# DEDICACE

**A mes très chers parents et tous les membres de ma famille.**

Ce document est le fruit de tous les sacrifices que vous avez toujours et encore consentis afin de faire de moi la femme que je suis.

Trouvez dans ce travail un témoignage de mon profond amour et une éternelle reconnaissance, car sans vous je n’aurais certainement pas eu la chance et l’opportunité d’y arriver.

Je vous dédie ce travail…

Fifen Maimouna…

# REMERCIEMENTS

Avant de présenter ce travail, je tiens à exprimer ma grande reconnaissance envers les personnes qui m’ont aidé à le réaliser et qui est l’aboutissement de mes études de licence en Informatique orienté à la gestion. Qu’ils trouvent ici collectivement et individuellement l’expression de toute ma gratitude.

Je rends d’abord grâce à ALLAH de m’avoir comblé de ses grâces et m’avoir donné la santé et la force pour accomplir ce travail.

Je remercie sincèrement l’entreprise « SK-Energie » et toute son équipe pour le cadre de travail exceptionnel et l’accompagnement dont j’ai bénéficié pour cette période de réalisation de mon projet de fin d’études. Mes vifs remerciements vont particulièrement à l’endroit de Mr. x, pour m'avoir donné l’honneur d'intégrer son équipe, sa motivation et sa confiance en mes capacités, de mon encadrant industriel Mr. x pour son professionnalisme et son extrême compréhension. J’ai appris énormément en complémentarité avec ma formation académique.

Mes remerciements s’adressent aussi à mon enseignant et encadrant académique Mme. SALMA KSIBI pour l’expérience et les conseils enrichissants mis à ma disposition et qui m’ont servi à améliorer mon travail et à la rédaction de ce rapport.

Je tiens également à remercier le corps enseignant et administratif de l’Institut Polytechnique des Sciences Avancées de Sfax qui est sans doute le meilleur institut de formation dans la ville.

A tous ceux qui, de près ou de loin m’ont apporté leur soutien sur tous les plans durant tout mon cycle de licence et mon séjour en Tunisie : je pense ici à M. FIFEN Inoussa, son épouse et ses enfants, … (Ajouter nom et communauté) …, mes amis, ma famille par affiliation tant au Cameroun qu’en Tunisie.

Que les membres de jury trouvent ici, l’expression de mes remerciements pour l’honneur qu’ils me font en acceptant de juger ce travail.

# INTRODUCTION GENERALE

Dans le panorama complexe et en perpétuelle mutation de l'industrie énergétique, les entreprises doivent constamment s'adapter aux exigences techniques et opérationnelles pour demeurer compétitives. C'est dans cette optique de répondre à ces impératifs que ce rapport de stage de Projet de Fin d'Études (PFE) s'attelle à explorer et à développer une solution novatrice : une plateforme desktop dédiée à la gestion intégrale des opérations au sein de SKS-Energy. Cette société, éminente dans le domaine du photovoltaïque et du pompage, fait face à des défis spécifiques liés à la gestion de ses opérations, nécessitant une approche technologique avancée et sur mesure.

À travers cette initiative, nous nous engageons à explorer les subtilités et les nuances de la gestion opérationnelle dans un contexte énergétique particulièrement exigeant. Nous aspirons à concevoir une plateforme qui ne se contente pas seulement de répondre aux besoins immédiats de SKS-Energy, mais qui soit également adaptable et évolutive pour s'aligner avec les tendances et les innovations futures de l'industrie.

Ce rapport documentera minutieusement chaque étape du processus, depuis la phase initiale de recherche et d'analyse préliminaire jusqu'à la réalisation concrète de la plateforme et son intégration au sein des opérations de SKS-Energy. Nous nous efforcerons de mettre en évidence l'importance cruciale de cette solution sur mesure dans l'amélioration de l'efficacité opérationnelle, la réduction des coûts et la maximisation des performances dans le secteur du photovoltaïque et du pompage.

Dans le chapitre 1, intitulé étude préalable, nous poserons les bases en mettant en exergue les enjeux essentiels liés à la gestion opérationnelle dans le contexte spécifique de l'industrie photovoltaïque et du pompage. À travers cette étude initiale, nous visons à comprendre les exigences uniques de SKS-Energy et à établir des objectifs clairs pour le projet.

Dans le chapitre 2, nous ferons l’étude de l'existant et critique, nous entreprenons une analyse approfondie des systèmes et pratiques actuellement en place chez SKS-Energy. Cette évaluation critique vise à identifier les lacunes et les opportunités d'amélioration dans le cadre de la gestion des opérations, fournissant ainsi une base solide pour la conception de la nouvelle plateforme.

Le chapitre 3, sera l’étude Conceptuel, qui constitue le cœur du projet. Nous détaillerons ici le processus de conception de la plateforme, de la spécification des exigences fonctionnelles à la modélisation de l'architecture logicielle. Nous mettrons également en lumière les choix conceptuels et les stratégies adoptées pour répondre aux besoins spécifiques de SKS-Energy.

Dans le chapitre 4, consacré aux technologies et outils utilisés pour la réalisation, nous aborderons la mise en œuvre pratique de la plateforme, en nous concentrant sur les technologies et outils sélectionnés. Des exemples concrets et des démonstrations seront présentés pour illustrer la manière dont les choix technologiques ont été alignés avec les objectifs du projet.

Enfin, dans le chapitre 5, nous présenterons les Interfaces Homme-Machine, se penchera sur l'aspect ergonomique et fonctionnel de la plateforme, en mettant l'accent sur la conception d'une interface utilisateur intuitive et conviviale.

# CHAPITRE 1 : ETUDE PREALABLE

# Présentation du projet

Dans le contexte spécifique de l'industrie photovoltaïque et du pompage, les enjeux liés à la gestion opérationnelle revêtent une importance capitale pour les entreprises comme SKS-Energy. Tout d'abord, ces industries opèrent dans des environnements complexes et dynamiques, caractérisés par des exigences techniques rigoureuses et des contraintes environnementales spécifiques. La gestion efficace des opérations est donc essentielle pour assurer la fiabilité et la performance des installations photovoltaïques et des systèmes de pompage, tout en garantissant une utilisation optimale des ressources et une minimisation des coûts. De plus, la nature intermittente de l'énergie solaire et les variations saisonnières dans la demande d'eau pour l'irrigation agricole ajoutent une dimension supplémentaire à la complexité de la gestion opérationnelle. Ainsi, les entreprises telles que SKS-Energy sont confrontées au défi de maintenir une flexibilité opérationnelle tout en assurant une efficacité maximale dans leurs activités, ce qui nécessite des solutions innovantes et des processus de gestion robustes pour relever ces défis spécifiques à leur secteur d'activité.

L'introduction d'une plateforme de gestion intégrée chez SKS-Energy, spécialisée dans l'énergie renouvelable et le pompage, représente une initiative stratégique d'une importance capitale. En visant à optimiser les opérations internes, cette plateforme permettra à SKS-Energy de rationaliser ses processus, d'améliorer sa visibilité et son contrôle sur ses activités, tout en renforçant la collaboration entre ses différents départements. Ce projet revêt une importance significative car il promet d'accroître l'efficacité opérationnelle de l'entreprise, de réduire les coûts administratifs et d'offrir une base solide pour une croissance future.

Cependant, la réalisation de cette plateforme de gestion intégrée n'est pas sans défis. SKS-Energy doit relever des obstacles majeurs, notamment l'adaptation de la solution aux besoins spécifiques de l'entreprise. Il est crucial que la plateforme soit flexible et personnalisable pour répondre aux exigences évolutives de SKS-Energy et s'intégrer harmonieusement avec ses systèmes existants. De plus, garantir une adoption réussie par les utilisateurs finaux constitue un défi essentiel. SKS-Energy devra investir dans la formation et la sensibilisation des employés pour assurer une transition en douceur vers la nouvelle plateforme.

Malgré ces défis, le projet de plateforme de gestion intégrée demeure une étape stratégique essentielle pour SKS-Energy. En surmontant les obstacles associés à la conception et à la mise en œuvre de cette solution, l'entreprise pourra renforcer sa position sur le marché de l'énergie renouvelable et du pompage. Cette initiative témoigne de l'engagement continu de SKS-Energy envers l'innovation et l'amélioration continue, et elle est destinée à jouer un rôle crucial dans la réussite future de l'entreprise.

# Présentation de l'entreprise d'accueil

## Présentation de SKS-Energy

### Historique et Fondation

Fondée en 2023, SKS-Energy est le fruit de la vision d'un groupe d'entrepreneurs passionnés par les énergies renouvelables et les technologies de pointe. Forts de leur expertise dans le domaine, ils ont décidé de créer une entreprise innovante, capable de répondre aux besoins croissants en solutions énergétiques durables et efficaces. Depuis sa fondation, SKS-Energy s'est forgée une solide réputation dans le secteur, devenant un acteur incontournable grâce à son engagement envers l'innovation, sa qualité de service et sa capacité à anticiper et à répondre aux évolutions du marché.

### Domaines d'Activité

SKS-Energy opère principalement dans deux domaines clés :

* **Énergie Renouvelable :** L'entreprise se spécialise dans l'installation, la maintenance et la distribution de solutions d'énergie renouvelable, avec un accent particulier sur l'énergie solaire. SKS-Energy propose une gamme complète de services, allant de la conception et de l'installation de systèmes photovoltaïques à la distribution d'équipements et de solutions énergétiques respectueuses de l'environnement.
* **Pompage :** En parallèle de ses activités dans le domaine de l'énergie renouvelable, SKS-Energy propose également des solutions de pompage sur mesure pour une variété d'applications, y compris l'agriculture, l'industrie et les systèmes d'irrigation. Grâce à son expertise technique et à son savoir-faire, l'entreprise est en mesure de concevoir, d'installer et de maintenir des systèmes de pompage efficaces et fiables, répondant aux besoins spécifiques de ses clients.

### Siège Social et Présence Nationale

Le siège social de SKS-Energy est situé à Sfax, en Tunisie, une ville stratégiquement positionnée au cœur de l'activité économique et industrielle du pays. Depuis ce hub central, l'entreprise coordonne ses activités à travers un réseau étendu, couvrant l'ensemble du territoire tunisien. Grâce à sa présence nationale, SKS-Energy est en mesure de fournir des services de haute qualité à ses clients, où qu'ils se trouvent dans le pays.

### Valeurs et Engagement

Au cœur de la philosophie de SKS-Energy se trouvent des valeurs telles que l'innovation, l'intégrité, et l'engagement envers la durabilité environnementale. L'entreprise s'efforce constamment de repousser les limites de la technologie pour offrir à ses clients les solutions les plus avancées et les plus efficaces. De plus, SKS-Energy est profondément engagée à contribuer à la transition vers une économie plus verte, en proposant des solutions énergétiques respectueuses de l'environnement et en promouvant une utilisation responsable des ressources naturelles.

### Vision et Perspectives d'Avenir

Fort de son succès et de sa croissance continue, SKS-Energy se positionne comme un leader émergent dans le secteur de l'énergie renouvelable et du pompage. À l'avenir, l'entreprise prévoit d'étendre sa présence sur le marché national et de renforcer ses partenariats internationaux pour mieux servir ses clients et contribuer à la transition mondiale vers une énergie plus propre et plus durable.

## Services de SKS-Energie

SKS-Energy offre une gamme diversifiée de services, comprenant :

### Installation, Maintenance et Réparation de Systèmes Photovoltaïques :

SKS-Energy offre une expertise complète dans l'installation de systèmes photovoltaïques, qu'il s'agisse de petites installations résidentielles, de projets commerciaux de taille moyenne ou de grands projets industriels. Leur équipe spécialisée assure une installation précise et conforme aux normes, en utilisant des équipements de haute qualité. De plus, SKS-Energy propose des services de maintenance régulière pour garantir le bon fonctionnement et la durabilité des systèmes. En cas de dysfonctionnement ou de panne, l'entreprise propose également des services de réparation rapides et efficaces, minimisant ainsi les temps d'arrêt et assurant la fiabilité continue de l'alimentation électrique.

### Distribution d'Énergie Électrique :

SKS-Energy se distingue par son engagement envers les solutions énergétiques durables et respectueuses de l'environnement. L'entreprise propose une large gamme de produits et de services pour la distribution d'énergie électrique, comprenant notamment des solutions de stockage d'énergie, des systèmes de gestion de l'énergie intelligents et des infrastructures de recharge pour véhicules électriques. En mettant l'accent sur l'efficacité énergétique et la réduction des émissions de carbone, SKS-Energy contribue à promouvoir une transition vers une énergie plus propre et plus durable.

### Conception Sur Mesure et Mise en Œuvre de Solutions de Pompage :

Pour répondre aux besoins variés de ses clients, SKS-Energy propose des solutions de pompage sur mesure adaptées à des applications diverses telles que l'agriculture, l'industrie et les systèmes d'irrigation. Leur équipe d'ingénieurs expérimentés conçoit des systèmes de pompage efficaces et fiables, en tenant compte des spécificités géographiques, des exigences de débit et de pression, ainsi que des contraintes budgétaires. SKS-Energy assure également la mise en œuvre professionnelle de ces solutions, garantissant une installation optimale et une performance maximale des équipements de pompage.

### Services de Conseil en Énergie, Ingénierie et Gestion de Projet :

En tant qu'expert en énergie, SKS-Energy offre des services de conseil complets pour accompagner ses clients à chaque étape de leurs initiatives énergétiques. Leurs consultants expérimentés fournissent des conseils stratégiques sur la planification et la gestion de projets, l'optimisation des performances énergétiques, la conformité réglementaire et les meilleures pratiques en matière de durabilité. De plus, SKS-Energy propose des services d'ingénierie détaillée pour la conception et la réalisation de projets énergétiques complexes, garantissant une exécution efficace et un résultat final conforme aux attentes du client.

## L'organisation et coordonnées

L'organisation de SKS-Energy reflète sa volonté d'efficacité et de collaboration.

### Organigramme :

Son organigramme met en évidence plusieurs départements clés :

* Le département des ventes et du marketing, chargé de promouvoir les produits et services de l'entreprise et d'établir des relations durables avec les clients.
* Le département technique et d'ingénierie, responsable de la conception, de la mise en œuvre et de la maintenance des systèmes énergétiques.
* Le département de la logistique et des opérations, chargé de coordonner les activités de distribution et de gestion des stocks.
* Le département administratif et financier, responsable de la gestion des ressources humaines, des finances et de l'administration générale.

L'entreprise encourage également une culture d'innovation et de collaboration, favorisant les échanges d'idées et les initiatives transversales entre les différents départements.

### Coordonnées :

* BP :
* Téléphone :
* E-Mail :
* Facebook :



**Figure 1:** logo de SKS-Energy

Nous avons passé en revue l’environnement physique de travail dans lequel nous avons été accueillis pour effectuer notre stage de fin d’étude ; Nous allons par la suite faire une présentation du stage.

# CHAPITRE 2 : ETUDE DE L’EXISTANT

# L'existant et critique

En faisant une évaluation détaillée du système et les pratiques actuellement en place chez SKS-Energy pour la gestion des opérations nous constatons qu’actuellement, l'entreprise utilise principalement des méthodes manuelles, avec les informations sur les clients et les services consignées dans des feuilles Excel. Ce processus, bien qu'il ait pu répondre aux besoins de l'entreprise à un moment donné, présente désormais des limites évidentes. Les processus manuels sont sujets aux erreurs humaines, aux retards dans la communication et à une inefficacité opérationnelle croissante à mesure que l'entreprise se développe. Il est donc impératif de procéder à une analyse approfondie pour comprendre pleinement les défis auxquels l'entreprise est confrontée dans sa gestion opérationnelle actuelle.

Pour cela, nous entreprendrons une évaluation critique des systèmes et pratiques actuels chez SKS-Energy. L'évaluation critique des systèmes et pratiques actuels chez SKS-Energy révèle plusieurs lacunes et limitations importantes qui impactent la gestion opérationnelle de l'entreprise. Tout d'abord, la dépendance aux méthodes manuelles, telles que l'utilisation de feuilles Excel pour consigner les informations clients et les activités de service, expose l'entreprise à des risques significatifs d'erreurs humaines. Les saisies incorrectes de données ou les omissions peuvent entraîner des incohérences dans les informations, compromettant ainsi la qualité et la fiabilité des données essentielles à la prise de décision. De plus, la fragmentation des données entre différentes feuilles Excel peut entraver la visibilité et la cohérence des opérations, rendant difficile la surveillance globale des activités de l'entreprise.

En outre, la communication interne entre les départements de SKS-Energy est sujette à des retards et des inefficacités en raison de l'absence de systèmes centralisés. Les informations essentielles peuvent être dispersées et difficilement accessibles, ce qui entrave la collaboration et la coordination entre les équipes. Ces retards dans la transmission des informations peuvent entraîner des retards dans l'exécution des tâches, compromettant ainsi la réactivité et la capacité de l'entreprise à répondre aux besoins des clients de manière efficace et opportune.

De plus, la gestion opérationnelle manuelle présente des défis croissants à mesure que SKS-Energy se développe. Les processus non automatisés sont moins évolutifs et peuvent devenir rapidement obsolètes à mesure que la charge de travail augmente. Cette inefficacité opérationnelle croissante risque de limiter la capacité de l'entreprise à soutenir sa croissance future et à rester compétitive sur le marché.

En conclusion, l'analyse critique des systèmes et pratiques actuels chez SKS-Energy met en lumière la nécessité urgente d'une modernisation et d'une automatisation des processus de gestion opérationnelle. La transition vers une plateforme de gestion intégrée apparaît comme une solution essentielle pour surmonter les défis identifiés et pour renforcer la compétitivité et la durabilité de l'entreprise à long terme.

# Solution proposée

Pour surmonter les défis identifiés et améliorer la gestion des opérations chez SKS-Energy, plusieurs solutions potentielles peuvent être envisagées. Tout d'abord, la mise en place d'une plateforme de gestion intégrée apparaît comme une option prometteuse. Cette solution permettrait d'automatiser les processus clés, tels que la gestion des clients, des services et des stocks, ce qui contribuerait à réduire les erreurs et les retards associés aux méthodes manuelles. De plus, en centralisant les données, cette plateforme faciliterait la collaboration entre les différents départements de l'entreprise, en permettant un accès facile et rapide aux informations pertinentes pour toutes les parties prenantes.

L'adoption d'une telle plateforme présente de nombreux avantages potentiels pour SKS-Energy. En plus d'une meilleure efficacité opérationnelle, cette solution pourrait entraîner une réduction des coûts liés aux tâches administratives et une amélioration de la prise de décision grâce à des données plus précises et à jour. De plus, une plateforme de gestion intégrée permettrait à SKS-Energy de mieux répondre aux besoins changeants du marché et de rester compétitive dans un environnement concurrentiel en constante évolution.

Cependant, la mise en œuvre d'une telle solution n'est pas sans défis. Les implications financières, les délais de déploiement et la formation des employés sont autant de facteurs à prendre en compte lors de la planification de ce projet. Des investissements initiaux seront nécessaires pour le développement et la mise en place de la plateforme, ainsi que pour la formation du personnel à son utilisation efficace. De plus, le processus de transition des systèmes existants vers la nouvelle plateforme pourrait entraîner des perturbations temporaires dans les opérations de l'entreprise.

En conclusion, face aux défis actuels de gestion des opérations chez SKS-Energy, l'adoption d'une application web sur mesure pour la gestion des opérations apparaît comme une solution particulièrement pertinente et prometteuse. Cette solution permettrait à l'entreprise de moderniser ses processus, d'améliorer son efficacité opérationnelle et de renforcer sa compétitivité sur le marché. D'où notre sujet « Conception et Développement d'une Plateforme Desktop pour la Gestion Complète des Opérations chez SKS-Energy dans le domaine de Photovoltaïque et du pompage

# CHAPITRE 3 : ETUDES CONCEPTUEL

Dans ce chapitre, nous plongerons dans l'étude conceptuelle en comparant diverses méthodologies de conception, mettant en évidence leurs forces et leurs faiblesses. Nous commencerons par définir les besoins fonctionnels et non fonctionnels de notre projet, essentiels pour guider nos choix méthodologiques. Ensuite, nous introduirons l'utilisation des diagrammes UML pour représenter visuellement les concepts, les structures et les interactions du système.

# Etude comparative

Dans le domaine de l'ingénierie logicielle et de la conception systémique, une multitude de méthodes et méthodologies sont disponibles pour orienter le processus de conception. Parmi celles-ci, des approches telles que UML, Agile, Merise, RAD, Lean, et bien d'autres encore, se démarquent par leurs particularités et leurs avantages. Cette étude comparative s'engage à explorer ces méthodes sous différents angles, en examinant leurs avantages et limites respectives. Nous évaluerons chaque approche en fonction de critères essentiels tels que la clarté de la méthode, sa flexibilité pour s'adapter aux changements, sa cohérence dans la gestion des projets, ainsi que son efficacité globale. En comprenant mieux les forces et les limites de chaque méthode, nous pourrons éclairer les décisions de conception et sélectionner la méthode la plus adaptée à notre contexte spécifique, optimisant ainsi les chances de succès de notre projet.

## Agile :

La méthodologie Agile est largement reconnue pour sa clarté et sa flexibilité. Elle favorise une communication ouverte et régulière entre les membres de l'équipe, ainsi qu'avec les parties prenantes, ce qui garantit une compréhension commune des objectifs du projet. Sa structure itérative permet une adaptation continue aux changements et aux retours d'expérience, offrant ainsi une grande flexibilité pour répondre aux besoins évolutifs du projet. Cependant, cette approche peut parfois manquer de cohérence, surtout dans les équipes peu expérimentées ou mal encadrées, ce qui peut entraîner des difficultés de gestion et de planification. De plus, bien que l'Agile soit efficace pour la livraison continue de produits fonctionnels, sa focalisation sur les courtes itérations peut parfois négliger une vision à long terme du projet, nécessitant une vigilance accrue pour maintenir la cohérence globale du système sur la durée. En d’autres termes, l'Agile offre une grande clarté et flexibilité, mais peut présenter des défis en termes de cohérence et de gestion à long terme.

## RAD (Rapid Application Development) :

La méthode RAD (Rapid Application Development) se distingue par sa rapidité d'exécution et sa capacité à produire rapidement des prototypes fonctionnels. Son approche itérative et collaborative favorise un développement rapide et des cycles de livraison courts, ce qui peut conduire à des résultats tangibles dans un laps de temps relativement court. Cette rapidité est particulièrement avantageuse pour les projets où les délais sont serrés ou lorsque des solutions doivent être déployées rapidement sur le marché. De plus, RAD encourage une forte implication des parties prenantes tout au long du processus, ce qui renforce la transparence et la satisfaction client. Cependant, cette rapidité peut parfois compromettre la clarté et la robustesse du code produit, ce qui peut nécessiter des efforts supplémentaires pour la maintenance et l'évolutivité du système à long terme. De plus, la forte dépendance aux prototypes peut entraîner des lacunes dans la documentation et la planification, ce qui peut poser des défis pour la gestion de projet et la communication avec les parties prenantes. En résumé, RAD offre une rapidité d'exécution et une implication client élevée, mais peut présenter des inconvénients en termes de clarté, de documentation et de maintenance à long terme.

## Lean :

La méthodologie Lean se distingue par sa focalisation sur l'élimination des gaspillages et l'optimisation des processus. Elle met l'accent sur la livraison de la valeur client de manière efficace et efficiente. Cela se traduit par une clarté accrue dans les processus, car Lean encourage une analyse minutieuse de chaque étape du processus pour identifier et éliminer les activités non essentielles. Cette approche permet de réduire les délais de développement et d'améliorer la qualité globale du produit en éliminant les activités redondantes ou non rentables. De plus, la méthodologie Lean favorise une culture d'amélioration continue, ce qui conduit à une plus grande flexibilité pour s'adapter aux changements et aux évolutions du projet. Cependant, la focalisation sur l'efficience peut parfois entraîner une rigidité excessive, ce qui peut rendre difficile l'adaptation aux besoins changeants ou aux imprévus. De plus, la recherche constante de l'efficience peut parfois négliger la vision à long terme du projet, ce qui peut compromettre la capacité du système à évoluer et à s'adapter aux nouveaux défis. Pour conclure, Lean offre une clarté et une efficacité accrues, mais peut présenter des défis en termes de flexibilité et d'adaptabilité à long terme.

## UML (Unified Modeling Language) :

UML (Unified Modeling Language) est une méthode de conception visuelle largement utilisée dans l'ingénierie logicielle pour représenter et spécifier les différents aspects d'un système. L'un de ses principaux avantages est sa clarté : en fournissant un ensemble standardisé de symboles et de notations, UML permet une communication précise et unifiée entre les membres de l'équipe ainsi qu'avec les parties prenantes du projet. De plus, UML offre une grande flexibilité, car il permet de modéliser divers aspects d'un système, tels que la structure, le comportement et les interactions, à différents niveaux d'abstraction. Cela permet aux concepteurs de s'adapter facilement aux besoins changeants du projet et d'itérer rapidement sur les modèles. Cependant, UML peut parfois manquer de cohérence dans sa mise en œuvre, en raison de la complexité des modèles et de la diversité des interprétations possibles. De plus, sa richesse fonctionnelle peut également entraîner une surcharge d'informations, rendant les diagrammes UML complexes et difficiles à comprendre pour les personnes non familières avec la notation. En outre, UML offre une clarté et une flexibilité considérables dans la modélisation des systèmes, mais peut présenter des défis en termes de cohérence et de complexité pour les équipes de développement.

## Merise

Merise est une méthode de conception structurée utilisée principalement dans le domaine de l'ingénierie des systèmes d'information. L'un de ses principaux avantages est sa clarté dans la modélisation des systèmes d'information en utilisant une approche en couches, ce qui facilite la compréhension des différentes parties du système et de leurs interactions. De plus, Merise offre une approche méthodique et structurée pour l'analyse des besoins, la conception et la mise en œuvre des systèmes d'information, ce qui favorise une gestion efficace du projet et une réduction des risques liés au développement. Cependant, Merise peut parfois manquer de flexibilité pour s'adapter aux besoins changeants du projet, en raison de sa nature rigide et de sa focalisation sur la planification et la documentation détaillée. De plus, la complexité des modèles Merise peut rendre difficile leur compréhension pour les personnes non familières avec la méthode, ce qui peut entraîner des défis en termes de communication et de collaboration au sein de l'équipe. Nous pouvons donc dire que, Merise offre une clarté et une méthodologie structurée pour la conception des systèmes d'information, mais peut présenter des limitations en termes de flexibilité et de complexité pour les projets nécessitant une adaptation rapide aux changements.

Après avoir minutieusement évalué les diverses méthodes de conception, nous préconisons une approche hybride combinant la méthodologie RAD pour la gestion de projet et UML pour la conception du système d'information. En adoptant RAD, les équipes peuvent bénéficier de sa rapidité d'exécution et de sa focalisation sur la livraison de résultats tangibles, ce qui est particulièrement avantageux pour respecter des délais serrés. Parallèlement, l'utilisation d'UML pour la conception du système d'information garantit une clarté et une cohérence dans la modélisation des différentes parties du système, tout en offrant la flexibilité nécessaire pour s'adapter aux évolutions du projet. De plus, la large adoption d'UML dans l'industrie en fait un choix pratique pour la collaboration et la standardisation des pratiques de conception, renforçant ainsi la viabilité de cette approche hybride pour les projets de conception systémique et logicielle.

# Les besoins fonctionnels

Les besoins fonctionnels définissent les fonctionnalités spécifiques qu'un système informatique doit fournir pour répondre aux exigences de ses utilisateurs. Ils constituent une composante essentielle de la phase d'analyse des besoins dans le processus de conception logicielle, guidant la création de solutions adaptées aux besoins des utilisateurs finaux. Dans cette partie, nous examinerons en détail les besoins fonctionnels de notre système, en utilisant trois outils de modélisation principaux : le diagramme de cas d'utilisation, le diagramme de classe et le diagramme de séquence. Ces outils nous permettront de visualiser et de spécifier les interactions entre les utilisateurs et le système, les structures des entités impliquées, ainsi que les séquences d'actions nécessaires à l'exécution des fonctionnalités.

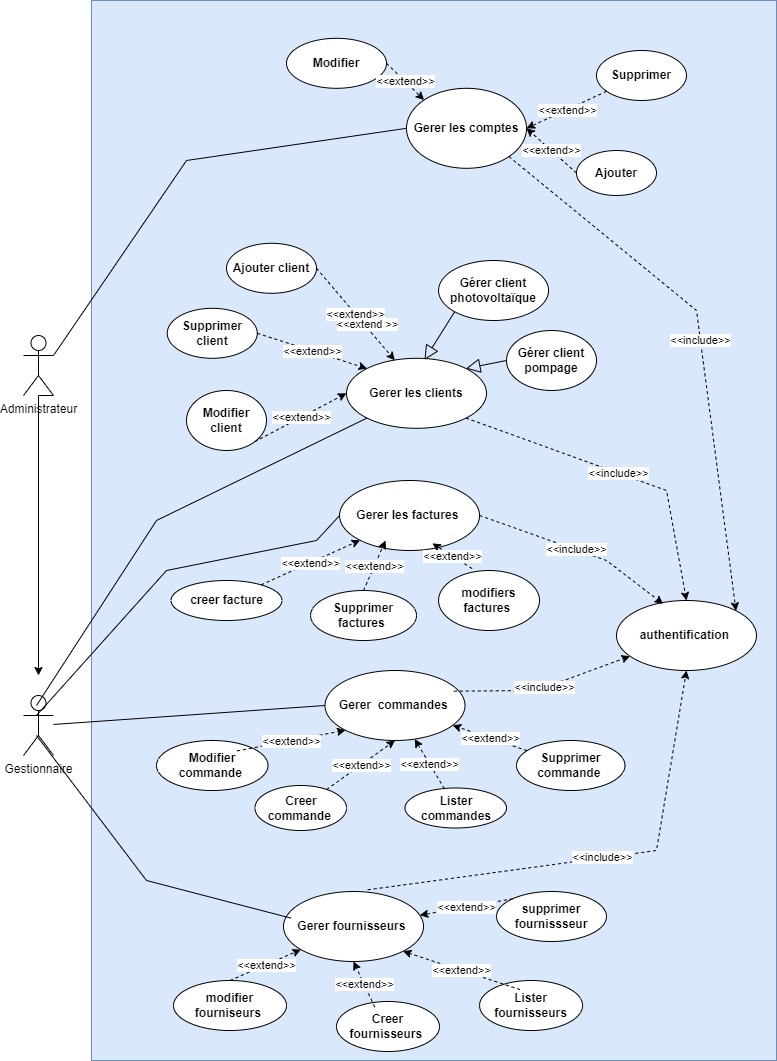
## Diagramme de cas d’utilisation du system

Le diagramme de cas d'utilisation est un outil de modélisation dynamique utilisé dans le génie logiciel pour représenter les interactions entre les acteurs (utilisateurs) et un système. Il met en évidence les différents scénarios d'utilisation d'un système en identifiant les actions que les utilisateurs effectuent et les réponses du système à ces actions. Les acteurs sont représentés sous forme de stickman ou de boîtes, tandis que les cas d'utilisation sont représentés par des ellipses. Les relations entre les acteurs et les cas d'utilisation sont illustrés par des flèches.

Dans notre contexte spécifique, le diagramme de cas d'utilisation (Figure 2) représente les diverses fonctionnalités que notre système doit fournir pour répondre aux besoins des utilisateurs. Dans notre système, nous identifions deux acteurs principaux : l'administrateur et le gestionnaire.

**L'administrateur** : est doté de tous les privilèges et a un accès complet au système. Il est responsable de la gestion globale du système, y compris la création et la suppression des comptes utilisateur, et toute autre tâche d'administration nécessaire.

**Le gestionnaire** : a des privilèges moins étendus que l'administrateur. Bien qu'il puisse contrôler toutes les actions et fonctionnalités du système, à l'exception de la création des comptes utilisateur, il est responsable de superviser les opérations quotidiennes et d'assurer le bon fonctionnement des processus et des flux de travail du système.



**Figure 2:** de Diagramme cas du system

## Description des cas d’utilisation

Nous explorerons en détail les différents cas d'utilisation de notre système, mettant en lumière les interactions entre les acteurs et les fonctionnalités offertes par le système. Les cas d'utilisation représentent les divers scénarios d'interaction entre les utilisateurs et le système, décrivant les actions que les utilisateurs entreprennent et les réponses du système à ces actions.

**Tableau 1:** description du cas d'utilisation "Gestion de compte"

|  |  |
| --- | --- |
| Cas d’utilisation | Gestion de compte |
| Acteur principal | Administrateur |
| Description | L’administrateur doit pouvoir gérer un compte |
| Précondition | L’administrateur est authentifié et a accès au système.  Des comptes d'utilisateurs ont déjà été créés dans le système. |
| Postcondition | Les informations sur les comptes sont mises à jour selon les actions effectuées par l'administrateur.  Les autorisations et les rôles des utilisateurs sont actualisés dans le système |
| Scenario nominal | 1. L'administrateur se connecte au système de gestion des comptes en utilisant ses identifiants d'accès. 2. Le système affiche la liste des comptes d'utilisateurs enregistrés dans le système, généralement classés par nom d'utilisateur, rôle ou statut. 3. L'administrateur peut rechercher un compte d'utilisateur spécifique en utilisant des filtres ou en parcourant la liste des comptes. 4. L’administrateur peut effectuer diverses actions de gestion sur le compte utilisateur 5. L'administrateur enregistre les modifications apportées au compte utilisateur |
| Scenario alternatif | A.1 : Si un compte d'utilisateur est désactivé en raison de problèmes de sécurité   1. L’administrateur peut règles ce problème grâce a la vérification d’authentification   A.2 : Si un utilisateur oublie son mot de passe   1. L’administrateur peut être amené à utiliser une procédure de réinitialisation de mot de passe pour aider l'utilisateur à regagner l'accès à son compte.   A.3 : Si un utilisateur demande des accès supplémentaires   1. L’administrateur peut être amené à examiner la demande pour mettre à jour les autorisations du compte utilisateur en conséquence. |

**Tableau 2:** description du cas d'utilisation "gestion de commande"

|  |  |
| --- | --- |
| Cas d’utilisation | Gestion de commande |
| Acteur principal | Gestionnaire |
| Description | Le gestionnaire doit pouvoir gérer les commandes dans le system |
| Précondition | Le gestionnaire est authentifié et a accès au système.  Des commandes ont été préalablement soumises par les clients. |
| Postcondition | Les commandes sont traitées selon les actions effectuées par le gestionnaire.  Les informations sur les commandes sont mises à jour dans le système. |
| Scenario nominal | 1. Le gestionnaire se connecte au système en utilisant ses identifiants. 2. Le système affiche la liste des commandes en attente de traitement. 3. Le gestionnaire examine les commandes en attente et choisit une commande à traiter. 4. Le gestionnaire sélectionne une commande spécifique pour afficher les détails de la commande, les articles commandés, les informations du client et l'état actuel de la commande. 5. Le gestionnaire met à jour le statut de la commande après validation. |
| Scenario alternatif | A.1 : Si une commande comporte des informations incomplètes ou incorrectes   1. Le gestionnaire peut contacter le client pour demander des clarifications avant de poursuivre le traitement de la commande.   A.2 : Si un article commandé n'est plus disponible en stock   1. Le gestionnaire peut soit contacter le client pour proposer un article de remplacement 2. Soit annuler cet article de la commande et procéder avec le reste de la commande.   A3 : si un article est en rupture de stock et doit être commandé auprès d'un fournisseur  4. le gestionnaire peut ajouter une note ou une tâche associée à cette commande pour un suivi ultérieur. |

**Tableau 3:** description de cas d'utilisation "gestion de fournisseur"

|  |  |
| --- | --- |
| Cas d’utilisation | Gestion du fournisseur |
| Acteur principal | Gestionnaire |
| Description | Utilisateur autoriser à gérer les fournisseurs dans le system |
| Précondition | Le gestionnaire est authentifié et a accès au système de gestion des fournisseurs.  Des fournisseurs ont déjà été enregistrés dans le système. |
| Postcondition | Les informations sur les fournisseurs sont mises à jour selon les actions effectuées par le gestionnaire.  Les relations entre les fournisseurs et les produits/services sont actualisées dans le système. |
| Scenario nominal | 1. Le gestionnaire se connecte au système en utilisant ses identifiants d'accès. 2. Le système affiche la liste des fournisseurs enregistrés. 3. Le gestionnaire peut rechercher un fournisseur spécifique en utilisant des filtres ou en parcourant la liste des fournisseurs. 4. Ajouter un nouveau fournisseur au système en saisissant ses informations. 5. Supprimer un fournisseur du système s'il n'est plus pertinent ou actif. |
| Scenario alternatif | a.1 : Si un fournisseur doit être ajouté mais que ses informations complètes ne sont pas disponibles.  1. le gestionnaire contacte le fournisseur pour compléter ses informations.  a.2 : Si un fournisseur change ses conditions de paiement  1. le gestionnaire met à jour les informations du fournisseur.  a.3 : Si un fournisseur doit être supprimé du système  1. le gestionnaire peut prendre des mesures pour s'assurer qu'il n'y a pas de commandes |

**Tableau 4:** description de cas d'utilisation "gestion de facture"

|  |  |
| --- | --- |
| Cas d’utilisation | Gérer facture |
| Acteur principal | Gestionnaire |
| Description | Utilisateur autoriser à gérer les factures dans le system |
| Précondition | Des factures ont déjà été générées et enregistrées dans le système. |
| Postcondition | Les paiements associés aux factures sont suivis et enregistrés dans le système.  Les informations sur les factures sont mises à jour |
| Scenario nominal | 1. Le gestionnaire se connecte au système. 2. Le gestionnaire peut également générer des rapports sur les factures 3. Le gestionnaire peut effectuer diverses actions de gestion sur la facture. |
| Scenario alternatif | a.1. Si un paiement est en attente   1. Le gestionnaire peut recontacter le client pour plus de détails concernant la commande.   a.2. Si une facture est contestée par le client en raison d'erreurs   1. Le gestionnaire peut ouvrir un processus de résolution de litiges pour résoudre la situation. |

Tableau 5: description de cas d'utilisation "gestion de client"

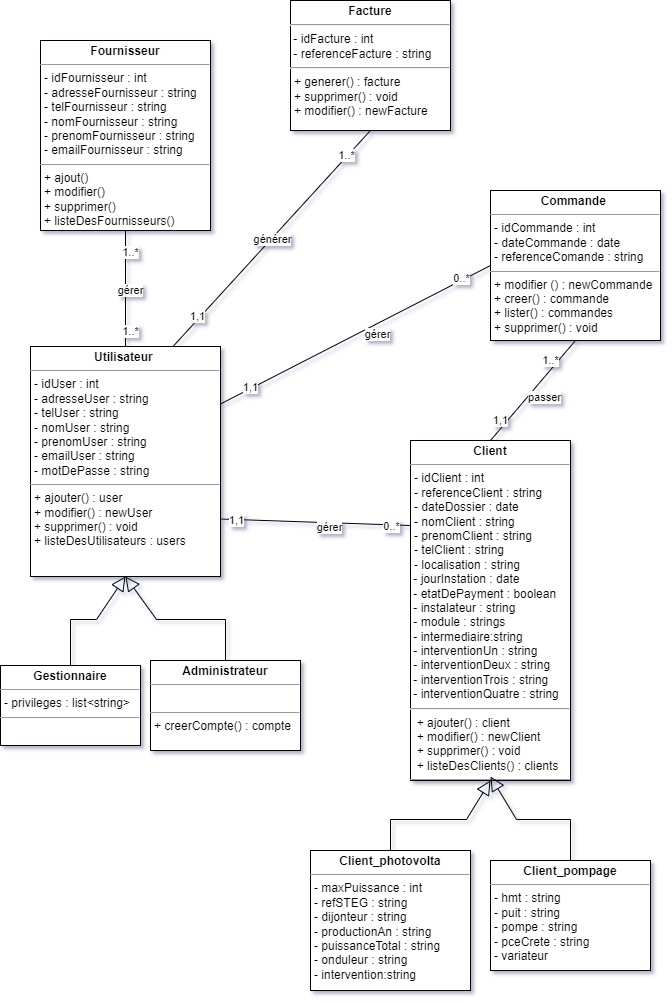
|  |  |
| --- | --- |
| Cas d’utilisation | Gérer client |
| Acteur principal | Gestionnaire |
| Description | Utilisateur autorisé à gérer les clients dans le système. |
| Précondition | Le gestionnaire est authentifié et a accès au système de gestion des clients.  Des informations sur les clients ont déjà été enregistrées dans le système. |
| Postcondition | Les informations sur les clients sont mises à jour selon les actions effectuées par le gestionnaire.  Les relations client-entreprise sont actualisées dans le système si nécessaire. |
| Scénario nominal | 1. Le gestionnaire se connecte au système de gestion des clients en utilisant ses identifiants d'accès 2. Le système affiche la liste des clients enregistrés dans le système 3. Le gestionnaire recherche un client spécifique 4. Le gestionnaire sélectionne un client pour afficher les détails du client, y compris ses informations personnelles, ses coordonnées, son historique d'achat, 5. Le gestionnaire peut effectuer diverses actions de gestion sur le client 6. Le gestionnaire enregistre les modifications apportées au client |
| Scénario alternatif | A1 : Si un client demande la suppression de ses informations personnelles   1. Le gestionnaire peut être amené à coordonner la suppression sécurisée des données   A2 : Si un client présente des problèmes ou des plaintes   1. Le gestionnaire peut être amené à enquêter sur la situation et à travailler avec le service client pour résoudre les problèmes de manière satisfaisante.   A3 : Si un client souhaite mettre à jour ses préférences de communication   1. Le gestionnaire peut être amené à mettre à jour les paramètres du client dans le système et à s'assurer que ses préférences sont respectées. |

## Diagramme de classe

Le diagramme de classe est un outil de modélisation statique utilisé pour représenter la structure interne d'un système d'information. Il met en évidence les différentes classes d'objets qui composent le système, ainsi que leurs attributs et leurs relations. Chaque classe est représentée par une boîte contenant le nom de la classe, suivie d'une liste de ses attributs et de ses opérations.

Ce diagramme est particulièrement utile pour comprendre l'architecture et la conception d'un système logiciel. Il nous permet de visualiser les entités clés du système et leurs interactions, offrant ainsi une vue d'ensemble de sa structure statique. En identifiant les classes principales, ainsi que leurs attributs et leurs relations, le diagramme de classe fournit un cadre clair pour la conception et le développement du logiciel. De plus, le diagramme de classe sert de documentation précieuse pour les membres de l'équipe et les parties prenantes, en offrant une représentation visuelle de la structure du système.

Notre diagramme de classe spécifique est présenté dans la Figure 3.



**Figure 3:** Diagramme de classes du système d’information de SKS-Energie

## Diagramme de séquences

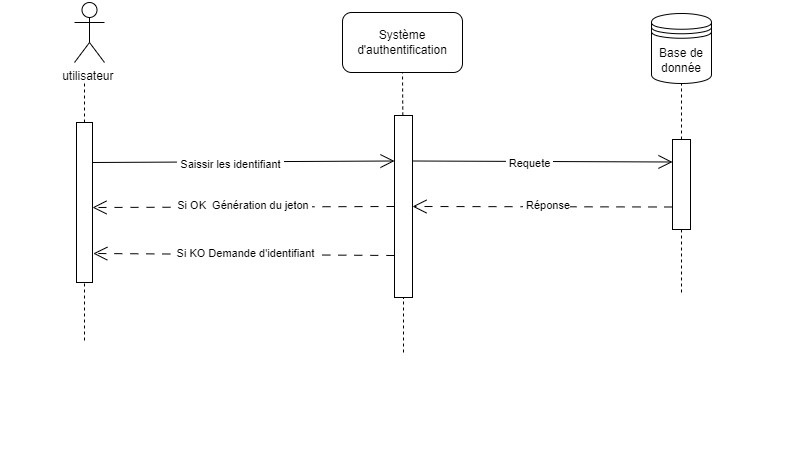
Le diagramme de séquence est un outil pour représenter les interactions entre les objets ou les acteurs dans un système, dans un ordre séquentiel. Il met en évidence la manière dont les messages sont échangés entre les différents composants du système au fil du temps, en illustrant les séquences d'actions et les changements d'état.

Ce diagramme est particulièrement utile pour comprendre le comportement dynamique d'un système et pour visualiser les interactions entre ses composants pendant l'exécution. En identifiant les messages échangés entre les objets ou les acteurs, ainsi que leur chronologie, le diagramme de séquence offre une représentation visuelle claire des processus et des flux de données au sein du système. Il est également utile pour la communication entre les membres de l'équipe et les parties prenantes, en offrant une représentation visuelle des interactions système dans un format compréhensible.

Dans les figures 4, 5 et 6, nous avons réalisé respectivement les diagrammes de séquence de l'authentification, de l'ajout de compte et de la suppression de client afin d'illustrer les interactions entre nos deux acteurs du système.

### Diagramme de séquence « Authentification »

Ce diagramme de séquence illustre de manière séquentielle les interactions entre l'utilisateur et le système pendant le processus d'authentification.



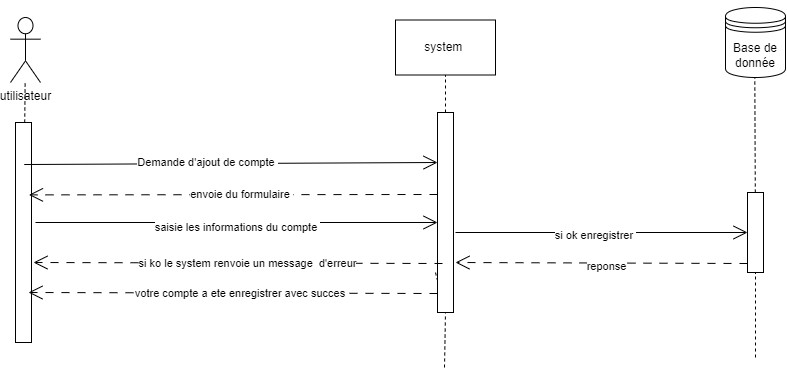
**Figure 4:** Diagramme de séquence « Authentification » dans le system

Pour cette séquence nous suivrons progressivement le processus décris comme suit :

1. L'utilisateur envoie une demande d'authentification au système en saisissant ses identifiants (email et mot de passe).
2. Le système reçoit la demande d'authentification et fait une requête à la base de données.
3. Le système examine la réponse fournis par la base de données et renvoie une réponse à l'utilisateur, indiquant si l'accès est autorisé ou non.
4. Si l'accès est autorisé, l'utilisateur peut accéder au système.
5. Sinon, l'accès est refusé et les identifiants sont demandés à nouveau à l’utilisateur.

### Diagramme de séquence « ajouter compte »

Ce diagramme de séquence illustre de manière séquentielle les interactions entre l'administrateur et le système lors du processus d'ajout de compte.



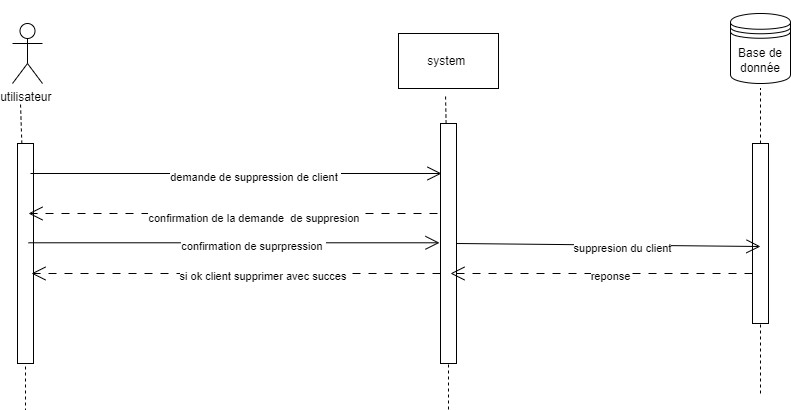
**Figure 5:** Diagramme de séquence « ajouter compte » dans notre système

En ce qui concerne la séquence d’ajout de compte par d’administrateur elle suit les étapes suivantes :

1. L'administrateur envoie une demande d'ajout de compte.
2. Le système reçoit la demande d'ajout de compte et envois le formulaire de création de compte.
3. L’administrateur saisis les données et le système vérifie les informations fournies pour s'assurer qu'elles sont complètes et valides.
4. Si elles sont ok le système crée un compte et envois une requête à la base de données avec les informations fournies pour enregistrer.
5. La basse de données enregistre le compte et envois une réponse au système pour informer que le compte a été créé.
6. Si un problème survient pendant l’enregistrement le système notifie l’administrateur.

### Diagramme de séquence « supprimer client »

Ce diagramme de séquence illustre de manière séquentielle les interactions entre l'administrateur et le système lors du processus de suppression d'un client.



**Figure 6:**Diagramme de séquence « supprimer client » dans notre système

Pour la séquence de suppression d’un compte par d’administrateur elle suit la chronologie suivante :

* 1. L'administrateur envoie une demande de suppression d'un client au système.
  2. Le système reçoit la demande de suppression et renvois une demande de confirmation.
  3. Si l’administrateur confirme, le système envois une requête à la base de données de supprimer les données du client.
  4. La base de données procède à l’opération de suppression et renvois une réponse au système une fois terminé.
  5. Si ok le système confirme la suppression du compte à l’administrateur.
  6. Sinon le système renvois le message d’erreur à l’administrateur.

# Les besoins non fonctionnels

Les spécifications non fonctionnelles d'un système informatique décrivent les caractéristiques qui ne sont pas directement liées aux fonctionnalités spécifiques du système, mais qui sont tout aussi importantes pour son bon fonctionnement, sa performance, sa sécurité, sa convivialité, etc. Voici quelques exemples de spécifications non fonctionnelles :

1. **Performance :**

* Temps de réponse : Le temps qu'il faut pour que le système réponde à une requête utilisateur.
* Capacité : Le nombre maximal d'utilisateurs ou de transactions que le système peut gérer simultanément.
* Évolutivité : La capacité du système à s'adapter et à évoluer pour supporter une charge croissante.
* Disponibilité : Le temps pendant lequel le système est opérationnel et accessible aux utilisateurs, souvent mesuré en pourcentage d'uptime.

1. **Sécurité :**

* Confidentialité : Assurer que seules les personnes autorisées ont accès aux données sensibles.
* Intégrité : Garantir que les données ne sont ni modifiées ni altérées de manière non autorisée.
* Disponibilité : Protéger le système contre les attaques visant à le rendre inaccessible.
* Authentification et autorisation : Mécanismes permettant de vérifier l'identité des utilisateurs et de contrôler leur accès aux ressources du système.

1. **Fiabilité :**

* Sauvegarde et récupération : Mécanismes pour prévenir la perte de données et restaurer le système en cas de sinistre.
* Stabilité : Capacité du système à maintenir des performances cohérentes dans des conditions variables.

1. **Convivialité :**

* Ergonomie : Facilité d'utilisation du système par les utilisateurs finaux.

1. **Maintenabilité :**

* Facilité de modification : La facilité avec laquelle le système peut être modifié pour répondre à de nouveaux besoins ou corriger des problèmes.
* Cohérence : Maintenir une structure et des conventions de codage cohérentes pour faciliter la maintenance par les développeurs.

# CHAPITRE 4 : REALISATION

Dans ce chapitre, notre attention sera portée sur la phase pratique de la mise en œuvre de la plateforme, où nous plongeons dans le cœur technologique de notre projet. Nous explorerons les choix de technologies et d'outils qui ont été sélectionnés avec soin pour transformer notre concept en une réalité fonctionnelle. Ce chapitre se révélera essentiel pour comprendre l'environnement de travail dans lequel nous avons évolué, ainsi que les langages de programmation et les Framework qui alimentent notre projet.

# Environnement de travail

## Environnement matériel

Pour mener à bien notre mission de développement, nous avons eu recours à la gamme variée de matériels indispensables suivante :

* **Système d’exploitation :** Windows 10
* **Processeur :** Intel Core i5, 5eme génération
* **Mémoire vive :** 8GB

## Environnement logiciel

Parmi les outils possibles pour développer notre projet, trois se démarquent par leur utilité et leur efficacité : Draw.io, Visual Studio Code (VS Code), GIT et MongoDB Compass. Nous découvrirons comment ces logiciels contribuent à notre flux de travail, facilitant la conception, le développement et la gestion de notre solution logicielle.

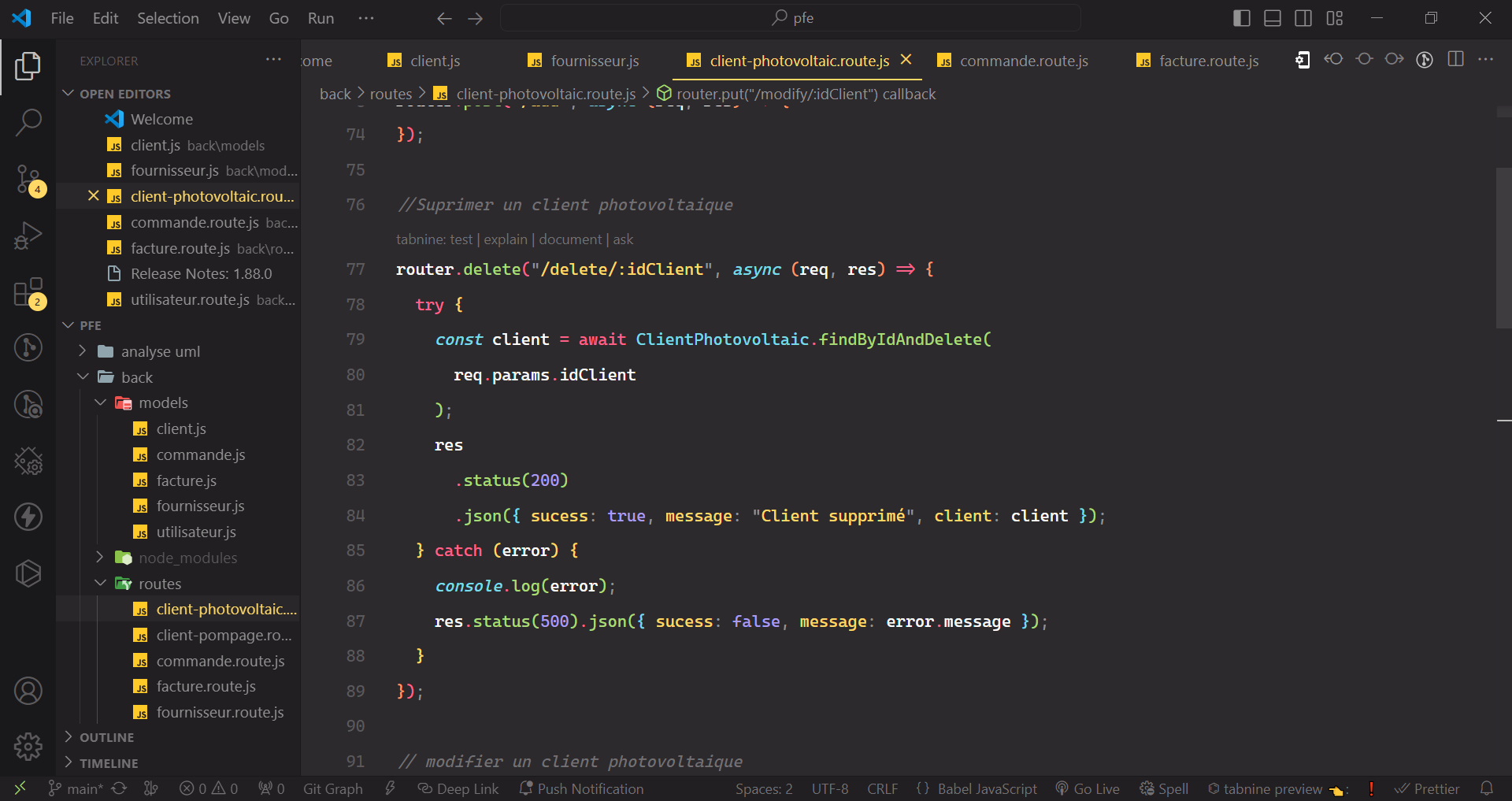
### Visual studio code

Visual Studio Code (VS Code) est un éditeur de code source polyvalent et largement apprécié, reconnu pour sa légèreté, sa rapidité et sa convivialité. Son interface utilisateur intuitive offre une expérience de développement fluide, avec des fonctionnalités avancées telles que la coloration syntaxique, l'autocomplétion intelligente et la mise en forme automatique du code. La capacité de personnalisation et d'extension de VS Code nous a permis d'adapter l'éditeur à notre besoin spécifique. Son intégration avec d'autres outils de développement et services en fait un choix idéal pour notre projet. En tant que pierre angulaire de notre environnement de travail, Visual Studio Code joue un rôle crucial dans le processus de réalisation.



**Figure 7:** Logo visual studio code

Visual Studio Code a été utilisé pour développer à la fois l'interface utilisateur et l'API de l'application SKS-Energy.



**Figure 8 :** Utilisation concrète de VS Code

### mongoDBCompass

MongoDB Compass est un outil de gestion graphique puissant pour MongoDB, offrant une interface utilisateur intuitive et des fonctionnalités avancées pour explorer, manipuler et visualiser les données stockées dans notre base de données MongoDB. Avec ses capacités de gestion de données, il permet d'effectuer des opérations telles que l'insertion, la mise à jour et la suppression de documents, ainsi que la création et la modification d'index pour optimiser les performances des requêtes. En outre, MongoDB Compass nous a offert des fonctionnalités de visualisation de données qui ont facilité la compréhension de la structure et de la distribution des données. En résumé, MongoDB Compass joue un rôle essentiel dans la gestion et l'analyse des données MongoDB, contribuant ainsi à optimiser l'efficacité et la qualité de notre travail dans le développement de l'application SKS-Energy.



**Figure 9 :** Logo mongoDBCompass

MongoDB Compass a été l'outil principal utilisé pour explorer et gérer la base de données MongoDB de l'application SKS-Energy.

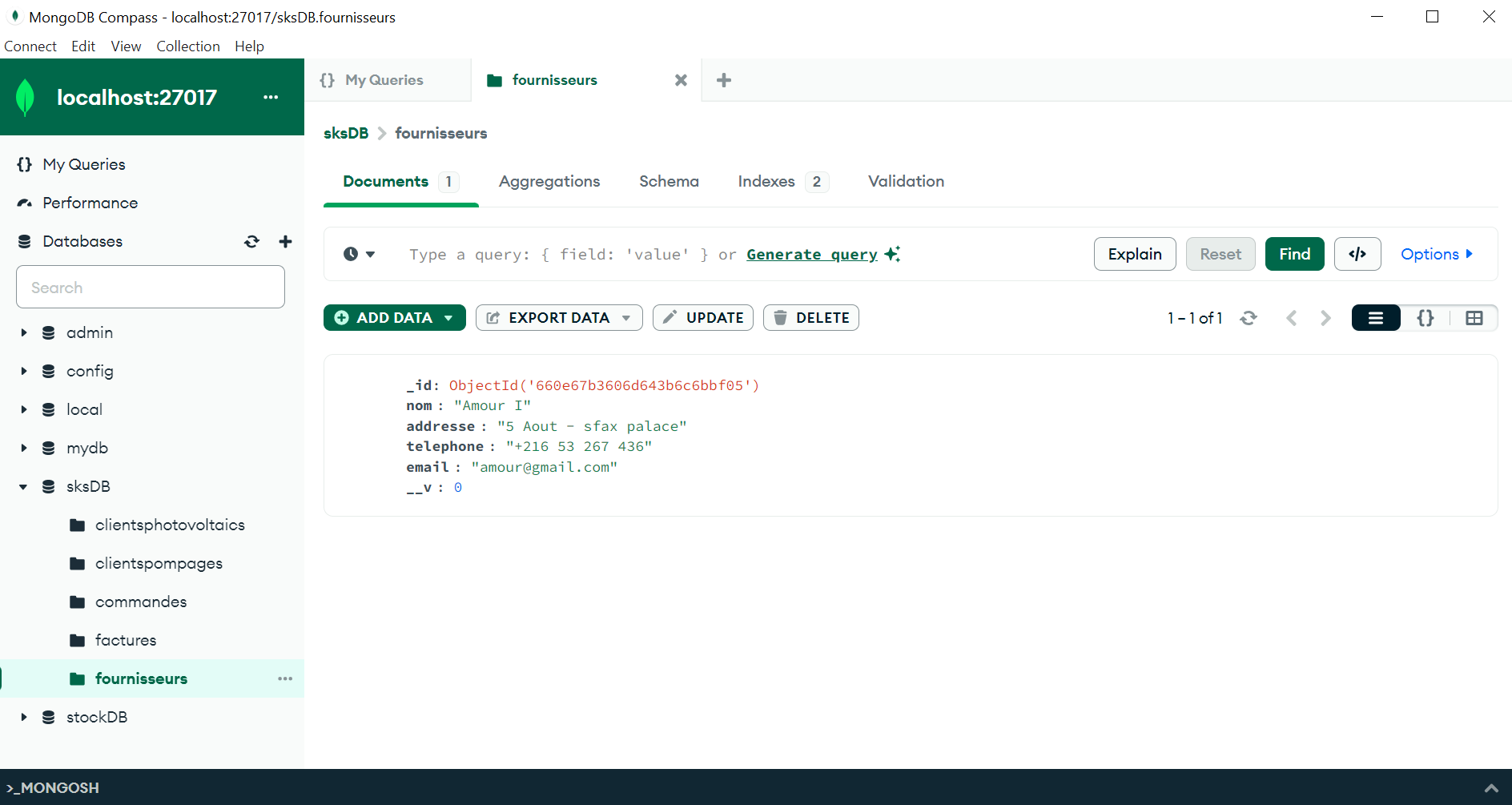


Figure 10 : Utilisation concrète de MongoDB Compass

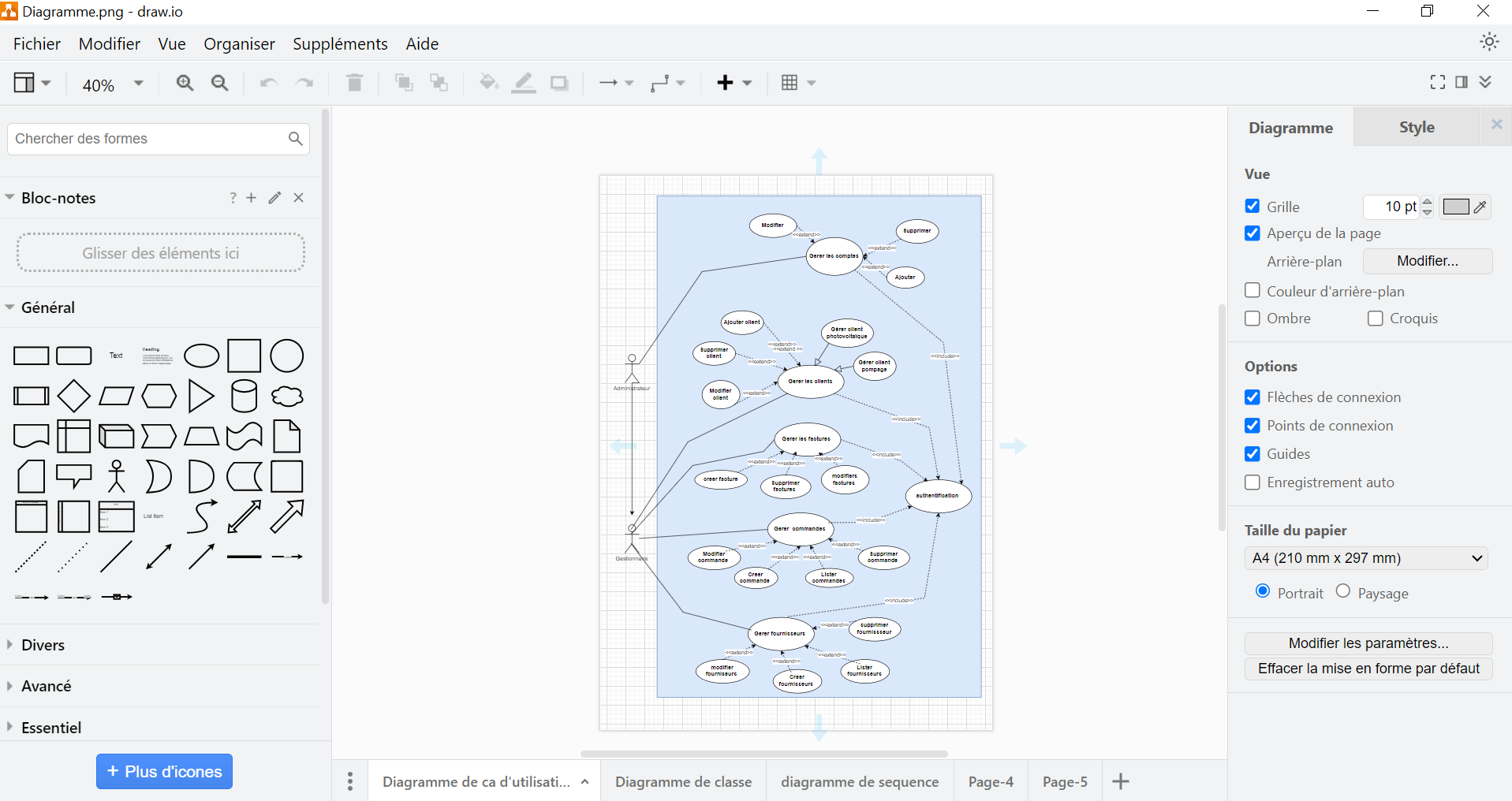
### Draw.io

Draw.io est un outil de dessin polyvalent et convivial largement utilisé pour créer une variété de diagrammes et de schémas dans le cadre du développement de l'application SKS-Energy. Avec son interface intuitive, Draw.io nous a permis grâce à ses fonctionnalités de créer le diagramme de cas d’utilisation, de classe et de séquence. De plus, son intégration avec d'autres outils, en font un choix idéal pour la création et la gestion de diagrammes dans le cadre de notre projet SKS-Energy.



Figure 11: Logo de Draw.io

Draw.io a été utilisé pour créer les diagrammes essentiels à la conception et à la documentation de l'application SKS-Energy, y compris des diagrammes de flux de processus et des schémas d'architecture.



**Figure 12 :** Utilisation de Draw.io

### GIT

Git est un système de contrôle de version distribué largement utilisé dans le développement logiciel pour suivre les modifications apportées au code source et faciliter la collaboration entre les développeurs travaillant sur l'application SKS-Energy. En tant qu'outil essentiel, Git nous a permis de gérer efficacement les versions et les fichiers de notre projet, tout en fournissant un flux de travail flexible pour la création, la gestion et la synchronisation des dépôts locaux et distants. Avec ses fonctionnalités avancées et son intégration avec des plateformes de gestion de projets telles que GitHub, Git a joué un rôle crucial dans la coordination des contributions au sein de l'équipe SKS-Energy.



**Figure 13 :** Logo de Git

Git, avec Git Bash, a été notre outil principal pour la gestion de versions dans le développement de l'application SKS-Energy. En utilisant Git Bash, une interface en ligne de commande basée sur Bash, nous avons pu exécuter les commandes Git pour suivre les modifications du code source, collaborer entre mes encadrants et moi. L'intégration de Git avec Visual Studio Code nous a permis d'optimiser notre flux de travail en développant à la fois l'interface utilisateur et l'API de notre application, garantissant ainsi une gestion de versions fluide et efficace tout au long du processus de réalisation.



**Figure 14 :** Utilisation de Git Bash

# Langages de programmation et Frameworks

## HTML5 (HyperText Markup Language)

HTML (HyperText Markup Language) est un langage de balisage utilisé pour créer et structurer le contenu des pages Web. En tant que norme de base du World Wide Web, HTML offre une structure et une sémantique permettant de définir les éléments d'une page Web, tels que les titres, les paragraphes, les listes, les liens et les médias. HTML utilise des balises pour marquer le début et la fin des éléments, ce qui permet aux navigateurs Web de comprendre la structure et de présenter le contenu de manière appropriée aux utilisateurs. Avec l'évolution vers HTML5, de nouvelles fonctionnalités ont été introduites, telles que les balises sémantiques pour une structuration plus claire du contenu, les formulaires améliorés pour une saisie de données plus précise, et la prise en charge intégrée des médias et de la géolocalisation. HTML est un élément essentiel du développement Web moderne, offrant une base solide pour la création d'interfaces utilisateur interactives et accessibles.



**Figure 15:** Logo de HTML

HTML5, en tant que norme de balisage, a joué un rôle central dans la création de l'interface utilisateur de l'application SKS-Energy. Avec ses fonctionnalités avancées telles que les balises sémantiques pour une structuration plus claire du contenu, les formulaires améliorés pour une saisie de données plus précise, et la prise en charge intégrée des médias, HTML5 a fourni un ensemble complet d'outils pour concevoir des expériences utilisateur interactives et engageantes. En utilisant HTML5, nous avons pu créer des interfaces utilisateur moderne et adaptable, offrant une expérience utilisateur fluide et cohérente sur une variété de navigateurs et de périphériques, ce qui contribue à l'efficacité de SKS-Energy.

## CSS3 (Cascading Style Sheets)

CSS (Cascading Style Sheets) est la dernière version du langage de style utilisé pour définir la présentation visuelle des pages Web écrites en HTML et XHTML. CSS3 offre une gamme étendue de fonctionnalités pour la conception et la mise en forme des éléments HTML, notamment la gestion des couleurs, des polices, des marges, des bordures, des arrière-plans et des mises en page. Les améliorations apportées par CSS3 incluent les transitions, les transformations, les animations et les effets visuels avancés, permettant une personnalisation plus poussée et une expérience utilisateur plus riche. Avec CSS3, les développeurs Web peuvent créer des interfaces utilisateur modernes, adaptatives et esthétiquement attrayantes, tout en maintenant une structure HTML claire et sémantique.



**Figure 16:** logo de CSS

CSS3, en tant que langage de style, a été crucial dans la conception de l'esthétique et de l'expérience utilisateur de l'application SKS-Energy. Grâce à ses fonctionnalités avancées telles que les sélecteurs et les propriétés étendus, les mises en page flexibles avec Flexbox et les grilles CSS avec CSS Grid, CSS3 a permis une personnalisation précise et une disposition dynamique des éléments sur les pages Web. De plus, les effets de transition, d'animation et visuels avancés offerts par CSS3 ont enrichi nos interfaces utilisateur en ajoutant du mouvement, de l'interactivité et des éléments visuels attrayants.

## JavaScript

JavaScript est un langage de programmation polyvalent, principalement utilisé pour développer des applications Web interactives et dynamiques. Contrairement à HTML et CSS, qui sont des langages de balisage et de style respectivement, JavaScript est un langage de programmation à part entière, permettant aux développeurs de créer des fonctionnalités avancées telles que des animations, des validations de formulaire, des effets visuels, des calculs dynamiques et des interactions utilisateur en temps réel. En plus de son utilisation dans les navigateurs Web, JavaScript est également utilisé côté serveur avec des plateformes telles que Node.js, ce qui permet aux développeurs de créer des applications Web à part entière, du frontend au backend, en utilisant un seul langage de programmation. JavaScript est un langage interprété, ce qui signifie qu'il est exécuté par un moteur JavaScript intégré dans le navigateur Web de l'utilisateur. Cela lui permet d'interagir avec le DOM (Document Object Model) pour modifier la structure et le contenu d'une page Web en réponse aux actions de l'utilisateur, telles que les clics de souris, les saisies au clavier et les mouvements de la souris. Grâce à sa popularité et à sa large adoption, JavaScript dispose d'une vaste communauté de développeurs, de bibliothèques et de frameworks, tels que React.js, AngularJS, Vue.js et jQuery, qui simplifient le processus de développement et permettent de créer des applications Web modernes, réactives et performantes.



**Figure 17:** Logo JavaScript

Dans le cadre de notre projet, nous avons largement exploité les capacités du langage JavaScript pour le développement de notre application. Cette utilisation de JavaScript s'est déployée sur deux fronts principaux : côté serveur et côté client. D'une part, nous avons tiré parti de Node.js, un environnement d'exécution JavaScript côté serveur, pour la mise en place de notre backend. Node.js nous a offert une plateforme flexible et performante pour gérer les requêtes, accéder à la base de données et exécuter des opérations côté serveur. D'autre part, pour la construction de l'interface utilisateur dynamique et réactive de notre application, nous avons utilisé React.js. Cette bibliothèque JavaScript frontale nous a permis de créer des composants réutilisables et modulaires, facilitant ainsi le développement et la maintenance de l'interface utilisateur. En combinant l'utilisation de Node.js côté serveur et de React.js côté client, nous avons pu développer une application Web complète et robuste, offrant une expérience utilisateur fluide et immersive.

## Node.JS

Node.js est un environnement d'exécution JavaScript côté serveur, basé sur le moteur JavaScript V8 de Google Chrome. Il permet d'exécuter du code JavaScript côté serveur, ce qui signifie qu'il peut être utilisé pour développer des applications Web complètes, en utilisant un seul langage de programmation. Les caractéristiques clés de Node.js incluent sa non-blocage et son modèle d'E/S asynchrone, qui lui permet de gérer de manière efficace de nombreuses connexions simultanées sans bloquer le thread d'exécution. Cela le rend particulièrement adapté aux applications Web en temps réel, telles que les applications de chat, les tableaux de bord en temps réel et les jeux en ligne. Node.js est également connu pour son écosystème robuste de modules, accessible via le gestionnaire de paquets npm (Node Package Manager). Avec npm, les développeurs peuvent facilement installer, gérer et partager des modules JavaScript réutilisables, accélérant ainsi le processus de développement. En outre, Node.js est extensible et peut être utilisé pour une variété de tâches, notamment la création de serveurs HTTP, la manipulation de fichiers, l'accès aux bases de données, la création d'API Web, la création de microservices et bien plus encore. Grâce à sa popularité croissante et à son large éventail d'applications, Node.js est devenu un choix populaire pour le développement d'applications Web modernes, offrant une performance élevée, une extensibilité et une facilité de développement.



**Figure 18:** logo de Node.js

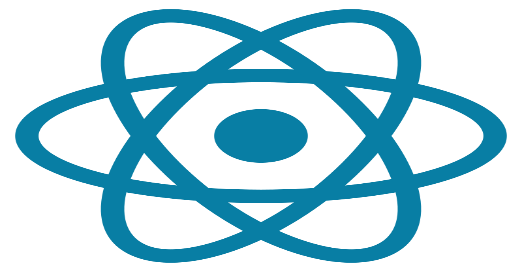
Pour ce qui est de notre projet, Node.js a été notre choix privilégié pour le développement des API et la communication avec la base de données MongoDB. Node.js nous a permis de créer des API robustes et efficaces, offrant des fonctionnalités telles que la gestion des requêtes HTTP, la manipulation des données et la mise en œuvre de la logique métier. Grâce à son modèle asynchrone et non bloquant, Node.js a été capable de gérer efficacement de multiples connexions simultanées, assurant ainsi des performances optimales pour notre application. De plus, Node.js s'est parfaitement intégré avec MongoDB, une base de données NoSQL populaire, permettant une interaction fluide et sécurisée avec la base de données. Cette combinaison de Node.js pour le développement des API et MongoDB pour le stockage des données a contribué à la création d'une architecture backend puissante et évolutive pour notre application.

## ReactJS

ReactJS est une bibliothèque JavaScript open-source développée par Facebook, qui est largement utilisée pour la construction d'interfaces utilisateur interactives et dynamiques dans les applications Web. ReactJS se concentre sur la création de composants réutilisables et modulaires, permettant ainsi de structurer efficacement le code et de simplifier le processus de développement. Les caractéristiques clés de ReactJS incluent son approche déclarative pour la création d'interfaces utilisateur, sa gestion efficace du DOM virtuel, et son utilisation du concept de "one-way data flow" pour la gestion des données. Avec ReactJS, les développeurs peuvent définir des composants qui représentent des parties isolées de l'interface utilisateur, ce qui facilite la réutilisation du code et la maintenance de l'application.

ReactJS utilise également JSX (JavaScript X’ML), une extension de syntaxe qui permet d'écrire du code HTML directement dans JavaScript. Cela facilite la création de composants React en combinant la logique JavaScript avec la structure HTML, ce qui rend le code plus lisible et plus facile à comprendre. Une autre caractéristique importante de ReactJS est son efficacité dans le rendu des composants. Grâce à son utilisation du DOM virtuel, ReactJS minimise les manipulations directes du DOM, ce qui améliore les performances de l'application en réduisant les re-rendus inutiles et en optimisant les mises à jour de l'interface utilisateur. En outre, ReactJS est accompagné d’un riche écosystème de bibliothèques et de frameworks complémentaires, tels que Redux pour la gestion de l'état de l'application, React Router pour la navigation dans l'application, et Material-UI pour les composants d'interface utilisateur prêts à l'emploi.

En résumé, ReactJS est une bibliothèque JavaScript puissante et flexible, qui offre aux développeurs les outils nécessaires pour créer des interfaces utilisateur modernes, réactives et performantes dans les applications Web.



**Figure 19:** Logo de React

Nous avons fait le choix d'utiliser ReactJS pour le développement de nos interfaces homme-machine. ReactJS, une bibliothèque JavaScript créée par Facebook, a été notre solution de prédilection en raison de ses nombreuses fonctionnalités et de son approche déclarative du développement d'interfaces utilisateur. Grâce à ReactJS, nous avons pu concevoir des composants réutilisables et modulaires, offrant ainsi une structure claire et efficace à nos interfaces. Cette approche modulaire nous a permis de simplifier le processus de développement et de maintenance, en nous permettant de réutiliser facilement les composants à travers différentes parties de notre application. De plus, la gestion efficace du DOM virtuel par ReactJS a contribué à optimiser les performances de nos interfaces, en minimisant les re-rendus inutiles et en assurant des mises à jour fluides et rapides de l'interface utilisateur. En utilisant ReactJS, nous avons pu offrir à nos utilisateurs des interfaces homme-machine modernes, réactives et conviviales, répondant ainsi à leurs besoins et attentes avec efficacité.