet »
La figure 16 montre le diagramme de séquence système du cas d'utilisation gérer la catégorie de projet

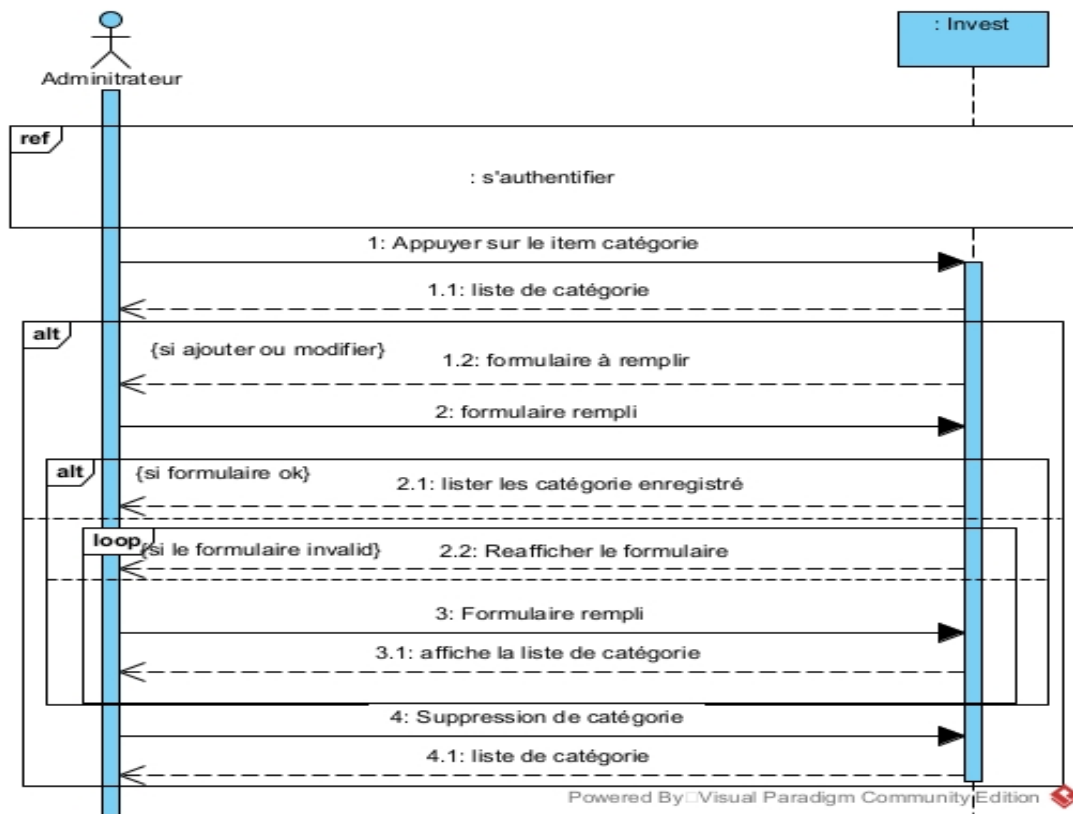


Figure 16: Diagramme de séquence système du cas d'utilisation gérer catégorie de projet

- Cas d'utilisation « S'inscrire »
La figure 17 montre le diagramme de séquence système du cas d'utilisation s'inscrire

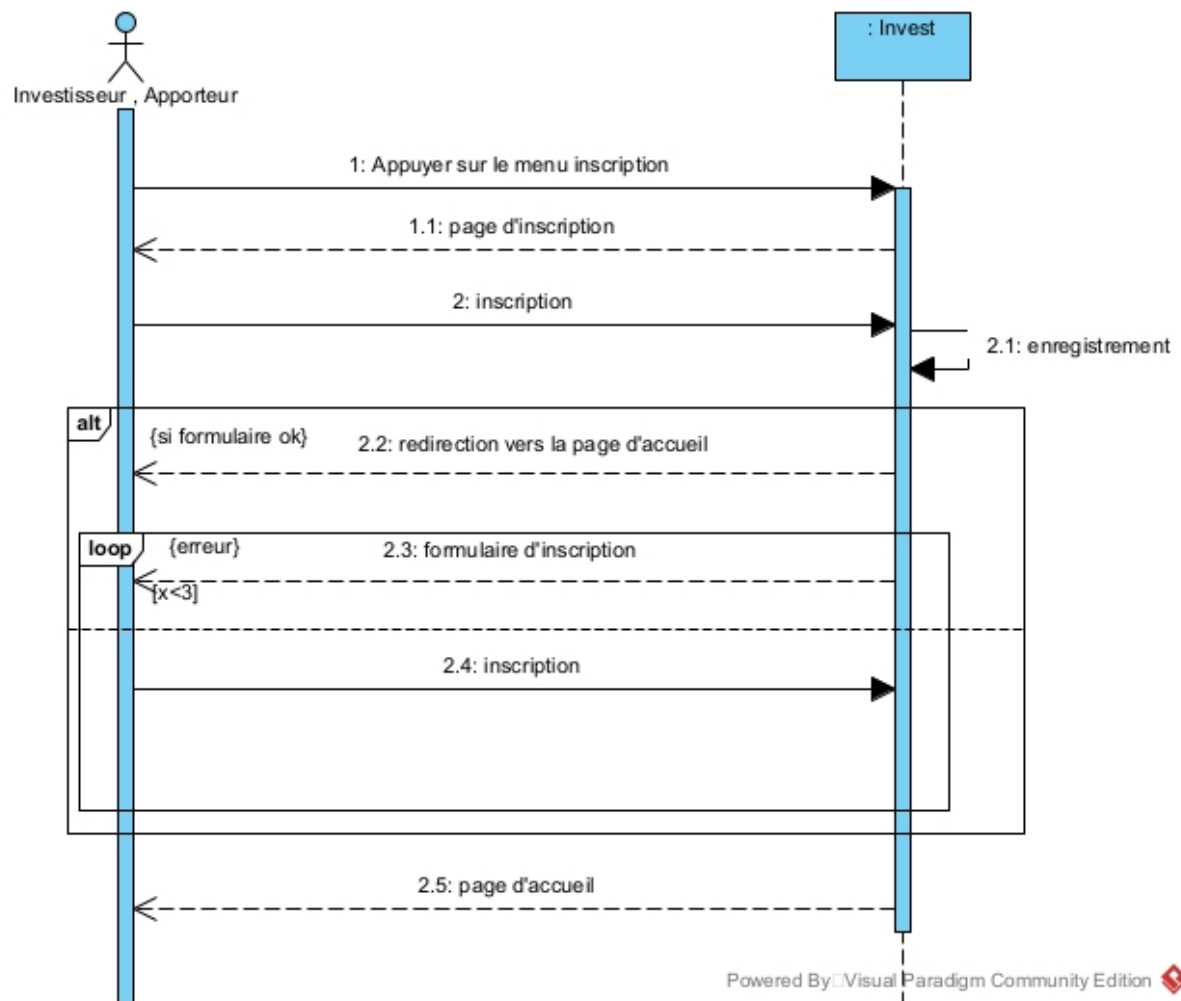


Figure 17: Diagramme de séquence s'inscrire

- Cas d'utilisation « Création de projet »

La figure 18 montre le diagramme de séquence système du cas d'utilisation création de projet

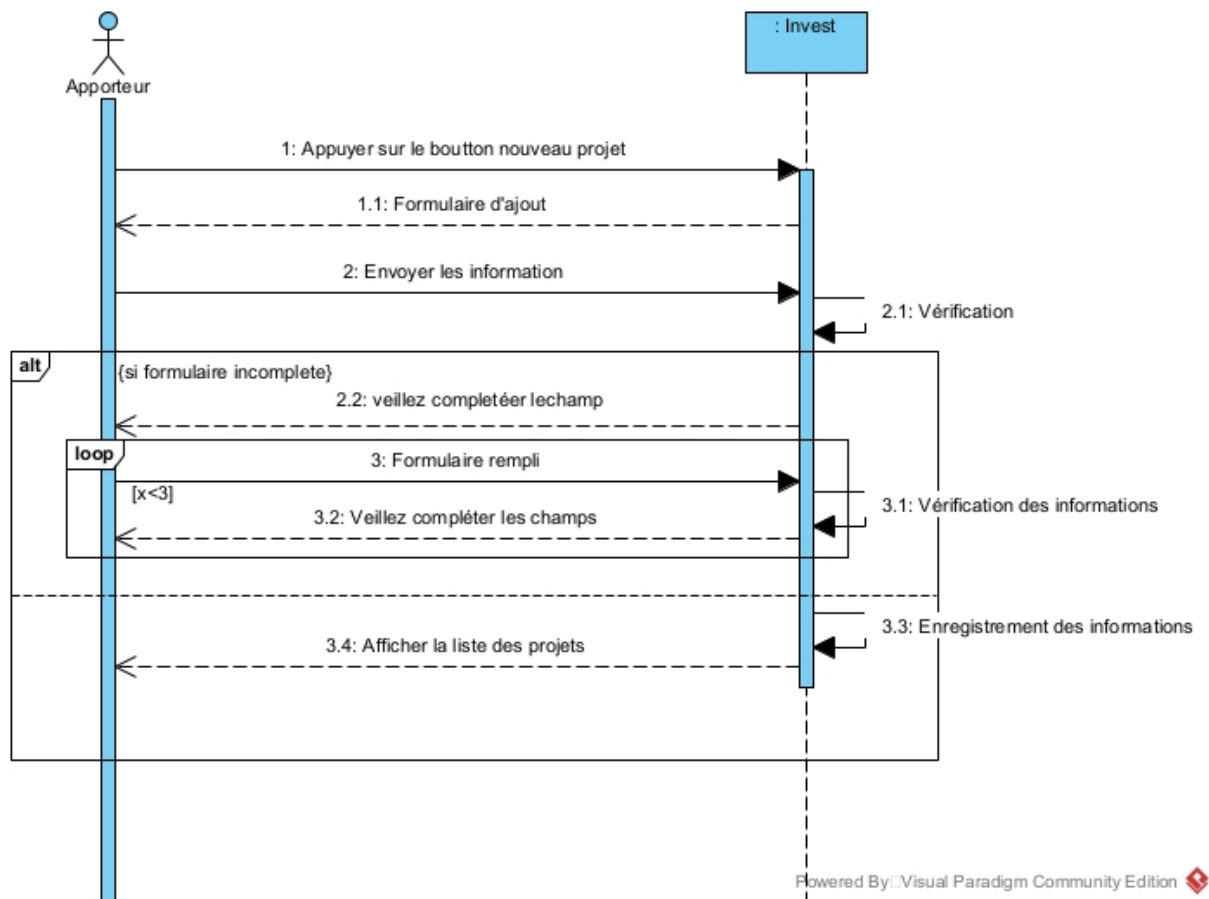


Figure 18: Diagramme de séquence création de projet

- Cas d'utilisation « Postuler un projet »

La figure 18 montre le diagramme de séquence système du cas d'utilisation Postuler un projet

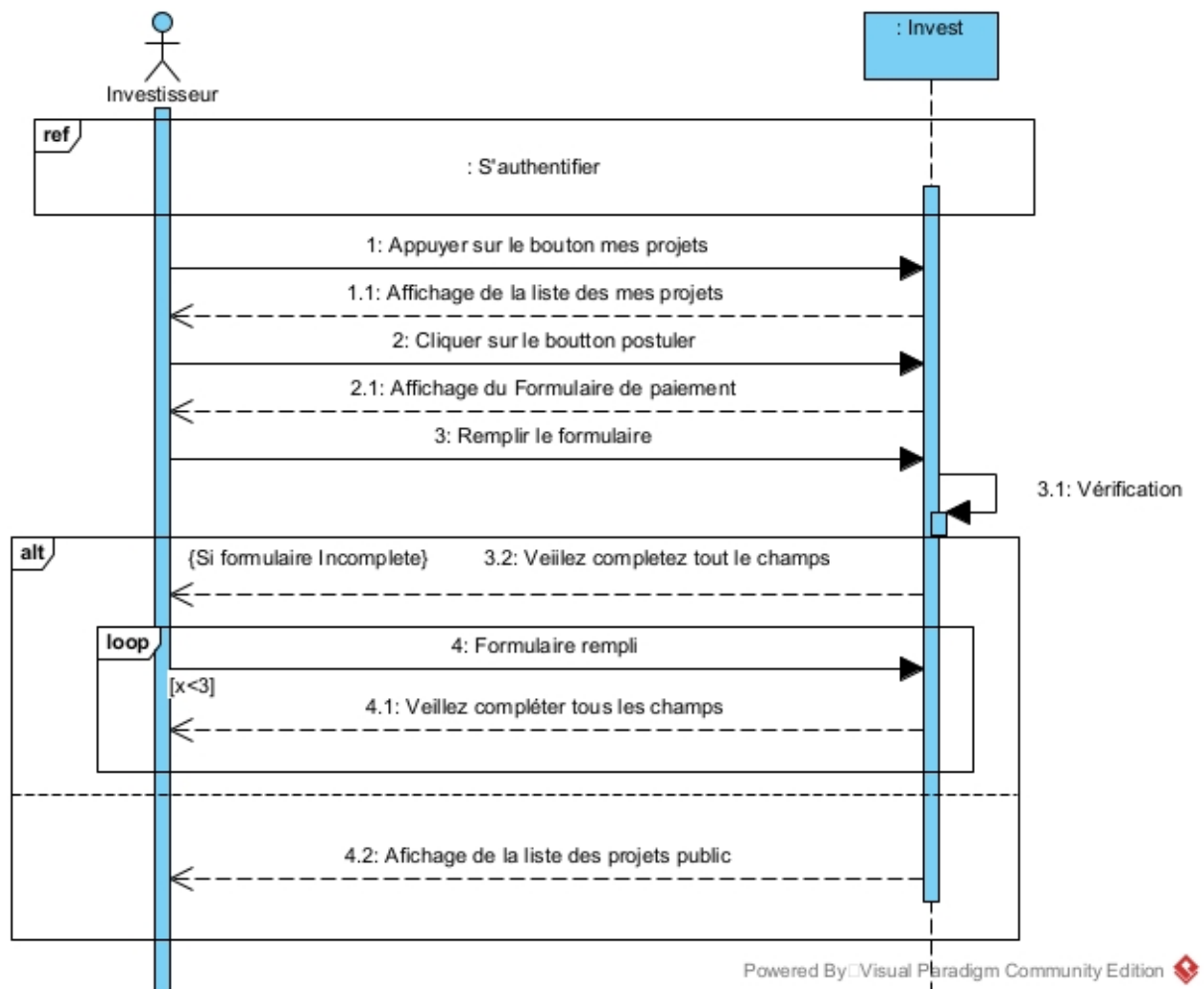


Figure 19: Diagramme de séquence Postuler un projet

- Cas d'utilisation « Postuler un projet »

La figure 20 montre le diagramme de séquence système du cas d'utilisation Gérer Investissement

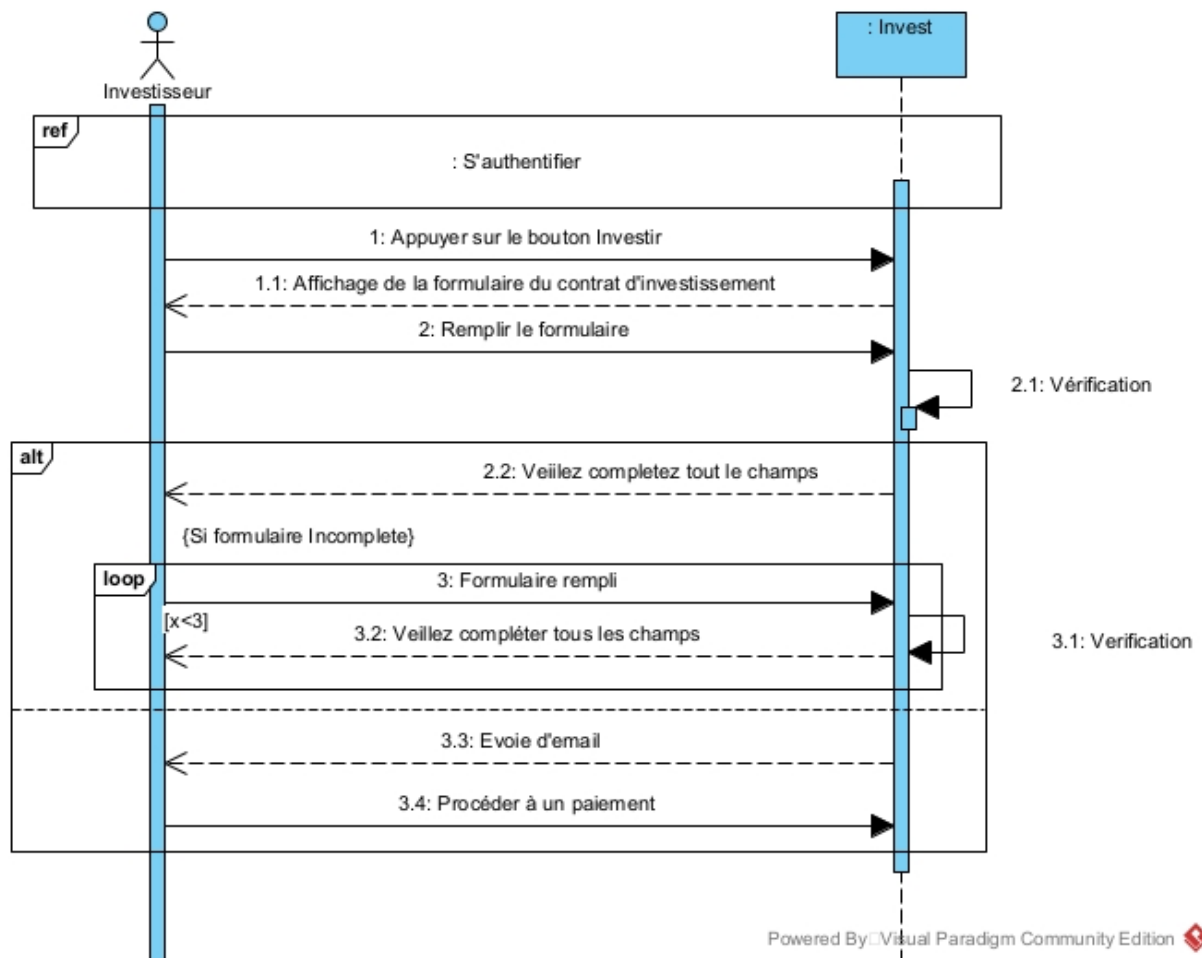


Figure 20: Diagramme de séquence gérer investissement

5.7.2. Modélisation du domaine

Le figure 21 Montre le modèle du domaine

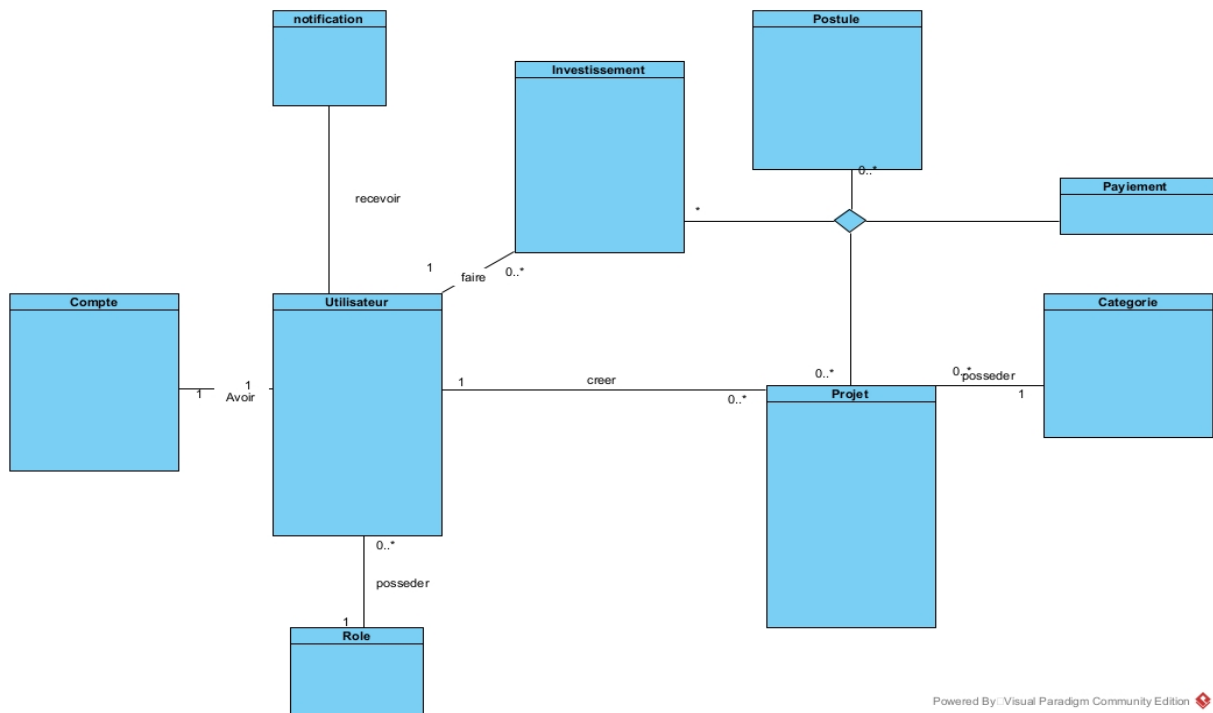


Figure 21Modèle du domaine

Chapitre 6. Conception détaillée

La conception détaillée est un document essentiel qui, s'il est bien étayé, nous permettra de restituer une réalisation de qualité.

L'objectif de ce document est de fournir la feuille de route exhaustive et détaillée nécessaire à la réalisation technique du projet.

6.1. Architecture du système

Pour l'architecture du projet, on a choisi d'adopter l'architecture MVC (Modèle Vue Contrôleur) couplée à une architecture 3-tiers pour structurer et fractionner le code du projet. L'architecture MVC est un concept très puissant qui intervient dans la réalisation d'une application. Son principal intérêt est la séparation des données (modèle) de l'affichage (vue) et des actions (contrôleur).

Ce concept peut très bien être appliqué à un site internet. De nombreux sites internet disposent d'une plateforme d'administration pour modifier facilement les contenus. Il est bien évident que l'utilisateur qui va utiliser ce système pour créer des pages ne doit pas modifier la structure complète du site, ni les actions, ni l'affichage. Pour qu'il ne modifie que les données, celles-ci doivent être alors isolées.

C'est sur ce constat que se base le système MVC. Chaque partie est bien distincte. Les données peuvent provenir d'une source quelconque et la vue peut être conçue par des webdesigners n'ayant pas de connaissance en programmation.

L'architecture MVC est donc décomposée en trois couches :

- **Le modèle :** Le modèle correspond aux données, la plupart du temps stockées dans une base de données. Mais celle-ci peut également être contenue dans un fichier XML, ou dans des fichiers texte. Les données peuvent être exploitées sous forme de classes dans un langage de programmation orientée objet.
- **La vue :** La vue correspond à un fichier HTML, peut-être un fichier XML, avec une feuille de style XLS la plupart du temps. Elle se contente d'afficher le contenu qu'elle reçoit sans avoir connaissance des données. Elle ne contient que les informations liées à l'affichage.
- **Le contrôleur :** le contrôleur est l'élément qui va utiliser les données pour les envoyer à la vue. Son rôle est donc de récupérer les informations, de les traiter en fonction des paramètres demandés par la vue (par l'utilisateur, exemple : affiche les derniers articles), puis de renvoyer à la vue les données afin d'être affichée. Le contrôleur peut

donc instancier différents objets (class user, class articles ...) qui enverront des requêtes vers la base de données ou récupéreront des données XML la transmission à la vue des données récupérées se fait généralement à l'aide d'un Template.

La figure 22 montre l'architecture MVC.

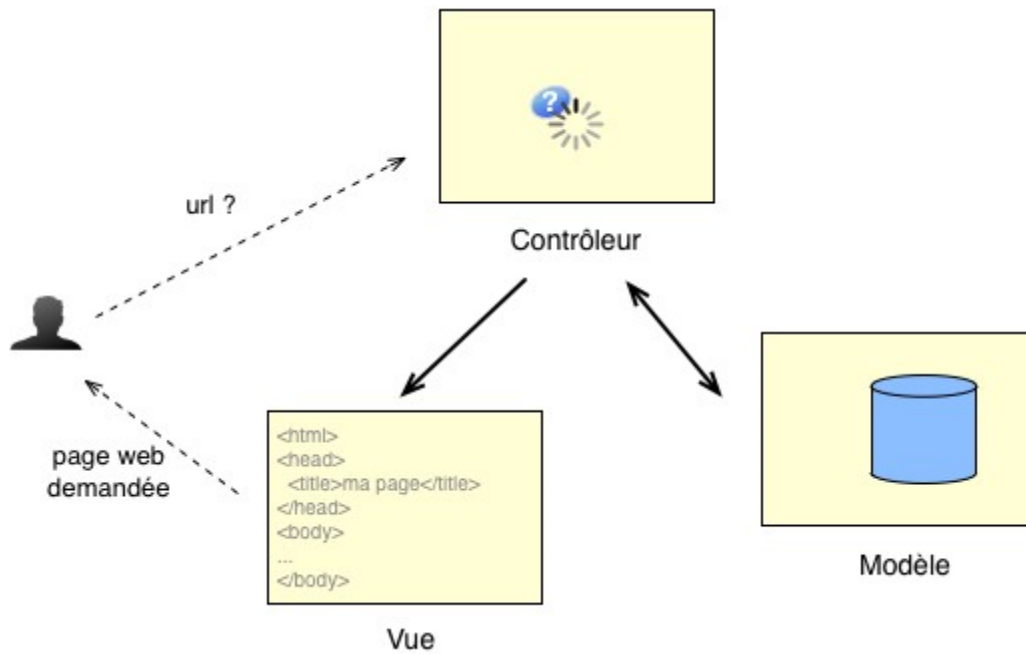


Figure 22 Architecture MVC

6.2. Diagramme de séquence de conception

➤ Cas d'utilisation s'authentifier

Figure 23 illustre le diagramme de séquence de conception du cas d'utilisation s'authentifier.

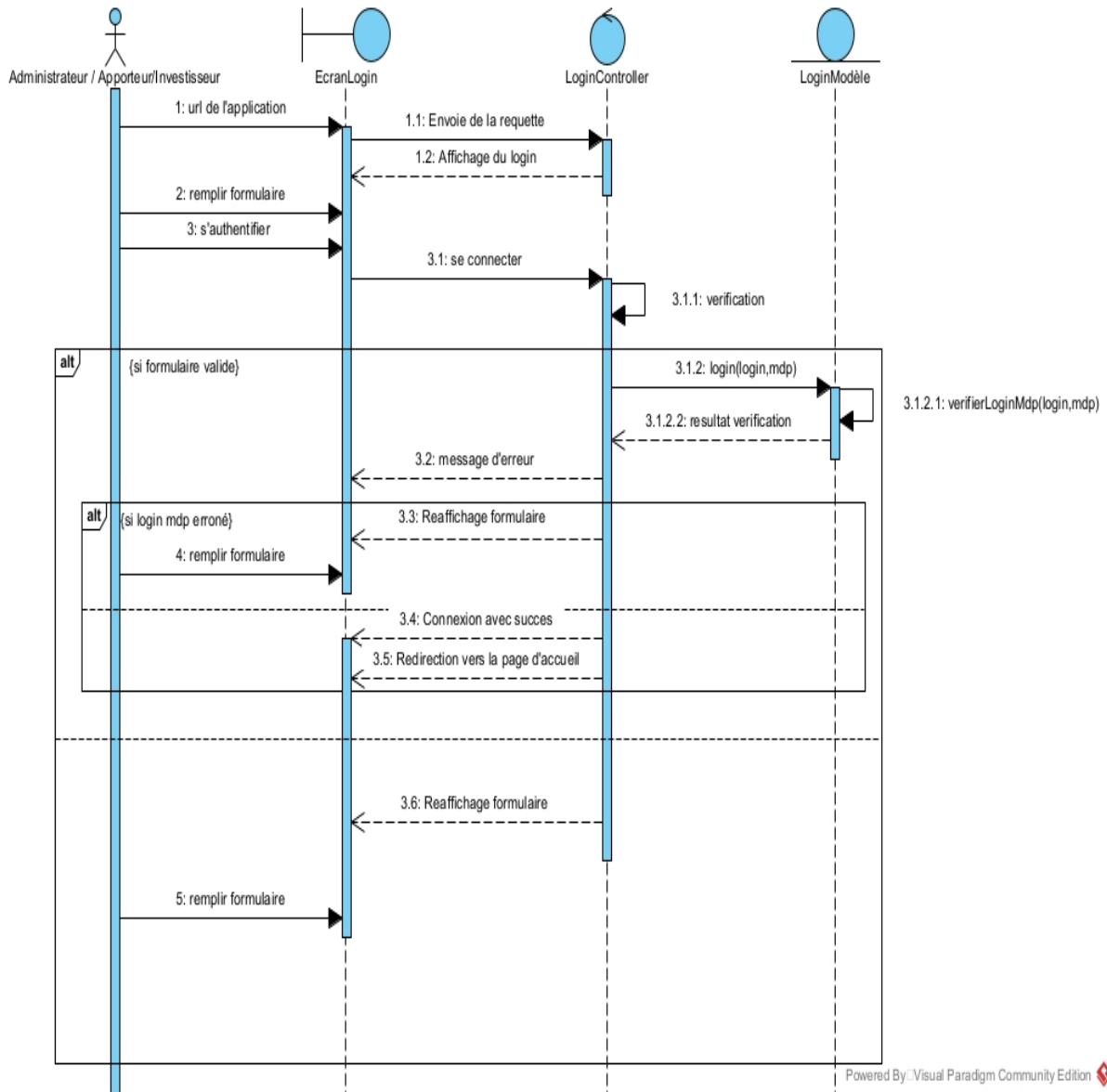


Figure 23:Diagramme de séquence de conception du cas d'utilisation s'authentifier

➤ Cas d'utilisation Creation projet

Figure 24 illustre le diagramme de séquence de conception du cas d'utilisation Création de projet.

Figure 24Diagramme de séquence de conception du cas d'utilisation création projet

6.3. Diagramme de classe de conception

L'application implémentera le pattern Modèle-Vue-Contrôleur. Les classes des modèles de conception seront donc stéréotypées « Modèle », « View » ou « Controller ».

- Cas d'utilisation s'authentifier

La figure 25 illustre le diagramme de classe de conception du cas d'utilisation s'authentifier

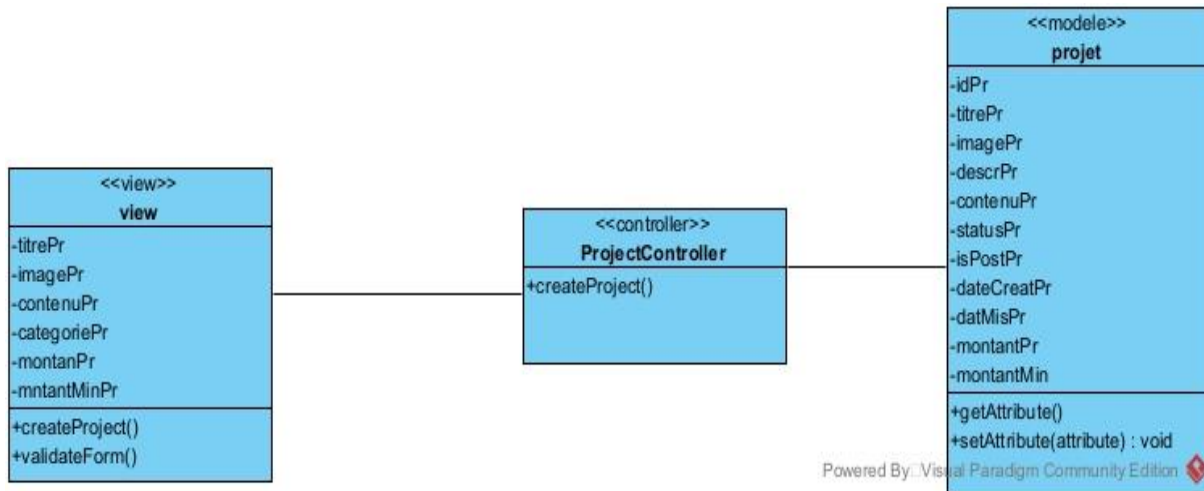


Figure 25:Diagramme de classe de conception du cas d'utisatoin s'authentifier

- Cas d'utilisation création de projet

La figure 26 illustre le diagramme de classe de conception du cas d'utilisation Création d'un projet

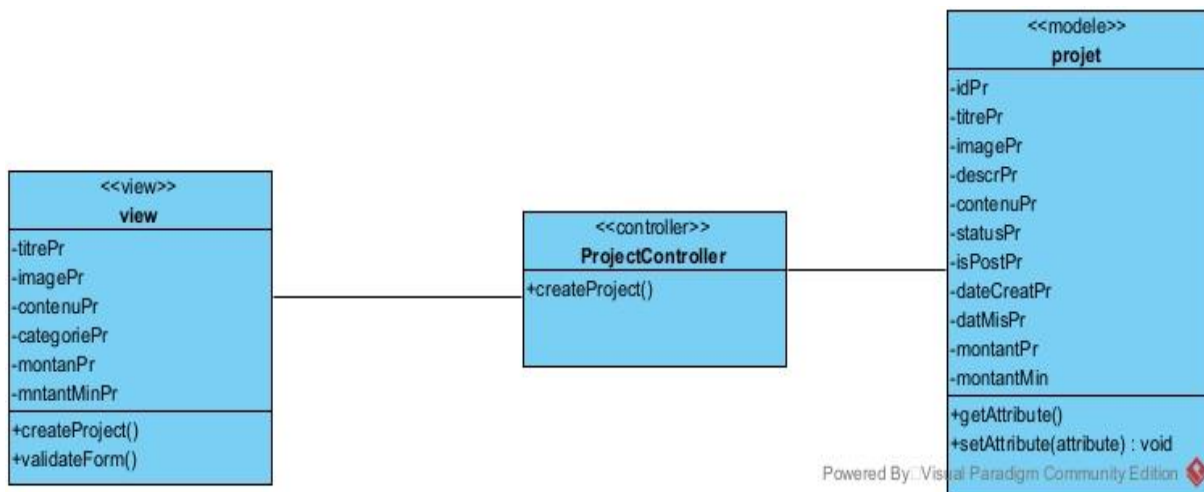


Figure 26:diagramme de classe de conception du cas d'utilisation Création d'un projet

- Cas d'utilisation postuler projet

La figure 27 illustre le diagramme de classe de conception du cas d'utilisation postuler un projet

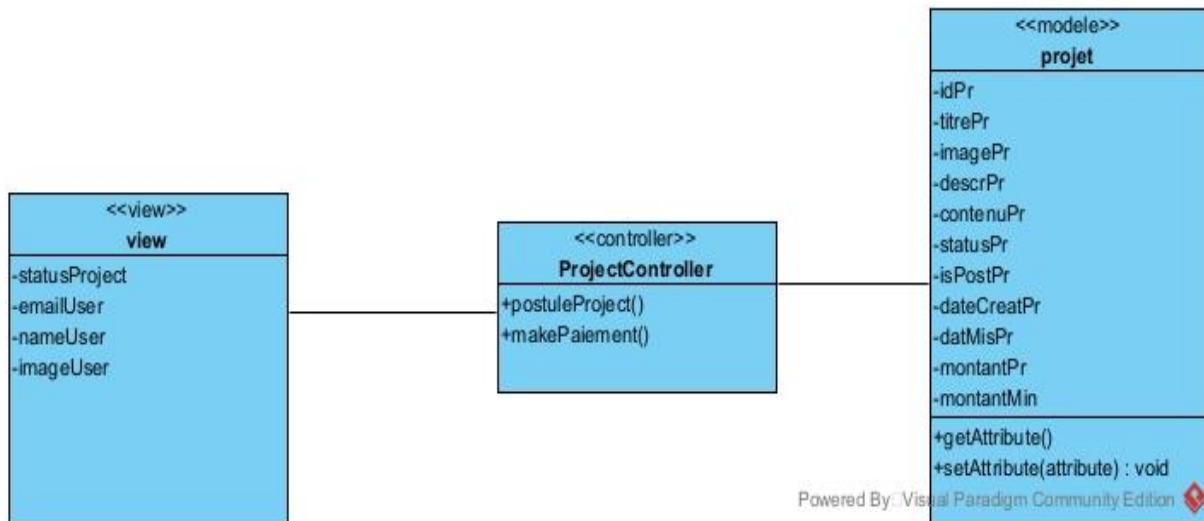


Figure 27:diagramme de classe de conception du cas d'utilisation postuler un projet

➤ Cas d'utilisation investir à un projet

La figure 28 illustre le diagramme de classe de conception du cas d'utilisation investir à un projet

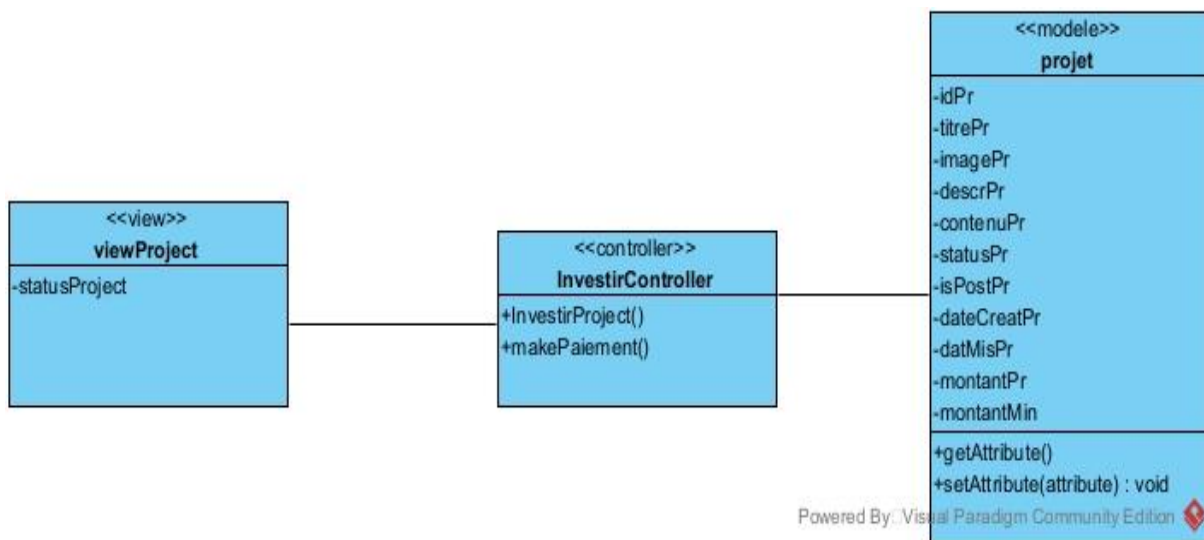


Figure 28: diagramme de classe de conception du cas d'utilisation investir à un projet

6.4. Diagramme de classe de conception global

Le figure 29 illustre le diagramme de classe de conception global

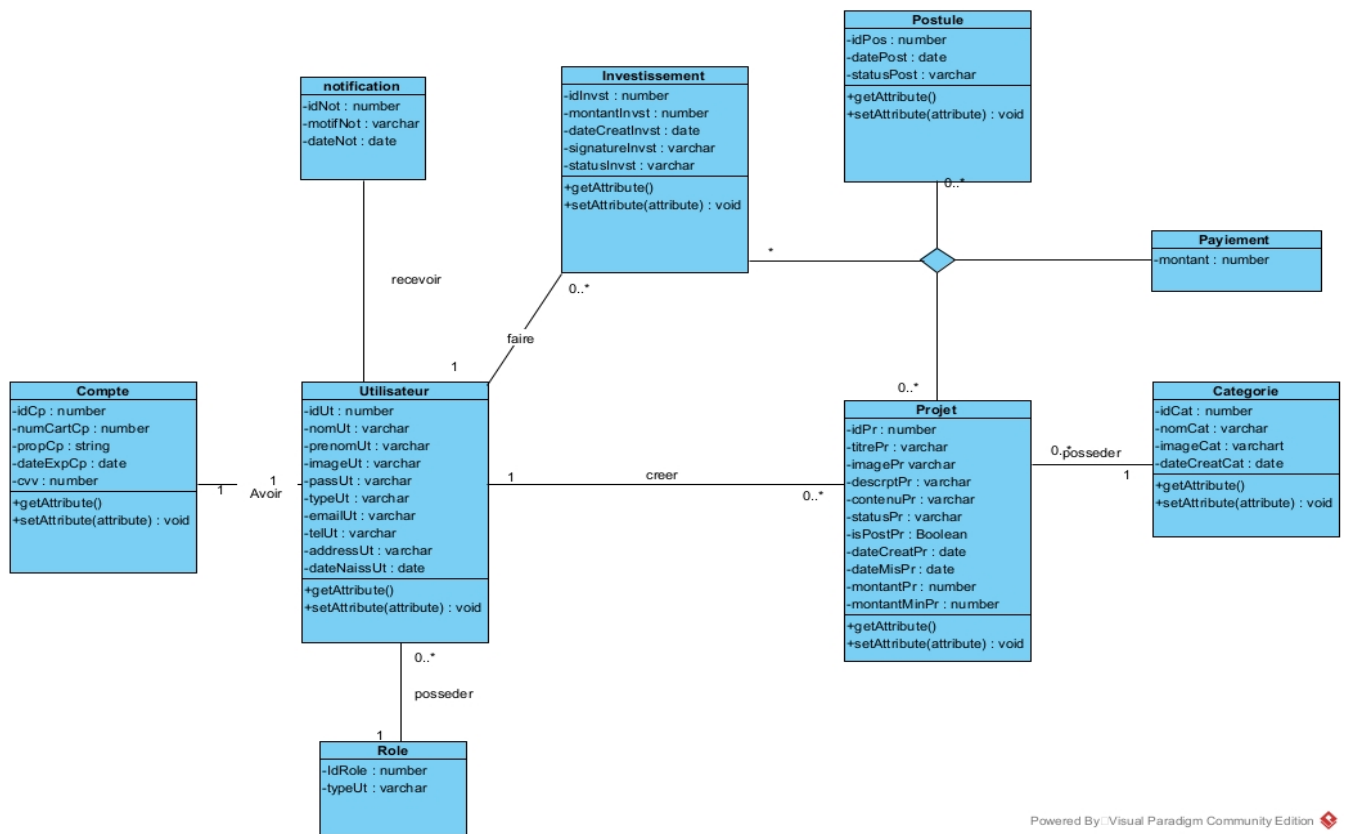


Figure 29: Diagramme de classe de conception globale

6.5. Diagramme de paquetage

Figure 30 illustre le diagramme de paquetage

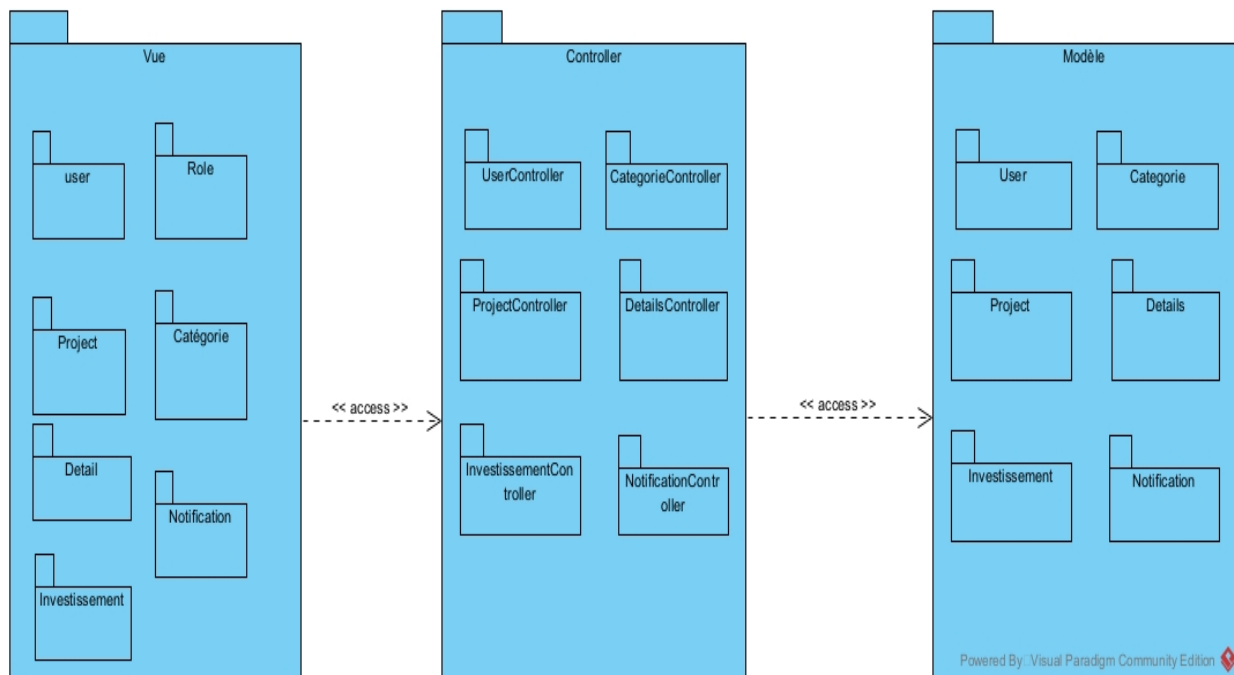


Figure 30: le diagramme de paquetage

6.6. Diagramme de déploiements

Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels et les ressources matérielles sont représentées sous forme de nœuds. Les nœuds sont connectés entre eux, à l'aide d'un support de communication. La nature des lignes de communication et leurs caractéristiques peuvent être précisées. Les diagrammes de déploiement peuvent montrer des instances de nœuds (un matériel précis), ou des classes de nœuds.

La figure 31 montre le diagramme de déploiement de l'application

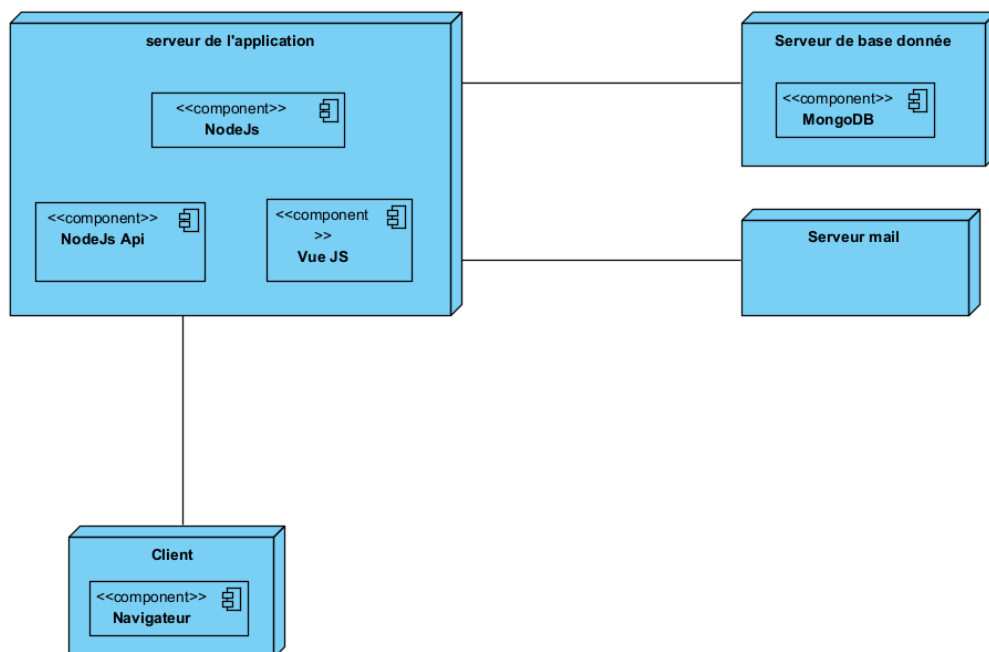


Figure 31: Diagramme de déploiement de l'application

PARTIE III : REALISATION

Chapitre 7. Mise en place de l'environnement de développement

7.1. Installation et configuration des outils

On va procéder maintenant à l'installation de et configuration des outils :

7.1.1. Visual Paradigm sur windows

Visual paradigm est utilisé pour la partie conception qui aboutit surtout à l'obtention de la base de données à partir du diagramme de classe.

Figure 32 montre la toute première étape de son installation

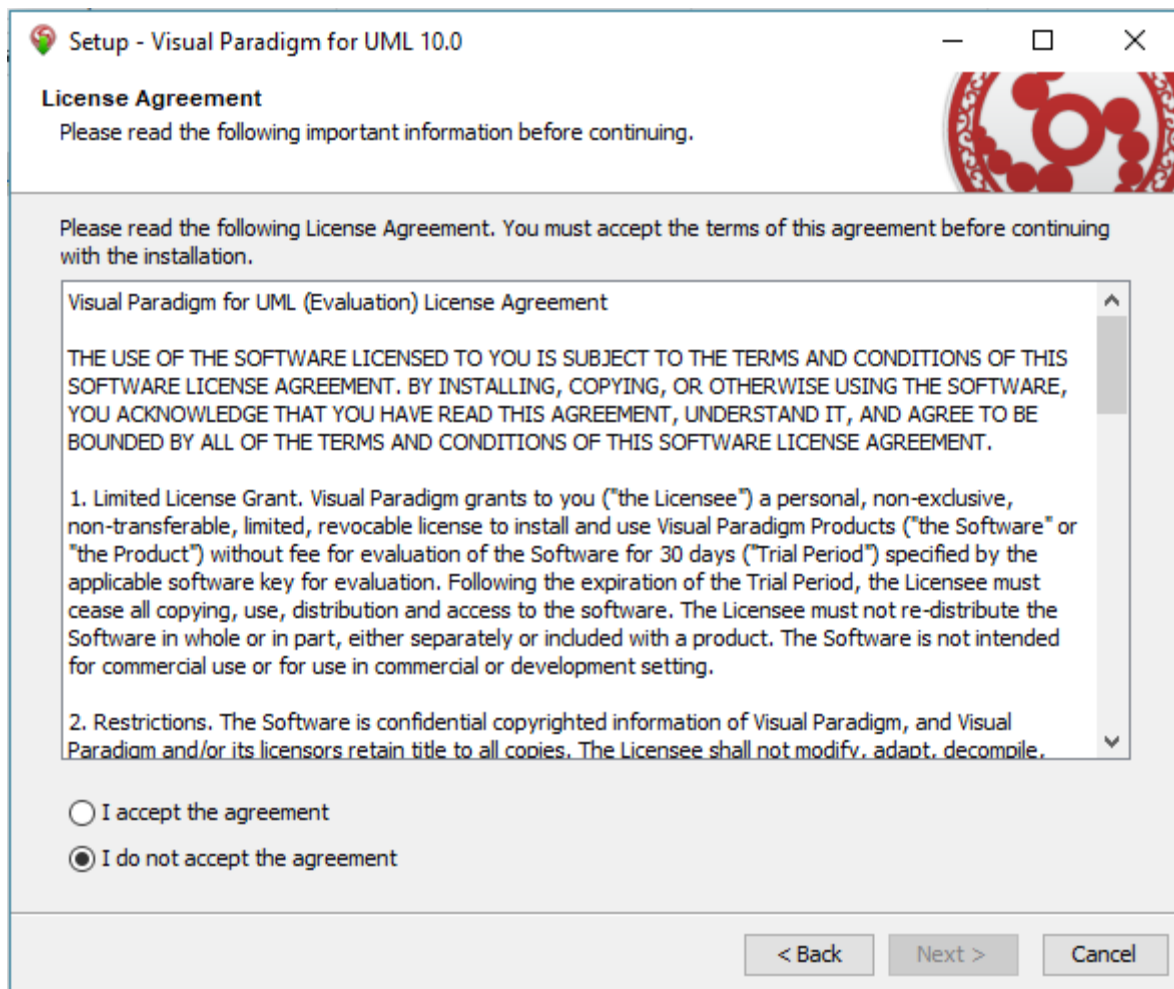


Figure 32: Installation de visual paradigm

7.1.1. Installation de NodeJs , Mongo DB , Visual Studio Code , Robo 3T sous windows

❖ NodeJS

Figure 33 Page de téléchargement de node Js

<div> <div>  <p>Windows Installer node-v0.10.21-x86.msi</p> </div> <div>  <p>Macintosh Installer node-v0.10.21.pkg</p> </div> <div>  <p>Source Code node-v0.10.21.tar.gz</p> </div> </div>		
Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.exe)	32-bit	64-bit
Mac OS X Installer (.pkg)	Universal	
Mac OS X Binaries (.tar.gz)	32-bit	64-bit
Linux Binaries (.tar.gz)	32-bit	64-bit
SunOS Binaries (.tar.gz)	32-bit	64-bit
Source Code	node-v0.10.21.tar.gz	

Figure 33 page de téléchargement de Node JS

Après le téléchargement du fichier d'installation Node JS , juste double Click sur le fichier d'installation , puis procéder à l'installation de Node JS.

Figure 34 montre l'installation de Node JS

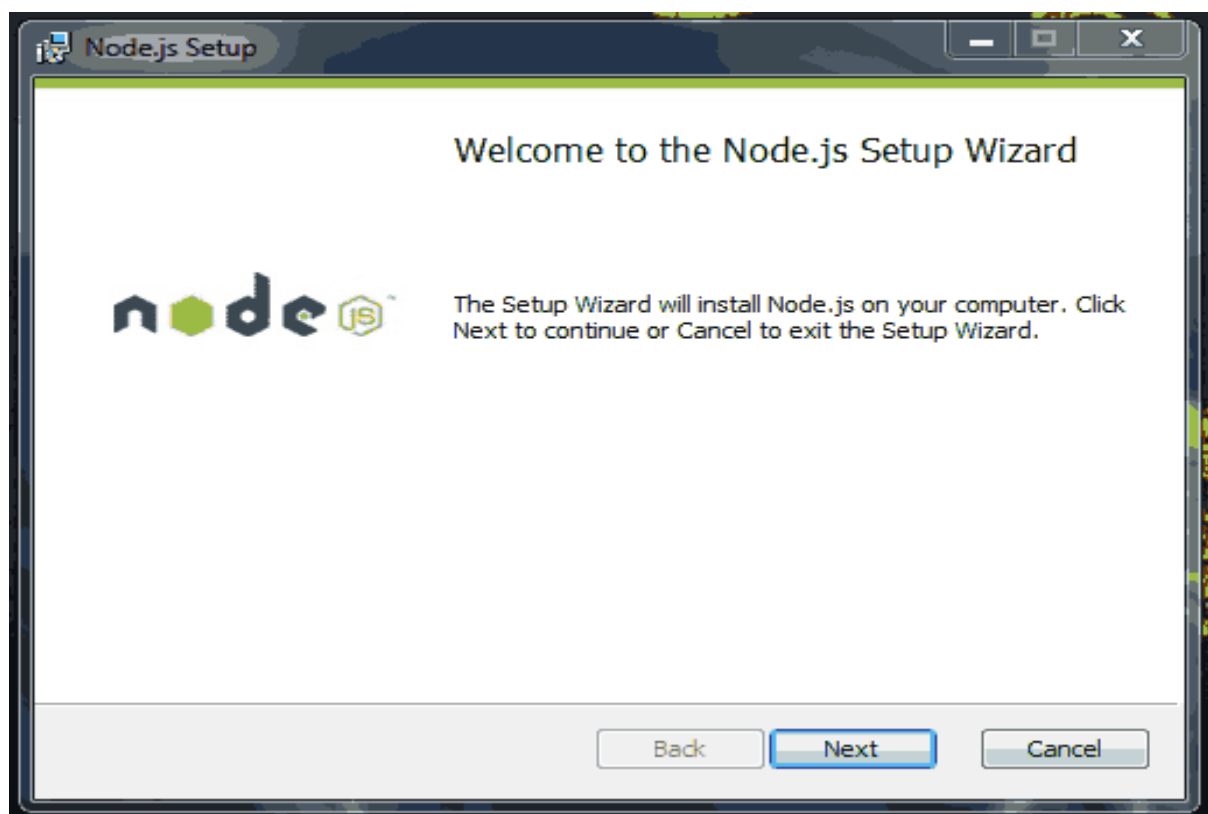


Figure 34 Installation De Node JS

❖ Mongo DB

Après téléchargement sur la page officielle de Mongo DB , voici l'installation de mongoDB.
Figure 35 Montre l'installation de Mongo DB

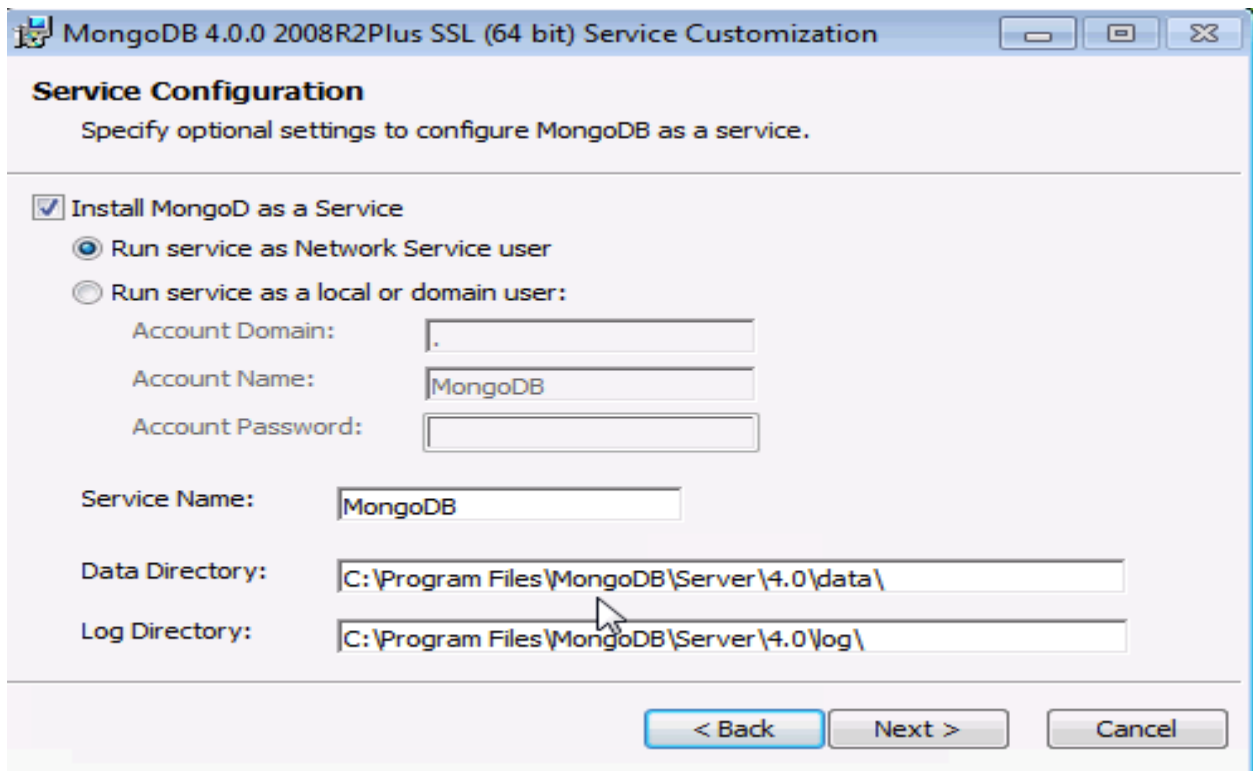


Figure 35 Installation de MongoDB

❖ Visual Studio Code

Figure 36 Montre le procédé d'installation de Visual studio Code.

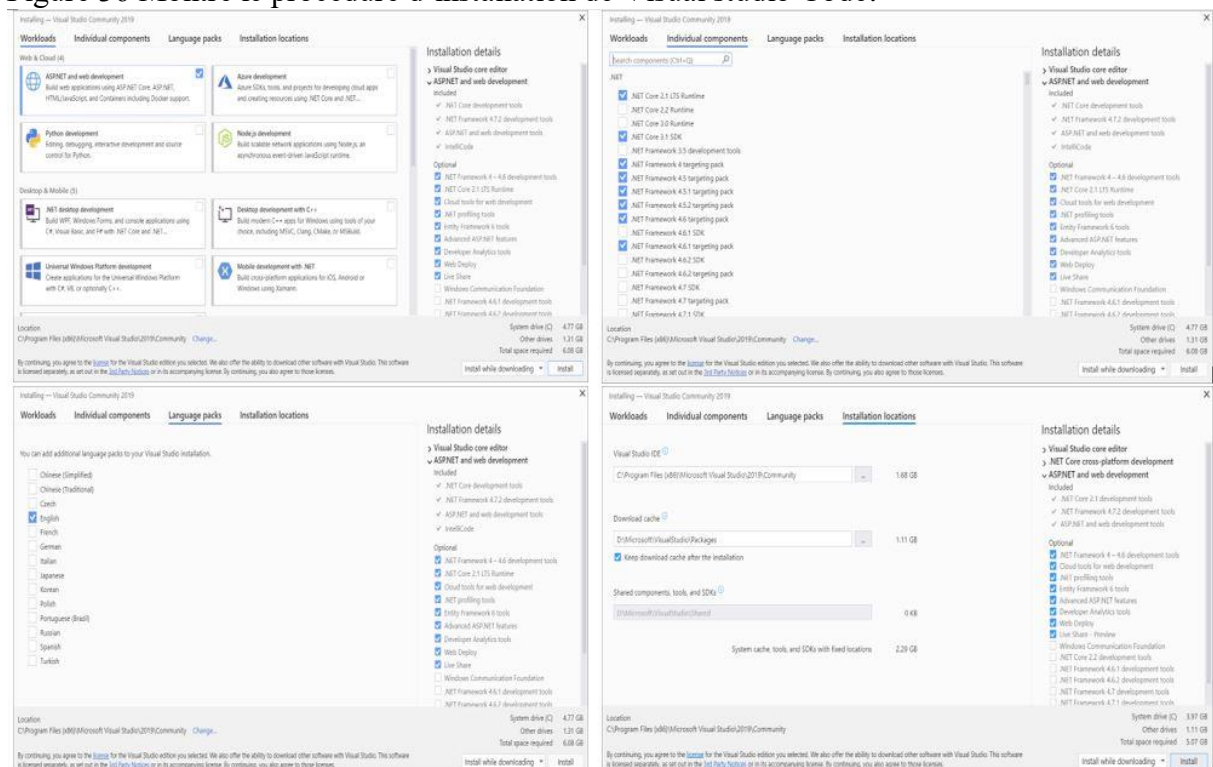


Figure 36 Etapes pour installer VS code

À la fin de l'installation, on devra obtenir la fenêtre indiquée sur la figure 37.

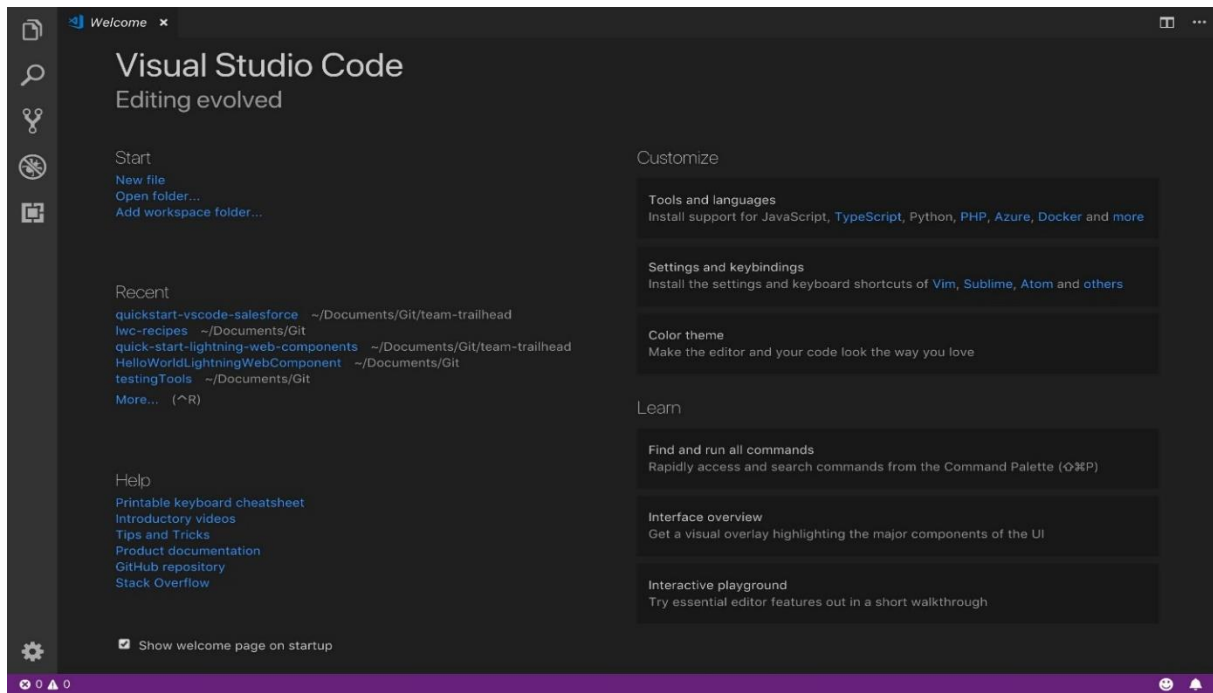


Figure 37 Fenêtre d'ouverture de VScode.

❖ Robo 3T

Figure 38 montre l'installation de Robo 3T

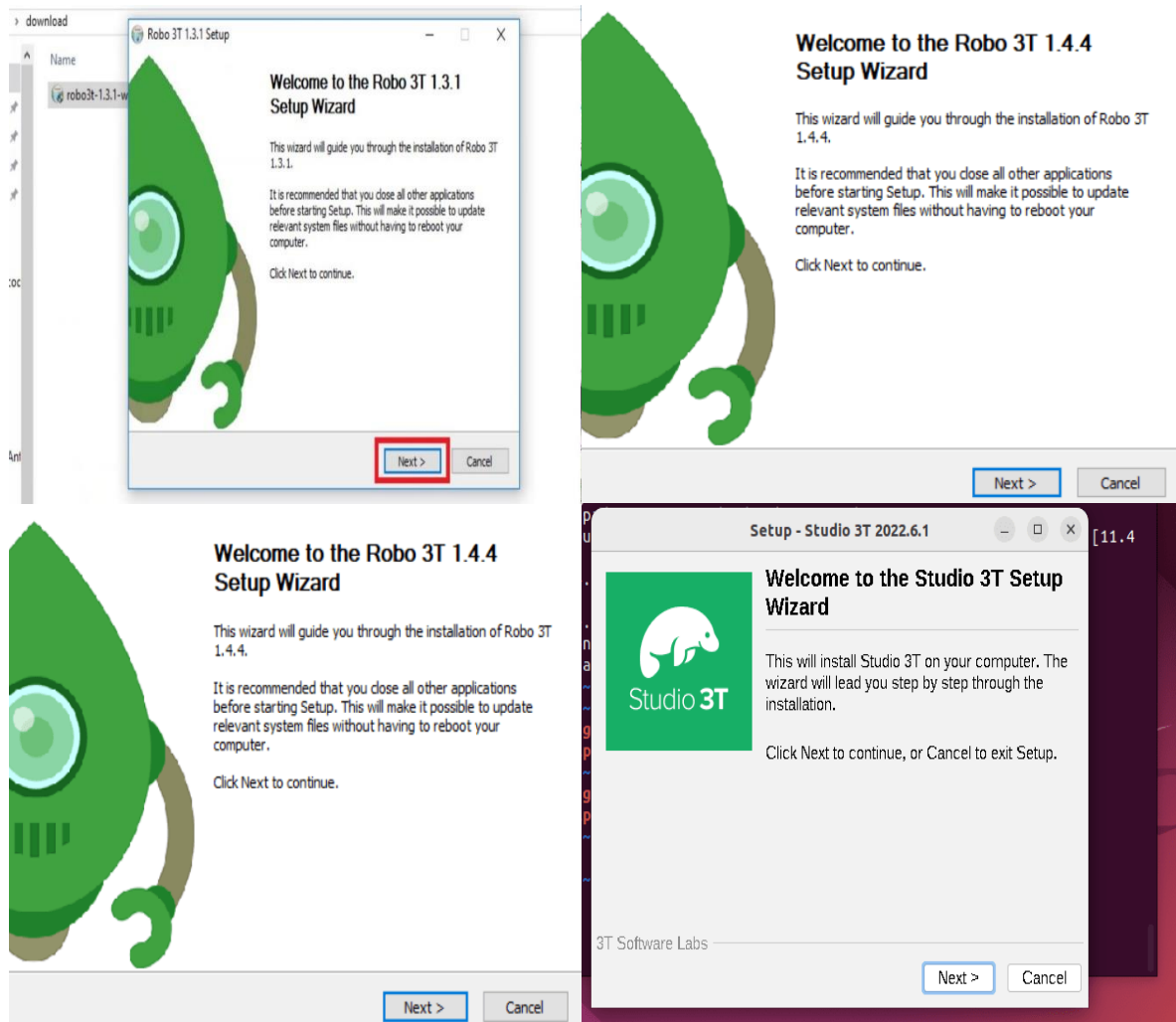


Figure 38 Installation de Robo 3T

7.2. Architecture de l'application

Vu que les besoins de la société adoptent le style, architecture 3-tiers, il sera préférable d'adopter ce type d'architecture, cela économisera le cout et le temps.

L'architecture trois tierce architecture 3-tiers, architecture à trois niveaux ou architecture à trois couches est l'application du modèle plus général qui est le multi-tiers. L'architecture logique du système est divisée en trois niveaux ou couches :

- Couche de présentation ;
- Couche Métier ;
- Couche accès aux données.

La figure 39 illustre l'architecture 3-tiers

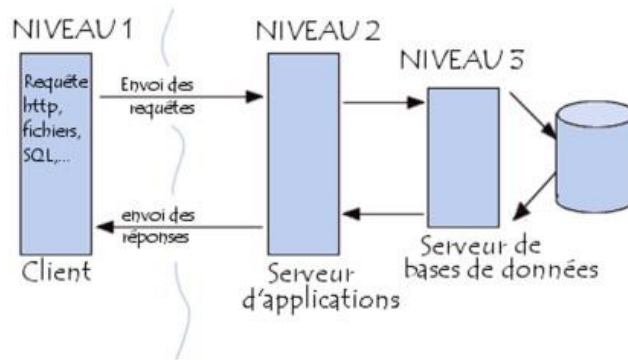


Figure 39 Architecture 3 tiers

Chapitre 8. Développement de l'application

On va voir maintenant les étapes du développement

Dans cette projet , On a utilisé Mongoose pour faire une relation au MongoDB.

Avec un composant dotenv on écrira la configuration de la base de données.

DATABASE_URL = « mongodb://localhost:27017/invest »

Configuration du serveur Node JS:

8.1. Création de la base de données

```

import express from "express";
import mongoose from "mongoose";
import auth from "./src/routes/authRoutes/index.js";
import categorie from "./src/routes/categorie/categorie.js";
import project from "./src/routes/project/project.js";
import detail from "./src/routes/detail/detail.js";
import cors from "cors";

import dotenv from "dotenv";

dotenv.config();
import subs from "./src/routes/sub/subs.js";
//express function
const app = express();

//connect to mongoDB database

mongoose.connect("mongodb://localhost:27017/invest", {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});
  
```

```
const db = mongoose.connection;
db.on("error", (error) => console.error(error));
db.once("open", () => console.log("Database Connected"));
```

Configuration de tout le routes

```
app.use(cors());
app.use(express.json({ limit: "50mb" }));
app.use("/auth", auth);
app.use("/categorie", categorie);
app.use("/project", project);
app.use("/subs", subs);
app.use("/detail", detail);
```

Configuration du port d'écoute du server:

```
app.listen("3002", () => console.log("Server Running at port : 3002"));
```

Pour lancer le serveur: Nodemon index.js

8.2. Codage de d'application

Voici quelque ligne de code avec quoi toute est centré dans le modèle.

8.2.1. Code modele

➤ Code de notre modèle Utilisateur et projet

```
import mongoose from "mongoose";

const { Schema } = mongoose;

const userSchema = new Schema(
  {
    firstName: {
      type: String,
      trim: true,
      required: true,
      unique: false,
    },
    lastName: {
      type: String,
```

```

    trim: true,
    required: true,
    unique: false,
  },
  password: {
    type: String,
    required: true,
    min: 5,
  },
  email: {
    type: String,
    trim: true,
    required: true,
    unique: true,
  },
  image: {
    type: String,
    trim: true,
    required: true,
  },
  typeUser: {
    type: String,
    trim: true,
    required: true,
  },
  customerStripeId: {
    type: String,
    required: true,
  },
},
{ timestamps: { createdAt: "created_at" } }
);

export default mongoose.model("User", userSchema);

import mongoose from "mongoose";

const { Schema } = mongoose;

const projectSchema = new Schema(
  {

```



```
user: {
  type: Object,
  required: true,
},

title: {
  type: String,
  trim: true,
  required: true,
},
categorie: {
  type: String,
  trim: true,
  required: true,
},
description: {
  type: String,
  required: true,
},
content: {
  type: String,
  required: true,
},
amount: {
  type: Number,
  required: true,
},
amountMin: {
  type: Number,
  required: true,
},
status: {
  type: String,
  default: "En attente",
},
isPotuled: {
  type: Boolean,
  default: false,
},

image: {
  type: String,
```

```

    trim: true,
    required: true,
  },
},
{ timestamps: { createdAt: "created_at" } }
);

export default mongoose.model("Project", projectSchema);

```

8.2.2. Code du controller

- Code notre controller utilisateur

```

import User from "../models/user.js";
import bcrypt from "bcryptjs";
import JWT from "jsonwebtoken";

import { stripe } from "../Utils/stripe.js";
import { validationResult } from "express-validator";
export const signUp = async (req, res) => {
  const { firstName, lastName, email, password, image, typeUser } =
req.body;
  const validationError = validationResult(req);
  if (!validationError.isEmpty()) {
    const errors = validationError.array().map((error) => {
      return {
        msg: error.msg,
      };
    });
  }
  return res.json({ errors, data: null });
}

const user = await User.findOne({ email });
if (user) {
  return res.json({
    errors: [
      {
        msg: "Email déjà utilisé",
      },
    ],
    data: null,
  });
}

```

```

}
const hashedPassword = await bcrypt.hash(password, 10);
console.log(process.env.STRIPE_SECRET_KEY);
const customer = await stripe.customers.create(
  {
    email,
  },
  {
    apiKey: process.env.STRIPE_SECRET_KEY,
  }
);

const newUser = await User.create({
  firstName,
  lastName,
  typeUser,
  image,
  email,
  password: hashedPassword,
  customerStripeId: customer.id,
});

console.log(process.env.JWT_SECRET);
const token = await JWT.sign(
  { email: newUser.email },
  process.env.JWT_SECRET,
  {
    expiresIn: 360000,
  }
);

res.json({
  errors: [],
  data: {
    token,
    user: {
      id: newUser._id,
      firstName: newUser.firstName,
      lastName: newUser.lastName,
      email: newUser.email,
      typeUser: newUser.typeUser,
      image: newUser.image,
      stripeCustomerId: customer.id,
    }
  }
});

```

```

    },
  },
});
};

export const login = async (req, res) => {
  const { email, password } = req.body;

  const user = await User.findOne({ email });
  if (!user) {
    return res.json({
      errors: {
        msg: "Invalids credentials",
      },
      data: null,
    });
  }
  const isMatch = await bcrypt.compare(password, user.password);

  if (!isMatch) {
    return res.json({
      errors: [
        {
          msg: "Invalid password",
        },
      ],
      data: null,
    });
  }
  const token = await JWT.sign({ email: user.email },
process.env.JWT_SECRET, {
  expiresIn: 3600000,
});
  return res.json({
    errors: [],
    data: {
      token,
      user,
    },
  });
};

```

➤ Code notre controller Projet

```
import Detail from "../models/detail.js";

export const createDetail = async (req, res) => {
  const detail = new Detail(req.body);

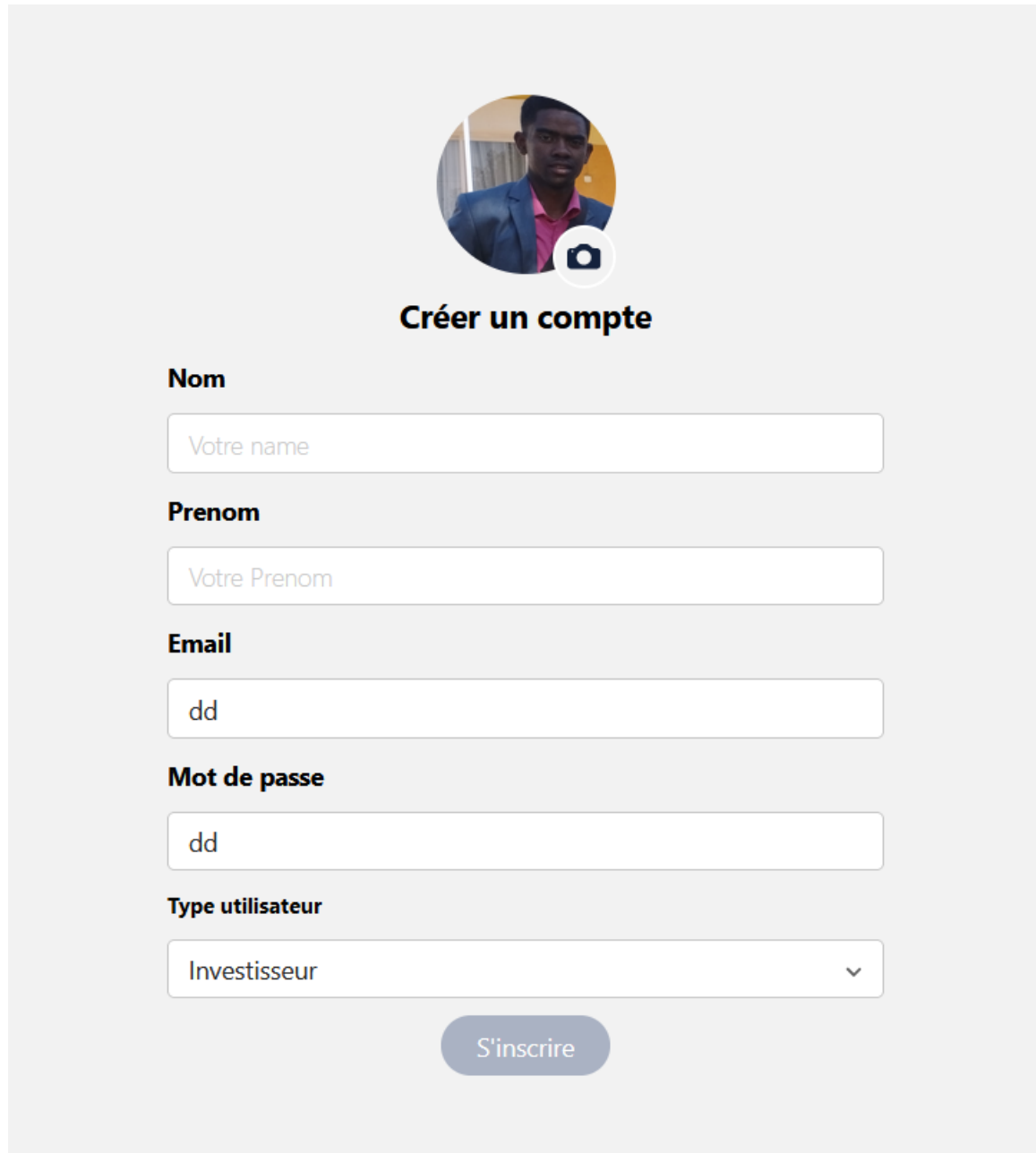
  try {
    const createdDetail = await detail.save();
    res.status(201).json({
      error: [],
      data: createdDetail,
    });
  } catch (error) {
    res.status(400).json({ msg: error.message });
  }
};

export const getDetail = async (req, res) => {
  try {
    const data = await Detail.find();
    return res.status(200).json({
      error: [],
      data,
    });
  } catch (error) {
    res.status(500).json({ msg: error.message });
  }
};
```

8.3. Présentation de l'application

- Page d'inscription

Figure 40 montre la Page d'inscription

The image shows a registration page with a light gray background. At the top center is a circular profile picture of a man in a blue suit and pink shirt, with a camera icon overlaid. Below the picture is the text "Créer un compte". The form consists of several labeled input fields: "Nom" with a placeholder "Votre name", "Prenom" with a placeholder "Votre Prenom", "Email" with the text "dd", "Mot de passe" with the text "dd", and "Type utilisateur" with a dropdown menu showing "Investisseur". At the bottom center is a blue button labeled "S'inscrire".

Créer un compte

Nom

Prenom

Email

Mot de passe

Type utilisateur

S'inscrire

Figure 40 Page d'inscription

➤ Page d'authentification

Figure 41 Montre la page d'authentification

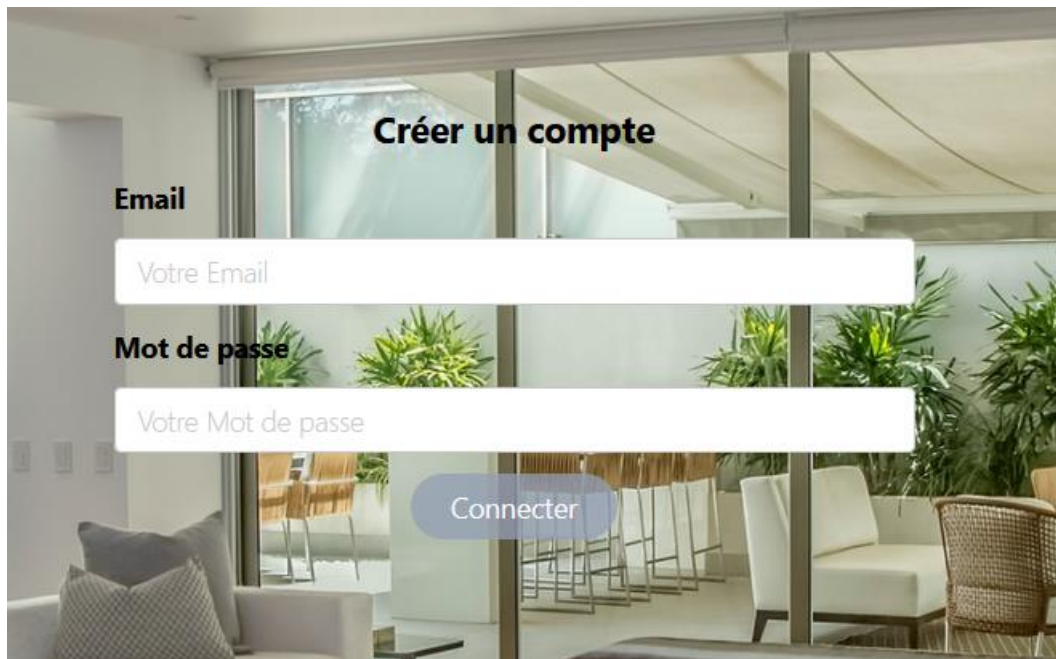


Figure 41: Page d'authentification

➤ Gestion de catégorie de projet

Figure 42 Gestion de catégorie de projet

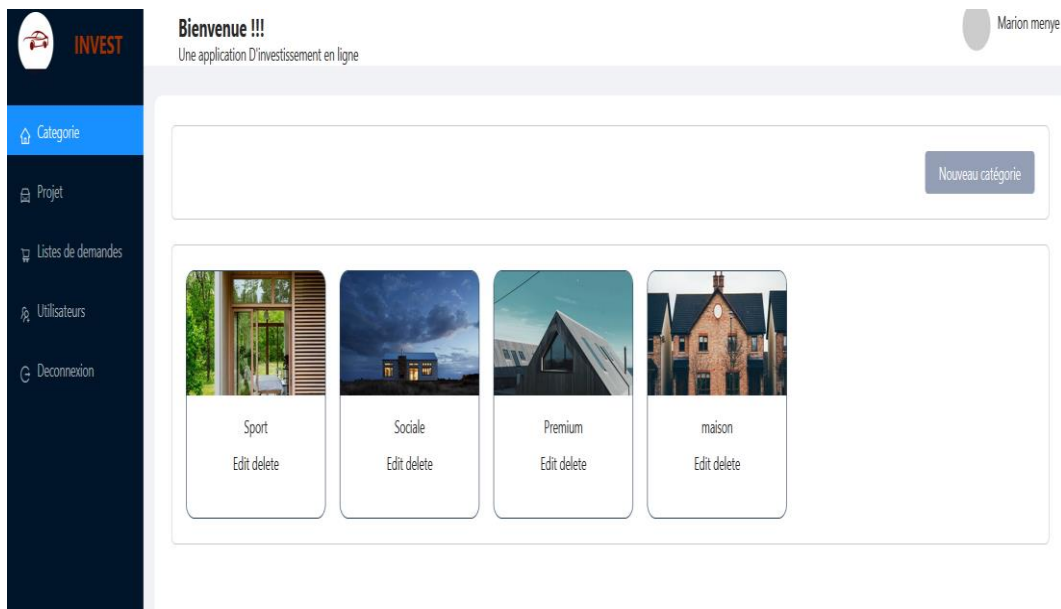


Figure 42 Gestion de catégorie de projet

❖ Liste des projets

Figure 43 Présente la liste des projets

INVEST Accueil Projets Facture Investisseurs

marion1menye
marion1@gmail.com

Role user Apporteur

Adresse Tana

Mes Projets

Mes Investisseurs

Mes Factures

Chiffre d'affaire

Chart graphique

Mes Projets

Nouveau Projet Rechercher

Filtrer Catégorie: Trier Montant: Filtrer Postule:

select select select

marion1 menye
marion1@gmail.com

PROJET	Montant	Statut
PROJET 1	100 000 000	En attente
PROJET 2	200 000 000	En attente
PROJET 3	300 000 000	En attente
PROJET 4	400 000 000	En attente
PROJET 5	500 000 000	En attente

Maison Sociale

En attente

Montant Totale: 3000000000 £

marion1 menye
marion1@gmail.com

immeuble Sociale

En attente

Montant Totale: 5666654 £

Figure 43 : la liste des projets

➤ Liste des projets affichage public

Figure 44 Montre la liste des projets en affichage public

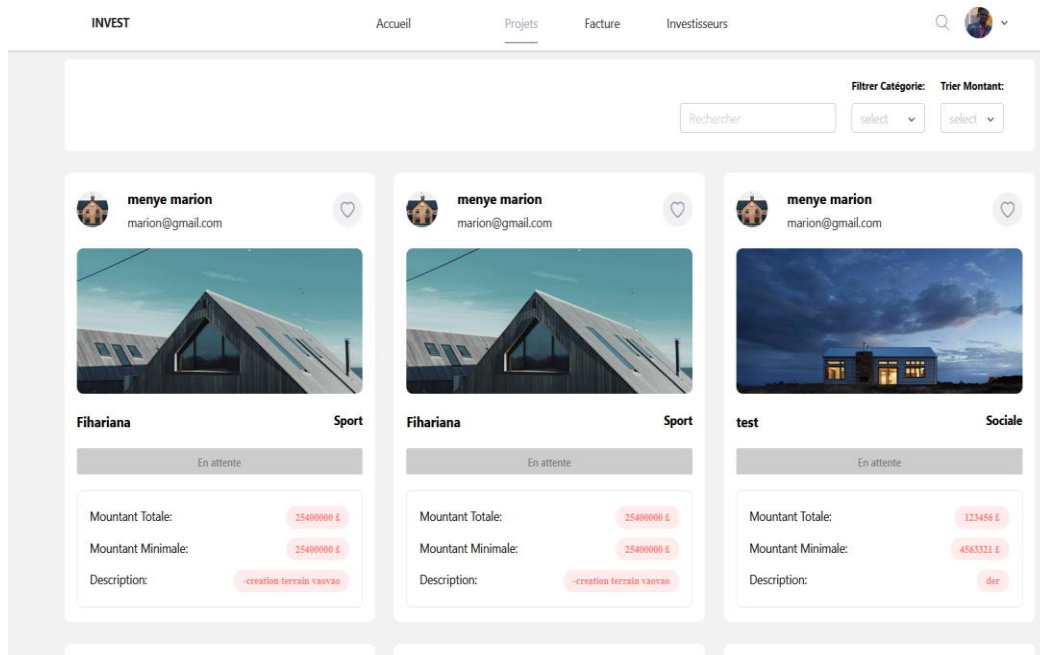


Figure 44: La liste des projets en affichage public

➤ Page d'abonnement pour voir la détail

Figure 45 Montre la page d'abonnement pour voir la détail de projet

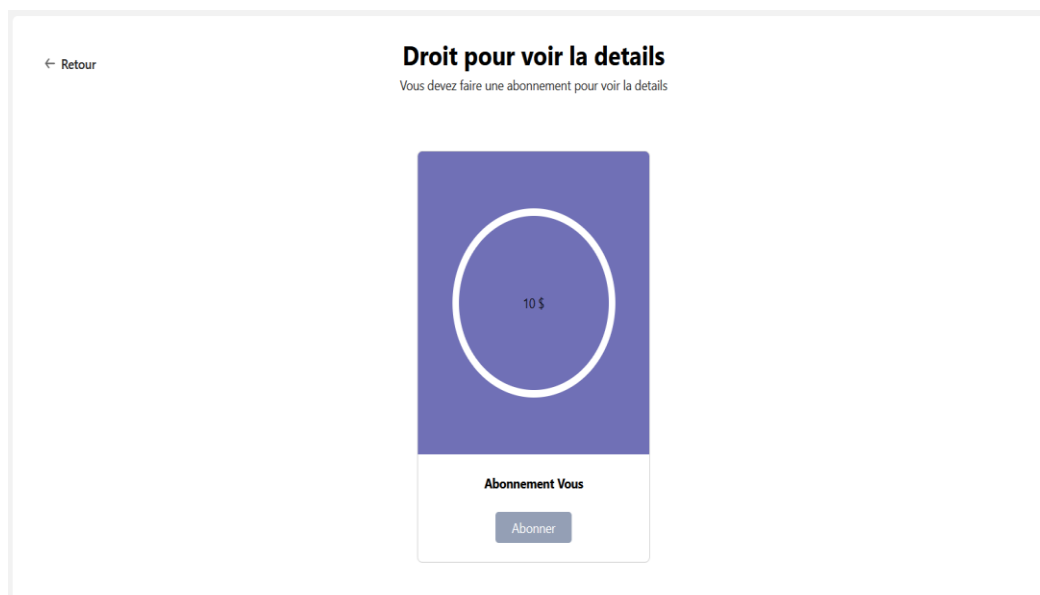


Figure 45 la page d'abonnement pour voir la détail de projet

➤ Page de contrat d'investissement d'un projet

Figure 46 illustre la page de contrat d'investissement d'un projet

contrat investissement

Formulaire d'investissement


Montant à payer

Montant à payer

Date de paiement

jj / mm / aaaa


Votre Signature



✓
✗

menye marion

marion@gmail.com



Fihariana

Sport

En attente

Montant Totale: 25400000 €

Montant Minimale: 25400000 €

Description: -creation terrain
YAOYAO

Edit

Supprimer

Postulée

Figure 46 Page de contrat d'investissement d'un projet

CONCLUSION

Ce stage s'est déroulé au sein de la société MyAgency. Le présent logiciel a été réalisé en se basant sur le pratique habituel de faire un investissement a des chose, pour réaliser le projet d'associer leur pratique aux nouvelles technologies de l'information et de la communication, MyAgency nous a recrutés en tant que stagiaire. Rappelons que l'objectif du projet est de faire développer une application web pour la gestion d'investissement d'un projet en ligne.

L'application permet en général d'apporter un idée à un projet au Platform qui a pour but de trouver des investisseurs pour la réalisation de ce projet. On a intégré un système de paiement à cette application pour pouvoir faire tout le virement au niveau d'investissement et d'abonnement au platform.

Pour sa réalisation, on a utilisé 2TUP comme méthode de conception, NodeJS comme langage de programmation, SCRUM pour la gestion de projet. Pour l'hébergement, on a utilisé le Server Node JS et en ce qui concerne la SGBD, on a utilisé MongoDB et pour la conception dusystème, on a utilisé UML avec visual paradigm pour les différents diagrammes.

Cette application est opérationnelle et répond au besoin du client. Néanmoins dans la perspective, on envisage de mettre à disposition des recruteurs ce puissant outil afin de les aider à recruter des vrais talents lors des recrutements. Identifier les menteurs parce que de nos jours, les CV sont boostés par les candidats et ainsi trouvé une personne pour bien investir.

Ce stage nous a permis de nous familiariser avec le mode professionnel et d'appliquer les connaissances acquises à l'ENI. Elle nous a fait vivre une expérience formidable et importante dans le mode du développement.

BIBLIOGRAPHIE

- [1] P. Kruchten, 2000, Introduction au Rational Unified Process, Edition EYROLLES.
- [2] Pascal ROQUES, 2008, Les Cahiers du programmeur UML2 – Modéliser une application, Edition EYROLLES, 4è édition, 264 pages.
- [3] F. Juliard UML Unified Method Language, Journal Université de Bretagne Sud UFR SSI-IUP Vannes, 2001-2002.
- [4] Taha M'HAND, 2009, Modélisation Conceptuelle BD 2, 23 pages
- [5] Bacco, B. (2018). Développez votre site Web avec le framework Symfony 4. Eyrolles.
- [6] Saliha, Y. (2013). User-stories et Backlog de produit.
- [7] Foundikou, V. (2005). Modèle Conceptuelle orientée Objet, Eyrolles.
- [8] Piechoki, L. (2007). UML, le langage de modélisation Objet unifié. Corel.
- [9] Voisin, F. (2014). Les diagrammes de séquence en phase de conception. Polytech ParisSud
- [10] Roques, P. Vallée, F. (2007). UML2 en action. Eyrolles.
- [11] Aubrey, C. (2011). SCRUM le guide pratique de la méthode agile le plus populaire. Dunod.
- [12] Laurent AUDIBERT, 2007, UML 2, Edition 2007-2008, 178 pages
- [13] GABAY J. et GABAY D., DUNOD, 2008, « UML 2 Analyse et conception », page 2 et 46.
- [14] Joseph Gabay et David Gabay, Mise en œuvre guidée avec études de cas, édition Dunod, Paris 2008
- [15] ROQUES P., EYROLLES, 2006. « UML 2 par la pratique », page 76 et 118.

WEBOGRAPHIE

- [16] <http://laurent-audibert.developpez.com/Cours-UML/>, consulté le 22 Avril 2021
- [17] http://fr.wikipedia.org/wiki/Diagramme_sequence/, définition de diagramme de séquence, 22 avril 2022
- [18] <http://www.additeam.com/SSII/systeme-d%E2%80%99information-si/>, définition de Système consulté le 3 Mai 2022
- [19] <http://www.iro.umontreal.ca/~pift3901/Slides/A05-S05-ModDom.pdf>, Modèle du Domaine, consulté en 6 Mai 2022
- [20] <http://book.cakephp.org/2.0/fr/cakephp-overview/understanding-modelviewcontroller.html>, Comprendre le système MVC (Model-View-Controller) consulté le 12 Mai 2022
- [21] <http://www.letudiant.fr/etudes/tout-pour-reussir-son-memoire-son-rapport-destage.html>, Tout pour réussir son mémoire, son rapport de stage, Consulté le 12 Mai 2022.
- [21] <http://www.letudiant.fr/etudes/tout-pour-reussir-son-memoire-son-rapport-destage.html>, Tout pour réussir son mémoire, son rapport de stage, Consulté le 22 Mai 2022.
- [22] <https://www.journalducsm.com/trello-guide-utilisation/>, Trello, consulté 28 Mai 2022
- [23] https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions, Logiciel de gestion de versions, consulté le 02 Juin 2022
- [24] https://fr.wikipedia.org/wiki/Langage_de_programmation, Langage de programmation, consulté le 03 Juin 2022
- [25] <https://blog.lws-hosting.com/conseils-marketing/symfony-zend-cakephp-codeigniterlequelchoisir>, consulté le 12 Juin 2022
- [26] https://ineumann.developpez.com/tutoriels/alm/agile_scrum/, Présentation des méthodes agiles et Scrum, consulté le 23 Juin 2022
- [27] <http://bahit.fr/prod/2018/06/11/la-methode-agile-scrum/>, La méthode SCRUM AGILE, consulté le 5 Juillet 2022

GLOSSAIRE

Application Desktop : C'est un logiciel applicatif qui affiche son interface graphique dans un environnement de bureau, il est hébergé et exécuté par l'ordinateur de l'utilisateur

Base de données : C'est un conteneur informatique de fichier ou ensemble de fichier disque mémoire permettant le stockage permanent ou temporaire et l'accès à des informations structurées.

Langage de programmation : est un langage informatique, permettant à un être humain d'écrire un code source qui sera analysé par une machine, généralement un ordinateur. Le code source subit une transformation ou une évaluation dans une forme exploitable par la machine, ce qui permet d'obtenir un programme.

Requête : instruction permettant d'interroger une base de données

SGBD : logiciel permettant de stocker les données, de mettre à jour et de les restituer.

SQL : langage structure de requête pour l'interrogation des bases des données importantes.

JAVA : est un langage de programmation orienté objet avec interface graphique

Codification : Ecrire de code de façon logique et relationnel.

Invest: Nom de choisi pour cette application ou logiciel.

Apporteur : rôle de propriétaire de projet de la plateforme.

Investisseur : rôle d'investisseur de projet de la plateforme.

TABLE DES MATIERES

CURRICULUM VITAE.....	I
SOMMAIRE GENERAL	IV
REMERCIEMENTS	VI
LISTE DES FIGURES	VII
LISTE DES TABLEAUX	VIII
LISTE DES ABREVIATIONS.....	IX
INTRODUCTION GENERALE.....	1
PARTIE I : PRESENTATIONS	2
Chapitre 1. Présentation de l'Ecole Nationale d'Informatique	3
1.1. Information d'ordre générale.....	3
1.2. Missions et historiques.....	3
1.3. Organigramme institutionnel de l'ENI	6
1.4. Domaines et spécialisation	7
1.5. Architecture des formations pédagogiques	8
1.6. Relation de l'ENI avec les entreprises et organismes.....	10
1.7. Partenariat au niveau international	11
1.8. Débouchés professionnels des diplômés	13
1.9. Ressources humaines	16
Chapitre 2. Présentation de MyAgency	17
2.1. Historiques et missions	17
2.1. Situation juridique	17
2.2. Organigramme.....	17
2.3. Services.....	18
2.4. Contacts	18
Chapitre 3. Description du projet.....	19
3.1. Formulation.....	19
3.1. Objectif et besoin d'utilisateur.....	19
3.1.1. Objectif du projet	19
3.1.2. Besoin de l'utilisateur.....	19
3.2. Moyen nécessaire à la réalisation du projet	20
3.2.1. Moyens humains	20
3.2.2. Moyens Matériels.....	21
3.2.3. Moyens logiciels	22
3.2.4. Résultats attendus	22
3.2.5. Chronogrammes des travaux.....	23

PARTIE II : ANALYSE ET CONCEPTION	24
Chapitre 4. Analyse préalable.....	25
4.1. Analyse de l'existant	25
4.1.1. Organisation Actuelle.....	25
4.1.2. Inventaire des moyens matériels et logiciels	25
4.2. Critique de l'existant	26
4.3. Conception avant projet.....	26
4.3.1. Comparaison de solutions.....	26
4.3.2. Choix et justification	27
a) Méthode	28
b) Outils de modélisation	33
c) Langage de modélisation utilisé	35
d) Système de gestion de base de données	35
e) Serveur	39
f) IDE.....	39
g) Système de gestion de version.....	40
h) Choix de langage de programmation	42
i) Framework	44
j) Choix des outils de gestion de projet	46
k) Méthode de gestion de projet	47
Chapitre 5. Analyse Conceptuelle	51
5.1. Etapes de l'élaboration du backlog de produit.....	51
5.1.1. Vision globale de projets	51
5.1.2. Liste détaillée des acteurs avec fiche	51
5.1.3. Thème et regroupement de fonctionnalités.....	54
5.1.4. Vision du premier release	62
5.1.5. User stories	62
5.1.6. Backlog de produit	63
5.2. Dictionnaire de données	64
5.3. Règle de gestion	69
5.4. Capture des besoins fonctionnels	70
5.5. Analyse	70
5.5.1. Diagramme de cas d'utilisation.....	70
5.5.2. Priorisation et description des cas d'utilisation.....	73
a) Priorisation de cas d'utilisation	73
b) Détails de cas d'utilisation	73
5.6. Capture des besoins techniques.....	77

5.7. Conception préliminaire	77
5.7.1. Diagrammes des séquences système	77
5.7.2. Modélisation du domaine	84
Chapitre 6. Conception détaillée.....	85
6.1. Architecture du système	85
6.2. Diagramme de séquence de conception.....	87
6.3. Diagramme de classe de conception	88
6.4. Diagramme de classe de conception global	89
6.5. Diagramme de paquetage	90
6.6. Diagramme de déploiements.....	91
PARTIE III : REALISATION.....	92
Chapitre 7. Mise en place de l'environnement de développement	93
7.1. Installation et configuration des outils	93
7.1.1. Visual Paradigm sur windows	93
7.1.1. Installation de NodeJs , Mongo DB , Visual Studio Code , Robo 3T sous windows	93
7.2. Architecture de l'application.....	98
Chapitre 8. Développement de l'application	99
8.1. Création de la base de données	99
8.2. Codage de d'application	100
8.2.1. Code modele	100
8.2.2. Code du controller	103
8.3. Présentation de l'application.....	107
CONCLUSION	XI
BIBLIOGRAPHIE	XII
WEBOGRAPHIE.....	XIII
GLOSSAIRE.....	XIV
TABLE DES MATIERES	XV
RESUME	XVIII
ABSTRACT	XVIII

RESUME

Dans ce présent projet, nous avons conçu et élaboré une application destinée à la gestion d'investissement en ligne. Le projet consiste à mettre en œuvre un système de création d'un idée à un projet et de système de paiement pour investir à un projet. Dans ce document se trouvent tous les détails concernés, depuis l'étude de l'existant jusqu'à la réalisation de l'application, comme la présentation des outils utilisés et la modélisation du système. Nous avons comparé les outils utilisés par rapport à d'autres et étudié l'architecture de l'application après avoir relevé les besoins de l'utilisateur avec la méthode 2TUP liée à la notation UML et on a géré le projet avec la méthode AGILE SCRUM. L'application c'est abouti avec une interface facile à manipuler et les fonctionnalités requises par la cellule que nous avons illustrée dans des captures d'écran.

Mots-clés : NODEJS, Framework , Express JS, MongoDB , Robo 3T, modélisation, AGILE, SCRUM, UML, 2TUP.

ABSTRACT

In this present project, we have designed and developed an application for online investment management. The project is to implement a system of creating an idea to a project and a payment system to invest in a project. This document contains all the details concerned, from the study of the existing situation to the realization of the application, such as the presentation of the tools used and the modeling of the system. We compared the tools used compared to others and studied the architecture of the application after having identified the user's needs with the 2TUP method linked to UML notation and we managed the project with the AGILE SCRUM method. . The application is completed with an easy to handle interface and the functionalities required by the cell that we have illustrated in screenshots.

Keywords: NODEJS, Framework, Express JS, MongoDB, Robo 3T, modeling, AGILE, SCRUM, UML, 2TUP.