|  |  |  |
| --- | --- | --- |
|  |  |  |



par

Mémoire présenté  
MASTER de SCIENCES, TECHNOLOGIES, SANTE, mention INFORMATIQUE  
Spécialité MOBIQUITE, BASES DE DONNEES ET INTEGRATION DE SYSTÈMES (MBDS)

Jury :

Dr Olivier Robinson, président  
 M. Tahina Razafinjoelina, examinateur  
 M. Roger Dupont, encadreur professionnel

© ,

Table des matières

Liste des tableaux iv

Liste des figures v

Glossaire vi

Avant-propos 7

L’IT University 7

L’Institution/Entreprise d’accueil 8

Remerciements 8

Introduction 9

1 Présentation du stage 10

1.1 Présentation de l'entreprise 10

1.2 Présentation du sujet et objectifs du projet (problème traité) 10

2 État de l’art sur le sujet traité 11

2.1 Critères de comparaison 12

2.2 Etude de chaque solution au vu des critères 12

2.3 Tableau comparatif des solutions au vu des critères 12

3 Etude de l’existant et solution envisagée 13

3.1 Étude de l’existant 13

3.1.1 Description externe du système logiciel existant (vision utilisateur) 13

3.1.2 Description interne du système logiciel existant (vision développeur/conception) 13

3.2 Critique de l’existant 13

3.3 Solutions envisagées 13

3.4 Objectifs principaux et livrables 13

4 Démarche projet 14

4.1 Principes de la démarche projet 14

4.1.1 Activités d’ingénierie logicielle 14

4.1.2 Méthode de gestion de projet utilisée 14

4.1.3 Rôles et responsabilités 14

4.1.4 Outils 15

4.1.5 Gestion de la configuration 15

4.2 Contraintes et risques sur le projet 15

4.3 Démarche projet mise en œuvre 16

4.4 Budget du projet 16

5 Exigences réalisées dans le projet (vision externe/utilisateur) 17

5.1 Exigences fonctionnelles – Cas d’utilisation 17

5.1.1 Cas d’utilisation 1 17

5.1.2 Cas d’utilisation 2 17

5.1.3 Cas d’utilisation X 18

5.2 Exigences non fonctionnelles transverses 18

5.3 Interfaces détaillées 18

5.3.1 IHM 18

5.3.2 Interfaces avec d’autres systèmes 18

6 Architecture(s) système 19

7 Conception du système logiciel réalisée dans le projet (vision interne/développeur) 20

7.1 Plate-forme technique 20

7.2 Conception du logiciel développé 20

7.2.1 Conception du code source 20

7.2.2 Le code source – vue statique 20

7.2.3 Modélisation de données 20

7.2.4 Réalisation des cas d’utilisation 21

7.2.5 Les composants et leur déploiement 21

8 Tests du système logiciel 22

9 Conclusion générale 23

10 Références et Bibliographie 24

11 Annexes (à intégrer absolument si ce n’est pas déjà fait dans le rapport principal. Elles peuvent être dans des fichiers séparés) 25

NOTES IMPORTANTES (à supprimer sur le rapport final): 25

# Liste des tableaux

[Tableau 1 : Exemple tableau 12](#_Toc142476122)

# Liste des figures

[Figure 1: Exemple figure 19](#_Toc142476236)

# Glossaire

Astéroïde. Minuscule planète dont la taille varie entre 1 000 km à moins d'un km de diamètre. Les astéroïdes gravitent généralement autour de planètes plus grandes.

Atmosphère. Masse gazeuse entourant les planètes, dont la Terre.

Densité. Nombre de particules par unité de mesure.

# Avant-propos

Le présent mémoire présente les résultats du travail effectué lors de mon stage de fin d’études de Master en Informatique de l’IT University; stage effectué au département de durant 3 mois, de juin à Aout 2023.

Afin de poser clairement le contexte de ce mémoire, je vais présenter succincte d’une part de l’IT University et d’autre part de mon institution/entreprise d’accueil.

## L’IT University

Fondée en 2011, l’IT University (ou ITU) est une université privée, spécialisée en informatique, formant les jeunes bacheliers scientifiques :

* en trois ans, pour l‘obtention d’une Licence, option Développement d’applications, Réseaux et Bases de Données ou Web et Design
* en cinq ans, pour l’obtention d’un Master MBDS en coopération avec l’Université de Nice Sophia Antipolis - France

Étant une formation professionnalisante, l’ITU a tissé des liens forts avec ses partenaires industriels, dont l’opérateur convergent TELMA, le GOTICOM et la plupart des entreprises et institutions du secteur des TIC[[1]](#footnote-1). Ces partenaires participent effectivement à la formation par la fourniture de connexion Internet à haut débit, l’envoi de conférenciers ou encore par l’accueil des étudiants en stage.  
D’autre part, le corps enseignant de l’ITU est constitué intégralement de spécialistes de très haut niveau et obligatoirement actifs professionnellement dans leurs domaines respectifs.  
Enfin, l’ITU fait partie du programme Microsoft Imagine en plus d’être un Oracle Gold Partner.

## L’Institution/Entreprise d’accueil

Madamonitor a été créé en 2012, est l’unité de production ainsi que l’unité de développements informatiques de la société Belge Auxipress qui est une entreprise qui œuvre dans la collecte et le traitement des sources médias.

Auxipress fondée en 1919, a pour mission de surveiller tous les médias :

1.400+ journaux, magazines, flux de nouvelles ;

15 stations de radio belges ;

11 grandes stations de télévision +20 stations locales ;

+2.000 sites web professionnels d’information ;

2.000.000+ informations web importées.

L’équipe de développement informatique de Madamonitor développe les solutions informatiques dont Auxipress a besoin afin de satisfaire leurs besoins internes. Madamonitor a déjà développé plusieurs solutions informatiques depuis sa création qui sont déjà utilisé au sein d’Auxipress.

## Remerciements

L'auteur tient à exprimer sa sincère reconnaissance à MM. les Professeurs Smith et Jones pour leur aide lors de la préparation de ce manuscrit.   
Il souhaite également remercier tout particulièrement le Dr. Elsa Leavitt dont la maîtrise du cours, tant sur le plan des besoins que des idées exposées, a été d'une grande aide pour la mise en route de ce projet.   
Enfin, merci également aux membres du comité étudiant pour leur soutien.

# Introduction

De nos jours les utilisateurs d'internet ne cessent de s'accroitre, lire et voir des articles de presse ainsi que des nouvelles dans le monde du web est le moyen le plus rapide pour passer des informations. En tant que développeur en Informatique dans la société Madamonitor. Développer et Maintenir des applications concernant la surveillance des articles web est ma principale tâche.

Ayant fait des études en Informatique et en connaissant la principale activité de l'entreprise qui œuvre sur la récolte des sources de données des médias, traités plusieurs données des articles présente dans le web m'as poussé à choisir le projet sur le thème de Validation des Articles Web nommée Webselect qui consistent a développé une application web pour filtrer les articles et les validés vis à vis des demandes des clients, l'application webSelect existe déjà mais il a été développé dans une plateforme déprécie , mon objectif principal est de faire une migration de l'application vers une plateforme a jour et d'amélioré la performance sur les lectures et écritures des données des articles , pour faciliter la tâche des départements de valideurs web qui utilisent l'application et d'avoir une expérience accrue dans le domaine du développement de bases de données.

Ma mission principale consiste alors de développer et d'assurer cette migration ainsi que la maintenance et la stabilité de l'optimisation des données face à plusieurs situations comme la croissance des nombres des données dans l'ancien version.

Les valideurs web ont quelques soucient de visualiser les articles de certains clients lorsqu'ils présentent beaucoup de données, l'affichage devient très lent et cela augmentent leurs temps de travail et diminuent le nombre des taches qu'ils doivent traités par jour, cela a posé des soucis de performance et après étude de la situation mon projet de stage consistent à résoudre ce problème.

Pour bien rentrer dans le sujet il est nécessaire de développé ces points :

-la présentation détaillé du projet

-les étapes de la réalisation de l'application

-l'évaluation du projet et les connaissances acquises.

# Présentation du stage

WebSelect est un outil utile pour la production concernant la surveillance des articles, facilité le traitement est nécessaire pour faire évoluer la société et rendre les traitements des articles plus efficaces.

## Présentation de l'entreprise

Présentation sommaire de l'entreprise

## Présentation du sujet et objectifs du projet (problème traité)

La migration de WebSelect va être réorienté dans une toute nouvelle plateforme, en faisant plusieurs tests et en se mettant en place des utilisateurs une innovation importante doit être fait pour qu'ils soient à l'aise et bien motiver a utilisé cette application.

Le nombre des articles disponible en ligne ne cessent d'augmenter et ça à provoquer une régression de traitement de ses données, donc le projet de migration de Webselect a été conçu pour donner à nos utilisateurs une meilleure qualité que ce soient coté affichage ou coté fluidité des réponses.

WebSelect sert a analysé des articles de sources web et les sélectionnés pour vérifier si l'article est utile ou pas aux clients de Madamonitor, les valideurs peuvent alors validés ses articles ou les rejetés, le travail des valideurs est une étape nécessaire de traitement pour assurer les filtres des articles essentiels ainsi tous ce qui sont validés sont traités par une autre application qui se nomme WebClip qui va découper l'article en prenons en compte que les textes et image nécessaire aux clients.

Le principal but est de traiter plus rapidement les articles essentiels pour chaque client, après analyse le risque qui n'est pas encore gérer c'est au niveau de la concurrence, les valideurs ne doivent pas traiter ni validé le même article et au niveau des infrastructures, on ne connait pas encore le nombre maximum d'utilisateurs qui peuvent utiliser l'application en même temps pendant une durée finie.

# État de l’art sur le sujet traité

La validation des articles est un filtre des articles nécessaires a traité pour minimaliser les découpes des articles a effectué et pour bien :

-Ameliorer le filtrage des sources de données

-développer en interne,

-nécessaire pour améliorer la productivité des articles,

Le rôle de l'outils de validation est de :

-voir les listes des articles

-analyser les articles et vérifier si ça correspond aux attentes des clients

-valider ce qui sont nécessaires et rejetés ce qui sont inutiles

Cet outil multifonction va simplifier la vie de vos équipes et améliorer la qualité du service client, il peut être traiter en réseau local interne pas besoin d'internet

Il assure une traçabilité fiable des approbations et permet aux informations de circuler de manière sécurisée.

La digitalisation de ce traitement est un moyen aboutissant à un résultat plus performant aux attentes des clients.

L’état de l’art consiste à faire une étude comparée de différentes solutions concernant un domaine d’intérêt de votre stage. Par exemple vous souhaitez créer une nouvelle monnaie électronique. Une étude des monnaies électroniques existantes s’impose. Vous souhaitez implémenter une nouvelle technologie blockchain, une étude des technologies existantes s’impose.

Pour effectuer de façon efficace l’étude, vous devez dans un premier temps élaborer des critères de comparaison.

Ensuite vous étudiez de façon exhaustive chaque solution au regard de ces critères. Enfin vous produisez un tableau comparé des solutions y compris la vôtre.

Cela permet de se positionner, connaitre la concurrence et voir si on n’a pas oublié des services.

NE PAS CONFONDRE ETAT DE L’ART ET ETUDE DE L’EXISTANT.

## Critères de comparaison

Qualité vis à vis des autres validations :

Plus rapides au niveau des traitements,

Plus sophistiqués,

S’adapte en notre ère de ce temps,

Disponible partout et envoyer directement vers notre client cible après traitement

Sans digitalisation :

Délai d'attente élevé,

Inaccessible partout (limite de zone),

Nombre d'article limité

Information limitée

## Étude de chaque solution au vu des critères

Une étude approfondie concernant l'analyse de la base de données est nécessaire pour aboutir à cette satisfaction, pour la société une relation entre les entreprises médiatiques est nécessaire pour aboutir à ces informations et aussi un recrutement des emplois qui devrait traités ces articles

Concernant côté technique, une bonne conception de la base de données a été fait pour satisfaire les clients au niveau de la qualité des services

## Tableau comparatif des solutions au vu des critères

Tableau 1 : Exemple tableau

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Position Jan ‘14 | Position Jan ‘13 | Language | Share Jan ‘14 | Trend |
| 1 | 1 | Java | 26.2% | -0.6% |
| 2 | 2 | PhP | 13.2% | -1.6% |
| 3 | 6 | Python | 10.2% | +1.3% |
| 4 | 3 | C# | 9.6% | -0.4% |
| 5 | 4 | C++ | 8.9% | 0% |
| 6 | 5 | C | 8.1% | -0.2% |
| 7 | 7 | JavaScript | 7.6% | +0.3% |

# Étude de l’existant et solution envisagée

## Étude de l’existant

### Description externe du système logiciel existant (vision utilisateur)

Faire une description haut niveau des exigences déjà réalisées dans l’existant

L’interface pour l'affichage des listes des clients et ses commandes :

Des champs de saisie pour effectuer une recherche multicritère et un filtre des données des clients

Un bouton de pagination en bas de la liste pour passer les résultats suivants

Une icône de recherche pour valider les filtres de la recherche

Une page de liste des articles à valider avec l'extrait de texte, sa source, son auteur, le lien source de la page de l'article, sa date de publication et date de sélection.

Quand on sélectionne une de ses listes le client va voir quelque extrait qui contient le mot clés de l’article.

Un check box Ok et Not Ok pour valider ou pas les articles sélectionnés.

Un champ de saisie commentaire pour les remarques de l’utilisateur.

Un dialogue Box qui va afficher la liste des autres commandes qui possèdent le même article et ainsi que le texte complet de l’article

Il va aussi afficher La liste des descriptions des mots clés ainsi que les mots clés des autres profils sélectionnés

### Description interne du système logiciel existant (vision développeur/conception)

Faire une description haut niveau de la plate-forme technique existante et du logiciel déjà développé (les packages de code source ainsi que les tables existantes, les composants et leur déploiement)

Coté Migration des données :

Utilisation de l'Entity Framework pour accéder à la base de données SQL Server

Utilisation d'une fonction fait dans une console C# pour extraire les données nécessaires pour le projet WebSelect en plusieurs fichier Json qui va contenir 5000 données par fichier

Utilisation des extensions gratuits « MongoDB.Driver » et « MongoDB.Bson » pour traiter les données Json .

Utilisation d'une fonction qui utilise l'extension Mongo DB pour importer les données Json vers la Base de Données Mongo DB

Cote Base de Données Mongo DB :

Cote Front-End :

Utilisation 3 composants pour l’interface :

Article-to-validate :

-une boucle d’un cdk-accordion pour l’affichage des articles a validé

Une mat-pagination pour l’affichage par pages des données

-une icône qui permet d’afficher une page dans un dialogue des autres profils

Other-profil-article :

-un Mat-tab : pour afficher les deux onglets profil et l’onglet texte

- un Mat-table pour l’affichage des autres articles et ses descriptions pour la première page et ‘affichage du texte de l’article pour la deuxième page

-un div va afficher les textes complets de l’article

-une boucle d’un cdk-accordion pour l’affichage des mots clés

Utilisation de plusieurs modèles de classe pour l’affichage des données

Utilisation de plusieurs services pour l’injection des données

Utilisation d’Angular Material pour l’affichage des tableaux visuel, les Cards et les paginations, quelques styles CSS pour améliorer le design de la page

Cote Back-End: Node Js Express:

-utilisation de l’extension Mongoose pour la connexion du serveur d’application vers la base de données MongoDB et pour manipuler les données dans la base.

-Plusieurs classe sont nécessaire pour l’affichage des éléments du projet,

Dans le model ajout de plusieurs classes dont : ArticleKeyword, ArticleSelected, KeyWordArticle, KeywordDescription, KeywordGeneral, MapFlow, MapLocation, OrderSetting, VArticleToValidate, VArticleToValidateSource, VOrderToTreat, VWebOrders tous ces modèles de classe est essentiel pour la construction de l’objet à envoyer en tant qu’API.

Un dossier routes contenant plusieurs fichiers ayant plusieurs fonctionnalités qui va utiliser ces différents modèles citez ci-dessous pour le CRUD des données utiles dans la base MongoDB.

Un fichier Index.js pour appeler les routes en affichant les données en formats Json, il sert aussi à démarrer le serveur de l’API.

## Critique de l’existant

La critique doit être positive et négative

Elle peut se faire dans le chapitre "Étude de l'existant"

Une étude de la technologie Angular est nécessaire pour l'aboutissement de l'application

Ma compétence concernant l'utilisation de la technologie MEAN n'est pas encore parfait

L’application doivent bien convenir au besoin de l'utilisateur

Les fonctionnalités de l'application ne sont pas encore testées par les utilisateurs

Un test de l'application avec plusieurs utilisateurs ne sont pas encore faites

Les utilisateurs ne sont pas encore formés concernant l'utilisation de l'application

L’application est encore en phase de test et plusieurs bugs peuvent être produit

Les avis et remarques des utilisateurs ne sont pas encore demandés

## Solutions envisagées

Décrire les solutions envisagées et justifier la solution choisie

Une formation dans mon étude de MBDS concernant Angular a été fait pour améliorer la performance de l'application et connaitre certains convention et optimisation ainsi que les normes qu'on doit suivre pour bien construire une bonne application MEAN

Une formation en Analyse de données est aussi faite pour améliorer ma compétence en traitement des données qui est la principale tâche nécessaire coté performance.

Une guide et manuel ainsi qu’une formation pour les utilisateurs doit être faits pour faire savoir à eux l'utilisation de l'application et un retour et remarques serait un plus pour améliorer les différents bugs qui peuvent être produit en cas d'utilisation

Une simulation de se connecter avec plusieurs utilisateurs doivent être faites pour bien confirmer si l'application peut supporter plusieurs utilisateurs en même temps

## Objectifs principaux et livrables

# Démarche projet

(Attention !!! la démarche et les outils associés doivent être mis en place dès le début du stage)

## Principes de la démarche projet

### Activités d’ingénierie logicielle

Indiquer de façon synthétique (1/2 page max) les activités d’ingénierie logicielle (exigences, conception, codage, différents niveaux de test …) applicables au projet.

Le cas échéant, préciser la méthode de référence (Unified Process, XP …). En particulier, préciser la stratégie de test.

Parmi les activités d’ingénierie, préciser celles que vous réalisez.

### Méthode de gestion de projet utilisée

Méthode « en cascade » ou « itérative ». Le cas échéant, indiquer la méthode de référence, (Scrum, UP, …), sans la décrire en détails. Précisez votre rôle vis-à-vis de la gestion de projet. Présenter la démarche de communication : réunions, compte rendus …

### Rôles et responsabilités

Décrire les parties prenantes (client, équipe marketing …) et l'équipe projet.

Préciser en particulier votre rôle et vos relations avec les autres acteurs.

Le principal responsable du projet :

Le chef de projet est le responsable pour l'analyse technique des projets et aussi pour la conception de la base de données

Les départements RND :

Responsable pour le développement des applications de Madamonitor

Rôle de notre équipe :

En tant que membre de l'équipe du département RND

Notre tâche générale pour le développement d'un projet est repartie par taches en utilisant un application web nommées Jira

Il sert à construire et gérer les tickets de chaque membre de l'équipe.

Dans l'interface de Jira présente cinq grilles qui présentent des éléments qu'on appelle ticket, ces tickets sont respectivement ajoutés de gauche à droite de la grille pour voir l'évolution du ticket

La première grille affiche seulement la description principale du ticket ainsi que le responsable de la tache il sera encore en attente.

Dans la deuxième grille c'est une représentation du ticket qu'il est déjà traité par le responsable en cours

Dans la troisième grille on considère que le développement du ticket est fini mais un test doit être effectuer et c'est donc la grille de la phase de test de l'application

Dans la quatrième grille affiche aussi un test mais ce qui différencie c'est que le test doit être fait par le chef de projet pour assurer une bonne qualité de nos produits faits.

Dans la dernière grille confirme que le ticket c'est bien terminer et a bien été conclue.

En ses cinq phases si le résultat n'est pas concluant ou le testeur dans la phase de test a trouvé des anomalies, il peut remettre les taches en cours pour que le principal responsable recorrige les problèmes.

Dans l'interface présente aussi un délai maximum qu'on doit respecter pour chaque tache, on l'appel le « DeadLine ».

La cinquième grille est aussi une confirmation que l'application est bel et bien testée et valider par tous ce qui le concerne, les utilisateurs interne de l'application, le chef de projet et les développeurs web ceci pour dire que pour aboutir à une application de qualité, le travail d'équipe et une bonne organisation de l'équipe sont essentiels.

### Outils

Décrire les outils choisis pour mettre en œuvre la démarche (IDE, outil de gestion de la configuration, modeleur UML …). Préciser votre rôle dans le choix des outils.

Base de Données Et Back-End

C# reste un outil compatible avec la manipulation de la base de données SQL Server car tous les deux sont des produits de même source développer par Microsoft c'est aussi le principal langage que la société nous impose et divers extensions C# sont déjà acheté par la société.

C# fait partie intégrante de la plateforme .NET et est utilisé pour développer une variété d'applications, notamment des applications de bureau, des applications Web, des applications mobiles et bien d'autres.

C# lui-même est un langage open source, et sa mise en œuvre principale, appelée ".NET Core", est également open source et gratuite à utiliser. Microsoft a fait de gros efforts pour ouvrir le développement de .NET et C# à la communauté et aux développeurs, ce qui a conduit à un certain nombre de changements positifs pour les développeurs, tels que l'expansion de l'écosystème open source, la prise en charge multiplateforme.

Sur le plan commercial pour faciliter le développement de l'application on utilise l'IDE Visual Studio.

Certaines éditions de Visual Studio sont gratuites, comme Visual Studio Community, qui convient aux développeurs individuels et aux petites équipes. D'autres éditions, comme Visual Studio Professional et Visual Studio Enterprise, ont des fonctionnalités plus avancées et sont destinées aux équipes de développement plus importantes. De notre part nous utilisons le pack Visual studio professionnel

Front-End

Angular a été utilisé coté client, c’est un Framework open-source développé par Google, destiné à la création d'applications web dynamiques et interactives.

Il est principalement utilisé pour la construction de single-page applications (SPA), où le contenu est mis à jour dynamiquement sur une seule page, sans avoir besoin de recharger toute la page.

Angular facilite la création de ces applications en fournissant une structure et des outils pour gérer les composants, la gestion des données et les interactions utilisateur.

Angular est écrit en TypeScript, un sur-ensemble de JavaScript qui ajoute des fonctionnalités de typage statique.

Cela permet de détecter plus facilement les erreurs et de rendre le développement plus sûr et maintenable.

Il est important de noter qu'Angular ne doit pas être confondu avec AngularJS, qui est une version antérieure du framework. AngularJS a été largement utilisé, mais Angular (également connu sous le nom d'Angular 2 et versions ultérieures) a introduit des changements majeurs et une nouvelle architecture.

Il existe plusieurs environnements de développement intégrés (IDE) populaires que vous pouvez utiliser pour développer des applications Angular mais on a utilisé Visual Studio Code car c'est un IDE léger et très populaire, largement utilisé pour le développement d'applications Angular. Il propose une grande variété d'extensions et de plugins pour améliorer la productivité lors du développement avec Angular et il est un outils open Source.

Back-End

Node Js a été utilisé coté Back-End, c’est un environnement d'exécution côté serveur basé sur le moteur JavaScript V8 de Google.

Contrairement à JavaScript traditionnellement utilisé pour le développement côté client dans les navigateurs web, Node.js permet d'exécuter du code JavaScript côté serveur. Cela ouvre la porte à la création d'applications réseau évolutives et performantes.

Node.js est un choix populaire pour la création de micro services en raison de sa capacité à gérer un grand nombre de connexions et de demandes simultanées.

### Gestion de la configuration

Git a été utilisé Pour les contrôles de version c’est un système de contrôle de version distribué (DVCS - Distributed Version Control System) conçu pour suivre les modifications apportées au code source d'un projet au fil du temps. Il permet aux développeurs de collaborer efficacement, de gérer les versions du code et de suivre l'évolution des fichiers, en enregistrant chaque modification sous forme de "commit".

Git est largement utilisé dans le développement logiciel pour des projets de toutes tailles, du développement personnel aux projets open source et aux entreprises.

Pour faciliter l’utilisation de git on a utilisé une application de bureau GitHub Desktop.

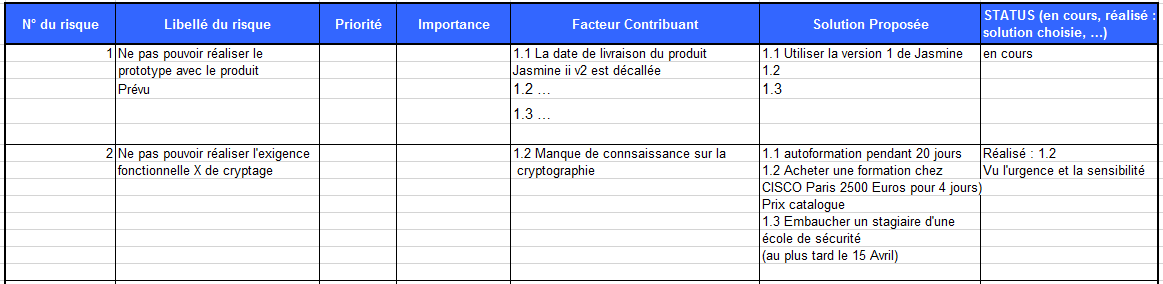
C’est une application de bureau conviviale qui facilite l'utilisation de la plateforme de gestion de code source GitHub.

GitHub est une plateforme de développement collaboratif qui permet aux développeurs de collaborer sur des projets, de gérer le contrôle de version de leur code source et de suivre les modifications apportées aux fichiers au fil du temps.

GitHub Desktop offre une interface utilisateur graphique (GUI) pour interagir avec les fonctionnalités principales de GitHub.

## Contraintes et risques sur le projet

Présenter les contraintes et les risques du projet.

Les risques peuvent être présentés dans un tableau. Par exemple :

Un plan de risque doit être exécutable. Les risques, les facteurs contributions et les solutions doivent être précis.

Être Malade n’est pas risque. C’est facteur contribuant.

Un risque c’est de ne pas pouvoir atteindre un de vos objectifs (une ou plusieurs de

vos exigences fonctionnelles ou non fonctionnelles, …).

Une solution proposée peut consister à faire quelque chose dans une des premières itérations pour lever le risque.

## Démarche projet mise en œuvre

Il s'agit de présenter ici le découpage du projet en étapes.

Dans le cas d'une méthode de gestion de projet itérative (Scrum, par exemple) :

* Définir un backlog avec des users stories et / ou les tâches. Chaque user story ou tâche doit être décrit de façon exhaustive (voir le chapitre 6)
* Présenter le découpage en sprints/itérations dans lesquels les Users stories et les tâches sont affectés. Chaque sprint doit être clairement défini
* Utiliser les outils de pilotages d’un projet scrum (vélocité, etc.)

Dans le cas d'une gestion de projet en cascade (i.e. non itérative), une étape est généralement la réalisation complète d'une activité d'ingénierie (par exemple "réaliser la spécification des exigences", "réaliser la conception"). Planification

Présenter un macro planning du projet. Ce planning doit être cohérent avec la démarche projet présentée au chapitre précédent. Mettre en avant les tâches sous votre responsabilité. Préciser également votre rôle dans l'élaboration de ce planning. S'il y a des écarts significatifs entre le planning initial et le planning réalisé, présenter les 2 plannings en expliquant les écarts.

Un diagramme de GANTT peut être utilisé. Mais dans ce cas, veillez à rester à un niveau macro, c’est-à-dire facilement lisible sur une page. Un diagramme de GANTT détaillé peut être présenté en annexe.

Vous pouvez vous appuyer sur d’autres méthodes de planification en fonction des outils de gestion de projet que vous allez utiliser.

## Budget du projet

Le coût de projet au moins par rapport à vous. Locaux, machines, salaires, …

Les outils nécessaires pour réaliser le projet sont :

Un ordinateur avec 16gb de ram et qui possèdent un processeur de marque Intel Core I5 8eme génération, il est utile pour bien se mettre à l'aise en développant l'application, pour éviter les plantages au niveau système et les lenteurs au niveau de la compilation de l'application

Un serveur de test pour la base de données pour bien simuler les performances de l'application

Un serveur de test de déploiement de l'application web utile aussi pour simuler un test de performance au côté utilisateur

Un ordinateur portable pour continuer le développement au cas où on a besoin de finir les taches urgentes

# Exigences réalisées dans le projet (vision externe/utilisateur)

Dans le cas d’un existant, décrire ici uniquement les nouvelles exigences réalisées au sein du projet.

S'inspirer de l’étude de cas du cours démarche de conception.

Notes : Les cas d’utilisations peuvent être remplacés par les User Stories si vous utilisez SCRUM par exemple.

## Exigences fonctionnelles – Cas d’utilisation

Donner ici le tableau de toutes vos cas d’utilisations.

Choisir ensuite 3 ou 4 cas d’utilisation représentatifs que vous allez décrire dans ce cette section. Mettre en annexe la description des autres cas d’utilisation. Expliquer la valeur de ces cas d'utilisation pour l'utilisateur.

### Cas d’utilisation

Un cas d’utilisation permet de décrire l’interaction entre les acteurs et le système. Il doit décrire exhaustivement les exigences fonctionnelles du système où chaque cas d’utilisation correspond à une fonction métier du système, selon le point de vue d’un de ses acteurs. Un cas d’utilisation se représente par un ovale dans lequel figure son intitulé.

#### Diagramme de séquence

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets. La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent à l'aide de messages.

#### Description textuelle

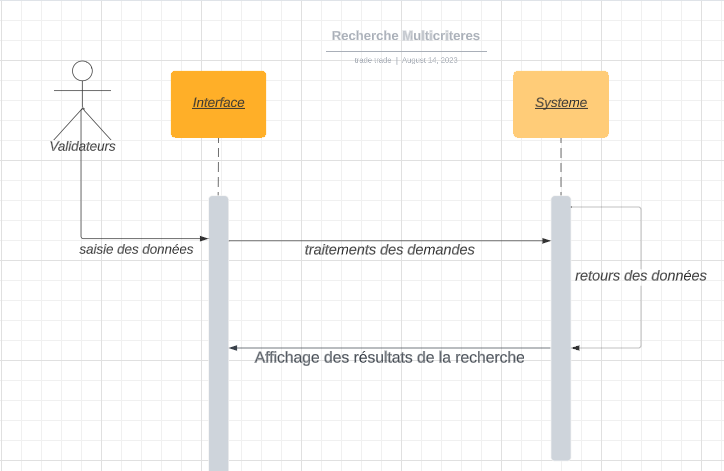
Dans cette partie, on va expliquer dans un tableau l’action en interaction pour chaque cas d’utilisation.

### Cas d’utilisation Recherche Multicritères

#### Description textuelle

|  |  |
| --- | --- |
| Titre | Recherche Multicritères |
| Acteur | Département validateurs |
| Résumé | Le validateur web veut filtrer la liste des commandes afficher en effectuant des recherches |
| Précondition | Le validateur web a navigué sur l’adresse du site et se trouve sur la page principale |
| Scenario nominale | L’utilisateur remplissent les champs de saisie des filtres qu’il veut rechercher :  -le champ de saisie Client ou son Id  - le champ de saisie Commande ou son Id  - le champ de saisie de la priorité  - le champ de saisie Origine  - le champ de saisie Localisation  L’utilisateur va ensuite cliquer sur l’icône de recherche |
| Scenario alternatif | Le clique du bouton recherche vient à afficher le résultat de la liste des commandes recherchés |

#### Diagramme des séquences systèmes



Description textuelle détaillée (voir un cours UML). Un cas doit contenir :

* Un nom
* Des préconditions
* Des postconditions
* Une description textuelle détaillée
* Des codes et messages d’erreurs.
* …

Un cas peut être accompagné de certains diagrammes UML :

* Des diagrammes de séquence (niveau système logiciel boîte noire, interactions entre le système et les acteurs extérieurs)
* Des diagrammes d’états
* Des diagrammes d’activités
* Etc.

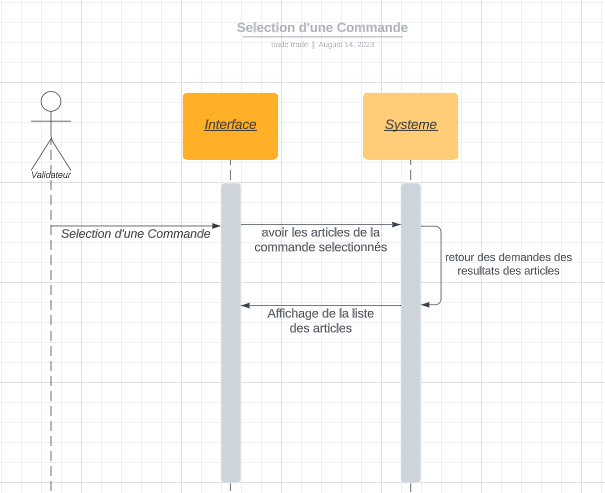
Un cas peut être accompagné d’écrans (voir la section 6.3)

### Cas d’utilisation sélection de commande

#### Description textuelle

|  |  |
| --- | --- |
| Titre | Sélection d’une commande |
| Acteur | Département validateurs |
| Résumé | Le validateur web veut sélectionner une commande |
| Précondition | Le validateur web a navigué sur l’adresse du site et se trouve sur la page principale |
| Scenario nominale | L’utilisateur clique sur l’une des listes de commande affiché |
| Scenario alternatif | La sélection de la commande vient à afficher la liste de ses articles |

#### Diagramme des séquences systèmes

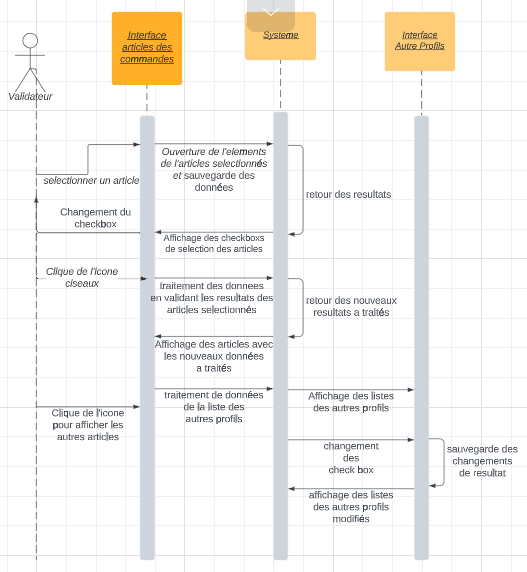


### Cas d’utilisation Validation des articles sélectionnés

#### Description textuelle

|  |  |
| --- | --- |
| Titre | Sélection d’une commande |
| Acteur | Département validateurs |
| Résumé | Le validateur web veut valider les articles sélectionnés |
| Précondition | Le validateur web doit sélectionnés une commande et voir la liste de ses articles |
| Scenario nominale | L’utilisateur sélectionne un article et un détail de l’article s’affiche avec un champs commentaire qu’ils peut saisir et deux check-box a coché ou pas (l’utilisateur peut sélectionner d’autres commande et faire le même scenario)  L’utilisateur peut cliquer sur l’icône qui permet de voir les autres profils et voit la liste des commande des autres profils qui possèdent l’article sélectionné, il peut cochez ou pas les deux checkbox de sélection des articles  L’utilisateur clique sur l’icône en forme de ciseaux  Un champ de dialogue s’affiche pour valider ou pas le traitement des articles modifiés |
| Scenario alternatif | Après acceptation du dialogue :  La validation de la sélection des Articles vient à afficher la liste des articles de la commande en excluant les articles traités  Sinon :  Annulation de traitement et affichage des articles par commandes sélectionné |

#### Diagramme des séquences systèmes



## Exigences non fonctionnelles transverses

Utilisabilité (charte graphique, ergonomie générale) Performances (temps de réponses, volumétrie) Robustesse (tolérances aux pannes, disponibilité) Sécurité

Supportabilité (maintenabilité, évolutivité)

Les exigences fonctionnelles doivent être décrites de façons concrètes et chiffrés. Par exemple

si le temps de réponse d’une page web.

## Interfaces détaillées

### IHM

CU d’utilisation X

écran 1

écran 2

Faire suivre chaque écran d’une description de ce qui s’y passe.

### Interfaces avec d’autres systèmes

Description d’API par exemple

# Architecture(s) système

## Back-End Migration des Données

Le back-end pour la migration des données du projet a été développé en C# console application et en utilisant Entity Framework et des nuggets Mongo DB il a été construit avec une architecture simple en un seul projet, où le code source est écrit dans un seul fichier ou dans plusieurs fichiers source au sein du projet.

## BackEnd traitement des données

Le back-end pour l'appel de la base de données est construit en Node.js Express avec une architecture en structure des dossiers :

Controller : Contient les contrôleurs qui gèrent la logique métier.

models : Contient les modèles de données ou les schémas de base de données.

routes : Contient les définitions des routes et les gestionnaires de requêtes.

Le dossier config peut contenir des configurations telles que les clés d'API, les configurations de base de données, etc.

node\_modules est le dossier où les dépendances de votre projet sont installées.

package.json est le fichier de configuration de votre projet Node.js, contenant les métadonnées du projet ainsi que les dépendances.

index.js (ou un autre nom) est le fichier principal de l'application où vous configurez et démarrez le serveur Express.

## Front-End

Le front-end coté utilisateur a été développer en Angular avec une architecture qui suit le modèle MVC (Modèle-Vue-Contrôleur) mais qui est souvent appelé MVVM (Modèle-Vue-VueModèle) dans le contexte d'Angular.

Voici une vue d'ensemble de cette architecture :

Modules : Les applications Angular sont organisées en modules.

Un module est un conteneur qui regroupe des fonctionnalités liées.

Chaque application Angular possède au moins un module racine, et d'autres modules peuvent être créés pour organiser des fonctionnalités spécifiques.

Composants : Les composants sont les blocs de construction fondamentaux d'une application Angular. Chaque composant combine une vue (template) avec la logique (contrôleur). Les composants peuvent être imbriqués pour créer des hiérarchies complexes.

Templates : Les templates sont écrits en HTML et définissent la structure de la vue. Ils contiennent des directives Angular pour ajouter de la logique et interagir avec les données.

Services : Les services sont des classes réutilisables qui fournissent des fonctionnalités spécifiques à l'application. Ils sont responsables de la logique métier, de la récupération des données depuis des sources externes (API, bases de données, etc.) et de la gestion des états.

Modules de routage : L'architecture par défaut d'Angular inclut également un module de routage pour gérer la navigation à travers les différentes vues de l'application. Le module de routage permet de définir des routes et d'afficher les composants correspondants en fonction de l'URL.

Observables et RxJS : Angular utilise RxJS (Reactive Extensions for JavaScript) pour gérer les opérations asynchrones, les flux de données et les événements. Les observables permettent de gérer les flux de données et de réagir aux changements de manière réactive.

Injection de dépendances : Angular dispose d'un système d'injection de dépendances intégré qui facilite la gestion des dépendances entre les différents composants, services et autres éléments de l'application.

Compilation et construction : Angular propose une chaîne d'outils de compilation et de construction pour transformer le code TypeScript, les templates et les styles en fichiers JavaScript, HTML et CSS optimisés pour le navigateur.

Dans ce chapitre vous devez décrire les architectures cibles de vos projets :

* L´architecture logicielle est une vue tournée sur l'organisation interne et le découpage en couches et modules du ou des logiciels du système informatique. Les responsabilités de chaque module et la nature et la structure des relations entre modules.
* L´architecture technique est une vue tournée vers les différents éléments matériels et l'infrastructure dans laquelle le système informatique s'inscrit, les liaisons physiques et logiques entre ces éléments et les informations qui y circulent.

Il est important de décrire ces architectures et de positionner votre contribution.



Figure 1: Exemple figure

# Conception du système logiciel réalisée dans le projet (vision interne/développeur)

Dans le cas d’un existant, faire référence à l’existant. Bien faire ressortir ce qui change ou ne change pas par rapport à l’existant. Justifier les choix. Préciser votre rôle dans ces choix. Dans le cas d'un existant, faire ressortir ce qui change par rapport à l'existant (si aucun changement sur un aspect, faire référence au chapitre décrivant l'existant).

(voir l’étude de cas du cours démarche de conception).

## Plate-forme technique

Architecture matérielle et framework logiciel

## Conception du logiciel développé

### Conception du code source

Expliquer les choix structurants sur concernant la structuration en unités de code (par exemple en Java, la structuration en classes et packages) et en tables.

Expliquer dans quelle mesure cette structuration applique les bonnes pratiques de conception. Expliquer les règles de nommage.

### Le code source – vue statique

Architecture du code (structuration en packages)

Pour chaque package: structuration en classes/fichiers de code source – des exemples d'implémentation de code source peuvent être présentés (ou bien mis en annexe et référencés ici).

### Modélisation de données

Décrire ici le modèle conceptuel MERISE (dictionnaire de données et MCD) ou UML (modèles de classes).

Décrire ensuite le modèle logique de données.

### Réalisation des cas d’utilisation

Choisir un cas d'utilisation significatif (1 de ceux présentés dans le chapitre précédent) et décrire le scénario de cas d’utilisation dans une vision interne/développeur (diagramme de séquence système logiciel boîte blanche)

### Les composants et leur déploiement

Par exemple, en JEE, la structuration en fichiers .war, ejb-jar …

Préciser les règles de nommage.

# Tests du système logiciel

En cohérence avec la stratégie de test décrite au chapitre 3, décrire les tests réalisés et leurs résultats. Préciser votre rôle dans la définition et l'exécution des tests.

# Conclusion générale

La conclusion est l’occasion pour vous de faire un bilan sur votre projet. Ce bilan doit

comporter les parties suivantes :

* Le bilan des résultats obtenus pour l’entreprise
* Le bilan des problèmes rencontrés et des solutions apportés
* Les perspectives du projet
* Le bilan personnel

**Le bilan des résultats obtenus pour l’entreprise**

Précisions sur le travail réalisé (nombre de programmes réalisés, nombre de classes implantées, ..., nombre de lignes de code, ...)

Précisions sur la situation finale de l’application réalisée (Le statu des livrables : en production, ..., en test, finis à X %, etc.)

Les livrables peuvent être : des releases d’une application, des uses cases traités, des users stories traitées, des sprints traités, des modules traités, etc.

**Le bilan des problèmes rencontrés et des solutions apportés**

Il s’agit ici de rapporter les principaux problèmes rencontrés et la façon dont ils ont été résolus.

**Les perspectives du projet**

Quelle est la suite possible à votre projet :

* Amélioration
* Nouveaux services
* Etc.

**Le bilan personnel**

Indiquer ici ce que ce stage et/ou projet a pu vous apporter.

# Références et Bibliographie

1. Auteur1, auteur 2, …, Titre\_du\_livre, Edition, date\_parution, lien s’il y a

…

1. Auteur1, auteur 2, …, Titre\_article, Revue de parution ou site Web, date\_parution, lien s’il y a …

Les liens web seuls ne peuvent être une bibliographie. Vous devez mettre dans la bibliographie : des articles de revues, des livres, des documents en respectant les exemples ci-dessus. Si vous avez un doute, merci d’aller voir les références et bibliographies de n’importe quel article WIKIPEDIA.

# Annexes (à intégrer absolument si ce n’est pas déjà fait dans le rapport principal. Elles peuvent être dans des fichiers séparés)

Annexe 1 : Exigences détaillées (correspond à peu près au cahier des charges)

Annexe 2 : Modèle conceptuel de données (diagramme de classes)

Annexe 3 : Dossier technique (Code réalisé y compris sa documentation)

Annexe 4 : Présentation des outils de développement utilisés

…

NOTES IMPORTANTES (à supprimer sur le rapport final):

* Les 40 pages maximums du rapport n’intègrent pas les annexes
* Concernant le stage, il y’aura trois soutenances (une première intermédiaire en Mars Orienté Organisation en présence du professeur de gestion de projet, un deuxième intermédiaire fin mai/début juin et la finale Début Septembre). Le professeur de gestion de projet donnera une note à lui au vu de la 1ère soutenance. Vous devez donc donner un maximum d’informations dans le rapport et ses annexes.
* Les éléments d’organisation du projet (risques, planning, coûts, etc.) sont des objets vivants. Ils doivent évoluer avec le projet. Ils doivent être mis à jour régulièrement. Ils doivent être utilisables.

1. TIC : Technologies de l’Information et de la Communication [↑](#footnote-ref-1)