

Réaliser par

Mr Malek BOUZAYANI

Réalisé à



Mise en place d'une application Dashboard It

Résumé

Cette solution consiste à mettre en place une application de 'Dashboard It' pour faciliter la gestion de la différente fonctionnalité de la société Qweeby comme : consultation et l'analyse du file d'attente de factures, la file d'attente de facture en cours de processing, le nombre de thread, l'état de la base (up or down) et le nombre des fichiers en attente, les fichiers logs, les informations du système (CPU , RAM ...), le nombre des fichiers qui en cours d'envoyer et les différents services (AS2, ALAD1 HTTPS, BOQ ALAD1, FTP, QBASE).

Mots clés: PHP, ReactJS, Postman, API, Axios

Abstract

This solution consists in setting up a 'Dashboard It' application to facilitate the management of the different functionality of the Qweeby company such as : consultation, analysis of the invoice queue, invoice queue being processed, number of threads, the status of the database (up or down) and the number of pending files, log files, system information (CPU, RAM, etc.), the number of files being sent and services (AS2, ALAD1 HTTPS, BOQ ALAD1, FTP, QBASE).

Keywords: PHP, ReactJS, Postman, API, Axios

Table des matières

Résumé	III
Liste des figures.....	VII
Liste des tableaux.....	VIII
Introduction générale	1
Chapitre 1 : Contexte général et étude préalable	2
Introduction.....	3
1.1. Cadre de projet : (Société du stage)	3
1.2. Présentation de l'organisme d'accueil	3
1.2.1.La mission de Qweeby	4
1.2.2.L'approche de Qweeby	4
1.3. Etude de l'existant.....	4
1.3.1.Contexte du Projet.....	4
1.3.2.Objectif du projet	5
1.3.3.Etude et analyse de l'existant.....	5
1.3.4.Critique de l'existant.....	6
1.3.5.Solution proposée	7
1.4. Méthodologie de travail.....	8
1.4.1.Méthodologie Agile	8
1.4.2.Pourquoi Scrum.....	8
1.5.Langage de modélisation.....	10
Conclusion.....	11
Chapitre 2 : Spécification des besoins.....	12
Introduction.....	13
2.1.Etude de contexte.....	13
2.1.1.Identification des acteurs	13
2.1.2.Spécification besoins fonctionnels	13
2.1.3.Spécification des besoins non fonctionnels	14
2.2.Product backlog	14
2.3.Modélisation des besoins fonctionnels	16
2.3.1.Diagramme de cas d'utilisation globale	16
2.3.2.La description fonctionnelle détaillée	17
2.3.2.1- Description de l'item « s'authentifier »	17
2.3.2.2- Description de l'item « Consulter file d'attente de factures »	18
2.3.2.2. Description de l'item « Consulter file d'attente de factures en cours de processing »	19

2.3.2.3. Description de l'item « Consulter les nombres de thread ».....	20
2.3.2.3 Description de l'item « Consulter fichier log ».....	22
2.3.2.4. Description de l'item « Consulter Service ».....	23
2.3.2.7. Description de l'item « Consulter information du système ».....	24
2.3.2.8. Description de l'item « Consulter le nombre des fichiers ».....	25
Conclusion.....	26
Chapitre 3 : Etude conceptuelle	27
Introduction.....	28
3.1. Planification des sprints	28
3.2. Conception du sprint 1	28
3.2.1. Sprint Backlog	28
3.2.2. Diagramme de séquence «s'authentifier»	29
3.3. Conception du sprint 2	30
3.3.1- Sprint Backlog	30
3.3.2- Diagramme de séquence «consulter les courbes»	31
3.4. Conception du sprint 3	32
3.4.1- Sprint Backlog	32
3.4.2- Diagramme de séquence « consulter les fichiers logs et les services »	34
3.5. Conception du sprint 4	35
3.5.1- Sprint Backlog	35
3.5.2- Diagramme de séquence «consulter les informations système et nombre des fichiers»	36
Conclusion.....	37
Chapitre 4 : Réalisation.....	38
Introduction.....	39
4.1. Architecture du système (Architecture Rest Api)	39
4.2. Environnement de travail	40
4.3. Langage et Framework de développement	41
4.3.1- Coté backend	41
4.3.2- Coté frontend	41
4.4. Présentation des interfaces de l'application	43
4.4.1. Interface de l'authentification	43
4.4.2. Interface du Dashboard	44
4.4.3. Interface des fichiers Logs	45
4.4.4. Interface du service (AS2, FTP ...)	45
4.4.5. Interface des informations du système	46
4.4.6. Interface du file en attente d'envoi	46
4.4.7. Interface de l'état de la base	47

4.4.8.Interface de déconnection	47
4.4.9.API (php pure)	48
Conclusion.....	48
Conclusion générale et perspectives.....	49
Bibliographie	50

Liste des figures

Figure .1: Logo QWEEBY	3
Figure .2: Processus de SCRUM	9
Figure .3: Diagramme de cas d'utilisation global.....	16
Figure .4: Diagramme de cas d'utilisation « S'authentifier»	17
Figure .5: Diagramme de cas d'utilisation «consulter courbe du file d'attente»	18
Figure .6: Diagramme de cas d'utilisation «Consulter file d'attente de factures en cours de processing »	19
Figure .7: Diagramme de cas d'utilisation «Consulter les nombres de thread».....	21
Figure .8: Diagramme de cas d'utilisation «Consulter fichier log».....	22
Figure .9: Diagramme de cas d'utilisation «Consulter Service»	23
Figure .10: Diagramme de cas d'utilisation «Consulter information du système».....	24
Figure .11: Diagramme de cas d'utilisation «le nombre des fichiers»	26
Figure .12: Diagramme de séquence « S'authentifier »	29
Figure .13: Diagramme de séquence «consulter les courbes»	31
Figure .14: Diagramme de séquence «consulter les fichiers logs et les services».....	34
Figure .15: Diagramme de séquence «consulter les informations système et nombre des fichiers»	36
Figure .16: Architecture Rest API	39
Figure .17: Interface de l'authentification 1	43
Figure .18: Interface de l'authentification 2	43
Figure .19: Interface du dashboard 1	44
Figure .20: Interface du dashboard 2	44
Figure .21: Interface des fichiers Logs	45
Figure .22: Interface du service (AS2, FTP ...).....	45
Figure .23: Interface des informations du système	46
Figure .24: Interface de la file en attente d'envoi	46
Figure .25: Interface de l'état de la base	47
Figure .26: Interface de déconnection.....	47
Figure .27: API (php pure).....	48
Figure .28: réponse json.....	48

Liste des tableaux

Tableau .1: Product backlog	15
Tableau .2: Description textuelle de cas d'utilisation « S'authentifier»	17
Tableau .3: Description textuelle de cas d'utilisation «Consulter file d'attente de factures»....	18
Tableau .4: Description textuelle de cas d'utilisation «Consulter file d'attente de factures en cours de processing».....	20
Tableau .5: Description textuelle de cas d'utilisation «Consulter les nombres de thread».....	21
Tableau .6: Description textuelle de cas d'utilisation «Consulter fichier log»	23
Tableau .7: Description textuelle de cas d'utilisation «Consulter Service».....	24
Tableau .8: Description textuelle de cas d'utilisation «Consulter information du système»....	25
Tableau .9: Description textuelle de cas d'utilisation «Consulter le nombre des fichiers»	26
Tableau .10: Planification des sprints	28
Tableau .11: backlog du sprint 1	28
Tableau .12: backlog du sprint 2	30
Tableau .13: backlog du sprint 3	32
Tableau .14: backlog du sprint 4	35
Tableau .15: Environnement logiciel.....	40
Tableau .16: Langages et frameworks de développement coté backen	41
Tableau .17: Langages et frameworks de développement coté frontend.....	41

Introduction générale

Aujourd'hui, il est essentiel de maintenir et de garantir la sécurité du système d'information d'une entreprise par une analyse exhaustive des intrusions possibles, et la protection de son environnement informatique interne et externe contre toute éventuelle intrusion ou surveillance malveillante.

Dans ce contexte, la société française QWEEBY nous a fait confiance pour améliorer un tableau de bord montre tous les éléments qui vous permettent d'évaluer et de juger la situation de l'entreprise. Il fournit un aperçu général du système, dans la mesure où il résume toutes les informations en quelques indicateurs clés. Cela aide à prendre des décisions judicieuses.

Il permet essentiellement de gagner du temps, car il donne une vue d'ensemble sur toutes les données chiffrées qui ne seront plus éparpillées dans les différentes branches de l'entreprise, mais centralisées dans un seul support.

De plus, le tableau de bord permet de déterminer si les objectifs ont été réalisés. Sinon, la réponse sera rapide si le tableau est mis à jour de façon régulière.

Ce rapport présente l'ensemble des démarches entreprises pour développer la solution proposée.

Chapitre 1 : Contexte général et étude préalable

Introduction

L'étude préalable est une phase cruciale pour réussir n'importe quel projet, et pour notre projet qui est la réalisation d'une application 'Dashboard It', nous allons commencer par l'analyse et l'évaluation de l'existant afin de le critiquer pour pouvoir dégager les besoins qui seront la base des contraintes à respecter durant l'élaboration de notre projet tout en proposant la liste des solutions possibles.

Dans ce chapitre nous allons présenter en premier lieu, l'organisme d'accueil dans lequel nous avons effectué notre sujet de travail proposé, le contexte et l'étude de projet ainsi que la méthodologie et le langage adoptés pour la réalisation du projet.

1.1. Cadre de projet : (Société du stage)

Ce stage est effectué au sein de la société "**QUESTS**". Ce dernier est une société française Crée en 2009 spécialisée dans l'édition des factures électroniques, est un « pure-player », sur le marché du B to B et B to G Dans ce cadre, nous allons concevoir et développer une application destinée aux employées de la société.

1.2. Présentation de l'organisme d'accueil :

QWEEBY, société par actions simplifiée, est active depuis 13 ans. Implantée à SAINT-ETIENNE (42000) France, elle est spécialisée dans le secteur d'activité de la programmation informatique.

Son effectif est compris entre 15 et 25 salariés. Sur l'année 2021 elle réalise un chiffre d'affaires de 500 600,00 €. Le total du bilan est resté stable entre 2020 et 2021.



Figure .1: Logo QWEEBY

1.2.1. La mission de Qweeby

Il existe plusieurs services au sein de Qweeby. Chaque service contribue à la vie quotidienne de l'entreprise à différents niveaux :

- Opérations : Il contient deux éléments principaux. Développement des affaires, qui est responsable de l'acquisition de nouveaux clients et de nouveaux projets. Et Customer Succès, pour assurer la satisfaction des clients Qweeby.
- Support technique : Il contient trois éléments principaux. Architecture applicative, qui définit les axes techniques pour améliorer le produit en fonction des demandes des clients internes (marketing, succès client ou exploitation). Développement, Développer les produits selon les demandes de l'architecte applicatif. et la Production, pour assurer la continuité des services proposés par Qweeby pour les clients et les internes.
- ERP : Mise en place et gestion de l'Enterprise Resource Planning.
- Marketing : Assurer la promotion et la valorisation de l'entreprise et de ses produits auprès des clients et prospects.

1.2.2. L'approche de Qweeby

Cette société assure La traçabilité détaillée pour suivre la facture à toutes les étapes de sa transmission et accès facilité aux les documents dans les outils ou aux étapes du parcours client, des données embarquées sur les factures pour informer les destinataires en direct aux bons endroits et des services embarqués sur les factures pour vous faciliter la vie et celle des destinataires.

1.3. Etude de l'existant

1.3.1. Contexte du Projet

Le présent projet est intitulé « Backoffice gestion de production IT ». Il est élaboré dans le cadre de la préparation du projet de fin d'études présenté en vue de l'obtention du diplôme du Mastère Professionnel Co-construit en Ingénierie des Applications Web-Nuragiennes à la L'ISET de Kairouan pour l'année universitaire 2021/2022.

1.3.2. Objectif du projet

Ce projet consiste à la mise en place d'un Dashboard avec des KPIs (indicateurs) sur tous les services fournis par la production et la synthèse des services

Implémentation d'un tracker d'incidents avec le Dashboard. Les incidents en cours doivent pouvoir être visualisés facilement et rapidement par les équipes internes ainsi que par nos clients. Il convient aussi d'avoir la possibilité de définir la visibilité de chaque incident (interne / clients). Les incidents devront être récupérés via API sur notre solution de gestion JIRA ITSM.

Centralisation des logs des différents applicatifs : Une stack ELK (Elastic Search, Logstash et Kibana) devra être utilisée pour centraliser les logs de différents services et de différents formats (texte, base de données, flux temps réel etc..). Les logs devront être structurés pour pouvoir être utilisés via le Dashboard de Production.

1.3.3. Etude et analyse de l'existant

Dans cette section, nous commençons nos recherches sur l'existant. Cette recherche nous permet d'ajouter de la crédibilité et de la valeur à notre travail et d'identifier les besoins et les faiblesses de chaque solution existante.

Pour faire cette recherche, nous avons été aidés par le gérant de la société « QWEEBY » lui-même, nous nous sommes donc comportés comme un utilisateur pour une vraie critique de l'avancement d'une commande de produits dans l'entreprise.

Les mesures prises au fil des ans par Qweeby incluent plusieurs types de systèmes d'accès aux données des clients.

Qweeby dispose d'une politique de confidentialité et d'une charte des bonnes pratiques, qui rassemble une liste de pratiques courantes concernant les actifs de l'entreprise. Mais sans une norme bien structurée, il n'y a pas de description réaliste et détaillée de la mise en œuvre des objectifs et des garanties.

Qweeby fournit une authentification traditionnelle pour ses employés, qui utilisent des noms d'utilisateur et des mots de passe pour accéder aux applications internes.

Par conséquent, l'entreprise ne dispose pas d'une solution de connexion sécurisée de contrôle d'accès pour ses applications internes.

1.3.4. Critique de l'existant

Les entreprises d'aujourd'hui s'appuient de plus en plus sur les données, car elles réalisent enfin à quel point un pourcentage de leurs données sur les revenus peut s'améliorer et les aider à comprendre leur entreprise et les besoins de leurs clients.

C'est pour cette raison que de nombreuses entreprises investissent dans des outils d'analyse de données.

Qweeby est bien conscient de l'importance des données et de leur capacité à aider et à améliorer leurs opérations commerciales.

Dans ce cadre, nous mettrons en place un système de rapporter complet dans Qweeby.

Dans un environnement très concurrentiel, cela aidera Qweeby à répondre aux besoins de l'entreprise et à offrir une large gamme de fonctionnalités à ces clients existants et potentiels.

Les données et les journaux de Qweeby proviennent exclusivement de l'application Qweeby Base, les membres du personnel s'appuient sur cette application pour contrôler le flux de données et recevoir des alertes pour les anomalies en cours dans la base de données.

Chaque fois qu'un membre du personnel souhaite obtenir une information, il doit accéder à cette application via la plate-forme de l'entreprise, rechercher les données, puis trouver le problème survenu. Cela peut également être fait lorsque l'équipe d'exploitation lance un ticket via JIRA.

Il y a plusieurs raisons pour lesquelles la méthode actuelle n'est pas optimale :

- ✓ Cela prend du temps
- ✓ Les données peuvent être difficiles à trouver
- ✓ L'absence d'une plate-forme de collecte de données claire

Sur la base de ces raisons, nous avons décidé de mettre à niveau la méthode actuelle vers une nouvelle plate-forme où nous centraliserons les données et implémenterons de nombreuses autres fonctionnalités.

Les fichiers journaux sont très importants pour une entreprise comme Qweeby, traitant des milliers de demandes et de factures chaque mois.

Cependant, les fichiers journaux peuvent être très délicats s'ils ne sont pas organisés, la société s'appuie actuellement sur l'application Qweeby Base pour générer tous ses fichiers journaux

dans un seul gros fichier journal, ce qui rend presque impossible l'identification des différentes factures dans ce fichier.

C'est dans le but de faciliter le traitement des journaux que nous allons créer une nouvelle méthode capable de transformer les fichiers journaux existants en fichiers journaux séparés contenant les informations d'un seul Facture au lieu d'un groupe de factures.

1.3.5. Solution proposée

Après l'étude que nous avons effectuée et selon le besoin de la société, nous opterons pour un nouveau site avec de nouvelles fonctionnalités. Il n'est pas facile de se débarrasser complètement des menaces et de la vulnérabilité des systèmes d'authentification.

Malgré cela, les organisations tentent de réduire ces risques en mettant en place des mécanismes de sécurité afin de les éviter. Dans notre cas, nous avons choisi de mettre en œuvre l'authentification unique (SSO) comme solution d'authentification et d'autorisation des utilisateurs.

Et afin d'avoir une meilleure vision et d'accéder facilement à leurs données, Qweeby s'est appuyé sur nous pour mettre en place un outil de reporter qui se compose de 3 grandes parties :

- Centraliser et collecter les données de Qweeby
- Transformez d'énormes fichiers journaux afin de faciliter la détection de factures séparées
- Récupérer les tickets existants dans JIRA afin de les visualiser

Pour atteindre ces points, nous avons choisi d'utiliser des API, cela nous donnera accès à de multiples fonctionnalités au sein de la pile qui nous permettront d'atteindre nos objectifs.

Tout d'abord, nous allons créer un connecteur entre la base de données Qweeby et notre site, puis nous allons créer un script php qui nous permettra de défragmenter les fichiers journaux en petits fichiers journaux séparés correspondant chacun à une certaine facture. Nous créerons ensuite des alertes en temps réel qui permettront au personnel d'être notifié par email chaque fois qu'une erreur se produit dans le système, nous procéderons ensuite à l'obtention de tous nos tickets auprès de JIRA et les stockerons dans un index qui permettra nous ensuite de faire un tableau de bord à partir des données présentes.

1.4. Méthodologie de travail

Pour la performance de notre système de gestion de l'information, afin de répondre aux exigences des clients dans un délai de moindre coût, nous avons utilisé une approche agile(16) du projet.

1.4.1. Méthodologie Agile

Le processus de gestion de projet est guidé par une méthodologie et le choix de la bonne aura un effet sur le flux de travail et la communication au sein de l'équipe.

Il existe de nombreuses méthodologies de gestion de projet, chacune a ses avantages et ses inconvénients et peut être appliquée à différents projets. :

- ✓ Scrum (1)
- ✓ Extreme Programming
- ✓ Adaptive Software Development (ASD)
- ✓ Dynamic System Development Method (DSDM)

Petit à petit sont regroupé sous le terme d'Agile Alliance, des personnes promettant l'utilisation de ces méthodes de développement.

1.4.2. Pourquoi Scrum

Le système de gestion de projet que nous avons utilisé dans ce projet est la méthodologie Agile, et SCRUM est la méthode. Scrum est un cadre de gestion de projet qui repose sur un développement incrémental et itératif afin d'atteindre un objectif spécifique.

L'objectif principal de SCRUM est de s'assurer que l'équipe accomplit un ensemble de tâches dans une itération temporelle bien définie, chaque itération se compose de deux à quatre semaines appelées sprints, où le but de chaque sprint est de développer une fonctionnalité afin de sortir avec un produit livrable.

Une réunion régulière est organisée chaque jour où les membres de l'équipe tentent de répondre à ces questions :

- Qu'est-ce que vous avez fait hier ?
- Que vas-tu faire aujourd'hui ?
- Avez-vous été confronté à quelque chose bloquant votre progression ?

En effet, Scrum définit trois rôles qui sont :

- **Product Owner** : Le Product Owner sert de contact entre l'équipe de développement et ses clients. Il gère le backlog produit et garantit que chacun connaît les priorités.

Le Product Owner dans ce projet est : M. Eric WANSCOOR

- **Scrum Master** : Le Scrum Master est au service de l'Equipe, ses rôles sont de diriger et diriger le groupe, il travaille en tant que superviseur et consultant. Il fixe les tâches et assume la responsabilité des délais et des révisions.

Le scrum master de ce projet est : M. Aymen BEN ABDALLAH

- **Équipe de développement** : L'équipe de développement est un groupe qui travaille ensemble pour produire et tester le produit fini pour la version finale.

L'équipe de développement de ce projet est : Youssef AZIZI.

Elle est « auto organisée » et elle reste inchangée pendant toute la durée du sprint. Son rôle principal est :

- Transformer les besoins exprimés dans le Sprint Backlog (est défini ci-dessous) en fonctionnalités utilisables.
- Livrer régulièrement une version fonctionnelle du produit.

La figure ci-dessous présente le déroulement de la gestion de projet par Scrum :

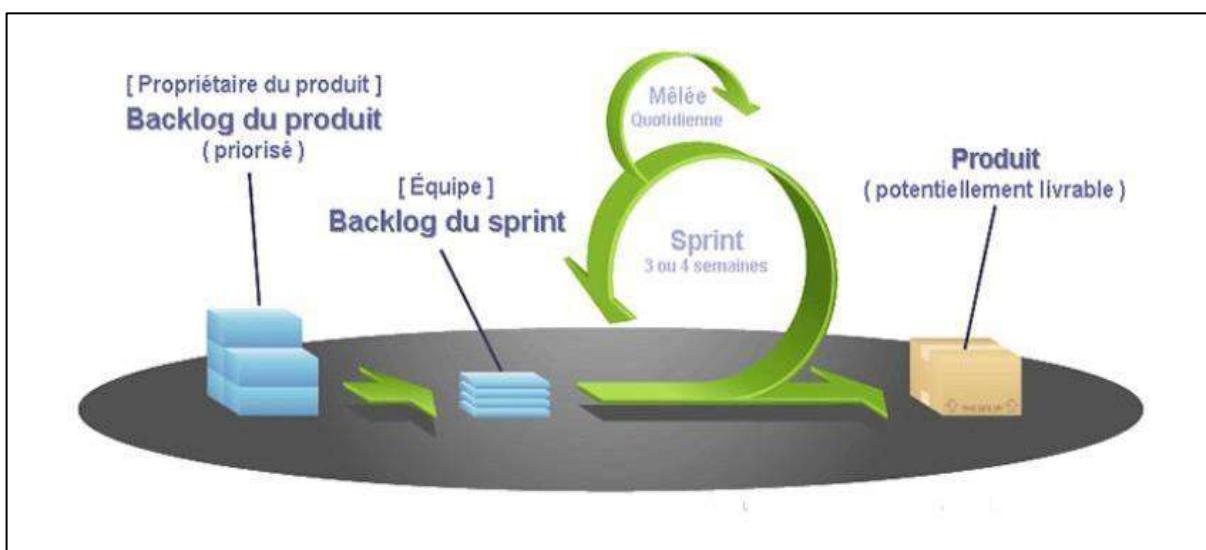


Figure .2: Processus de SCRUM

Backlog produit :

Le backlog de produit est la liste des fonctionnalités attendues d'un produit. Plus exactement, au-delà de cet aspect fonctionnel, il contient tous les éléments qui vont nécessiter du travail pour l'équipe. Les éléments sont classés par priorité, ce qui rend possible de définir l'ordre de mise en œuvre.

Sprint Backlog

Le Sprint Backlog c'est un plan fait par les gens qui effectuent le travail, c'est-à-dire. L'équipe de développement et surtout les développeurs.

Sprint

Un sprint désigne une brève période limitée dans le temps dont une équipe Scrum a besoin pour effectuer une quantité de travail donnée.

1.5. Langage de modélisation

La conception est la plus importante étape du cycle du développement logiciel. Elle se base essentiellement sur la bonne spécification et l'analyse des besoins. Notre démarche débute par la compréhension du problème. Ensuite nous analysons le problème pour donner une solution adéquate.

Dans notre travail nous allons utiliser L'UML (17) (Acronyme de Unified Modeling Language) comme un langage de conception et de modélisation graphique à base des diagrammes. Ce langage est destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. Il unifie à la fois les notations et les concepts orientés objets. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage.

UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis l'expression de besoin jusqu'au codage.

Un diagramme UML est une représentation graphique, qui permet de modéliser un aspect bien précis du système, il donne à l'utilisateur un moyen de visualiser et de manipuler des éléments de modélisation. Chaque type de diagramme UML possède une structure et des concepts prédéfinis

Conclusion

Dans ce premier chapitre, nous avons présenté l'entreprise d'accueil et détaillé le périmètre de notre étude afin de préciser les objectifs à atteindre, ce qui nous amène à expliquer d'avantage, dans le chapitre qui suit, les besoins fonctionnels et non fonctionnels de notre projet.

Chapitre 2 : Spécification des besoins

Introduction

C'est une phase essentielle dans le cycle de développement d'un projet. En effet, elle permet de mieux comprendre le contexte de système en dégageant tous les besoins fonctionnels et non fonctionnels des utilisateurs de notre projet ainsi que le product backlog suivi d'un diagramme de cas d'utilisation globale.

2.1. Etude de contexte

Le contexte d'un projet correspond à l'ensemble des informations qui caractérisent un projet et lui donnent de la profondeur en présentant ces différents acteurs, les besoins fonctionnels et non fonctionnels et les différentes tâches à effectuer.

2.1.1. Identification des acteurs

Les acteurs sont des entités extérieures au système qui interagissent avec le système. La notation UML pour un acteur est soit un dessin "simplifié" du personnage complété par le libellé ci-dessous, soit un dessin rectangulaire contenant le libellé du nom de l'acteur sous le libellé "Acteur Archétype". Notre notation graphique préférée est bien sûr le dessin du personnage, plus facile à repérer dans les schémas. Un acteur n'est pas toujours une personne, il peut s'agir d'un système externe interagissant avec le système décrit.

- Utilisateur

2.1.2. Spécification besoins fonctionnels

Le but d'un projet est de satisfaire un besoin. Il faut l'exprimer clairement avant d'imposer une solution.

Par conséquent, cette étape est nécessaire pour définir le projet et élaborer un plan pour bien piloter et réaliser ce que le client attend avant de commencer. Les besoins fonctionnels sont une expression de ce que le produit livré par le projet devrait être ou faire. Il s'agit donc de définir la fonctionnalité du système requise par le client sans aucune contrainte.

Nous commençons par identification des besoins fonctionnels :

➡ Employée :

- S'authentifié (avec votre compte Microsoft)
- Consulter courbe de la file d'attente de factures

- Consulter courbe file d'attente de facture en cours de processing
- Consulter courbe nombre de thread
- Consulter l'état de la base (up or down) et le nombre des fichiers en attente
- Consulter le fichier Log
- Consulter Service (AS2, ALAD1 HTTPS, BOQ ALAD1, FTP, QBASE)
- Consulter le nombre des fichiers qui en cours d'envoyée
- Consulter les informations du système (CPU, RAM ...)

2.1.3. Spécification des besoins non fonctionnels

Les besoins non fonctionnels définissent la qualité de l'application développée. Ils présentent une restriction ou une contrainte qui pèse sur un service, telle que les contraintes liées à l'environnement et à l'implémentation et les exigences en matière de performances, les dépendances de la plate-forme, la facilité la maintenance, l'extensibilité et de fiabilité

✓ Sécurité

Le système doit être sécurisé, en effet il doit garantir la fiabilité des données et leurs confidentialités, lors de la connexion, chaque utilisateur est identifié par un login et un mot de passe.

✓ Performance

L'application doit garantir un temps de réponse court.

✓ L'extensibilité

C'est la possibilité d'ajouter ou de modifier de nouvelles fonctionnalités dans de futures versions de l'application.

✓ L'ergonomie

La navigation doit être simple et claire pour bien guider l'utilisateur. L'application doit fournir une interface d'aide avec des informations riches et simples à utiliser.

2.2. Product backlog

Dans notre projet, nous allons créer un product backlog, qui vise à collecter tous les besoins clients auxquels l'équipe projet doit répondre. Par conséquent, il contient une liste de

caractéristiques impliquées dans la composition du produit. On a ci-dessous un tableau représentant le product backlog de notre projet.

ID	User Story	Description
1	S'authentifier	En tant qu'utilisateur l'application, je peux m'authentifier.
2	Consulter courbe de la file d'attente de factures	En tant qu'utilisateur de l'application, je peux voir l'analyse en temps réels des files en attente de factures.
3	Consulter courbe de la file d'attente de facture en cours de processing	En tant qu'utilisateur de l'application, je peux voir l'analyse en temps réels des files en attente de facture en cours de processing.
4	Consulter courbe nombre de thread	En tant qu'utilisateur de l'application, je peux voir l'analyse en temps réels le nombre de thread.
5	Consulter l'état de la base (up or down) et le nombre des fichiers en attente	En tant qu'utilisateur de l'application, je peux voir l'état de la base de données et le nombre des fichiers en attente.
6	Consulter le fichier Log	En tant qu'utilisateur de l'application, je peux consulter le fichier log.
7	Consulter Service (AS2, ALAD1 HTTPS, BOQ ALAD1, FTP, QBASE)	En tant qu'utilisateur de l'application, je peux consulter les services.
8	Consulter le nombre des fichiers qui en cours d'envoyée	En tant qu'utilisateur de l'application, je peux voir le nombre des fichiers qui est en cours d'envoie

9	Consulter les informations du système (CPU, RAM ...)	En tant qu'utilisateur de l'application, je peux voir les informations du système
---	--	---

Tableau .1: Product backlog

2.3. Modélisation des besoins fonctionnels

Cette section permet de modéliser les besoins fonctionnels et de les rendre compréhensible par ceux qui vont les couvrir en les représentants par les acteurs et leurs fonctionnalités à lesquelles doit répondre notre application.

2.3.1. Diagramme de cas d'utilisation globale

Chaque usage du système fait par les acteurs est représenté par un cas d'utilisation. Chaque cas d'utilisation représente une fonctionnalité qui leur est offerte afin de produire le résultat Attendu. Ainsi, le diagramme de cas d'utilisation (2) décrit l'interaction entre le système et l'acteur en déterminant les besoins de l'utilisateur et tout ce que doit faire le système pour l'acteur.

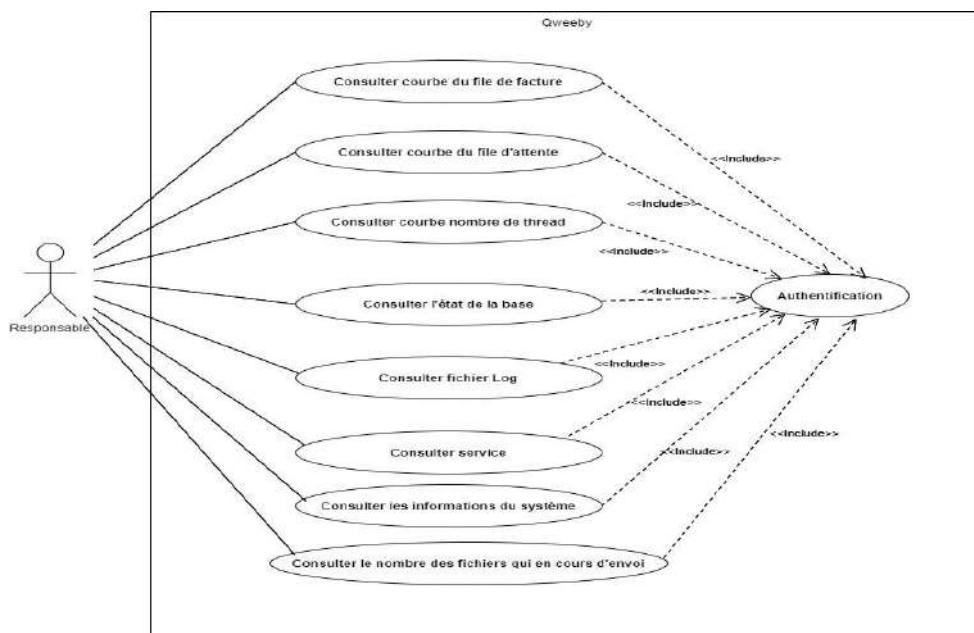


Figure .1: Diagramme de cas d'utilisation global

2.3.2. La description fonctionnelle détaillée

Dans cette partie on va décrire les fonctionnalités chargées afin de mieux comprendre ces missions.

2.3.2.1- Description de l'item « s'authentifier »

a) Diagramme de cas d'utilisation « S'authentifier »

Ce diagramme des cas d'utilisation de la figure 4 schématise l'authentification

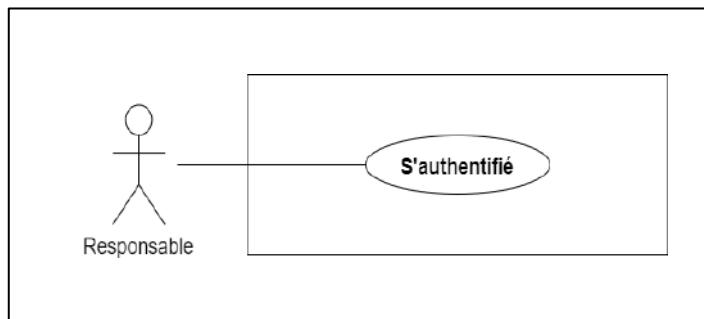


Figure .3: Diagramme de cas d'utilisation « S'authentifier »

b) Description textuelle de cas d'utilisation « s'authentifier »

Description d'Identification :

Acteur	Utilisateur
Objectif	Permet à l'utilisateur de s'authentifier
Préconditions	-Le système fonctionne -L'utilisateur saisit son login et son mot de passe
Post conditions	Ouverture de l'espace de l'acteur
Enchaînement nominal	1- Le système vérifie si les champs ne sont pas vides 2- Il vérifie ensuite si les informations sont valides 3- Le système redirige l'acteur vers son espace

Enchaînement alternatif	<ul style="list-style-type: none"> - En (1) : message d'erreur « un champ est vide » - En (2) : message d'erreur « login ou mot de passe incorrect »
--------------------------------	--

Tableau .2: Description textuelle de cas d'utilisation « S'authentifier»

2.3.2.2. Description de l'item « Consulter file d'attente de factures »

a) Diagramme de cas d'utilisation « Consulter file d'attente de factures »

La figure .5 illustre le diagramme de cas d'utilisation de consulter file d'attente de factures

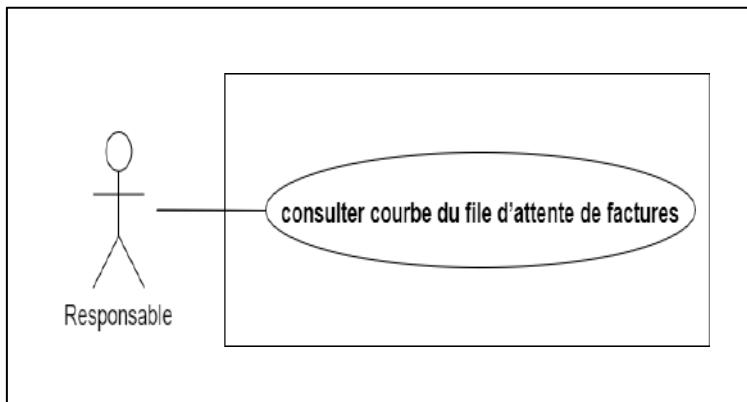


Figure .4: Diagramme de cas d'utilisation «consulter courbe du file d'attente»

b) Description textuelle de cas d'utilisation « Consulter file d'attente de factures »

Description de consulter file d'attente de factures :

Acteur	Utilisateur
Objectif	Permet à l'utilisateur de consulter l'analyse en temps réels des files d'attente de facture et de filtrer l'analyse entre deux dates précis
Préconditions	<ul style="list-style-type: none"> -Le système fonctionne -S'authentifier
Post conditions	Visualiser les analyses de file d'attente de factures

Enchaînement nominal	1- Le système affiche la courbe d'analyse des files d'attente de factures 2- L'acteur clique sur le bouton du date 1 3- Le système renvoie le calendrier 4- L'acteur choisit une date 5- Le système renvoie la courbe d'analyse 6- L'acteur clique sur le bouton du date 2 7- Le système renvoie le calendrier 8- L'acteur choisit une date 9- Le système renvoie la courbe d'analyse entre les deux dates
Enchaînement alternatif	- En (5) : « La courbe est vide » - En (9) : « La courbe est vide »

Tableau .3: Description textuelle de cas d'utilisation « Consulter file d'attente de factures »

2.3.2.3. Description de l'item « Consulter file d'attente de factures en cours de processing »

a) Diagramme de cas d'utilisation « Consulter file d'attente de factures en cours de processing »

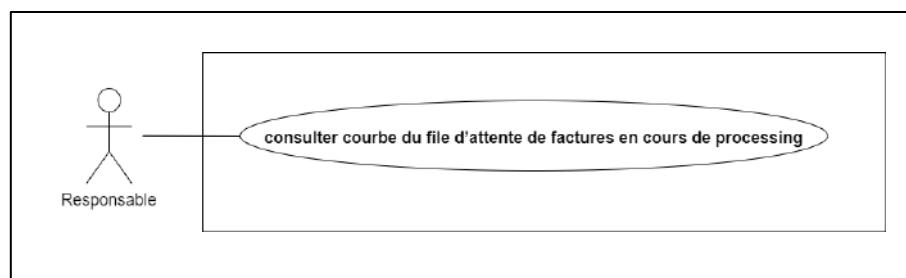


Figure .5: Diagramme de cas d'utilisation « Consulter file d'attente de factures en cours de processing »

b) Description textuelle de cas d'utilisation « Consulter file d'attente de factures en cours de processing »

Description de consulter file d'attente de factures en cours de processing :

Acteur	Utilisateur
Objectif	Permet à l'utilisateur de consulter l'analyse en temps réels des files d'attente de factures en cours de processing et de filtrer l'analyse entre deux dates précis
Préconditions	-Le système fonctionne -S'authentifier
Post conditions	Visualiser les analyses de file d'attente de factures en cours de processing
Enchaînement nominal	1- Le système affiche la courbe d'analyse des files d'attente de factures en cours de processing 2- L'acteur clique sur le bouton du date 1 3- Le système renvoie le calendrier 4- L'acteur choisit une date 5- Le système renvoie la courbe d'analyse 6- L'acteur clique sur le bouton du date 2 7- Le système renvoie le calendrier 8- L'acteur choisit une date 9- Le système renvoie la courbe d'analyse entre les deux dates
Enchaînement alternatif	- En (5) : « La courbe est vide » - En (9) : « La courbe est vide »

Tableau .4: Description textuelle de cas d'utilisation « Consulter file d'attente de factures en cours de processing »

2.3.2.4. *Description de l'item « Consulter les nombres de thread »*

a) *Diagramme de cas d'utilisation « Consulter les nombres de thread »*

La figure .7 illustre le diagramme de consulter les nombres de thread :

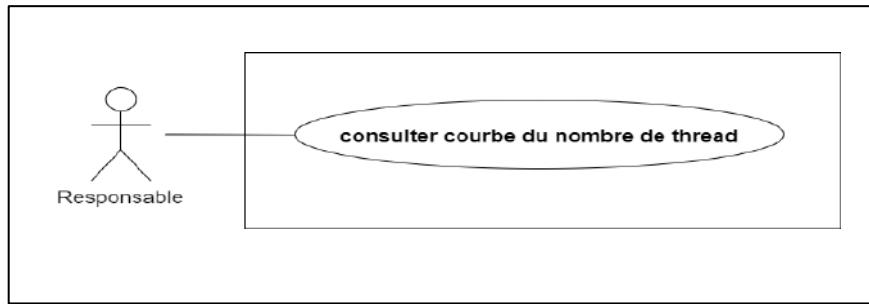


Figure .6: Diagramme de cas d'utilisation « Consulter le nombre de thread »

b) Description textuelle de cas d'utilisation « Consulter les nombres de thread »

Description de consulter les nombres de thread :

Acteur	Utilisateur
Objectif	Permet à l'utilisateur de consulter l'analyse en temps réels les nombres de thread et de filtrer l'analyse entre deux dates précis
Préconditions	-Le système fonctionne -S'authentifier
Post conditions	Visualiser les analyses des nombres de thread
Enchaînement nominal	1- Le système affiche la courbe d'analyse des nombres de thread 2- L'acteur clique sur le bouton du date 1 3- Le système renvoie le calendrier 4- L'acteur choisit une date 5- Le système renvoie la courbe d'analyse 6- L'acteur clique sur le bouton du date 2

	7- Le système renvoie le calendrier 8- L'acteur choisit une date 9- Le système renvoie la courbe d'analyse entre les deux dates
Enchaînement alternatif	- En (5) : « La courbe est vide » - En (9) : « La courbe est vide »

Tableau .5: Description textuelle de cas d'utilisation « Consulter les nombres de thread »

2.3.2.5. Description de l'item « Consulter fichier log »

a) Diagramme de cas d'utilisation « Consulter fichier log »

La figure .8 illustre le diagramme de cas d'utilisation de consulter fichier log :

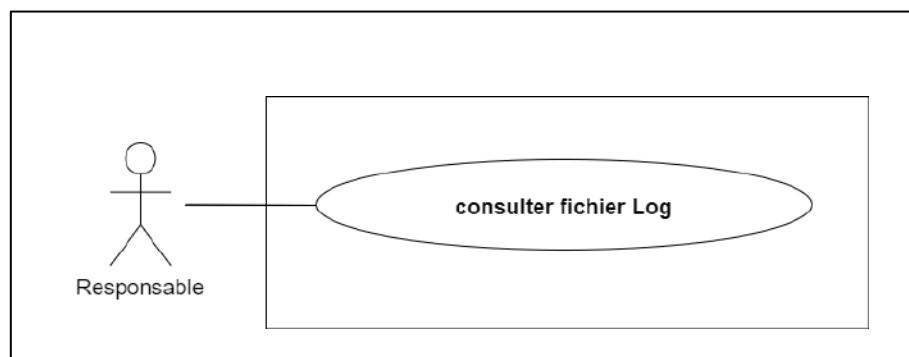


Figure .7: Diagramme de cas d'utilisation «Consulter fichier log»

b) Description textuelle de cas d'utilisation « Consulter fichier log »

Description de consulter fichier log :

Acteur	Utilisateur
Objectif	Permet à l'utilisateur de consulter le tableau des fichiers du log et d'effectuer des recherches par nom fichier et par code de l'entreprise.
Préconditions	-Le système fonctionne -S'authentifier
Post conditions	Visualiser le tableau des fichiers log.
Enchaînement nominal	1- Le système affiche le tableau des fichiers log. 2- L'acteur clique sur le champ de saisit du nom de fichier et le remplit 3- Le système renvoie la liste des fichiers log 4- L'acteur clique sur le champ de saisit du type de fichier et le remplit 5- Le système renvoie la liste des fichiers log
Enchaînement alternatif	- En (3) et (5) : « Le tableau est vide »

Tableau .6: Description textuelle de cas d'utilisation « Consulter fichier log »

2.3.2.6. Description de l'item « Consulter Service »

a) Diagramme de cas d'utilisation « Consulter Service »

La figure .9 illustre le diagramme de cas d'utilisation de consulter Service :

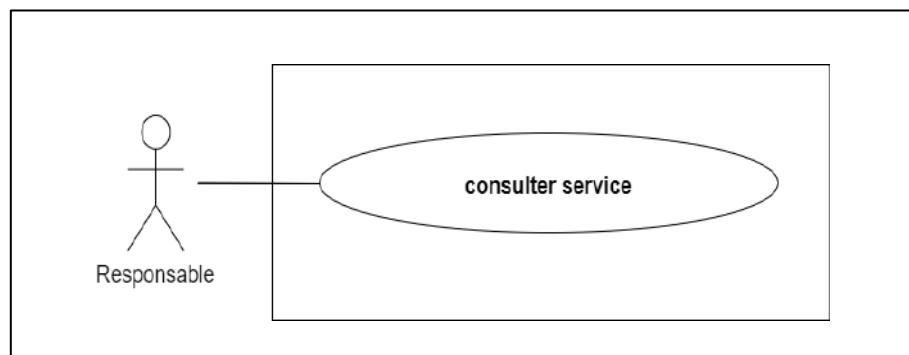


Figure .8: Diagramme de cas d'utilisation « Consulter Service »

b) Description textuelle de cas d'utilisation « Consulter Service »

Description de consulter Service :

Acteur	Utilisateur
Objectif	Permet à l'utilisateur de consulter l'état (Actif/Non actif) des service (AS2, ALAD1 HTTPS, BOQ ALAD1, FTP, QBASE).
Préconditions	-Le système fonctionne -S'authentifier
Post conditions	Visualiser l'état des services.
Enchaînement nominal	1- Le système affiche l'état de chaque service. 2- Le système affiche les états de chaque service (ALAD1 HTTPS, BOQ ALAD1, FTP)du derniers 90 jours.

Tableau .7: Description textuelle de cas d'utilisation « Consulter Service »

2. 3.2.7. Description de l'item « Consulter information du système »

a) Diagramme de cas d'utilisation « Consulter information du système »

La figure .10 illustre le diagramme de cas d'utilisation d'information du système :

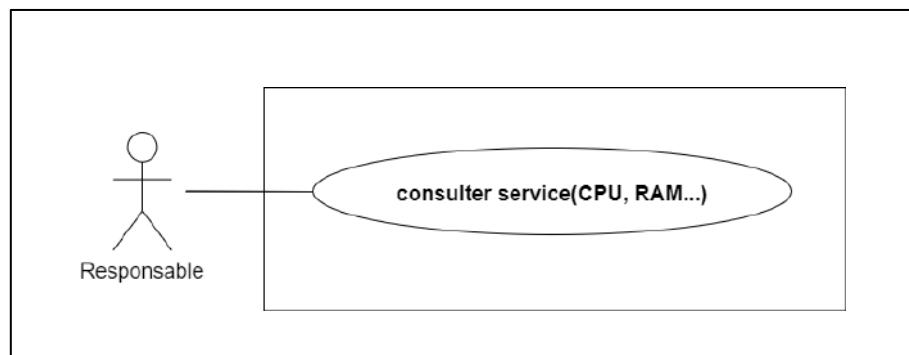


Figure .9: Diagramme de cas d'utilisation «Consulter information du système»

b) Description textuelle de cas d'utilisation « Consulter information du système »

Description de consulter information du système :

Acteur	Utilisateur
Objectif	Permet à l'utilisateur de consulter les informations (cpu, ram...) de votre système.
Préconditions	-Le système fonctionne -S'authentifier
Post conditions	Visualiser les informations du système.
Enchaînement nominal	1- Le système affiche les informations du système : ➡ Valeur du CPU ➡ Valeur du RAM ➡ Valeur du Stockage ➡ Statuts du batterie

Tableau .8: Description textuelle de cas d'utilisation « Consulter information du système »

2.3.2.8. Description de l'item « Consulter le nombre des fichiers »

a) Diagramme de cas d'utilisation « Consulter le nombre des fichiers »

La figure .11 illustre le diagramme de cas de consulter le nombre des fichiers :

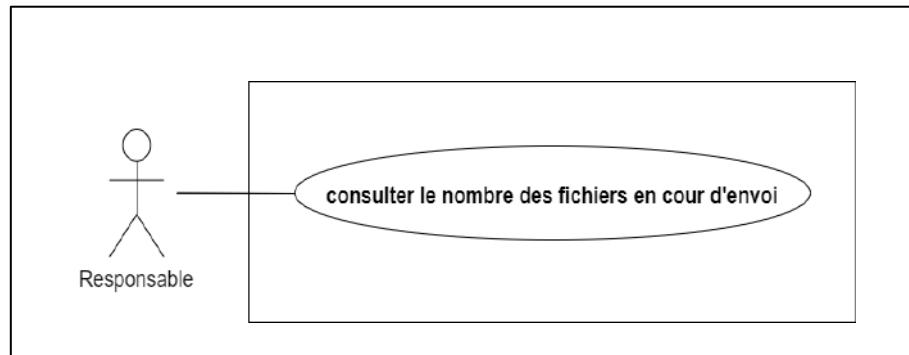


Figure .10: Diagramme de cas d'utilisation « le nombre des fichiers »

b) Description textuelle de cas d'utilisation « Consulter le nombre des fichiers »

Description de consulter le nombre des fichiers :

Acteur	Utilisateur
Objectif	Permet à l'utilisateur de consulter le nombre des fichiers qui est en cours d'envoi.
Préconditions	-Le système fonctionne -S'authentifier
Post conditions	Visualiser le nombre des fichiers en cours d'envoi par société.
Enchaînement nominal	1- Le système affiche le nombre des fichiers : ➡ Nombre par société ➡ Nombre totale

Tableau .9: Description textuelle de cas d'utilisation « Consulter le nombre des fichiers »

Conclusion

Ce chapitre nous a permis de bien définir le projet en présentant les différents acteurs ainsi que les besoins fonctionnels et non fonctionnels accompagnés du Product backlog et d'un diagramme de séquence global.

Chapitre 3 : Etude conceptuelle

Introduction

Après avoir planifié et découpé notre projet en ensemble de sprints tout en nous basant sur la méthode Scrum nous avons intéressé à la partie conceptuelle.

3.1. Planification des sprints

Le tableau .10 illustre la planification des sprints :

Titre	Description	Durée
Sprint 1	Développement de l'authentification	3 semaines
Sprint 2	-Développement de la partie backend d'analyse des courbes -Développement de la partie frontend d'analyse des courbes	3 semaines
Sprint 3	-Développement de la partie backend du fichier log et des services (AS2, ALAD1 HTTPS, BOQ ALAD1, FTP, QBASE) -Développement des interfaces du fichier log et les services	3 semaines
Sprint 4	-Développement de la partie backend de les informations du système (CPU, RAM ...) et le nombre des fichiers -Développement de la partie frontend de les informations du système et de la partie du nombre des fichiers	3 semaines

Tableau .10: Planification des sprints

3.2. Conception du sprint 1

3.2.1. Sprint Backlog

Sprint Backlog du sprint 1 :

ID	Item	User Story	Description
1	S'authentifier	S'authentifier	En tant qu'utilisateur l'application, je peux m'authentifier avec mon compte Microsoft fournit à partir de la société.

Tableau .11: backlog du sprint 1

3.2.2. Diagramme de séquence «s'authentifier»

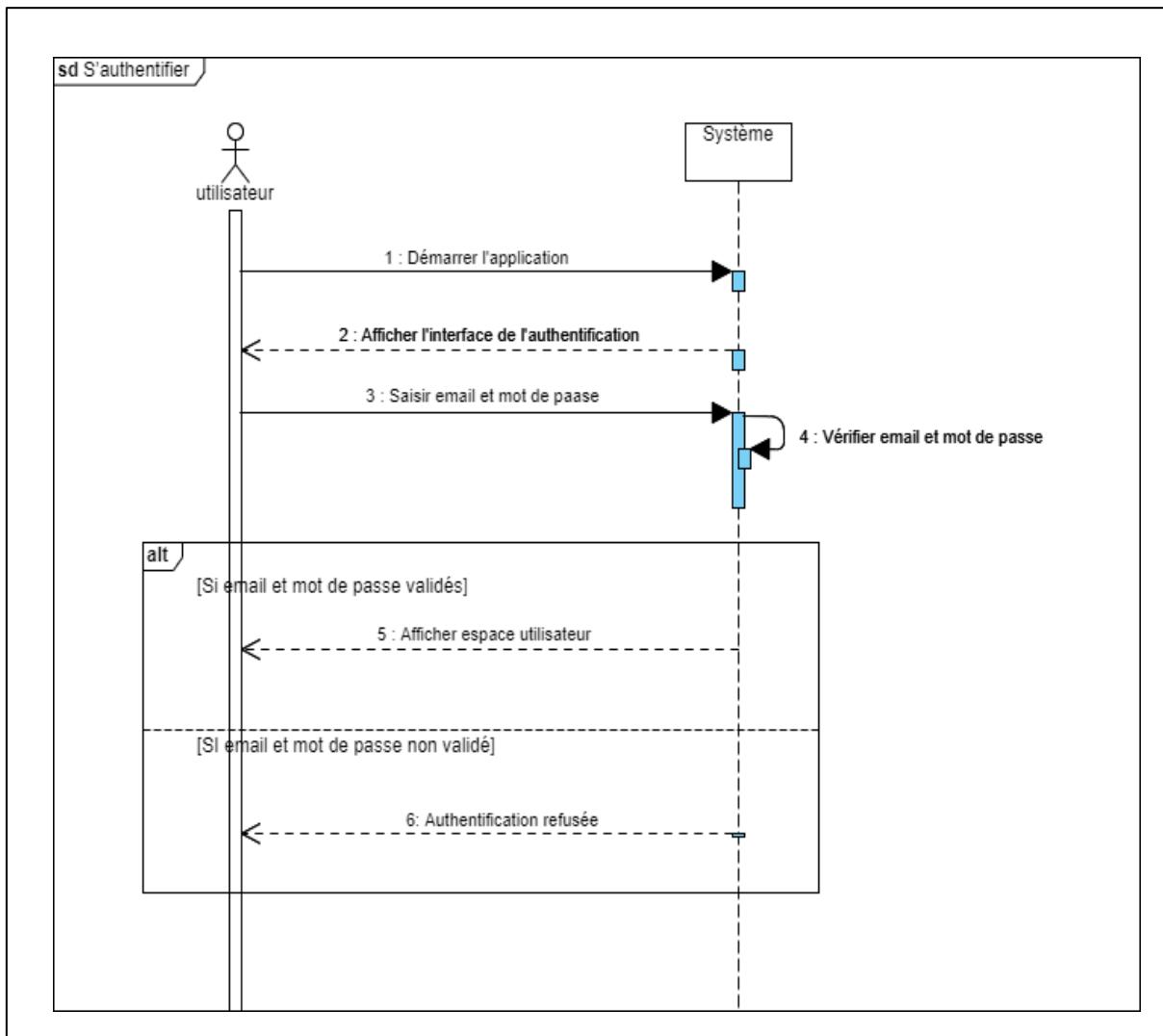


Figure .11: Diagramme de séquence « S'authentifier »

Quand l'utilisateur demande l'accès à l'application, il doit se connecter avec son identifiant et son mot de passe via l'interface d'authentification. Ainsi, les données seront envoyées au serveur azure (Microsoft) qui soutient la vérification et la consultation de la base de données.

Si elle est acceptée, elle donne accès au système et aux interfaces de menu correspondantes. Si non, le serveur d'application enverra un message d'erreur afin de corriger ses données.

3.3. Conception du sprint 2

3.3.1- Sprint Backlog

Sprint Backlog du sprint 2 :

ID	Item	User Story	Description
1	Consulter file d'attente de factures	Consulter l'analyse	En tant qu'utilisateur de l'application, je peux voir l'analyse en temps réels des files en attente de factures.
		Filtrer l'analyse	En tant qu'utilisateur de l'application, je peux filtrer l'analyse entre des date précis.
2	Consulter file d'attente de factures en cours de processing	Consulter l'analyse	En tant qu'utilisateur de l'application, je peux voir l'analyse en temps réels des files en attente de factures en cours de processing.
		Filtrer l'analyse	En tant qu'utilisateur de l'application, je peux filtrer l'analyse entre des date précis.
3	Consulter le nombre de thread	Consulter l'analyse	En tant qu'utilisateur de l'application, je peux voir l'analyse en temps réels les nombres de thread.
		Filtrer l'analyse	En tant qu'utilisateur de l'application, je peux filtrer l'analyse entre des date précis.

Tableau .12: backlog du sprint 2

3.3.2- Diagramme de séquence «consulter les courbes»

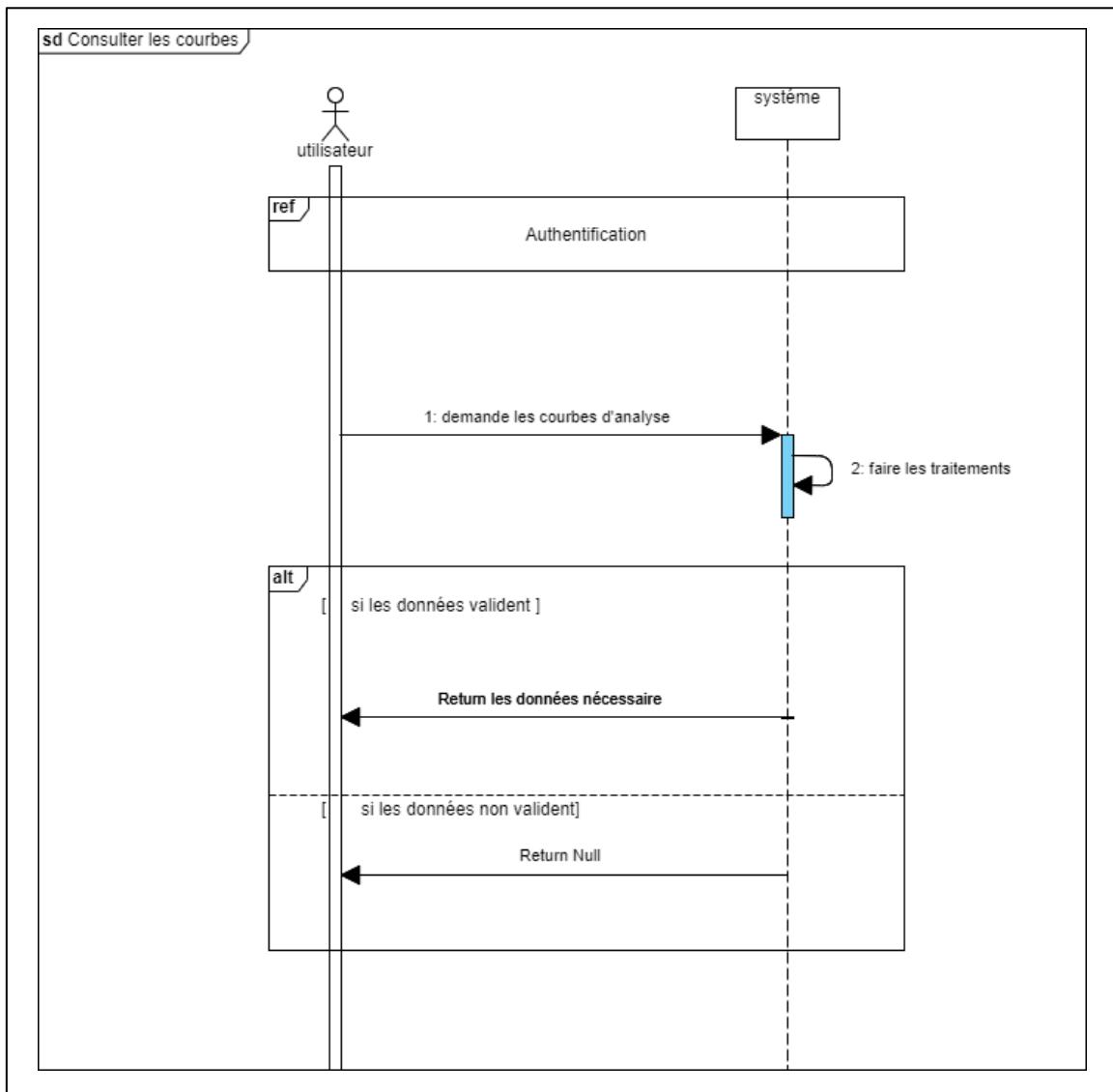


Figure .12: Diagramme de séquence «consulter les courbes»

Apres l'authentification, l'utilisateur a le droit de consulter les analyses du file d'attente de factures, file d'attente de factures en cours de processing et le nombre de thread

3.4. Conception du sprint 3

3.4.1- Sprint Backlog

Sprint Backlog du sprint 3 :

ID	Item	User Story	Description
1	Consulter les fichiers Log	Consulter le tableau des fichiers log	En tant qu'utilisateur de l'application, je peux voir le tableau des fichiers log.
		Filtrer le tableau	En tant qu'utilisateur de l'application, je peux faire des recherches avec le nom du fichier ou avec le type.
2	Consulter les Services (AS2, ALAD1 HTTPS, BOQ ALAD1, FTP, QBASE)	Consulter le service AS2	En tant qu'utilisateur de l'application, je peux voir l'état de la service AS2.
		Consulter le service ALAD1 HTTPS	En tant qu'utilisateur de l'application, je peux voir l'état de la service ALAD1 HTTPS.
		Consulter le service BOQ ALAD1	En tant qu'utilisateur de l'application, je peux voir l'état de la service BOQ ALAD1.
		Consulter le service FTP	En tant qu'utilisateur de l'application, je peux voir l'état de la service FTP.

	Consulter le service QBASE	En tant qu'utilisateur de l'application, je peux voir l'état de la service QBASE.

Tableau .13: backlog du sprint 3

3.4.2- Diagramme de séquence « consulter les fichiers logs et les services »

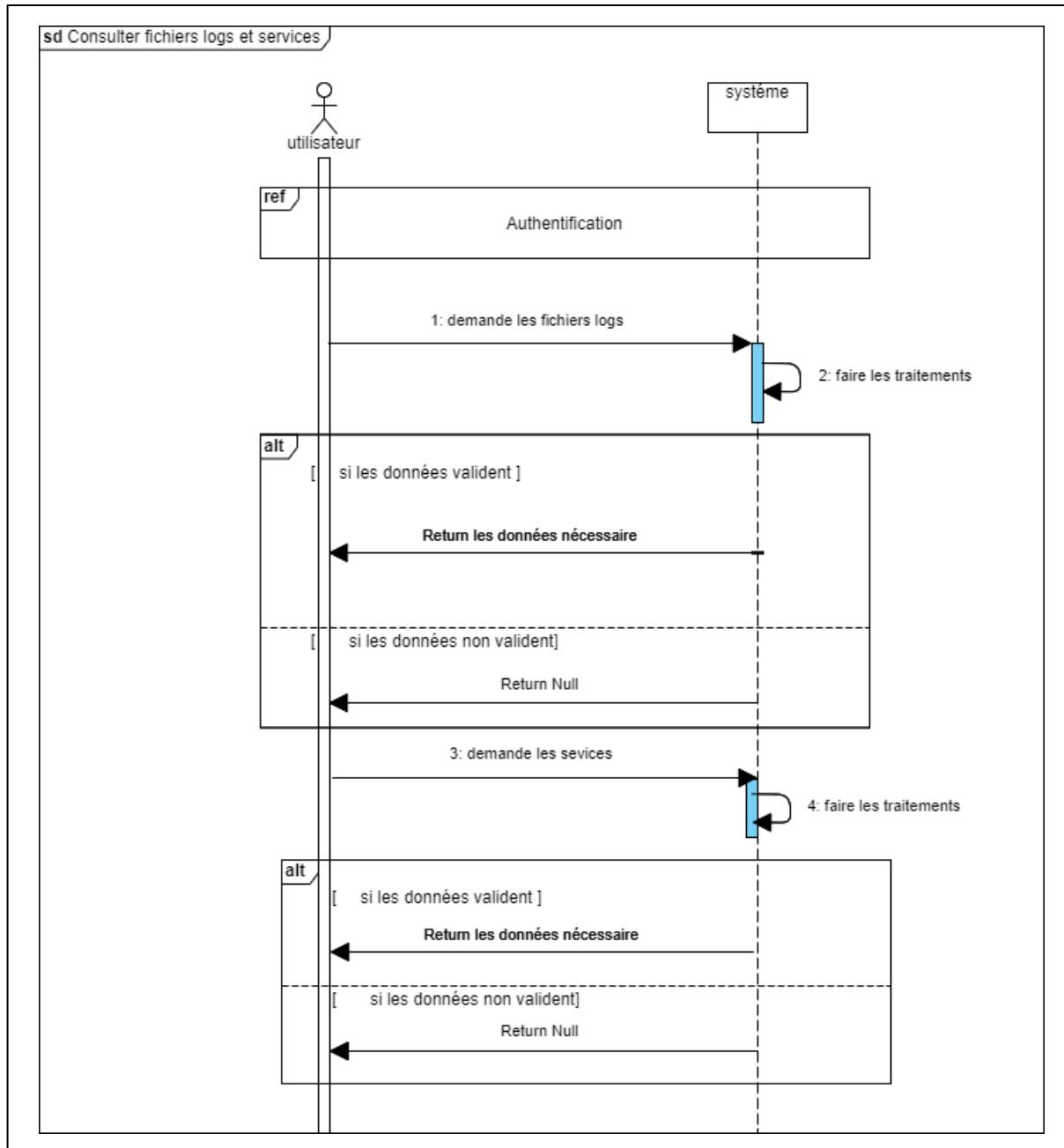


Figure .13: Diagramme de séquence «consulter les fichiers logs et les services»

Après l'authentification, l'utilisateur a le droit de consulter les fichiers logs et d'avoir les Services (AS2, ALAD1 HTTPS, BOQ ALAD1, FTP, QBASE).

3.5. Conception du sprint 4

3.5.1- Sprint Backlog

Sprint Backlog du sprint 4 :

ID	Item	User Story	Description
1	Consulter les informations du système (CPU, RAM ...)	Consulter CPU	En tant qu'utilisateur de l'application, je peux voir La valeur du CPU.
		Consulter RAM totale et RAM libre	En tant qu'utilisateur de l'application, je peux voir La valeur du RAM.
		Consulter Disc totale et Disc libre	En tant qu'utilisateur de l'application, je peux voir La valeur du Disc. En tant qu'utilisateur de l'application, je peux voir La valeur du Disc.
		Consulter Battery statuts et charge	En tant qu'utilisateur de l'application, je peux voir La valeur du Battery.
2	Consulter le nombre des fichiers qui en cours d'envoyée	Consulter le nombre des fichiers par société	En tant qu'utilisateur de l'application, je peux voir le nombre des fichiers par société qui en cours d'envoyée.
		Consulter le nombre des fichiers totale	En tant qu'utilisateur de l'application, je peux voir le nombre total des fichiers qui en cours d'envoyée.

Tableau .14: backlog du sprint 4

3.5.2- Diagramme de séquence «consulter les informations système et nombre des fichiers»

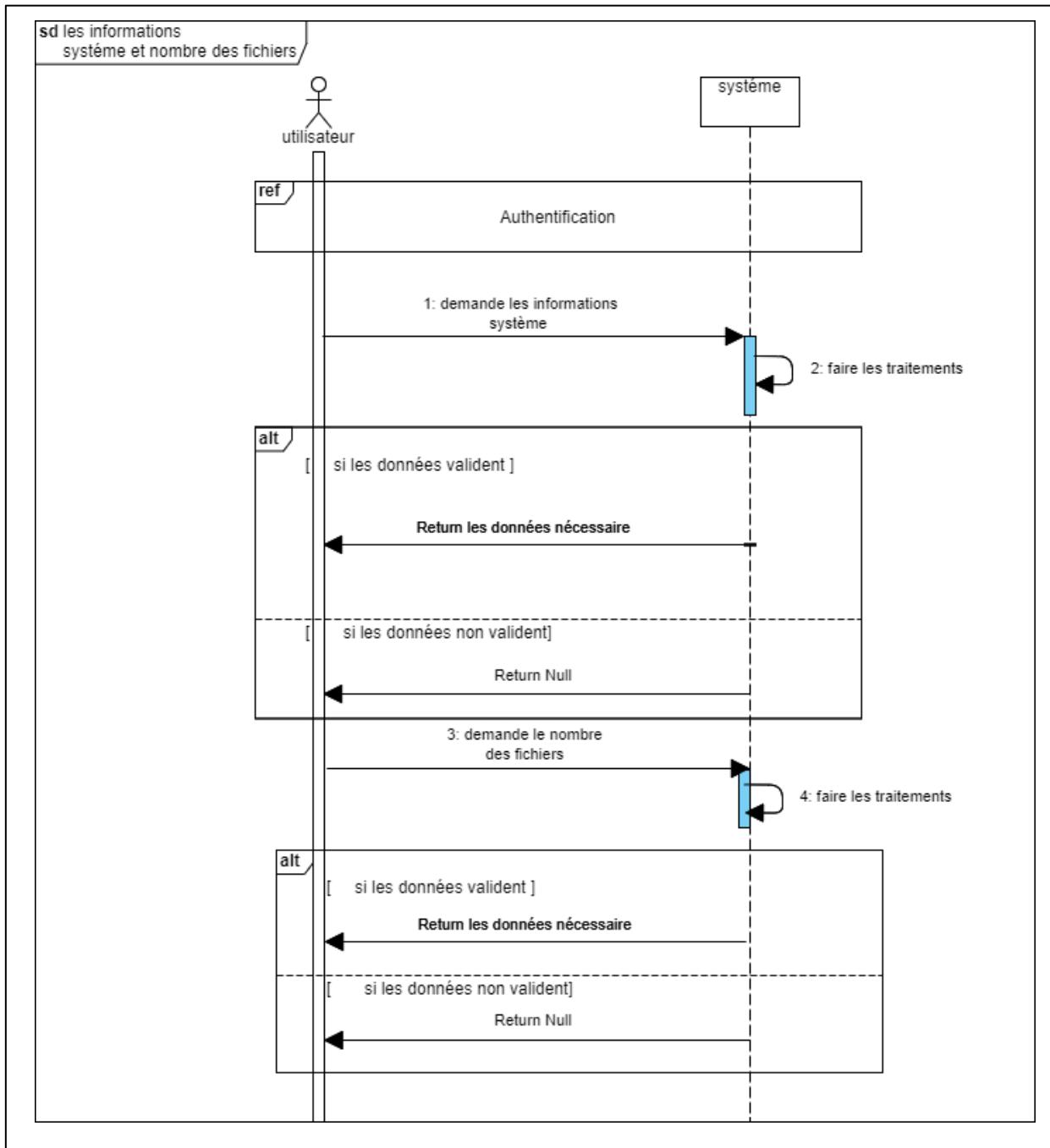


Figure .14: Diagramme de séquence «consulter les informations système et nombre des fichiers»

Après l'authentification, l'utilisateur a le droit de consulter les informations du système et le nombre des fichiers en cours d'envoi.

Conclusion

Dans ce chapitre nous avons présenté les sprints de l'application tout en faisant la description fonctionnelle détaillée et en nous appuyant sur la modélisation de chaque fonctionnalité.

Chapitre 4 : Réalisation

Introduction

Dans ce chapitre, nous allons présenter les aspects techniques mis en œuvre pour l'accomplissement de ce projet.

Pour ce faire, nous allons exposer l'architecture de notre application, l'environnement de travail dans lequel va être réalisé notre projet ainsi que la présentation des interfaces.

4.1. Architecture du système (Architecture Rest Api)

API (3) (interface de programme d'application) est un ensemble de règles permettant à différents programmes de communiquer entre eux. Il décrit la manière appropriée pour un développeur de logiciels de composer un programme sur un serveur qui communique avec diverses applications clientes.

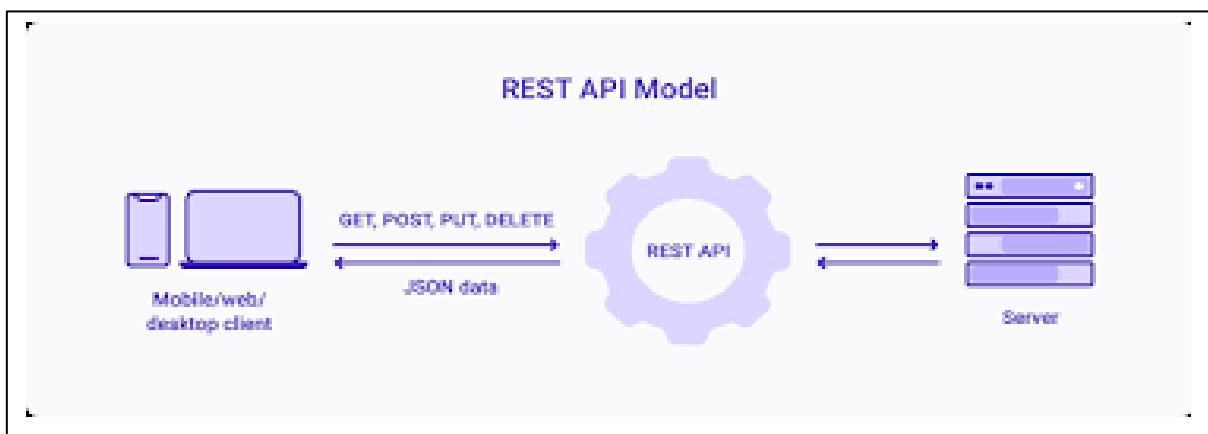


Figure .15: Architecture Rest API

4.2. Environnement de travail

Logo	Nom	Description
	Postman (4)	Postman sert à exécuter des appels HTTP (5) directement depuis une interface graphique. Vous pourrez simplement choisir l'URL, la méthode HTTP (le plus souvent GET, POST, PUT, PATCH et DELETE), les headers, les query params et dans certains cas le body de la requête.
	Visual Studio Code (6)	Un éditeur de code extensible développé par Microsoft.
	PhpMyadmin (7)	PhpMyAdmin est un outil logiciel gratuit écrit en PHP, destiné à gérer l'administration de MySQL sur le Web.
Azure Active Directory	Azure Ad (8)	Azure Active Directory est un programme capable d'intégrer un système d'identité local comme le fait Windows Server Active Directory avec Azure Active Directory. Cet outil d'intégration d'identité locale permet de connecter plusieurs utilisateurs à Microsoft Office 365, Azure ainsi que des milliers d'applications SaaS.
	Node JS (9)	Une plateforme logicielle libre en JavaScript. Nécessaire pour l'exécution de code.
	Visual Paradigm (10)	C'est un outil de création de diagrammes basé sur le Web qui prend en charge un grand nombre de diagrammes commerciaux et techniques.

Tableau .15: Environnement logiciel

4.3. Langage et Framework de développement

4.3.1- Coté backend

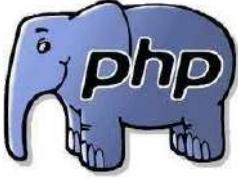
Logo	Nom	Description
	PHP (11)	PHP est un langage de script utilisé le plus souvent côté serveur : dans cette architecture, le serveur interprète le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS par exemple) et des données (JPEG, GIF, PNG par exemple) pouvant être interprétés et rendus par un navigateur web.

Tableau .16: Langages et Framework de développement coté backend

4.3.2- Coté frontend

Logo	Nom	Description
	HTML HTML (12)	Un langage universel utilisé pour les pages web lisibles par tous les navigateurs web. Il fonctionne suivant la combinaison de balises permettant de structurer les données et donner l'apparence voulue aux images, textes et multimédias.
	Mui material (13)	Material UI est une bibliothèque complète de composants qui présente notre implémentation du système Material Design de Google

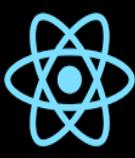
	JavaScript (14)	Un langage de script orienté objet utilisé principalement dans les pages HTML.
	React (15)	React est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.

Tableau .17: Langages et Framework de développement côté frontend

4.4. Présentation des interfaces de l'application

4.4.1. Interface de l'authentification

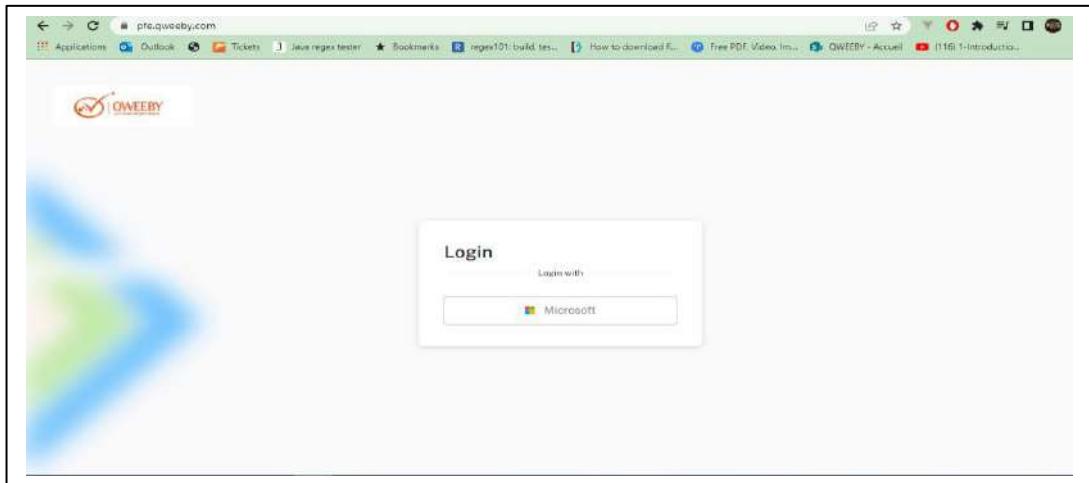


Figure .16: Interface de l'authentification 1

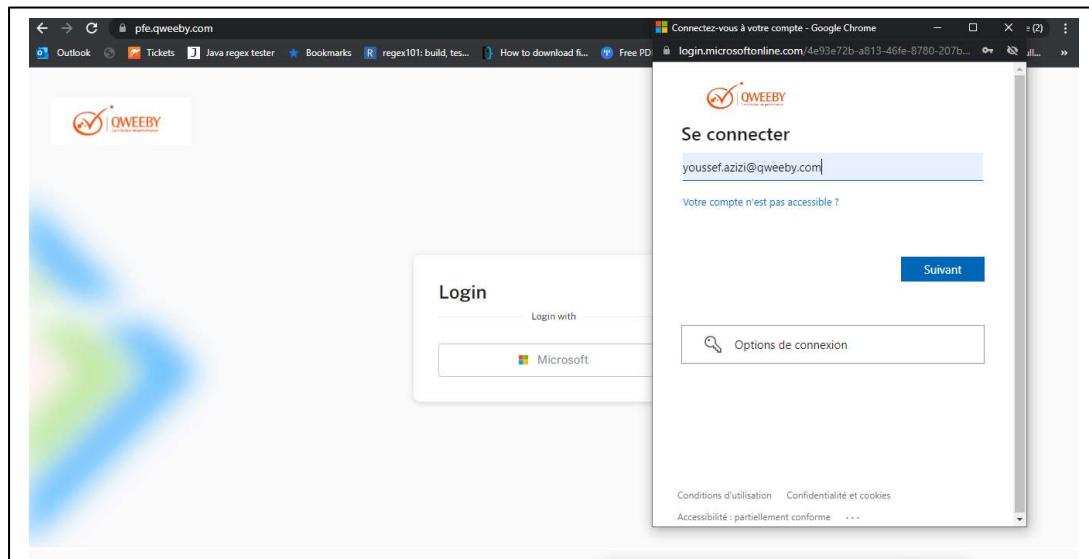


Figure .17: Interface de l'authentification 2

- ✓ Pour accéder à n'importe quelle page, le responsable doit s'authentifier avec son compte Microsoft fourni par la société. Pour cela nous avons réalisé cette interface qui est la page de connexion.

4.4.2. Interface du Dashboard

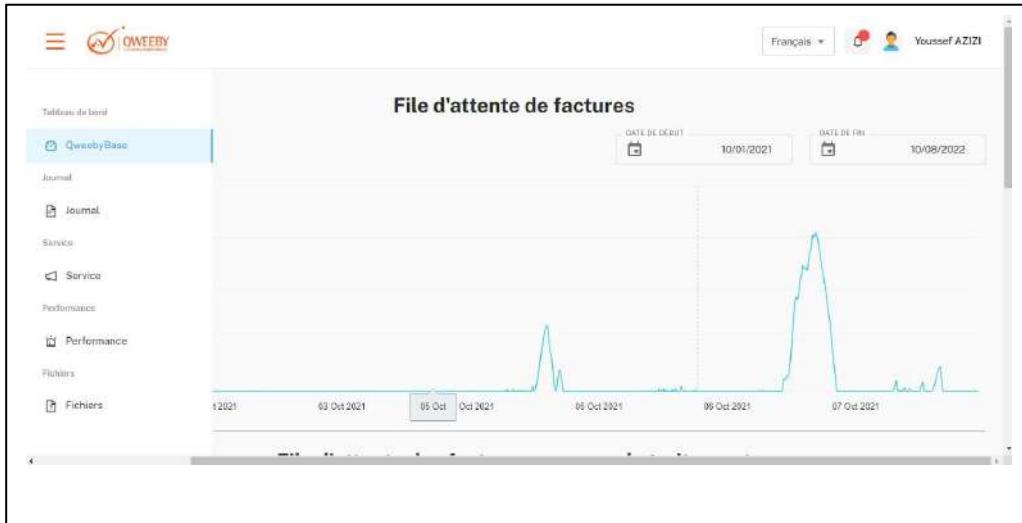


Figure .18: Interface du Dashboard 1

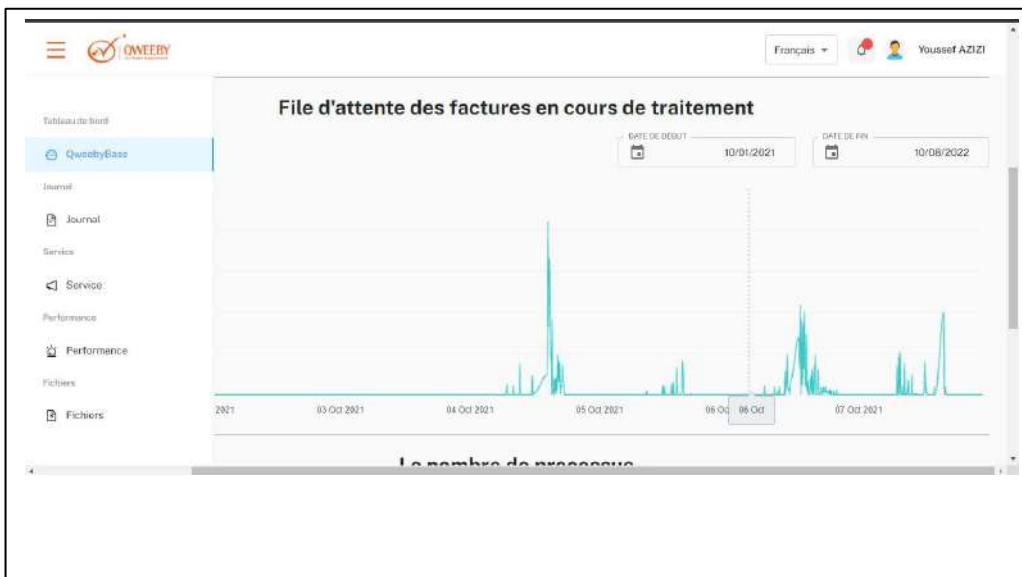


Figure .19: Interface du Dashboard 2

- ✓ Cette interface permet au responsable d'avoir les analyses des files d'attente de factures, des files d'attentes de factures en cours de processing et nombre de Thread.

4.4.3. Interface des fichiers Logs

Chemin	Nom de fichier	Taille	Date d'insertion	Type de fichier
/fr41381550...	fr413815520...	1887	2022-10-08 04:00:55	csv
/FR82330001...	FR82330001...	495877	2022-10-08 05:20:03	csv
/FR81381382...	FR813813892...	285	2022-10-08 06:00:03	csv
/FR45429373...	FR454293703...	4041	2022-10-08 06:00:34	csv
/fr41381554...	fr413815524...	1887	2022-10-08 06:00:55	csv
/5137008985...	51370089805...	2589	2022-10-08 07:55:46	csv
/fr41381556...	fr413815526...	1887	2022-10-08 08:00:55	csv
/fr41381557...	fr413815527...	1887	2022-10-08 10:00:55	csv

Figure .20: Interface des fichiers Logs

- ✓ Cette interface permet au responsable d'avoir les fichiers du log et d'effectuer des recherches par le nom du fichier ou le type du fichier.

4.4.4. Interface du service (AS2, FTP ...)

État du service
AS2 1 ✓
AS2 2 ✓
FTP ✓
HTTPS ✓
BOQ ✓
QweebyBase ✓

HTTPS 99.992 %	<div style="width: 99.992%;"></div>
BOQ 100.00 %	<div style="width: 100.00%;"></div>
FTP 100.00 %	<div style="width: 100.00%;"></div>

Figure .21: Interface du service (AS2, FTP ...)

- ✓ Cette interface permet au responsable d'avoir l'état des services (AS2, FTP, ALAD1 HTTPS, BOQ, QBASE).

4.4.5. Interface des informations du système



Figure .22: Interface des informations du système

- ✓ Cette interface permet au responsable d'avoir les informations du système (RAM, CPU...).

4.4.6. Interface du file en attente d'envoi

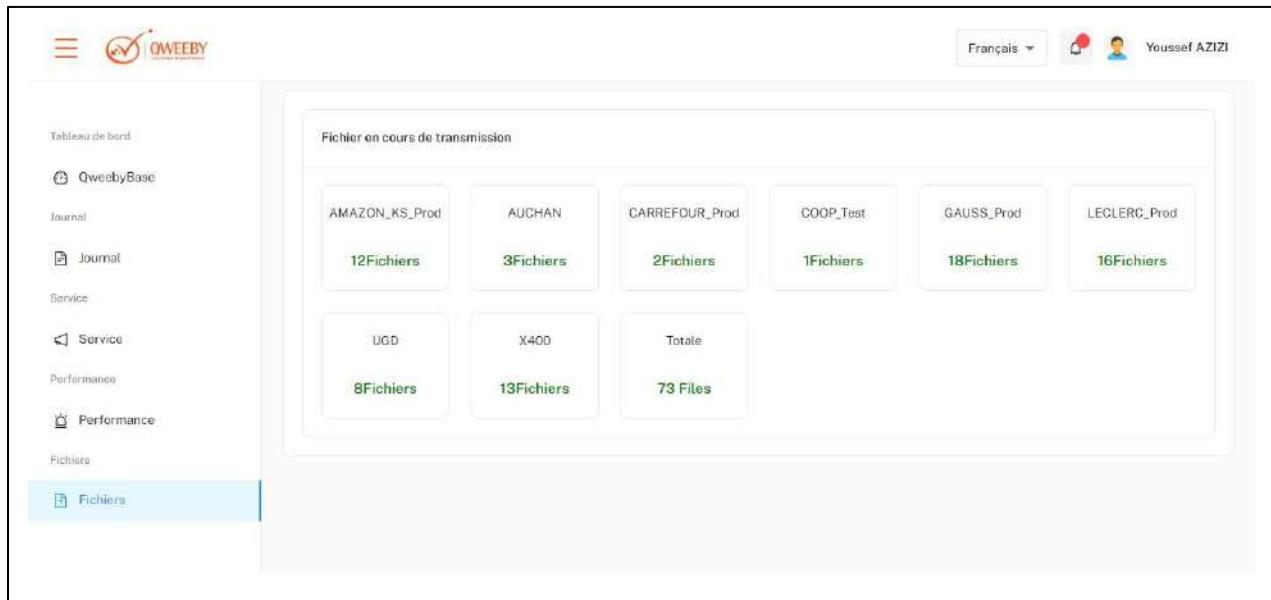


Figure .23: Interface de la file en attente d'envoi

- ✓ Cette interface permet au responsable d'avoir les files qui est en cours d'envoi.

4.4.7. Interface de l'état de la base

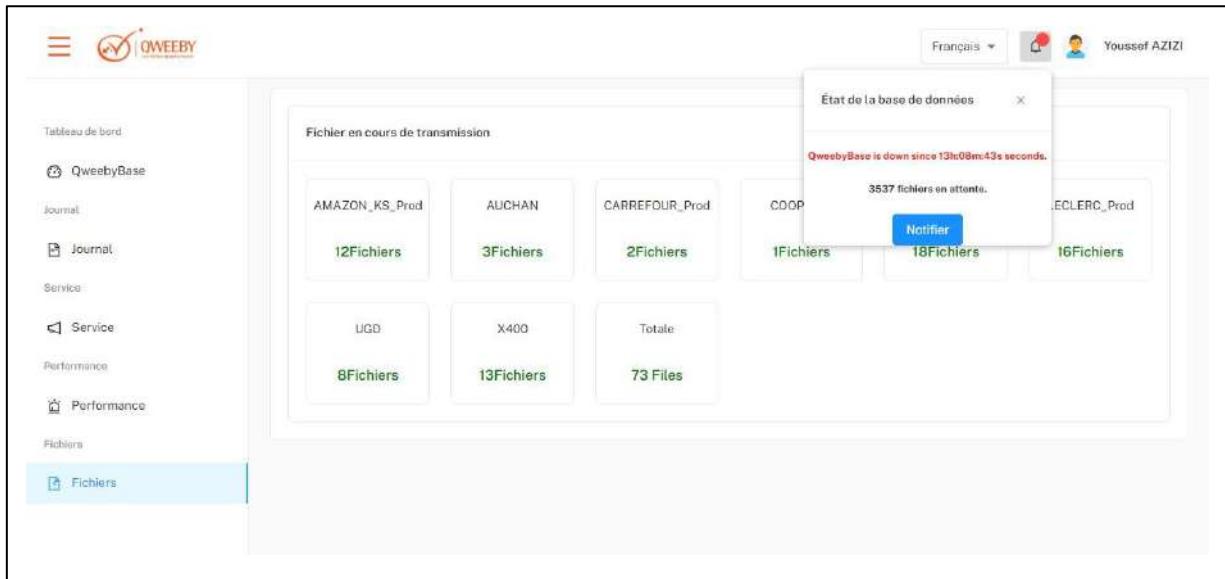


Figure .24: Interface de l'état de la base

- ✓ Cette interface permet au responsable d'avoir l'état de la base de données à l'aide de l'Icon de la notification.

4.4.8. Interface de déconnection

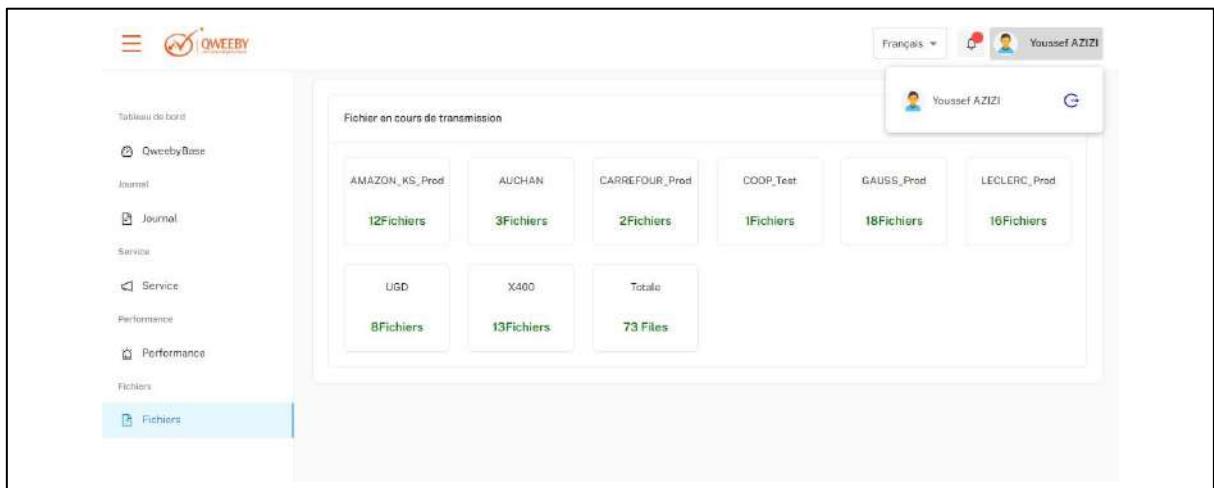


Figure .25: Interface de déconnection

- ✓ Cette interface permet au responsable de déconnecté de votre espace.

4.4.9. API (php pure)

```
> base
> SeviceStatus
BaseDown.php
config.php
DataFileFacture.php
DataFileFactureProces...
DataThread.php
dateDebut.php
FactureAttente.php
FilterNom.php
GetLog.php
mail.php
Nombrefichier.php
system.php
xferlog

1 <?php
2 include('config.php');
3 $dataQB = 0;
4 $handle = $link->prepare('SELECT TIMESTAMPDIFF(SECOND,sqb_insertion_date,NOW()) AS time,
5 $handle->execute();
6 $result = $handle->fetchAll(\PDO::FETCH_OBJ);
7
8 foreach($result as $row){
9     $dataQB = $row->time;
10 }
11
12 echo json_encode($dataQB, JSON_NUMERIC_CHECK);
13
14
15 ?>
```

Figure .26: API (PHP pure)

- ✓ Cette interface représente code source du partie backend de notre application.

Figure .27: réponse Json

- ✓ Cette interface représente un exemple du réponse Json de notre API.

Conclusion

Dans ce chapitre nous avons présenté l'architecture de notre application, les technologies et les outils utilisés dans le développement ainsi que les interfaces réalisées.

Conclusion générale et perspectives

À mesure que la technologie évolue, de nombreuses entreprises qui ne veulent pas être en retard accélèrent la transformation numérique et l'automatisation de leurs tâches en mettant en œuvre divers logiciels pour accélérer les processus quotidiens.

C'est dans ce cadre que s'inscrit notre projet effectué au sein de la société **Qweeby**, qui consiste à mettre en place une solution ‘Dashboard It’.

Afin de mener à bien ce travail, nous avons commencé notre projet par une phase d'apprentissage des nouvelles technologies et langages existants, il a fallu aussi étudier et comprendre ce qui a déjà existé dans l'industrie. Ce qui était également représenté dans le premier chapitre et la mention de l'existant puis élaborant une solution convenable à notre environnement industriel en se fixant à une méthodologie livrant de manière productive et créative des produits de la plus grande valeur possible.

Après, nous avons spécifié nos besoins fonctionnels et non fonctionnels. Cette phase nous a permis de dégager les cas d'utilisations qui nous ont permis à leur tour de modéliser notre solution pour avoir une vue claire. A cet égard, nous avons entamé la partie conception ou nous avons pu commencer à voir la forme réelle de notre application.

Bibliographie

1. Qu'est-ce que la méthodologie Scrum ? *Planzone*. [En ligne] 20 08 2017. [Citation : 03 05 2022.] <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1443834-scrum-guide-de-la-methode-agile-star/>.
2. Les diagrammes de cas d'utilisation UML. *Lucidchart*. [En ligne] 08 05 2022. [Citation : 01 07 2022.] <https://www.lucidchart.com/pages/fr/diagramme-de-cas-dutilisation-uml>.
3. Qu'est-ce qu'une API. *mulesoft*. [En ligne] 01 01 2022. [Citation : 09 07 2022.] <https://www.mulesoft.com/fr/resources/api/what-is-an-api>.
4. Postman: le guide complet de l'outil indispensable . *practicalprogramming*. [En ligne] 11 08 2020. [Citation : 11 07 2022.] <https://practicalprogramming.fr/postman/>.
5. HTTP - MDN Web Docs. *mozilla*. [En ligne] 16 03 2022. [Citation : 12 07 2022.] <https://developer.mozilla.org/fr/docs/Web/HTTP>.
6. Visual Studio Code. *wikipedia*. [En ligne] 14 11 2015. [Citation : 15 07 2022.] https://fr.wikipedia.org/wiki/Visual_Studio_Code.
7. phpMyAdmin. *Wikipédia*. [En ligne] 10 04 2022. [Citation : 17 07 2022.] <https://fr.wikipedia.org/wiki/PhpMyAdmin>.
8. Azure Active Directory documentation. *Microsoft Entra*. [En ligne] 05 02 2022. [Citation : 20 07 2022.] <https://docs.microsoft.com/en-us/azure/active-directory/>.
9. Node.js . *Wikipédia*. [En ligne] 27 04 2009. [Citation : 25 07 2022.] <https://fr.wikipedia.org/wiki/Node.js>.
10. Visual Paradigm. *wikipedia*. [En ligne] 01 09 2010. [Citation : 30 07 2022.] https://en.wikipedia.org/wiki/Visual_Paradigm.
11. php. *Wikipédia*. [En ligne] 04 04 2022. [Citation : 20 07 2022.] <https://fr.wikipedia.org/wiki/PHP>.
12. HTML. *wikipedia*. [En ligne] 05 04 2011. [Citation : 27 07 2022.] https://fr.wikipedia.org/wiki/Hypertext_Markup_Language.
13. MUI: The React component library you always wanted. *MUI*. [En ligne] 03 03 2022. [Citation : 30 07 2022.] <https://mui.com/>.
14. JavaScript. *Wikipédia*. [En ligne] 10 06 2022. [Citation : 01 08 2022.] <https://fr.wikipedia.org/wiki/JavaScript>.
15. React – Une bibliothèque JavaScript pour créer des interfaces . *React* . [En ligne] 21 01 2022. [Citation : 08 06 2022.] <https://fr.reactjs.org/>.
16. Qu'est-ce que la méthode Agile ? . *Planzone*. [En ligne] 11 07 2017. [Citation : 02 05 2022.] <https://www.planzone.fr/blog/quest-ce-que-la-methodologie-agile>.
17. UML (informatique). *lucidchart*. [En ligne] 01 12 2017. [Citation : 06 05 2022.] <https://www.lucidchart.com/pages/fr/langage-uml>.