



HONORIS UNITED UNIVERSITIES

2021 / 2022

SPÉCIALITÉ : SAE

Conception et développement d'une plateforme ESG « ESG Smarter »

Réalisé par : Mondher Haddad

Encadré par :

Encadrant ESPRIT : Mr. Bilel Charfi

Encadrant Entreprise : Mr. Omar Yaacoubi



Délivrée par la
Commission
des Titres



Dédicace

Remerciement

Table des matières

Introduction générale.....	1
CHAPITRE 1 : CADRE GENERALE DU PROJET	2
1. Introduction.....	3
2. Organisme d'accueil.....	3
2.1. Présentation :	3
2.2. Services :.....	3
3. Présentation du projet.....	3
3.1. Objectifs du projet :	3
3.2. Problématique :	4
3.3. Analyse de l'existant :.....	4
3.4. Critique de l'existant :.....	5
3.5. Solution proposée :.....	5
4. Méthodologie adoptée :	6
4.1. Plan de travail :	6
4.2. Choix de la méthode :	6
4.3. Présentation de la méthodologie :.....	6
5. Planification du projet :	7
6. Conclusion :	7
CHAPITRE 2 : ANALYSE ET SPECIFICATION DES BESOINS	8
1. Introduction.....	9
2. Identification des besoins	9
2.1. Besoins fonctionnels	9
2.2. Besoins non fonctionnels	10
2.3. Identification des acteurs :.....	10
3. Analyse:	10
3.1. Diagramme de cas d'utilisation global :	10
3.2. Diagrammes de cas d'utilisation raffinés :	11
3.3. Description procédure de la registration et authentification Utilisateur :	14
4. Maquettage de l'application :.....	16
5. Architecture :.....	18
5.1. Architecture Physique :.....	18
5.2. Architecture Logique :.....	18
6. Conclusion :	20
CHAPITRE 3 : ETUDE CONCEPTUELLE	21
1. Introduction :.....	22
2. Conception générale :.....	22
2.1. Design Pattern (Patron de conception) :	22

2.2. Architecture MVC :	22
3. Conception détaillée :	23
3.1. Vue statique :	23
3.2. Vue Dynamique :	25
4. Conclusion :	28
CHAPITRE 4 : REALISATION	29
1. Introduction :	30
2. Diagramme de déploiement :	30
3. Environnement Logiciel :	30
3.1. Outils de développement :	31
3.2. Outils d'intégration et collaboration :	33
3.3. Outils de conception et Design :	33
4. Interfaces de la plateforme :	34
5. Conclusion :	39
Conclusion générale et perspectives	Error! Bookmark not defined.

Table des figures

<i>Figure 1 : Logo DataWize</i>	3
<i>Figure 2 : Logo Streamlit</i>	4
<i>Figure 3: Logo Bloomberg</i>	5
<i>Figure 4 : Les différents facteurs ESG</i>	5
<i>Figure 5: Les phases et les itérations de la méthodologie PU</i>	6
<i>Figure 6: Diagramme de Gantt</i>	7
<i>Figure 7: Diagramme de cas d'utilisation global</i>	11
<i>Figure 8 : Diagramme de cas d'utilisation raffiné d'un client</i>	11
<i>Figure 9 : Diagramme de séquence système « Dashboard Admin »</i>	12
<i>Figure 10 : Diagramme de cas d'utilisation raffiné « Consulter scores ESG »</i>	13
<i>Figure 11 : Diagramme de séquence système « Consultation des scores ESG »</i>	14
<i>Figure 12 : Diagramme de séquence de la procédure de registration</i>	15
<i>Figure 13: Diagramme de séquence d'authentification</i>	16
<i>Figure 14 : Maquettage interface registration et authentification</i>	16
<i>Figure 15: Maquettage pour l'interface graphique « Dashboard Client »</i>	17
<i>Figure 16 : Maquettage pour l'interface graphique « Dashboard Admin »</i>	17
<i>Figure 17 : Architecture physique</i>	18
<i>Figure 18 : Architecture Logique</i>	19
<i>Figure 19 : Architecture MVC</i>	23
<i>Figure 20 : Diagramme de classe global</i>	24
<i>Figure 21 : Diagramme de séquence objets « Consultation des scores ESG »</i>	25
<i>Figure 22 : Diagramme de séquence objets « Gestion des utilisateurs »</i>	26
<i>Figure 23: Diagrammes d'activité « Gestion réclamation » et « Gestion Users »</i>	27
<i>Figure 24: Diagramme de déploiement</i>	30
<i>Figure 25: Logo STS</i>	31
<i>Figure 26 : Logo VS code</i>	31
<i>Figure 27: Logo MongoDB</i>	31
<i>Figure 28 : Logo ReactJS</i>	32
<i>Figure 29 : Logo GraphQL</i>	32
<i>Figure 30 : Logo Postman</i>	32
<i>Figure 31 : Logo ChartJS</i>	32
<i>Figure 32 : Logo GitHub</i>	33
<i>Figure 33 : Logo Jira</i>	33
<i>Figure 34 : Logo Slack</i>	33
<i>Figure 35 : Logo draw.io</i>	34
<i>Figure 36 : Logo Balsamiq</i>	34
<i>Figure 37 : Interface pour la registration utilisateur</i>	34
<i>Figure 38 : Interface pour s'authentifier</i>	35
<i>Figure 39 : Email de confirmation registration</i>	35
<i>Figure 40 : Tableau de bord de l'utilisateur par défaut</i>	36
<i>Figure 41 : Interface d'affichage des données ESG pour le client</i>	36
<i>Figure 42 : Affichage des informations ESG filtrés par secteur</i>	37
<i>Figure 43 : Affichage des informations ESG filtrés selon la localisation de l'entreprise</i>	37
<i>Figure 44 : Profil du client</i>	37
<i>Figure 45 : Interface de la gestion des utilisateurs</i>	38
<i>Figure 46 : Interface de la gestion des réclamations et les rapports de bug</i>	38
<i>Figure 47 : Notification envoyé pour le client concernant son réclamation</i>	39

Table des tableaux

<i>Tableau 1: Description Textuelle – Cas d'utilisation «Dashboard Admin»</i>	12
<i>Tableau 2: Description textuelle – Cas d'utilisation « Consultation des scores ESG »</i>	13
<i>Tableau 3: Description textuelle de la procédure de registration et authentification</i>	15

Introduction générale

Généralement, les investisseurs autour du monde mesurent les capacités et la force d'une telle entreprise selon l'état financière, la force de production, et surtout le positionnement de l'entreprise dans le marché par rapport aux autres concurrents.

Mais vu l'énorme concurrence dans les différents secteurs et le besoin des investisseurs aux études bien précises et décisifs pour prendre une décision et une vision globale sur l'état d'une entreprise choisie, les facteurs classiques mentionnés au-dessus deviennent insuffisante et surtout ne sont pas synchronisés avec l'évolution de la situation économiques mondiale.

Et sur cette base, notre plateforme « ESG Smarter » entre en jeu à travers des avantages non plus très importantes mais décisifs dans ce contexte, parce que notre plateforme se concentre sur 3 critères ESG principaux : les critères environnementaux, les critères sociales et les critères de gouvernance qui nous permet de découvrir des informations cachées dans le Data grâce à des solutions innovantes et intelligentes.

Les critères ESG sont utilisés pour évaluer la performance globale de l'entreprise. Ces critères vont au-delà du succès financier d'une entreprise et évaluent plutôt la manière dont une entreprise est gérée de manière durable et éthique. L'analyse des opérations non financières d'une entreprise peut déterminer si une organisation réussira à long terme ou non, et c'est l'objectif principal de notre plateforme pour guider notre client vers une décision finale décisif et satisfaisante pour lui.

CHAPITRE 1 : CADRE GENERALE DU PROJET

1. Introduction

Dans ce chapitre nous allons présenter en premier lieu l'organisme d'accueil au sein de laquelle j'ai eu l'opportunité de d'effectuer mon stage de fin d'études durant six mois, on va situer le projet dans son cadre général et les objectifs pour lesquels il a été conçu et développé. Nous allons par la suite décrire l'analyse de l'existant ainsi que la problématique et leur ampleur. Enfin, nous présenterons brièvement la solution proposée et la méthodologie de développement de notre plateforme.

2. Organisme d'accueil

2.1.Présentation :

DataWize est une startup franco-tunisienne fondée en 2019, par des compétences jeunes dans les domaines du Data, Digital et le cyber sécurité.

DataWize donne l'opportunité pour les clients à découvrir des informations importantes cachées dans le data et qui sert à identifier à moindre coût des nouvelles opportunités commerciales et générer plus de revenus.



Figure 1 : Logo DataWize

2.2.Services :

DataWize est un service qui aide les entreprises à évoluer et progresser à travers l'exploitation des données importantes extraites depuis plusieurs sources et avec différents méthodes ce qui donne le client l'opportunité à :

Découvrir des informations et corrélations cachées et de comprendre mieux les données afin d'identifier les opportunités en utilisant l'analyse avancée et l'intelligence artificielle.

Déploiement rapide des solutions I.A pour transformer les analyses rapides en vrai opportunités business.

3. Présentation du projet

Notre projet consiste à réaliser une conception, développement et le déploiement d'une plateforme ESG pour afficher les résultats des analyses et des algorithmes de scoring des sociétés ESG.

3.1. Objectifs du projet :

- Validation de l'architecture de la plateforme
- Conception et modélisation des différents modules de la solution

- Développement de la plateforme (back-end et une partie front-end)
- Tests et validation

3.2. Problématique :

Les investisseurs autour du monde utilisent de plus en plus les informations ESG pour éclairer leurs décisions d'investissements, mais ce besoin et l'intérêt porté à ces bilans d'évaluation basés sur les scores ESG n'a pas attiré l'attention ce qui rendre le choix d'une plateforme qui opère sur ce genre de choses très limitée.

En contrepartie, les différents bilans financières et juridiques deviennent insuffisants pour un investisseur qui veut consulter des données utiles et décisifs afin de prendre une décision d'investissement peut être vitale pour lui.

3.3. Analyse de l'existant :

Sur l'échelle national, il n'y a pas vraiment des solutions proposés qui opère sur ce genre de plateforme de scoring ESG vu que les investisseurs ne donnent pas l'importance à l'intelligence économique, autrement dit, il n'y a pas la culture de recherche en profondeur des données disponibles.

Par contre sur l'échelle internationale, ce genre d'analyse des données commence à devenir une tendance dans le domaine de l'intelligence économique, et la preuve qu'il y a des entreprises ont commencés à travailler sur des solutions d'approvisionnement en données ESG.

Parmi les plus importantes solutions existantes, on peut citer :

➤ **Streamlit :**

Streamlit est une librairie open source Python créée en 2018 par une équipe de Data-Science américaine. Cette fameuse et puissante librairie donne la possibilité aux clients de créer des applications Web de Machine Learning en Python pur sans besoin d'une partie frontend pour la gestion d'affichage des données.

ESG AI était l'un des applications crée par Streamlit dans le domaine des applications de scoring ESG, c'est une application avec une seule page Web qui peut générer des rapports ESG à l'utilisateur d'un entreprise donnée.



Figure 2 : Logo Streamlit

➤ **Bloomberg :**

Est un groupe financier américain spécialisé dans les services aux professionnels des marchés financiers et dans l'information économique et financière.

Bloomberg opère sur ses propres solutions ESG professionnelles, cette entreprise américaine

est considérée parmi les leaders sur le marché de l'industrie économiques, elle offre des solutions ESG uniques et complète afin de guider le client vers des décisions financières durable lors de la prise de décision stratégique.



Figure 3: Logo Bloomberg

3.4.Critique de l'existant :

Les solutions proposées par « Streamlit » restent limitées en termes de services disponibles pour le client afin de filtrer et naviguer à la profondeur de ses données pour obtenir une vision global sur l'état d'une entreprise choisie par et aussi afin d'exporter un rapport final qui lui permet de prendre une décision et comprendre mieux les informations importantes cachées dans les différentes sources de données.

D'une autre côté, il y a un autre critère qui est très important et décisif au niveau de choix entre ces solutions proposées, c'est le « Fake News » ou les fausses nouvelles, il faut toujours garantir l'aspect de la crédibilité et la transparence lors de la présentation de ces informations et scores ESG afin de gagner la confiance du client. Ce qui n'est pas confirmé dans la plupart des cas.

3.5. Solution proposée :

En se basant sur les points faibles des autres concurrents, l'équipe de DataWize a décidé d'implémenter une solution plus professionnelle et plus efficace par rapport aux autres outils et plateformes présentes sur le marché de l'intelligence économique. Notre plateforme ESG sera facile à utiliser et flexible basée sur les données externes du Web, des documents, des images, du DarkWeb avec une interface et score faciles à comprendre. La capacité ESG devrait permettre aux clients de fournir également leur propre «Score ESG» par rapport à une norme interne ou permettre une comparaison avec des normes externes.



Figure 4 : Les différents facteurs ESG

Pour résoudre le défi des fausses nouvelles, nous devons corréler les informations de 2 ou 3 articles ou sources d'informations avant de les analyser et les transférer vers notre client. De cette façon, on peut assurer la crédibilité et la transparence envers tous nos clients.

4. Méthodologie adoptée :

4.1. Plan de travail :

Comme toute équipe professionnelle en train de réaliser un projet complexe, il faut mettre en place un plan de travail qui nous aide à subdiviser le travail et séparer les tâches à réaliser et surtout de trouver la bonne méthodologie de travail afin d'achever les objectifs désignés de ce projet et bien sûr pour la livraison d'un produit final professionnel et qui couvre ces besoins du client.

4.2. Choix de la méthode :

Suite à une étude approfondie et la consultation de différentes sources de documentation, et surtout par rapport aux besoins de nos clients et la date limite pour présenter une première version de cette plateforme, nous avons optés pour «Unified Process» ou processus unifié comme une méthodologie pour la gestion de notre projet.

Notre choix était basé sur les perceptions de nos responsables envers ce projet, parce que il y a beaucoup des détails techniques et logistiques qui n'étaient pas encore traités lors de démarrage de projet et aussi la forte cohésion entre l'équipe développement et l'équipe des Data-Analystes parce que l'enchaînement de travail est collaboratif et c'était très difficile de synchronisés les tâches entre les différents parties.

Donc voilà notre choix, cette méthodologie se caractérise par plusieurs avantages importantes dans notre situation comme : Décomposition du temps disponible en un ensemble d'itérations, diminution des risques et centrée sur l'architecture.

4.3. Présentation de la méthodologie :

« Le processus unifié est un processus de développement logiciel, c'est-à-dire qu'il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel ».

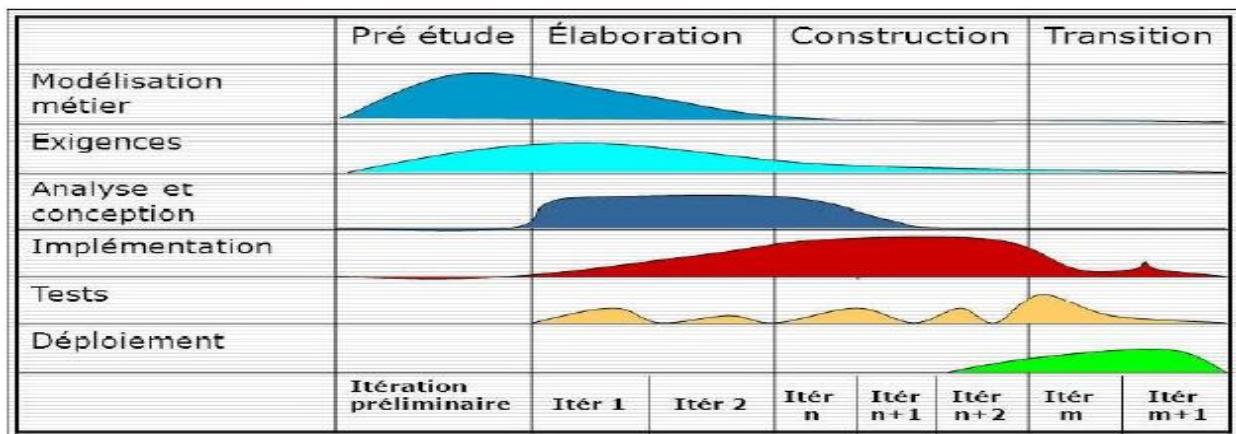


Figure 5: Les phases et les itérations de la méthodologie PU

Pour la réalisation de notre projet, nous nous adapterons au Processus Unifié, est un processus de conception et de réalisation d'un logiciel développé avec des langages de programmation orientés objet (Java par exemple).

C'est un guide méthodologique pour faire du logiciel en conseillant et en guidant l'équipe dans ses différents activités pour réduire la complexité du projet (on sait où on en est et ce que reste à faire).

Ce processus met en lumière « qui fait quoi » et permet à chacun de connaître sa place dans le processus de production du logiciel.

5. Planification du projet :

Pour la planification et le suivi de l'état d'avancement de notre projet, nous avons présentés les différentes phases du projet durant les six mois de stage sur le diagramme de Gantt.

Il s'agit d'une représentation graphique évalué et orienté qui nous permet de suivre notre avancement par rapport à la date fixé comme date limite.

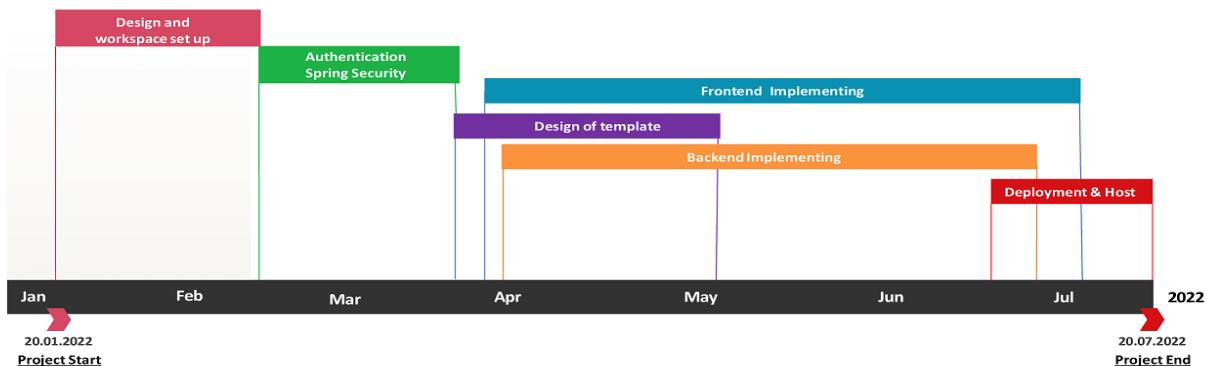


Figure 6: Diagramme de Gantt

6. Conclusion :

Après avoir présenté dans ce chapitre le cadre général du projet ainsi que la plateforme que nous allons proposer comme une solution.

Nous passons maintenant au chapitre suivant qui sera consacré à la partie de l'étude conceptuelle de notre projet.

CHAPITRE 2 : ANALYSE ET SPECIFICATION DES BESOINS

1. Introduction

Après avoir mis notre travail dans son cadre, nous énonçons en première partie les différents besoins auxquels devrait répondre notre application. Ensuite, en deuxième partie, nous allons passer par une phase de conception.

2. Identification des besoins

Dans cette partie, nous identifions les fonctionnalités de notre application. Plus précisément, on va identifier les différents besoins fonctionnels qui décrivent le comportement de fonctionnement de notre plateforme ainsi que les besoins **on** fonctionnels qui décrivent la qualité d'exécution des différents services **présentes**.

2.1.Besoins fonctionnels

- L'application doit permettre à notre client :
 - **S'identifier** : Pour être un client dans notre plateforme, il faut faire l'inscription et l'activation du compte réalisé par email.
 - **S'authentifier** : Après l'inscription, le client devient capable d'accéder aux différents espaces de notre plateforme selon les habilitations nécessaires.
 - **Consulter les données ESG disponibles** : Avec ce service, le client est capable d'accéder aux notations ESG de presque toutes les entreprises dans le monde.
 - **Suivre les scores ESG d'une entreprise choisie** : Cette fonctionnalité donne à l'investisseur la possibilité d'accéder aux données ESG d'une entreprise donnée et de faire une comparaison avec entreprises existantes.
 - **Rechercher des critères des informations ESG par entreprises** : Cette fonctionnalité permet notre client à chercher des informations importantes par entreprises, par secteur d'activité et par région présentés par des graphiques pertinentes de haut niveau.
 - **Fournir des rapports ESG détaillés** : Notre client est capable d'exporter des rapports ESG détaillés basés sur les règles ESG actuels depuis la plateforme.
 - **Envoyer des réclamations et des rapports de Bug** : Dans son Dashboard, le client peut ajouter des nouvelles sources de données personnels pour son propre environnement en fonction de ses besoins.
 - **Suivre la situation environnementale des règles ESG** : Notre client doit pouvoir configurer des alertes/notifications pour suivre la situation des règles ESG.
- L'application doit permettre à notre administrateur :
 - **Consulter la liste des utilisateurs** : à travers son Dashboard spécifique, l'administrateur doit consulter tous les utilisateurs inscrits sur notre plateforme.
 - **Gérer les utilisateurs** : L'administrateur a les privilèges nécessaires pour bloquer le compte d'un utilisateur en cas de besoin.
 - **Gérer les réclamations des utilisateurs** : L'administrateur est le responsable à la gestion des réclamations et les rapports de bug envoyés par les utilisateurs.

2.2.Besoins non fonctionnels

Plusieurs autres besoins doivent être respectés par l'application afin d'offrir une expérience d'utilisation confortable pour notre client et surtout pour respecter les normes et les fonctionnalités **demandées** dans le cahier **de charges** :

- **La performance** : Le temps de réponse du système doit être raisonnable pour assurer une continuité de fonctionnement.
- **L'ergonomie** : Côté UX/UI design, l'application doit contenir des interfaces graphiques bien présentées et ergonomique afin à l'utilisateur la tâche d'interprétation des informations à partit des graphes un peu compliqués.
- **La fiabilité** : Les sources de données dans notre plateforme doivent être fiables et mentionnées pour notre client afin de garantir la crédibilité de notre plateforme.
- **La sécurité** : Nous avons utilisés des algorithmes de cryptage spécifiquement pour protéger les mots de passe de nos clients au niveau de la base de données, ainsi l'accès à notre plateforme est basé sur un mot de passe bien sécurisé avec une politique de gestion des utilisateurs basée sur les rôles.

2.3.Identification des acteurs :

Un acteur représente un rôle joué par une entité externe qui interagit directement avec le système étudié, il peut consulter ou modifier l'état du système.

- ❖ **Les clients de la plateforme « ESG Smarter »** : Les utilisateurs qui ont les habilitations nécessaires pour consulter les différents données et bénéficier de tous les services disponibles dans notre solution.
- ❖ **L'administrateur** : Celui qui est responsable à la gestion des utilisateurs et leurs réclamations.

3. Analyse:

3.1.Diagramme de cas d'utilisation global :

Le diagramme de cas d'utilisation est un moyen qui nous aide à illustrer les fonctions générales et le comportement d'un système par rapport à ses utilisateurs, sans la description du mécanisme de fonctionnement en interne.

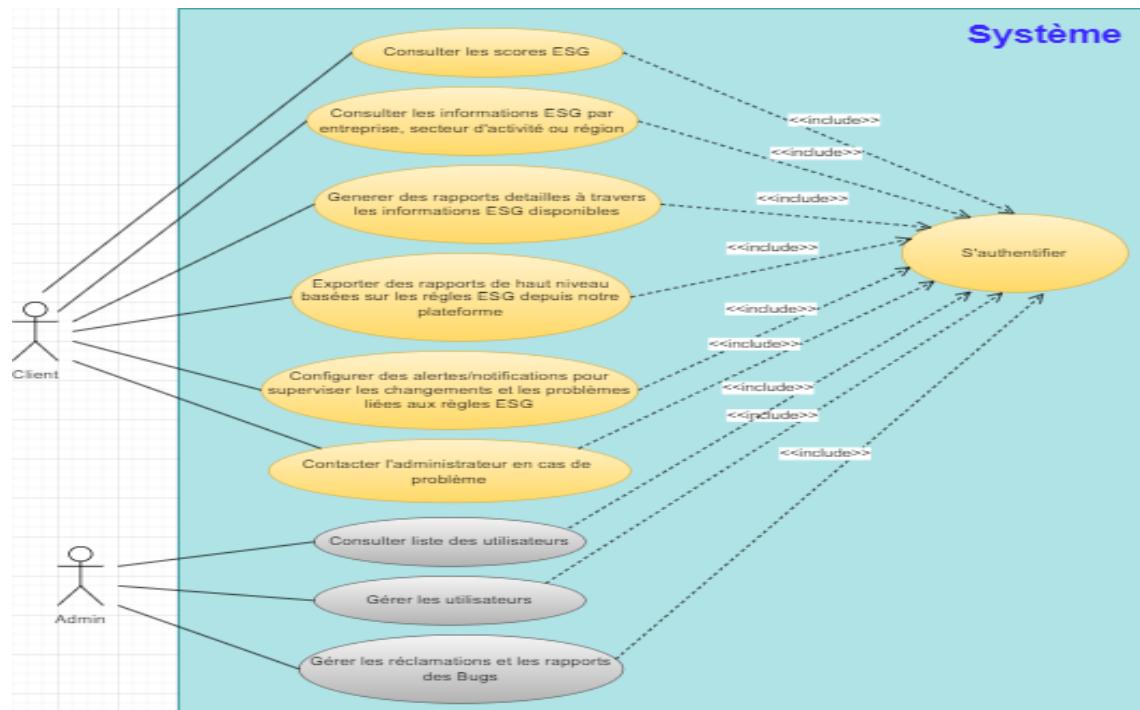


Figure 7: Diagramme de cas d'utilisation global

3.2. Diagrammes de cas d'utilisation raffinés :

Dans cette partie, nous allons diviser le diagramme des cas d'utilisation global en sous cas d'utilisation plus raffinés et plus détaillés afin de mieux comprendre les différents fonctionnalités dans cette plateforme et modéliser en détails comment l'acteur va interagir avec le système.

3.1.1. Cas d'utilisation «Dashboard Admin»

- ❖ Diagramme de cas d'utilisation détaillé :

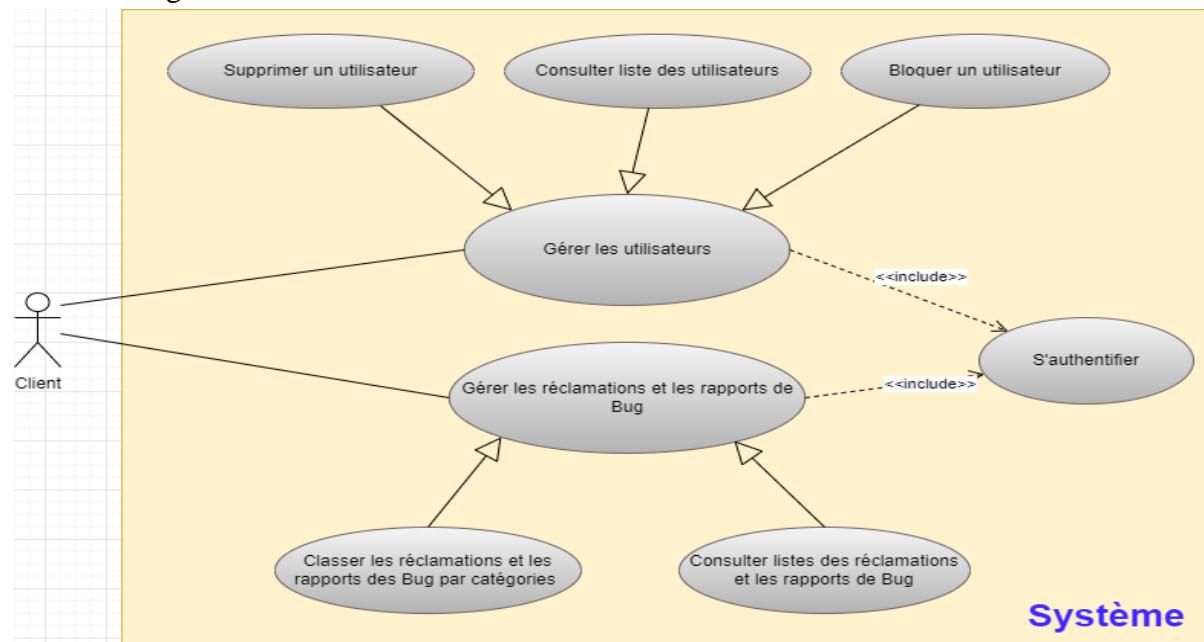


Figure 8 : Diagramme de cas d'utilisation raffiné d'un client

❖ **Description textuelle :**

Nom du cas : Gestion Dashboard Admin
Acteur : Administrateur
But : L'administrateur de la plateforme doit gérer les comptes des utilisateurs inscrits dans notre plateforme et gérer aussi les réclamations et les rapports des Bug envoyés par les utilisateurs
Pré-condition : Acteur identifié
Scenario : <ol style="list-style-type: none"> 1. L'administrateur consulte la liste de tous les utilisateurs inscrits. 2. L'administrateur peut supprimer les comptes non confirmés par l'utilisateur ou bien désactiver un compte d'un utilisateur bien précis. 3. L'administrateur peut consulter la liste des réclamations envoyées par l'utilisateur. 4. L'administrateur change le statut de la réclamation après la résolution du problème décris par l'utilisateur. 5. Opération effectué avec succès, l'utilisateur sera notifié aussi. Exceptions : Si l'email de l'utilisateur est introuvable ou bien son identifiant, un message d'erreur s'affiche

Tableau 1: Description Textuelle – Cas d'utilisation «Dashboard Admin»

❖ Diagramme de séquence système :

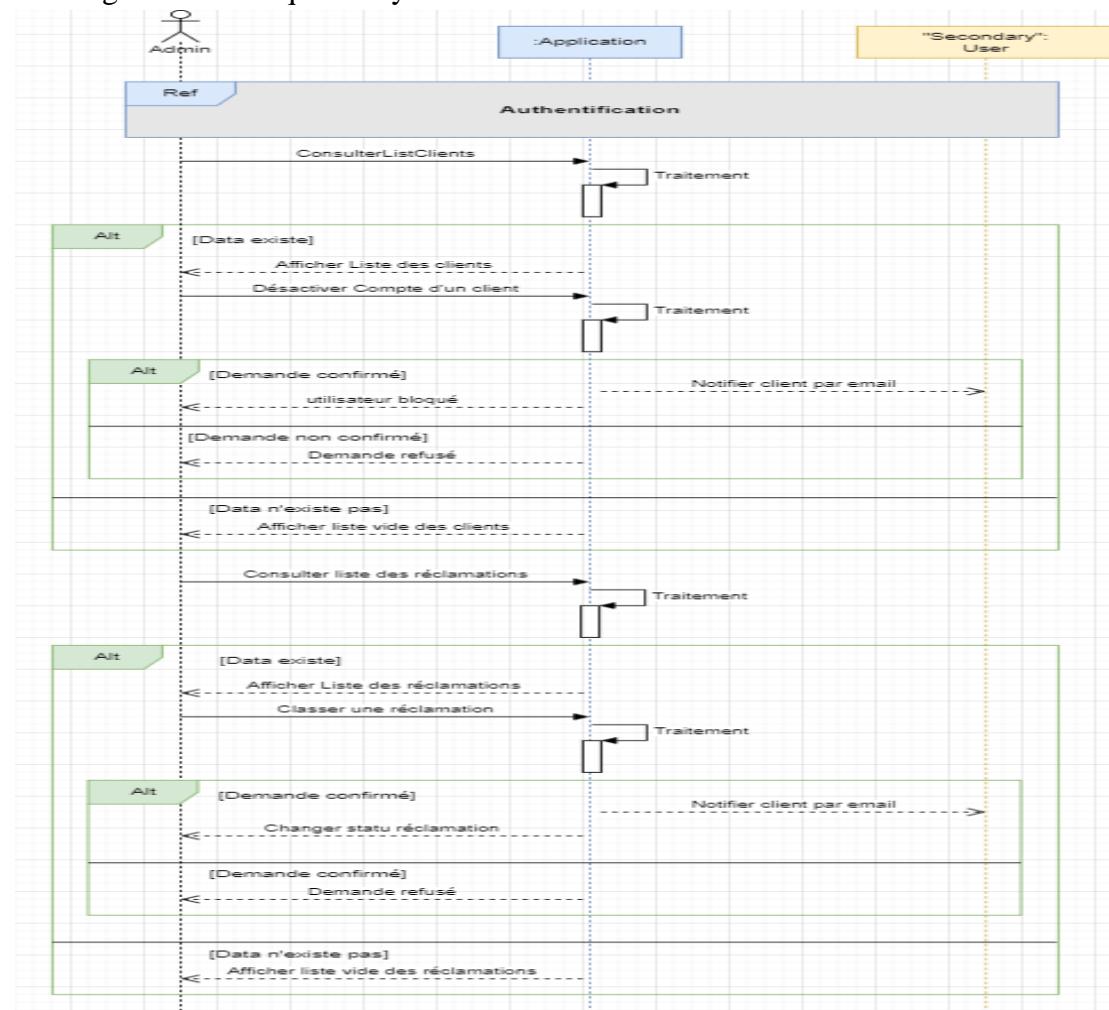


Figure 9 : Diagramme de séquence système < Dashboard Admin >

Le diagramme de séquence système ci-dessus décrit la procédure de la gestion des comptes des utilisateurs et les réclamations reçus de la part de l'utilisateur de la plateforme à travers le dashboard de l'administrateur. Après l'authentification, l'administrateur passe à une interface où il se trouve une liste des différents utilisateurs qui ont des comptes activés réellement, à travers cette interface l'administrateur a le droit de supprimer ou désactiver le compte de l'utilisateur en cas de besoin ou bien il peut passer vers la liste des réclamations envoyées par l'utilisateur pour lui-notifier si le problème précisé par l'utilisateur est vraiment résolu.

3.1.2. Cas d'utilisation « Consultation des scores ESG » :

- ❖ Diagramme de cas d'utilisation détaillé :

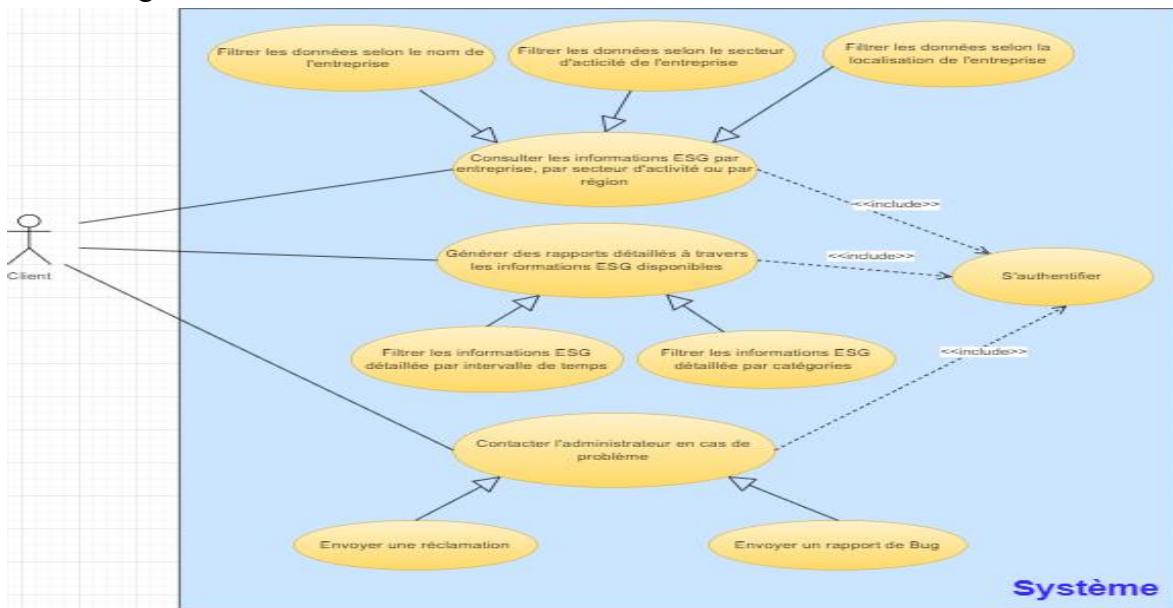


Figure 10 : Diagramme de cas d'utilisation raffiné « Consulter scores ESG »

- ❖ Description textuelle :

Nom du cas : Consulter des scores ESG
Acteur : Client
But : L'utilisateur peut consulter les scores ESG de n'importe quelle entreprise choisie. Il peut aussi filtrer et visualiser les rapports ESG détaillés et vérifier les sources de l'information donnée
Pré-condition : Acteur identifié
Scénario : <ol style="list-style-type: none"> 1. Le client va choisir une entreprise depuis une liste qui contienne plusieurs entreprises autour du monde. 2. Le client va obtenir des scores ESG et des graphes explicatifs. 3. Le client peut consulter et visualiser les différentes sources des données affichées. 4. Le client a l'occasion de filtrer les données ESG selon des critères bien définis comme la localisation, secteur d'activité... 5. Opération effectué avec succès.
Exceptions : Si un type de données est erroné ou des informations sont manquantes, un message d'erreur s'affiche.

Tableau 2: Description textuelle – Cas d'utilisation « Consultation des scores ESG »

❖ Diagramme de séquence système :

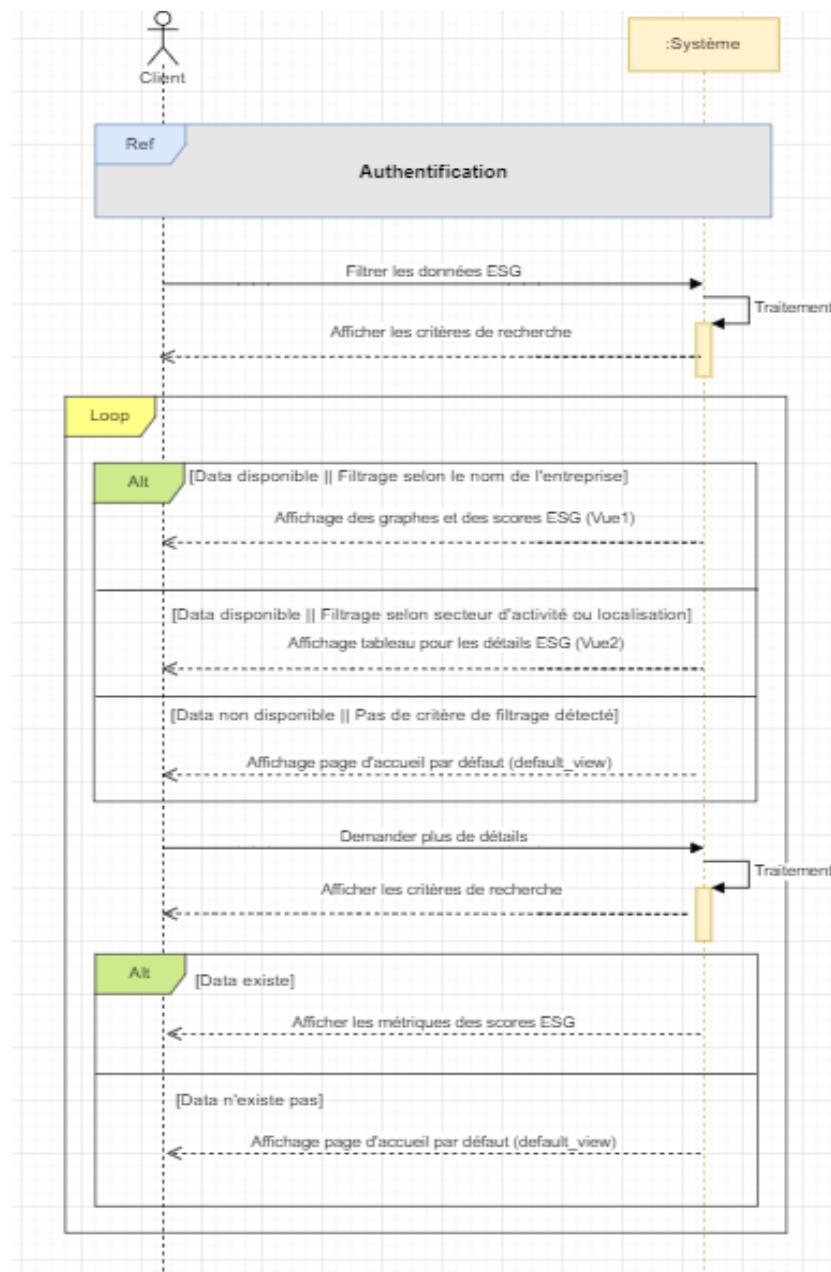


Figure 11 : Diagramme de séquence système « Consultation des scores ESG »

Le diagramme de séquence système ci-dessus décrit la procédure de consultation des scores ESG et les rapports détaillés par un client de la plateforme « ESG Smarter ». Après l’authentification, l’utilisateur va passer vers une interface dynamique qui réagit selon les critères de recherche et les critères de filtrage des données insérés par l’utilisateur lui-même.

3.3. Description procédure de la registration et authentification Utilisateur :

Dans cette partie, nous allons essayer de décrire la procédure de la création d’un compte au sein de notre plateforme par un utilisateur ainsi que la procédure d’authentification lors de lancement d’une nouvelle session de connexion.

❖ **Description textuelle :**

Nom du cas : Registration et authentification utilisateur
Acteur : Utilisateur
But : L'utilisateur peut créer un compte dans notre plateforme pour bénéficier des différents services offerts à travers la plateforme « ESG Smarter ».
Pré-condition : Données personnelles valides de l'utilisateur
Scenario : <ol style="list-style-type: none"> 1. L'utilisateur va créer un nouveau compte avec son email et un mot de passe sécurisée. 2. L'utilisateur doit confirmer la création de son compte à l'aide d'un lien de confirmation envoyé vers son adresse email donnée. 3. L'utilisateur peut lancer une nouvelle session de connexion et accéder à tous les données disponibles sur la plateforme. 4. Opération effectué avec succès. Exceptions : Si l'email donné par l'utilisateur est non valide ou introuvable, un message d'erreur s'affiche comme une alerte.

Tableau 3: Description textuelle de la procédure de registration et authentification

- Diagramme de séquence système de registration :

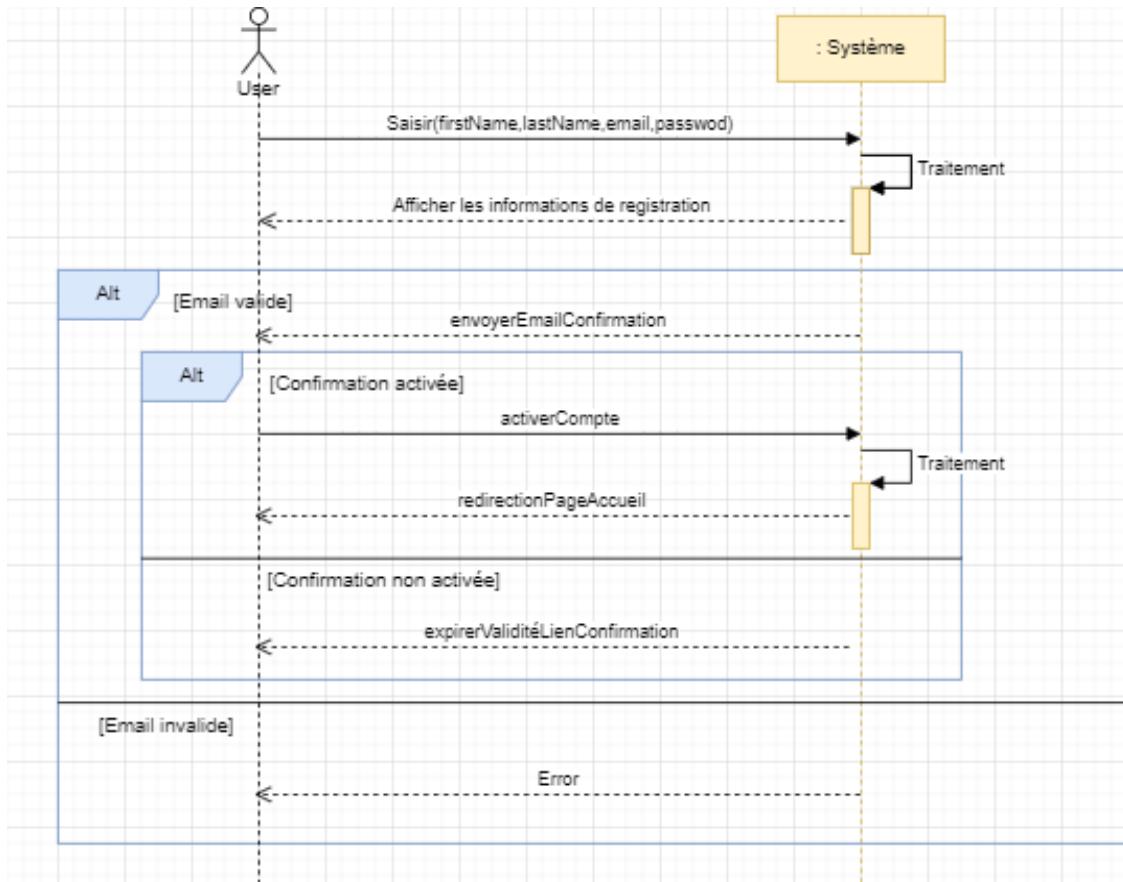


Figure 12 : Diagramme de séquence de la procédure de registration

La figure ci-dessus, nous montre le processus suivi par l'utilisateur afin de créer un compte avec des priviléges bien précis dans cette plateforme.

- Diagramme de séquence système d'authentification :

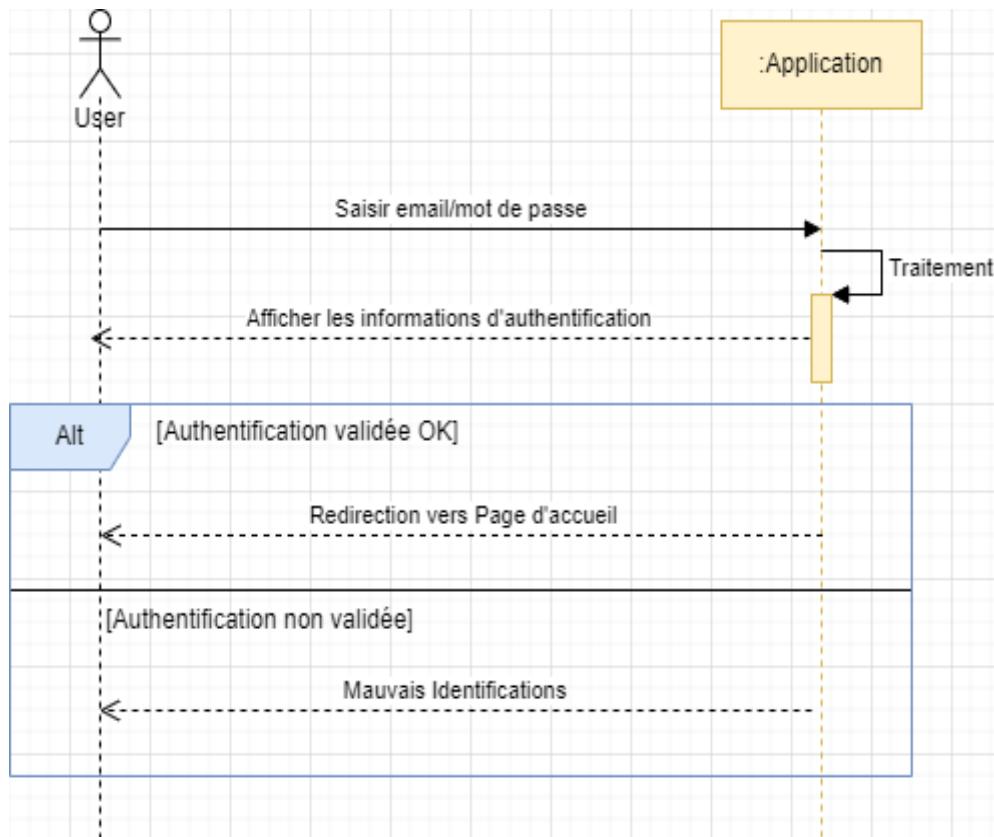


Figure 13: Diagramme de séquence d'authentification

4. Maquettage de l'application :

Nous avons réalisé des maquettes d'interfaces graphiques qui donne une vue approximative sur le système en question, pour cela nous avons utilisé **Balsamiq Mockups** comme outil de créations des interfaces graphiques, il est multiplateforme et gratuit.



Figure 14 : Maquettage interface registration et authentification

La figure ci-dessus représente les interfaces graphiques de la page de registration du client

ainsi que la page d'authentification. Nous avons essayé de créer des interfaces simples et guidés pour une expérience paisible pour tous les utilisateurs.

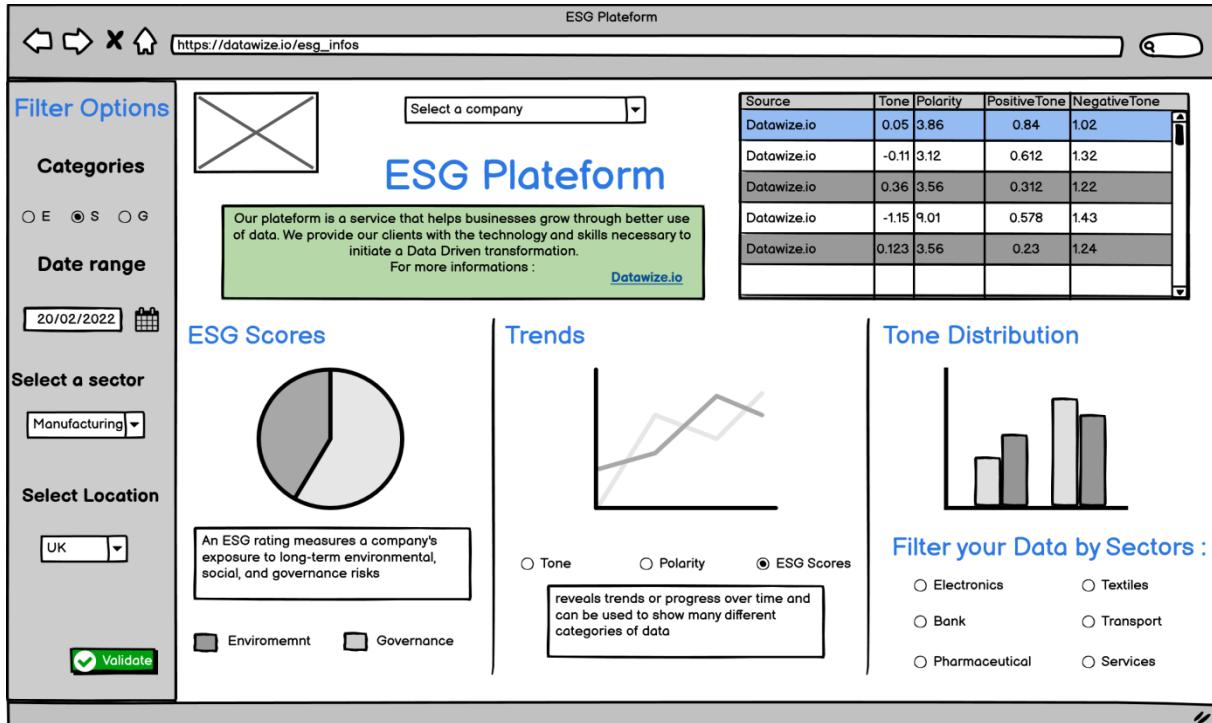


Figure 15: Maquette pour l'interface graphique « Dashboard Client »

La figure ci-dessus représente l'interface graphique où notre client sera redirigé après une authentification validée, dans cette interface notre client peut trouver tous les informations ESG demandés pour une entreprise choisie et les scores ESG ainsi que les différentes sources des données afin de garantir la transparence.

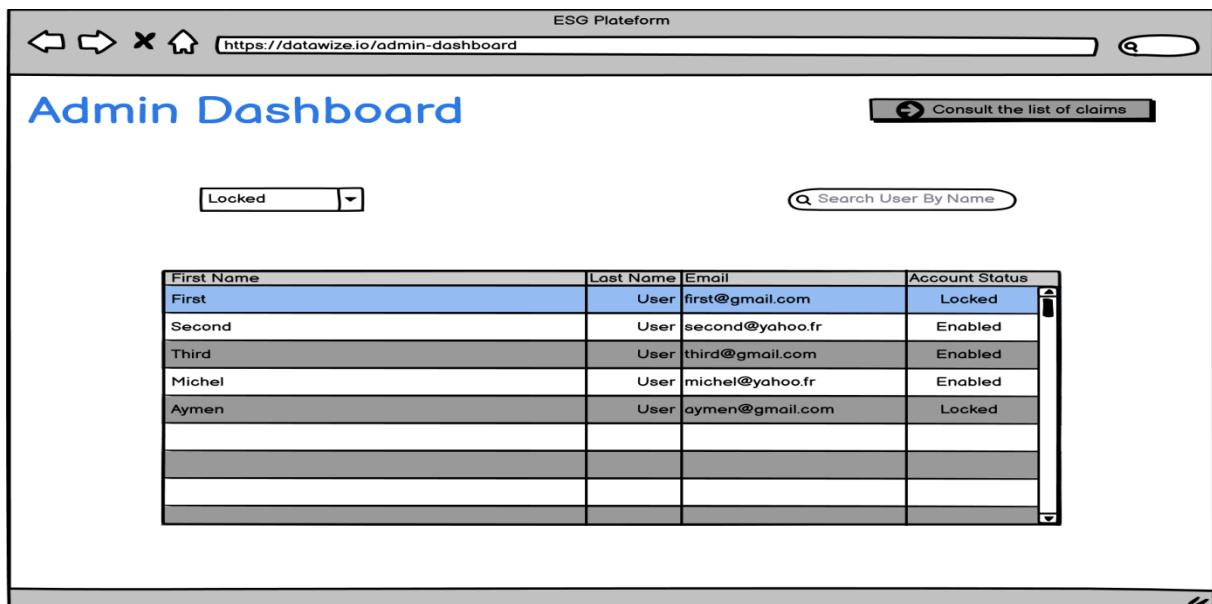


Figure 16 : Maquette pour l'interface graphique « Dashboard Admin »

La figure ci-dessus représente le bord administratif qui est géré par l'administrateur de

notre plateforme. A travers cette interface, l'administrateur peut contrôler tous les utilisateurs qui sont réellement des membres dans la plateforme et il peut passer vers d'autres interfaces pour la gestion des réclamations et d'autres services administratifs.

5. Architecture :

5.1. Architecture Physique :

En général, l'architecture technique d'une application décrit l'ensemble des composants matériels constituant l'application. Dans notre contexte, notre plateforme est constituée d'un client, serveur Tomcat pour l'application du back-end, serveur proxy de NodeJS pour l'application du front-end et un serveur de base de données MongoDB. Nous pouvons visualiser l'architecture physique de notre solution dans la figure ci-dessous avec plus de détails :

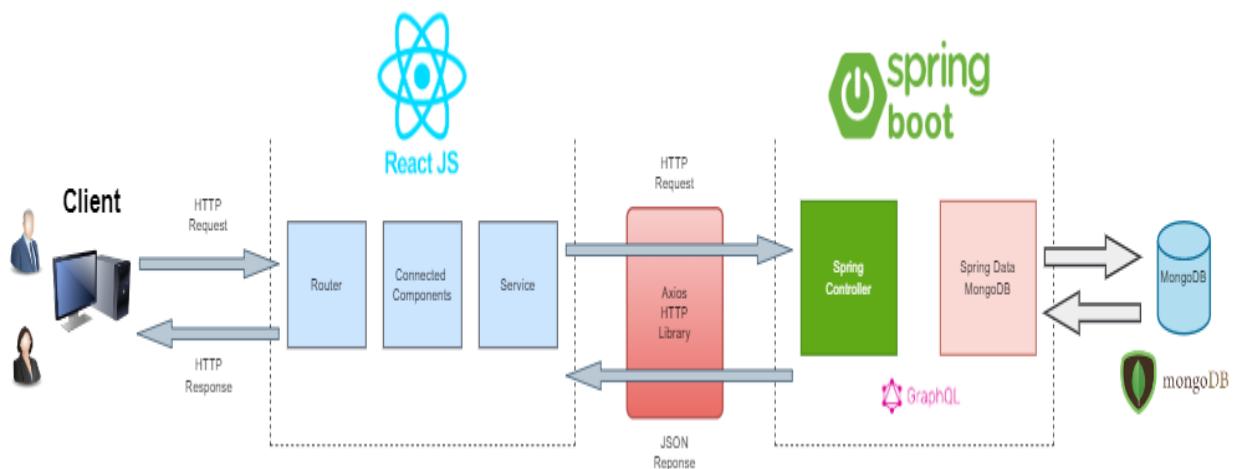


Figure 17 : Architecture physique

Cette architecture est composée par :

- Un niveau client qui représente la couche présentation
- Un niveau serveur d'applicatif qui est responsable à la gestion des données
- Un niveau physique qui représente le serveur de notre base de données qu'il va être le responsable à la persistance des données.

5.2. Architecture Logique :

L'architecture logique est une conception structurelle qui nous permet de visualiser la composition interne de notre solution et surtout de décrire le mécanisme suivi lors de l'interaction entre les différentes applications présentes dans notre architecture (bases de données, applications, systèmes middleware...).

La figure ci-dessous nous permet d'avoir une meilleure visibilité sur les différentes parties principales de notre application.

Nous avons séparés cette architecture en deux couches l'une pour la partie back-end Spring (en Vert), et l'autre couche pour la partie front-end ReactJs (en Violet).

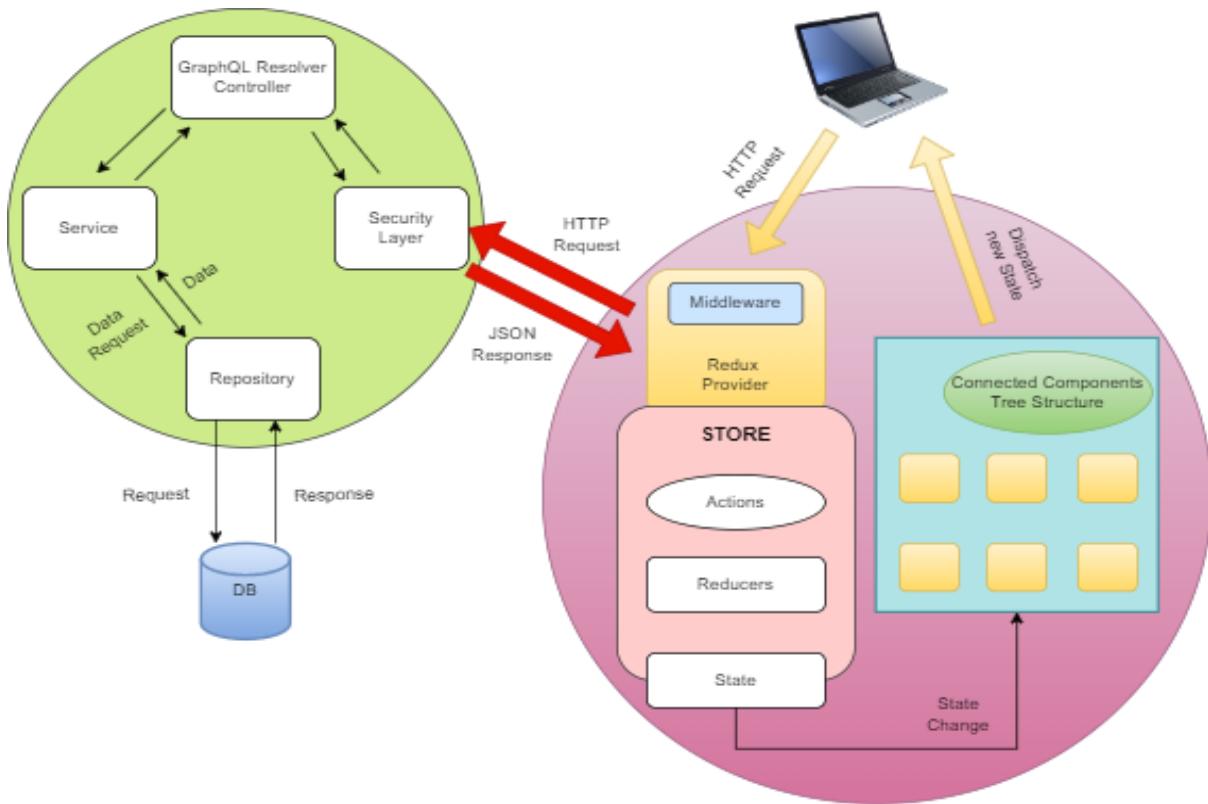


Figure 18 : Architecture Logique

- **Partie backend :** Elle est composée de 4 partie principales qui sont la couche de sécurité, contrôleur, couche métier et la couche de persistance.
- **Couche sécurité :** à l'aide de Spring_Security, nous avons défini une liste spéciale qui contiennent tous les routes qui ne nécessitent pas une authentification pour les accéder, par conséquent tout demande d'accessibilité à un ressource précis nécessite automatiquement l'authentification en fonction du rôle de l'utilisateur, donc cette couche représente le filtre des requêtes http envoyés par les utilisateurs.
- **Le résolveur GraphQL :** C'est le contrôleur dans notre cas, il sert à gérer les requêtes HTTP envoyés et de renvoyer les réponses qui sont réellement les données demandés par le client sous forme JSON. Ce contrôleur un peu spécial par rapport aux contrôleurs
- Rest, dans ce cas, il joue le rôle d'un exécuteur de requêtes gérées par GraphQL afin de limiter la quantité de données renvoyées.
- **Couche métier :** Elle contienne tous les opérations nécessaires pour la gestion des données, nous définissons toutes notre logique métier à l'intérieur des classes de service séparés. La logique code de tous les API déclarés est développés au sein de ces classes services.
- **Couche persistance :** C'est la partie à travers laquelle nous définissons nos différentes entités et les associations entre eux.
- **Partie front-end :** Cette partie est composée par 2 parties principales l'arborescence des composants de la plateforme et le store Redux.
- **Les composants :** C'est l'outil qui va définir la logique d'affichage de notre interface utilisateur, ces composants sont des classes ou des fonctions JavaScript qui prennent des données en entrées et retourne la forme d'affichage à travers la méthode « Render ».

- Redux store : C'est un conteneur de gestion d'état global pour notre application. Tous les composants seront indexés dans ce conteneur et en écoute à tout changement possible affecté par des actions qui peuvent changer le comportement actuel de n'importe quel composant.

6. Conclusion :

Dans ce chapitre, nous avons identifié les besoins fonctionnels et non fonctionnels de notre plateforme, les acteurs présents dans cette plateforme et les diagrammes de cas d'utilisation ainsi que les diagrammes de séquence système afin de préciser l'aspect dynamique de notre plateforme.

Par la suite, nous avons choisi l'architecture physique et logique de notre solution avec quelques exemples des maquettages des interfaces graphiques pour commencer à imaginer le résultat finale de notre travail.

CHAPITRE 3 : ETUDE CONCEPTUELLE

1. Introduction :

Dans ce chapitre, nous énonçons la partie conception de notre application. En première partie, nous allons commencer par une conception prémilitaire, ensuite on va élaborer une conception détaillée en utilisant le diagramme de séquence objet, diagramme de classe et le diagramme d'activité.

2. Conception générale :

Dans cette partie, nous présentons l'étape de conception la plus importante étape dans le cycle de développement de notre solution. Cette conception se base sur la spécification des besoins fonctionnels identifiés afin de livrer une solution optimale.

2.1.Design Pattern (Patron de conception) :

Les patrons de conception (design patterns) sont des solutions classiques à des problèmes récurrents de la conception de logiciels. Chaque patron est une sorte de plan ou de schéma que vous pouvez personnaliser afin de résoudre un problème récurrents dans votre code [1].

Cette organisation du code nous facilite la réutilisation et la maintenance des services utilisés dans notre système. Dons à l'aide de cette structure de code, on peut obtenir un modèle d'application contient des services réutilisables dans autres application facilement.

2.2.Architecture MVC :

L'architecture Modèle/Vue/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue, et le contrôleur ayant chacun un rôle précis dans l'interface [2].

Dans notre cas, l'architecture MVC sera l'architecture suivie lors de développement de notre plateforme afin de garantir une architecture bien organisé avec des parties bien distinctes et fonctionnels d'une façon autonome pour la séparation des données.

Parmi les avantages de l'architecture MVC :

- Gain de temps lors de la maintenance ou l'évolution du site
- Séparation entre les trois parties de l'architecture Modèle, Vue et le contrôleur.

Rôle du modèle :

Celui qui gère les données dans l'application, il permet de récupérer les informations dans la base de données, de les structurer pour qu'elles puissent ensuite être gérables et exploiter par le contrôleur.

Rôle de la vue :

C'est la partie de l'affichage graphique des données pour les présenter à l'utilisateur.

Rôle du contrôleur :

Celui qui gère la logique du code, il représente l'intermédiaire entre le modèle et la vue

à travers l'organisation de tous les évènements déclenchés par l'utilisateur, il synchronise le changement des données au modèle ensuite il avertit la vue qu'il y a des modifications au niveau de l'affichage pour que celle-ci se mette à jour.

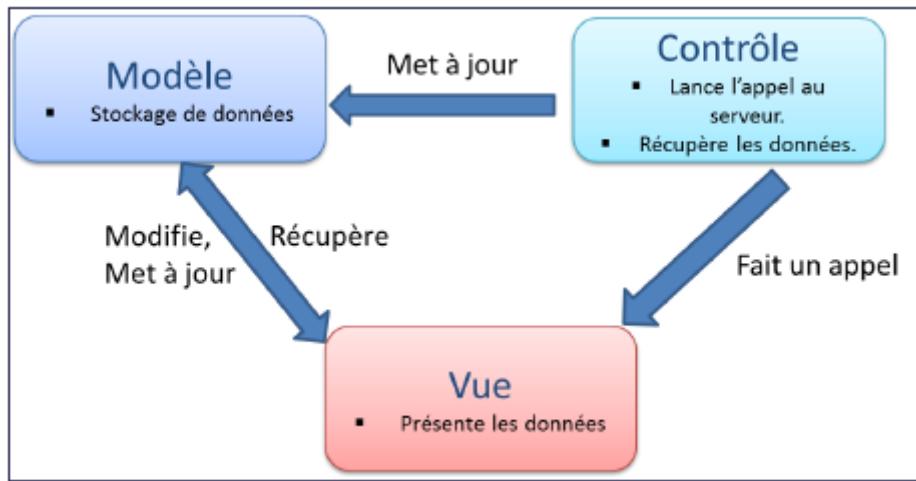


Figure 19 : Architecture MVC

L'architecture MVC est une solution efficace et importante pour faciliter la réutilisation de code dans d'autres projets, ce qui présente un gain de temps remarquable pour les développeurs.

L'indépendance entre les trois couches facilite la tâche de maintenance en cas où il y a des problèmes aussi la facilité au niveau de développement des tests unitaires.

3. Conception détaillée :

Pour montrer la façon de fonctionnement de notre système, nous allons migrer vers les diagrammes dynamiques qui nous aident à illustrer les actions effectués par les utilisateurs en temps réel comme les diagrammes de séquence objets et les diagrammes d'activités.

Vue le grand nombre d'actions dans le système, nous nous contenterons de présenter quelques cas d'utilisation de l'application.

3.1. Vue statique :

Avant de passer vers l'aspect dynamique dans notre conception, nous allons essayer de déterminer les différents entités présentes dans notre application ce qui nous aide à organiser notre code.

❖ Modèle de données :

Le diagramme de classe est une vue statique, car il ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classe modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application [3].

A l'aide de cette modélisation, on peut regrouper les données dans des classes qui représentent les entités nécessaires pour assurer la partie de la persistance des informations, et le plus important dans cette modélisation c'est l'association entre ces différents classes

« Mapping » sachant que chaque classe représente une projection d'une table dans notre base de données.

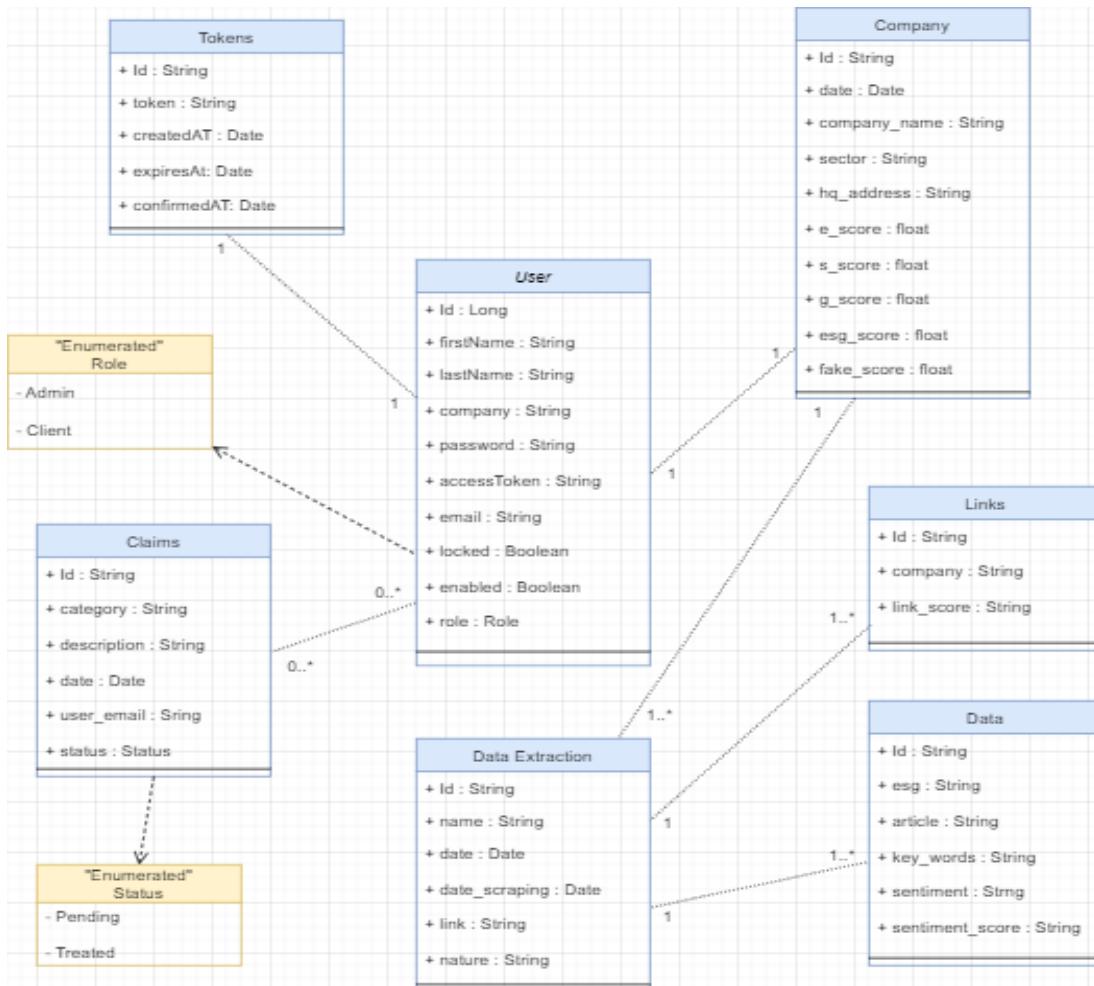


Figure 20 : Diagramme de classe global

Description des classes :

- La classe User : Cette classe représente les données d'un utilisateur inscrit dans notre plateforme que ce soit un client ou bien un administrateur.
- La classe Rôle : Cette classe représente tous les rôles possibles des utilisateurs dans notre application.
- La classe company : Cette classe représente les données propres de tous les entreprises stockés dans la base de données et qui seront présentés sous forme d'une liste regroupant tous les noms de ces entreprises.
- La classe Data-Extraction : Cette classe est une présentation des données extraites et collectées depuis plusieurs sources par notre équipe des Data Analystes.
- Les classes Data et Links : ce sont le noyau de la structure de la classe Data-Extraction, autrement dit ces deux classes sont des rapports ESG détaillées et qui seront affichés en cas d'une demande envoyée par le client.
- La classe Token : Cette classe regroupe tous les jetons assignés aux clients de

la plateforme « ESG Smarter » lors de l'inscription.

- La classe Claims : Cette classe représente les réclamations et les rapports de bug envoyés par le client vers le service administratif.

3.2.Vue Dynamique :

Dans cette partie, nous allons présenter l'aspect dynamique de notre application. En premier lieu, nous présentons des diagrammes des séquences afin de mettre en œuvre les différentes couches de l'application dans quelques cas. En deuxième lieu nous allons exposer les diagrammes d'activités pour mettre en œuvre quelques scenarios de développement.

3.2.1. Diagramme de séquence objet :

Afin de montrer l'aspect dynamique de notre application, nous allons présenter quelques exemples des Opérations effectués par l'utilisateur en temps réel.

Ce diagramme est une présentation des interactions en temps réel entre l'utilisateur et les différents objets présentes dans notre application et les messages qui échangent entre eux pour exercer une opération précise avant la fin de la ligne de vie.

❖ Diagramme de séquence objets « Consulter les scores ESG » :

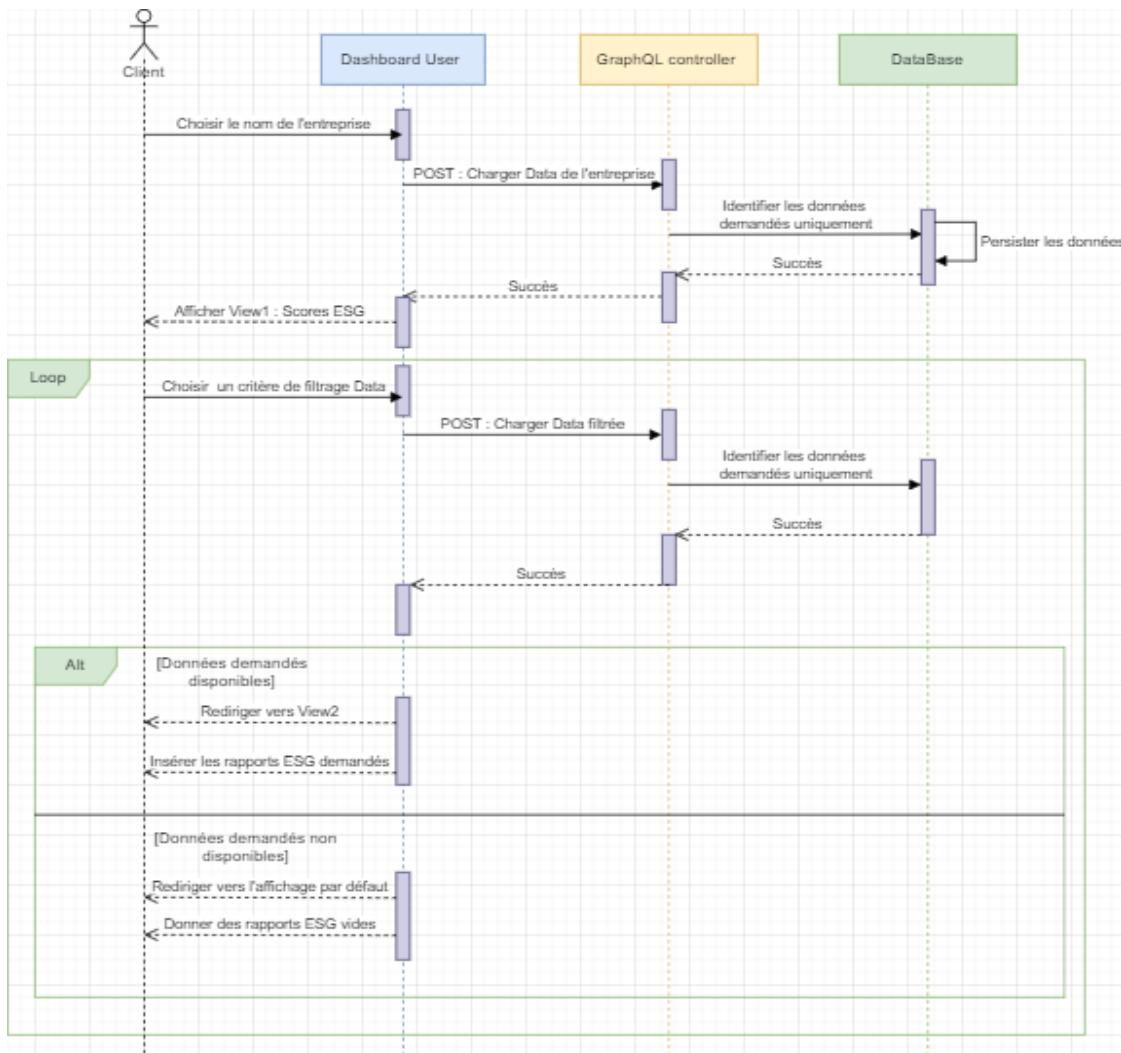


Figure 21 : Diagramme de séquence objets « Consultation des scores ESG »

Le diagramme de séquence objets ci-dessus décrit le scenario de la consultation des scores et les rapports ESG par le client à travers son interface spécifique et en temps réel.

En premier lieu, le client accède à l'interface approprié à travers lequel il peut découvrir les scores ESG d'une entreprise choisie et il peut aussi naviguer en profondeur de ces informations à travers des composants graphiques destinés au filtrage des données selon des critères différents comme le secteur d'activité de l'entreprise, la localisation...

❖ Diagramme de séquence objets « Gestion des utilisateurs » :

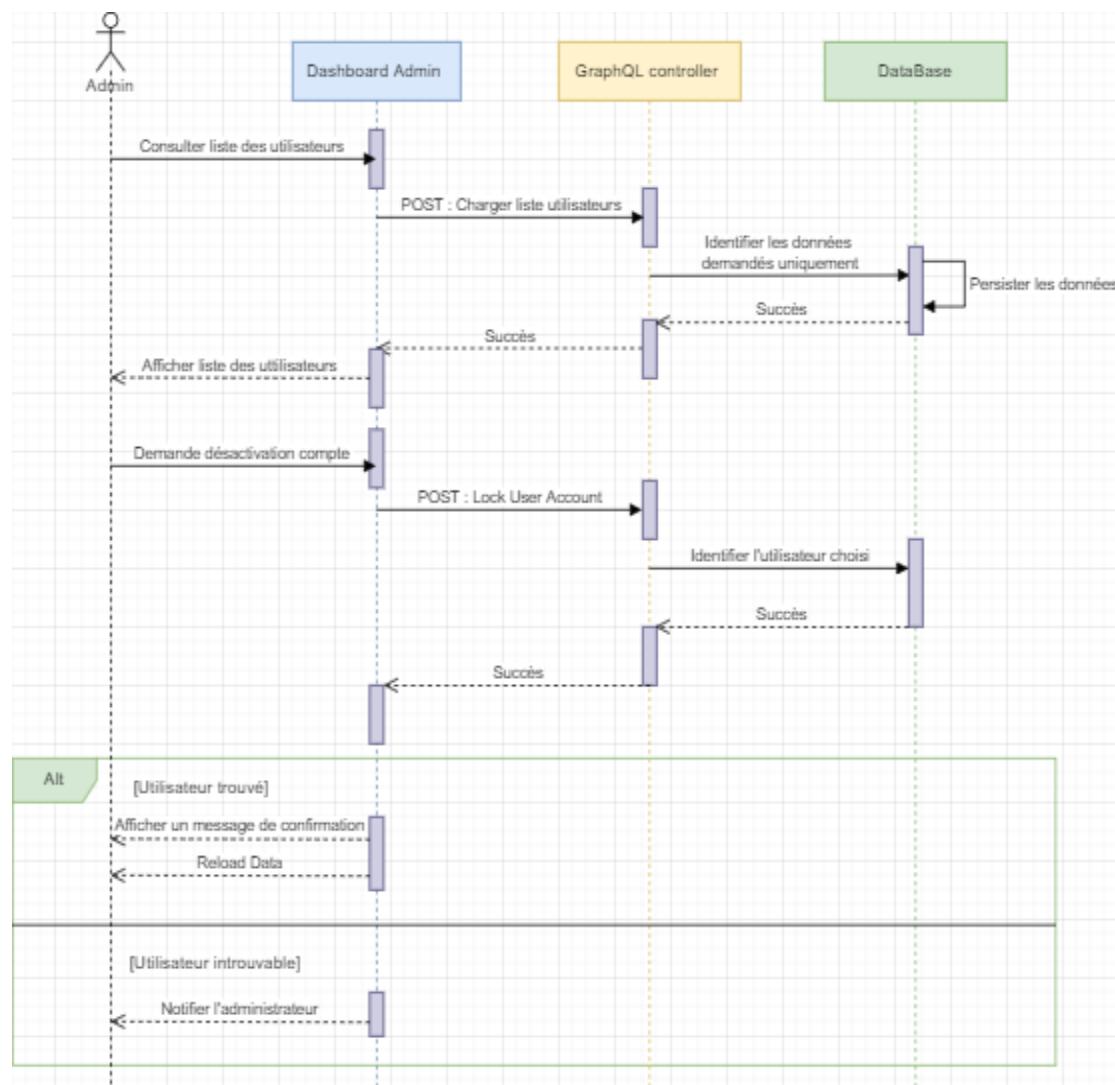


Figure 22 : Diagramme de séquence objets « Gestion des utilisateurs »

Le diagramme de séquence objets ci-dessus décrit le scenario de la gestion des utilisateurs par l'administrateur en temps réel.

L'administrateur ou le service administratif a son propre interface graphique qui nécessite des priviléges bien précis et exigeants pour l'accéder, l'administrateur peut consulter la liste de tous les utilisateurs inscrits dans l'application et il a les habilitations nécessaires pour désactiver ou bien supprimer un utilisateur en cas de besoin.

3.2.2. Diagramme d'activité :

Le diagramme d'activité nous aide à présenter d'une façon dynamique les mécanismes complexes dans notre système.

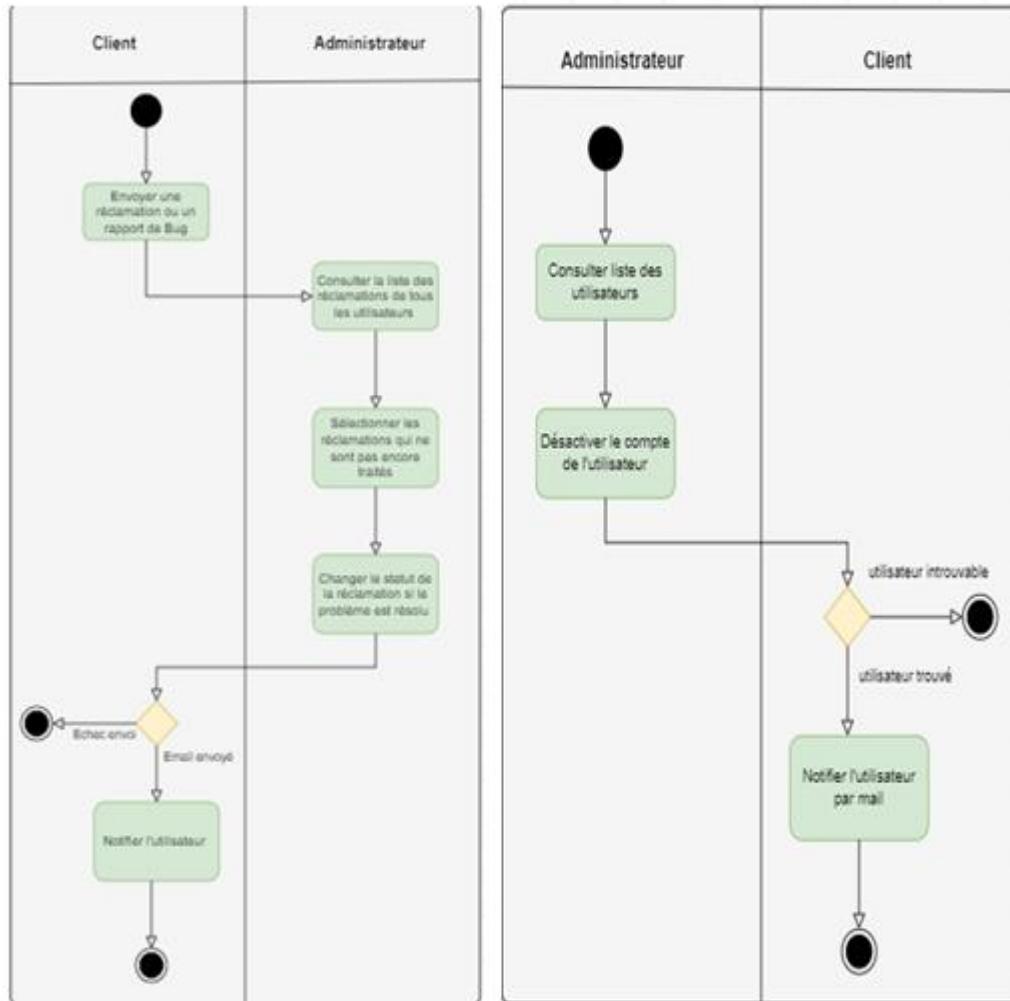


Figure 23: Diagrammes d'activité « Gestion réclamation » et « Gestion Users »

Les deux diagrammes ci-dessus visent à décrire les étapes relatives à la procédure de la gestion des utilisateurs par le service administratif (figure à gauche) et la gestion des réclamations (figure à droite).

Diagramme d'activité « Gestion des utilisateurs » : L'administrateur consulte la liste des utilisateurs, il a le droit de désactiver le compte d'un utilisateur choisi ou de le supprimer si le compte de l'utilisateur n'était pas activé réellement. En cas de la désactivation, le client sera notifié à travers un email.

Diagramme d'activité « Gestion des réclamations » : L'administrateur consulte la liste des réclamations et les rapports des Bug, après la résolution du problème, l'administrateur va changer les statuts de la réclamation et notifier le propriétaire de cette réclamation par un email.

4. Conclusion :

Durant ce chapitre, nous avons présenté les deux aspects statique et dynamique de notre application respectivement à travers le diagramme de classe général et les diagrammes de séquence et les diagrammes d'activités.

CHAPITRE 4 : REALISATION

1. Introduction :

Après avoir achevé la partie conception de l'application, dans cette partie nous allons présenter la phase de la réalisation de notre plateforme. Pour cela, nous allons présenter quelques interfaces graphiques de notre application afin de montrer le fonctionnement de nos services avec la présentation des différents outils de développement utilisés durant la période de la réalisation de notre application.

2. Diagramme de déploiement :

Les diagrammes de déploiement sont constitués de plusieurs formes UML. Les boîtes en trois dimensions. Appelées nœuds, représentent les composants du système, qu'ils soient logiciels ou matériels. Les lignes entre les nœuds indiquent les relations et les petites formes à l'intérieur des boîtes représentent les artefacts logiciels qui sont déployés [4].

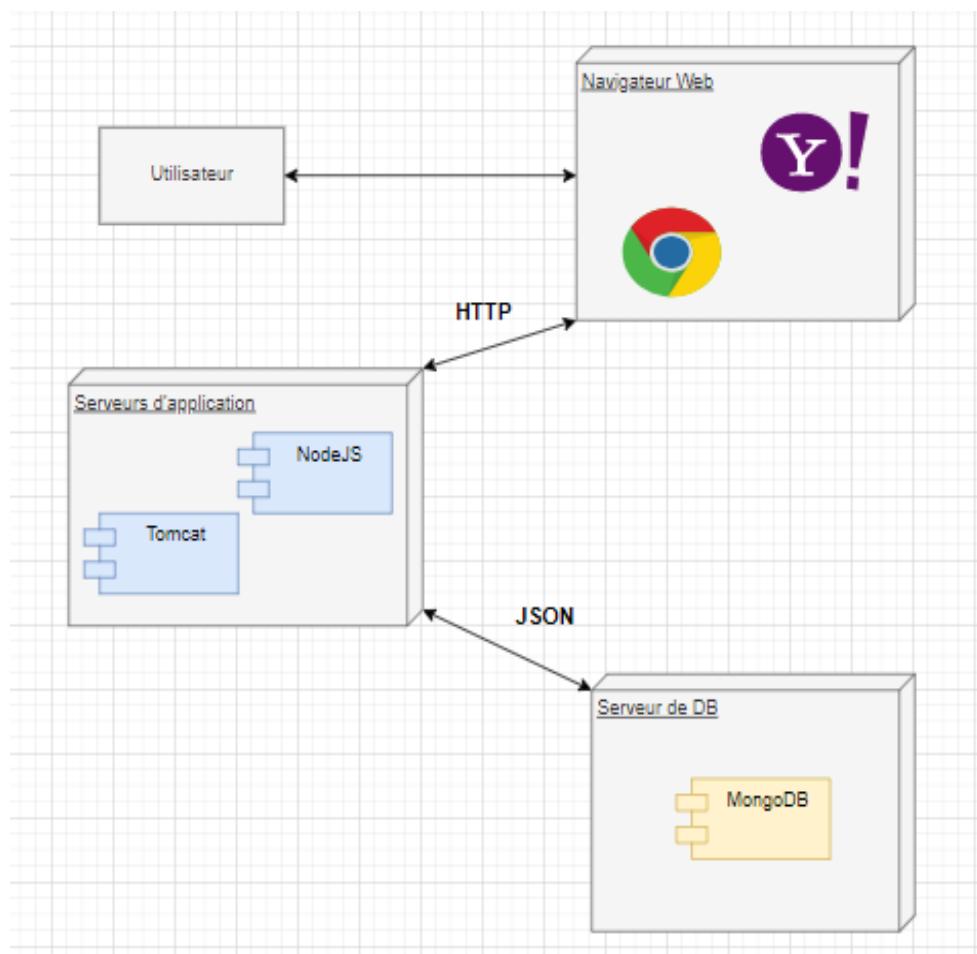


Figure 24: Diagramme de déploiement

3. Environnement Logiciel :

Dans cette partie, nous allons présenter les différents outils utilisés durant le développement de ce projet.

3.1.Outils de développement :

Spring Tool Suit (STS) :

Est un IDE étendu pour Eclipse. Il se spécialise dans le développement des applications Spring.



Figure 25: Logo STS

Visual Studio Code (VScode):

Est un éditeur de code développé par Microsoft, il nous permet de développer des projets avec plusieurs langages de programmation comme Javascript, TypeScript...



Figure 26 : Logo VS code

MongoDB :

Est une base de données NoSQL orientée document, nous avons choisi de travailler avec cette base vu la quantité massif des données qu'on va utiliser et surtout qu'elle ne nécessite pas une structure prédéfinis des données.



Figure 27: Logo MongoDB

ReactJS :

Est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création des applications Web monopage, via la création de composants dépendant d'un état et générant une page HTML à chaque changement d'état [5].

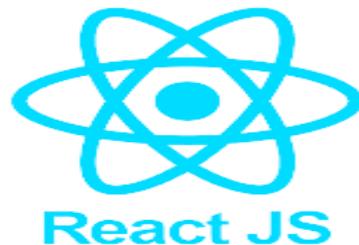


Figure 28 : Logo ReactJS

GraphQL :

Est un langage de requête et un environnement d'exécution côté serveur pour les API qui s'attache à fournir aux clients uniquement les données qu'ils ont demandés et rien de plus. GraphQL est conçu pour mettre à la disposition des développeurs des API rapides, flexibles et faciles à utiliser. Il est utilisé à la place des API Rest, il permet aux utilisateurs de créer des requêtes qui extraient les données de plusieurs sources à l'aide d'un seul appel d'API [6].



Figure 29 : Logo GraphQL

Postman :

Afin de vérifier le bon fonctionnement des API développés et avant de les implémenter dans notre code source, nous avons utilisés Postman comme un outil de test des différents services.



Figure 30 : Logo Postman

ChartJS :

Est une bibliothèque JavaScript qui nous permet de faire la visualisation de données graphiques et interactive. Elle lie nos données à un objet et nous les affiche dans un graphiques afin de les rendre plus facilement compréhensibles et analysables [7].



Figure 31 : Logo ChartJS

3.2.Outils d'intégration et collaboration :

GitHub :

C'était notre outil de stocker et partager les différentes versions de notre projet que soit entre les différents membres de l'équipe ou bien avec nos tuteurs afin de contrôler et suivre les changements au niveau du code étape par étape.



Figure 32 : Logo GitHub

Jira :

Est une plateforme multifonction qui vise à faciliter la gestion de projet en aidant à suivre les tâches, identifier les blocages et partager des informations entre les membres d'une équipe [8].



Figure 33 : Logo Jira

Slack :

Est une plateforme de communication collaborative, C'était l'outil de communication principal entre tous les membres de l'équipe DataWize et pour la gestion de notre projet.



Figure 34 : Logo Slack

3.3.Outils de conception et Design :

Draw.io :

Est un logiciel open source et gratuit destiné pour la réalisation des dessins graphiques multiplateformes, nous avons utilisé Draw.io durant la phase de conception pour la réalisation des différents type de diagrammes.



draw.io

Figure 35 : Logo draw.io

Balsamiq :

Est un outil de création des prototypes et des maquettes qui facilite la tâche aux Webdesigners pour créer des interfaces graphiques d'une manière très simple.



Figure 36 : Logo Balsamiq

4. Interfaces de la plateforme :

Pour pouvoir accéder à l'application, l'utilisateur aura besoin de créer un compte avant de se rediriger vers la page d'authentification pour vérifier son identité.

Hack your ESG investing success

Take advantage of your data, DataWize supports you in exploring new avenues for growth and optimization...

About Us

Welcome to ESG Smarter, let's get started!

Please use the form to register.
If you are a member, please login.

First Name

Last Name

Email

Password

I have read and accept the [terms and conditions](#).

Signup

Figure 37 : Interface pour la registration utilisateur

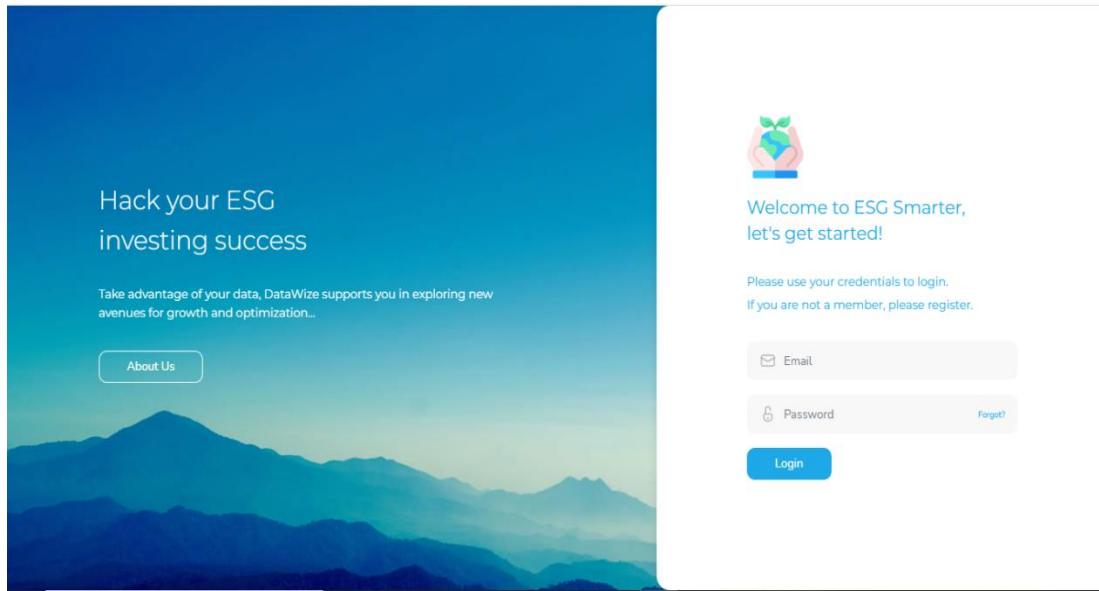


Figure 38 : Interface pour s'authentifier

Après la registration, l'utilisateur va recevoir un email envoyé par l'administration de la plateforme qui contient un lien de confirmation de registration pour que le client peut activer son compte crée et valider son registration.

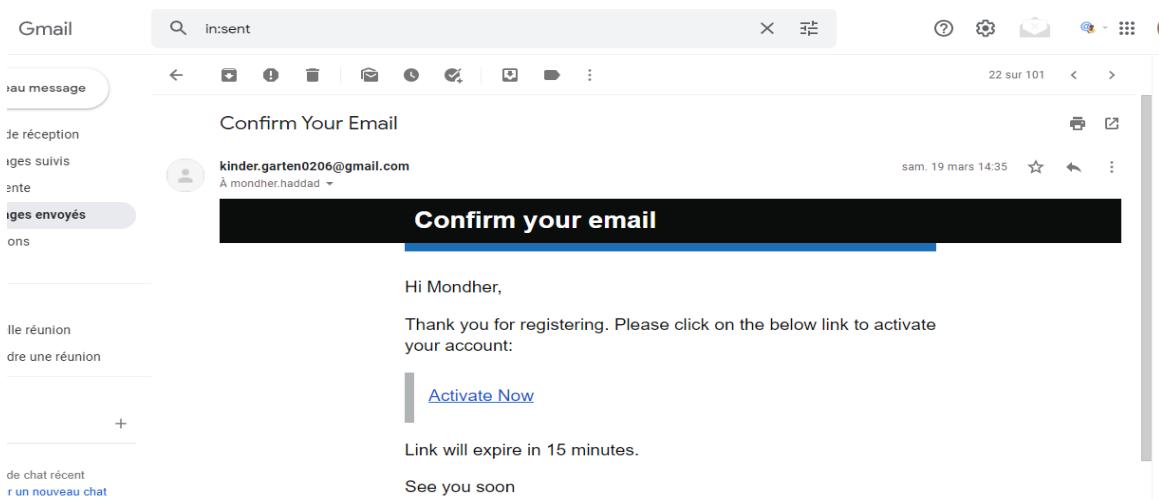


Figure 39 : Email de confirmation registration

➤ .Consulter les scores ESG :

Apres une opération d'authentification valide, l'utilisateur peut accéder à son tableau de bord spécifique pour l'affichage des scores et les rapports ESG détaillées d'une entreprise choisie.

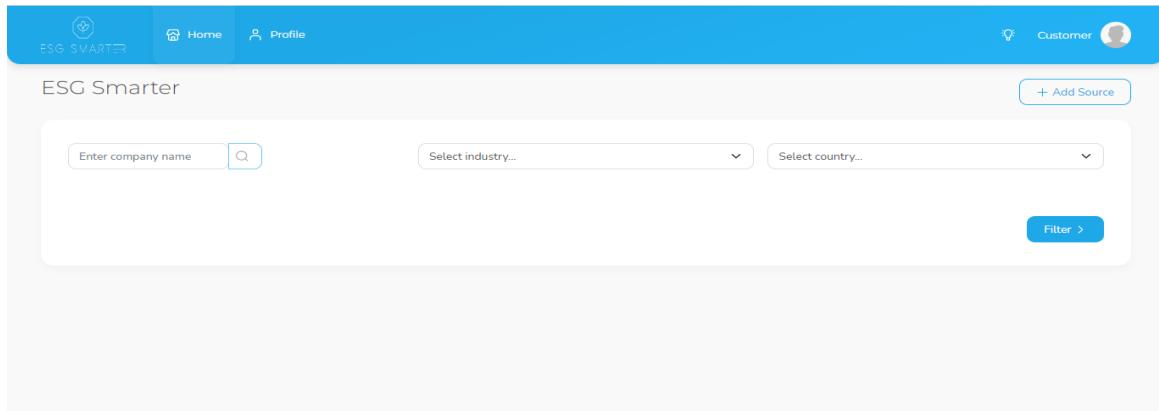


Figure 40 : Tableau de bord de l'utilisateur par défaut

Dans ce niveau, le client va sélectionner une entreprise qu'il veut consulter leur rapports et leur score ESG, ou bien il a le droit de filtrer la totalité des données selon des critères de filtrage bien définis comme le secteur d'activité ou la localisation afin d'obtenir un ensemble des entreprises qui correspondent aux critères choisies.

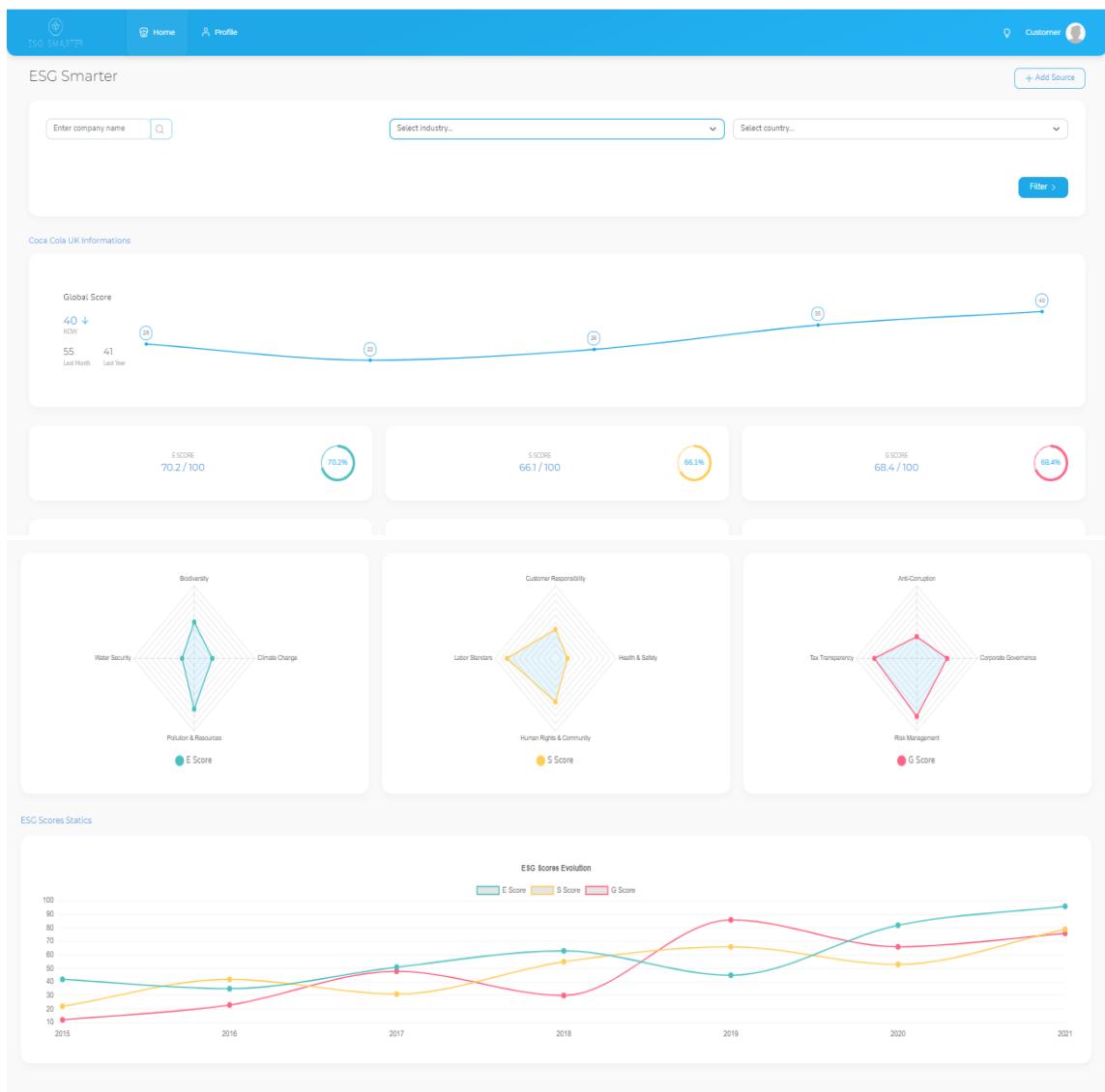


Figure 41 : Interface d'affichage des données ESG pour le client

La figure ci-dessous représente le résultat de filtrage des données par le client selon le secteur d'activité des entreprises, on peut remarquer ici qu'il y a une seule entreprise correspondante au secteur choisi.

The screenshot shows the ESG Smarter platform's interface. At the top, there is a navigation bar with 'Home' and 'Profile' buttons, and a 'Customer' profile icon. Below the navigation bar, the title 'ESG Smarter' is displayed. On the left, there is a sidebar with a search bar labeled 'Enter company name' and a magnifying glass icon. To the right of the sidebar are two dropdown menus: 'Select industry...' and 'Select country...'. Below these dropdowns is a blue button labeled 'Filter >'. Underneath the filter section, the heading 'ESG Details' is shown, followed by a table with the following data:

Company	Industry	Country	E	S	G	Esg Score	Fake Score
PwC	Audit	UK	76.3	59.9	66.8	70.1	67.1

Figure 42 : Affichage des informations ESG filtrés par secteur

This screenshot shows the same ESG Smarter platform interface as Figure 42. The 'Select industry...' and 'Select country...' dropdowns have been used to filter the results. The table now displays multiple companies located in the UK:

Company	Industry	Country	E	S	G	Esg Score	Fake Score
Accenture UK	Technology	UK	30.16	73.56	41.22	68.3	0.41
Total Energy	Oil	UK	47.8	41.1	56.3	47.3	0.611
Coca Cola UK	Manufacturing	UK	70.2	66.1	68.4	69	51.9
PwC	Audit	UK	76.3	59.9	66.8	70.1	67.1

Figure 43 : Affichage des informations ESG filtrés selon la localisation de l'entreprise

Dans la figure ci-dessus, le client a filtré les données disponibles de toutes les entreprises selon la localisation, ce tableau contient tous les entreprises qui se trouvent à « UK ».

This screenshot shows the 'User Profile' section of the ESG Smarter platform. At the top, there is a navigation bar with 'Home' and 'Profile' buttons, and a 'Customer' profile icon. The main area is titled 'User Profile'. On the left, there is a sidebar with a user icon and the name 'Mondher Haddad' along with the email address 'hunkingdom.store@gmail.com'. Below this, there are four navigation links: 'Overview', 'Projects', 'Permissions', and 'About'. On the right, there is a form titled 'Contact Administrator' with fields for 'Category' (a dropdown menu with 'Select category...') and 'Description' (a text input field with placeholder text 'Describe your problem here'). A blue 'Send' button is located at the bottom of the form.

Figure 44 : Profil du client

L'interface ci-dessus représente l'interface graphique d'un profil d'un client inscrit dans notre application. Le client peut contacter l'administration de la plateforme à travers l'envoi des réclamations et des rapports de bug en cas de besoin.

➤ Session d'administration :

Dans cette partie, nous allons présenter les interfaces d'administration de notre plateforme « ESG Smarter » à travers lesquelles l'administrateur peut consulter la liste des utilisateurs inscrits ainsi que la gestion des réclamations et les rapports de bug reçus.

First Name	Last Name	Email	Account Status	Actions
Mondher	Haddad	huntkingdom.store@gmail.com	Activated	<button>Lock</button>
Omar	Yaacoubi	omar.yaacoubi@yahoo.fr	Locked	<button>Lock</button>
Mohsen	Haddad	mondher.haddad2017@gmail.com	Locked	<button>Lock</button>

Figure 45 : Interface de la gestion des utilisateurs

La figure ci-dessus représente la page d'accueil pour l'administrateur, à l'aide de cette interface l'administrateur peut consulter la liste des utilisateurs inscrits et il a les priviléges nécessaires pour désactiver le compte d'un utilisateur choisi en cas de besoin.

Category	Description	Date	User Email	Claim Status	Actions
Claiming	First Test	2022-05-31	huntkingdom.store@gmail.com	PENDING	<button>Classify</button>
Bug Report	Second Test	2022-05-26	huntkingdom.store@gmail.com	DONE	<button>Classify</button>
Bug Report	scoring charts are not tables	2022-05-21	mondher.haddad2017@gmail.com	DONE	<button>Classify</button>
Claiming	filter bar is not flexible	2022-05-18	mondher.haddad2017@gmail.com	PENDING	<button>Classify</button>

Figure 46 : Interface de la gestion des réclamations et les rapports de bug

Cette interface représente le gestionnaire des réclamations et les rapports des bugs envoyés par les utilisateurs. A l'aide de cette interface, l'administrateur peut contrôler les problèmes et superviser la résolution des bugs ou des complications rencontrés par le client.

Après le traitement du problème désigné par le client, l'administrateur va classifier la réclamation comme un problème résolu et en même temps une notification sera envoyée vers le client par email pour l'informer de l'évolution de son réclamation.

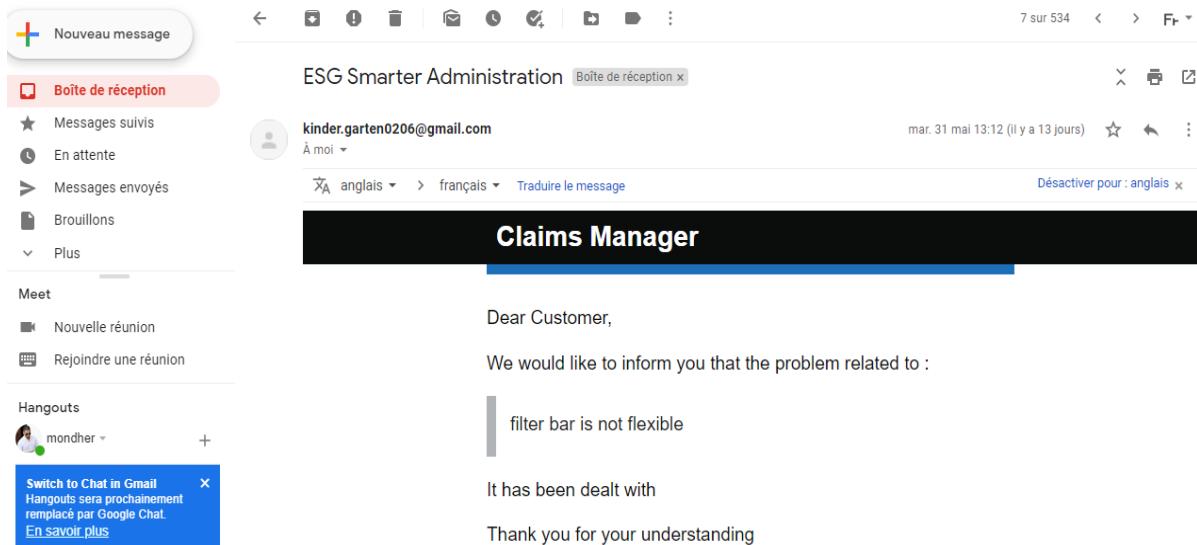


Figure 47 : Notification envoyé pour le client concernant son réclamation

5. Conclusion :

Dans ce chapitre, nous avons présenté l'environnement de travail, exposé les différents outils de développement utilisés pour la mise en œuvre de notre projet, nous avons décrit les principales fonctionnalités implémentées en se basant sur des imprimés écrans.

Conclusion générale et perspectives

Durant la période de stage, j'ai eu l'occasion des travailler sur des technologies qui sont considérés comme une tendance dans le domaine de développement Web, comme la librairie React, l'exécuteur des Api GraphQL, Framework Spring Boot...

Mais le plus important que la réalisation de ce projet c'était un vrai défi pour moi, j'ai suivi la réalisation de ce projet dès la partie de conception jusqu'à le développement l'implémentation de la solution, c'était une expérience très enrichissante pour moi à travers laquelle j'ai eu l'opportunité pour apprendre beaucoup des nouvelles technologies et des compétences interpersonnelle.

D'une autre côté, le contexte de la réalisation de notre solution était une occasion pour découvrir le monde de l'intelligence économique et précisément les critères ESG, vu que pour prédire la performance et la croissance à long terme d'une entreprise, les investisseurs ont commencés à considérer ses stratégies environnementales, sociales et gouvernementales. La performance d'une entreprise devient de plus en plus dépendante ses opérations non financières. Par conséquent, la demande de données ESG a augmenté parmi les fournisseurs des données, les investisseurs potentiels et les entreprises, ce qui confirme que notre plateforme « ESG Smarter » est une vrai solution efficace qui répondre aux besoins de ses investisseurs dans nos jours.

Aussi vu le grand volume de données et les délais de livraison de la première version de notre plateforme, nous n'avons eu pas l'occasion pour améliorer et intégrer des différent services dans notre plateforme, mais dans le futur, on peut travailler sur ça, l'application que nous avons développé pourrait être enrichie par des fonctionnalités avancées et intéressantes pour notre client comme l'extraction des scores et données ESG sous forme des rapports PDF, l'ajout des sources personnalisés par le client...

Résumé

Webographie

- [1] : <https://refactoring.guru/fr/design-patterns>
- [2] : <https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/Cours/Swing/mvc.html>
- [3] : <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes>
- [4] : <https://www.lucidchart.com/pages/fr/diagramme-de-deploiement-uml>
- [5] : <https://fr.wikipedia.org/wiki/React>
- [6] : <https://www.redhat.com/fr/topics/api/what-is-graphql>
- [7] : <https://www.numendo.com/blog/front/d3-js-ou-chart-js-comment-choisir-sa-bibliotheque-graphique/>
- [8] : <https://www.cadrememploi.fr/editorial/conseils/conseils-carriere/detail/article/jose-pas-demandez-mais-quest-ce-que-jira.html>



ESPRIT SCHOOL OF ENGINEERING

www.esprit.tn - E-mail : contact@esprit.tn

Siège Social : 18 rue de l'Usine - Charguia II - 2035 - Tél. : +216 71 941 541 - Fax. : +216 71 941 889

Annexe : 1-2 rue André Ampère - 2083 - Pôle Technologique - El Ghazala - Tél +216 70 250 000 - Fax +216 70 685454