

RAPPORT DE STAGE CURSUS INGÉNIEUR

TOM FOLTIER

Production d'un logiciel de création de block-diagram.

2022-2023



TABLE DES MATIÈRES

Glossaire

Remerciements

Résumé

Abstract

I. Introduction

II. Présentation de l'entreprise

2.1 Qu'est ce qu'un distributeur ?

2.2 Center Of Excellence

III. Contexte du stage

3.1 Projet "System Solution Designer"

3.2 Découverte du projet

3.3 Méthodologies de travail

IV. Contenu du stage

4.1 Technologies utilisées et développement

4.1.1 Présentation des technologies

4.1.2 Structure de données et interface graphique

4.1.3 Aspect sécuritaire

4.2 Formations

4.3 Déploiement

V. Bilans

5.1 Bilan personnel

5.2 Bilan professionnel

VI. Conclusion

VII. Ouverture

VII. Liste des références

GLOSSAIRE

DC	Demand Creation
Part Number	Référence unique d'un composant
AWS	Amazon Web Services
AEU	Advanced Engineering University
UI	User Interface (interface utilisateur)
UX	User eXperience (expérience utilisateur)
Design Center	Espace de création de block-diagrams
API	Application Programming Interface (interface de programmation d'application)
SGBD	Système de Gestion de Bases de Données
CLI	Command Line Interface (interface en ligne de commande)
BDM	Business Development Manager

REMERCIEMENTS

Je tenais tout d'abord à remercier mon tuteur industriel, Etienne LANOY qui m'a permis de réaliser ce stage dans les meilleures conditions, il a su être présent tout au long de ce stage et dans tous les moments clés du projet en répondant à mes questionnements et besoins. Je tenais également à le remercier de m'avoir permis de participer à l'AEU en Avril 2023, évènement durant lequel j'ai pu échanger avec beaucoup de personnes provenant de différents horizons et ayant des compétences diverses.

Je souhaitais également remercier toute l'équipe de l'agence de Lyon pour leur accueil chaleureuse et leur présence quotidienne.

J'aimerais aussi remercier Sandra Perdrigeon, la directrice des ressources humaines des agences françaises, pour son précieux accompagnement durant ce stage. Ses conseils avisés, son soutien constant et sa bienveillance ont grandement contribué à la qualité de mon apprentissage. Merci pour la confiance accordée et pour avoir été un pilier essentiel de cette aventure professionnelle.

Pour finir, je souhaiterais remercier toutes les personnes avec qui j'ai pu échanger tout au long de ce stage pour leurs conseils et leur bienveillance.

RÉSUMÉ

Fondée en 1968, Future Electronics s'est constamment adaptée à l'évolution technologique, en développant des logiciels spécifiques pour répondre à ses besoins croissants. Cependant, avec le progrès rapide des technologies de l'information, certains de ces logiciels ne répondent plus adéquatement aux besoins actuels de l'entreprise.

La société détient d'importantes bases de données, englobant les informations relatives à ses clients, commandes, stocks, entre autres. Une équipe IT, basée à Montréal, est dédiée au développement et à la maintenance de ces logiciels internes, garantissant ainsi la gestion optimale et l'analyse de ces données volumineuses.

Face à ces enjeux, la direction européenne a initié une campagne de recrutement. L'objectif est d'accompagner la croissance de Future Electronics par une refonte complète des outils informatiques mis à la disposition des collaborateurs.

La "demand creation" (DC), ou "création de la demande", est le cœur de métier des employés chez Future Electronics. Ainsi, l'entreprise a élaboré un système permettant la création de blocs-diagrammes pour chaque projet attribué à un employé.

Ces diagrammes intègrent toutes les informations essentielles relatives aux besoins du client, tels que les pièces nécessaires, la quantité, la date de lancement du projet, etc. Face à l'obsolescence de ce logiciel et à son utilisation décroissante, la refonte de cet outil est devenue le focus principal de mon stage.

L'enjeu majeur de mon intervention était de conceptualiser et de mettre en œuvre une application web moderne et ergonomique, nommée "System Solution Designer". Basée sur plusieurs API et intégrant un système de recommandations piloté par l'intelligence artificielle, cette nouvelle solution vise à accroître significativement la productivité des collaborateurs.

ABSTRACT

My industrial tutor recruited me with the aim of upgrading the company's existing software, starting with the block diagram creation software which, over time, had become outdated and no longer really met the company's internal needs.

I therefore used my knowledge about web technologies (framework, database, hosting, security, ...) and I did a lot of research to find the best stack to start developing this software "System Solution Designer". I therefore presented multiple technologies to my tutor, and we finally chose a stack based on React, NestJS and MongoDB.

I then started coding the application referring to some templates that my tutor did to show me the UI design he wanted to see on the application.

Key-words : Javascript, React, Node.js, NestJS, MongoDB, Docker, AWS.

I – INTRODUCTION

Dans le cadre de ma formation d'ingénieur à l'ISEN Toulon, j'ai réalisé mon stage de fin d'études dans l'entreprise Future Electronics (agence de Lyon) à Bron. Ce stage s'est déroulé en deux parties, une première partie d'octobre 2022 à Mars 2023 durant laquelle j'effectuais un roulement de deux semaines entre l'école et l'entreprise. Puis une deuxième partie de Mars 2023 à Octobre 2023 durant laquelle j'étais présent à temps plein en entreprise.

Future Electronics possède 4 grandes agences en France (Paris, Lyon, Toulouse, Bretagne). Chaque agence regroupe un certain nombre d'employés qui se partagent le marché français selon leur position géographique, il s'agit majoritairement de posts orientés "business (nous verrons plus tard comment s'organise l'entreprise dans le détail). Pour ma part, j'ai rejoint une branche orientée "technique", il s'agissait en plus d'une création de post (qui n'existait pas encore au niveau Européen), ce qui m'a permis de choisir l'agence de Lyon pour mener à bien la mission qui m'allait être confiée.

Durant mon stage de 4ème année (également réalisé chez Future Electronics), j'ai été amené à échanger avec Etienne LANOY "Senior Director AEG EMEA" (nous verrons plus tard l'organisation de l'entreprise) qui m'a fait part de sa recherche de nouveaux profils pour développer et améliorer les logiciels informatiques de l'entreprise et c'est ainsi que je me suis retrouvé à occuper ce poste.

L'objectif premier du poste que j'ai rejoint était de développer de nouveaux outils qui seront utilisés, en interne, par les employés de Future Electronics (au niveau Européen dans un premier temps). Dans le but d'augmenter la productivité des employés afin de continuer à faire grossir l'entreprise, à travers des logiciels utilisant des technologies plus performantes que celles utilisées jusqu'à maintenant et en créant des interfaces utilisateurs modernes et intuitives pour apporter une réelle plus-value aux utilisateurs.

Ce rapport est composé de quatre parties. La première partie présente brièvement l'entreprise. La seconde partie est une introduction du contexte du stage dans laquelle nous découvrirons le projet "SystemSolution Designer", les technologies utilisées ainsi que la méthodologie de gestion de projet employée. La troisième partie expose de façon détaillée les différentes missions effectuées. Enfin, une analyse de mon expérience professionnelle et personnelle est présentée.

II – PRÉSENTATION DE L'ENTREPRISE



**Future Electronics à Pointe Claire,
futureelectronics.com**



Robert Miller, forbes.com

Future Electronics est une société spécialisée dans la distribution de composants électroniques et électromécaniques dont le siège social est situé à Pointe-Claire au Québec. L'entreprise a été fondée en 1968 par Robert Miller et est à ce jour l'une des plus importantes entreprises privées du Québec.

Au niveau mondial, Future Electronics est actuellement le troisième distributeur de composants électroniques. Après Arrow et Avnet qui sont ses principaux concurrents et qui occupent encore à ce jour une place bien plus importante sur le marché, bien que Future Electronics ne cesse remarquablement de croître depuis les 20 dernières années.

L'entreprise opère dans le monde entier, à travers 170 sites situés dans 44 pays répartis sur les 5 continents. Dans le cadre de mon stage, j'ai intégré la région Européenne ou "EMEA" (Europe Middle East Africa) et qui est sous la direction de Karim Khebere.

Au niveau Francophone, l'entreprise possède des bureaux à Paris, Toulouse, en Bretagne, à Strasbourg ainsi qu'à Lyon (Bron) qui est l'agence dans laquelle j'ai effectué mon stage. C'est à Paris que la majeure partie du marché Français est gérée et que l'on retrouve les managers-France et une partie des managers-Europe.

A sa création, l'entreprise fonctionnait comme un Brocker (achat/revente de composants) puis avec l'évolution du marché et la montée en puissance de ses concurrents, l'entreprise a été amenée à évoluer elle aussi en devenant petit à petit ce qu'elle est aujourd'hui.

QU'EST CE QU'UN DISTRIBUTEUR ?

Dans de nombreux secteurs, dont l'électronique, l'alimentaire et l'automobile, la vente de produits ne se réalise pas toujours en direct entre fabricants et consommateurs.

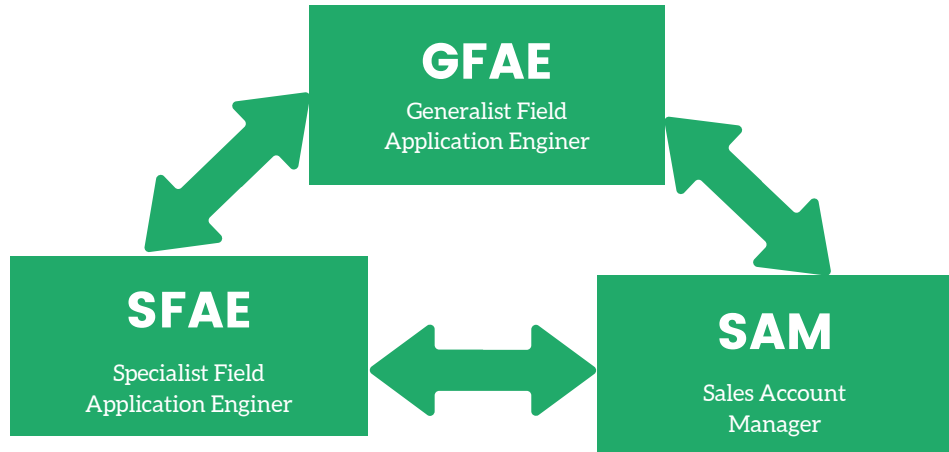
Prenons l'exemple de GlobalFoundries, spécialisée dans la production de composants électroniques à base de silicium, et collaborant avec Future Electronics. Une fois ces composants produits, GlobalFoundries a deux options :

- Vendre directement ces semi-conducteurs sur le marché, quel que soit le volume demandé par les clients. Cela requiert la gestion des commandes, le suivi des stocks, la logistique, le marketing et la promotion via des experts techniques, des équipes commerciales et des campagnes publicitaires. Cette démarche demande d'importantes ressources financières et humaines.
- Déléguer cette responsabilité à des distributeurs, comme Future Electronics, qui assument alors la distribution des produits fournis par les fabricants. Ce modèle est similaire à celui de la grande distribution. À titre illustratif, les enseignes comme Leclerc ou Carrefour achètent en grande quantité des produits fabriqués par divers fournisseurs et les revendent ensuite aux consommateurs à des prix majorés, leur permettant ainsi de réaliser des bénéfices.

Dans cette chaîne, le rôle du distributeur est primordial, assurant une interface efficace entre fabricants et consommateurs, tout en optimisant la logistique, le stockage et la vente.

N.B Il est important de préciser que certaines entreprises de grande envergure qui consomment des millions voir des milliards de composants (comme Apple ou Samsung) travaillent en général en direct avec les fabricants sans passer par des distributeurs. Les fabricants ont dans ce cas la possibilité de gérer la logistique nécessaire car la livraison par exemple se fait en un seul point et les quantités utilisées par ces entreprises sont assez importantes pour que cela reste bénéfique pour les fabricants.

Organisation d'une équipe de vente chez Future Electronics :



Avant toute chose, le premier contact avec le client s'effectue par le biais **BDM** (Business Développement Manager) qui a pour rôle de "dénicher" de nouveaux clients en leur proposant les services de Future Electronics.

Une fois ce travail accompli, le client est généralement attiré à un GFEA. A ce stade, le rôle du GFEA est de comprendre le besoin du client, le guider dans le choix des composants et si possible, lui proposer des produits complémentaires dans le but d'améliorer son produit final et donc de vendre plus de composants. C'est ce que l'on appelle la **Demand Creation**, abrégée "**DC**" et qui est **le principe fondamental de fonctionnement de l'entreprise**.

La Demand Creation est une stratégie qui consiste à créer une prise de conscience et une demande pour un produit ou service. Concrètement, il s'agit d'accroître la notoriété de la marque et de capter l'attention des personnes qui sont conscientes du problème mais pas encore de la solution.

Le travail des Généralistes (GFEA) est complexe. En effet, ils doivent connaître (avec une précision technique plus ou moins élevée) les centaines de composants proposés par les divers fabricants du portfolio de Future Electronics, cela requiert un bagage technique important dans les différentes technologies électroniques. Au quotidien, ils sont amenés à interagir avec différents profils travaillant pour les entreprises clientes, à savoir les ingénieurs et techniciens travaillant en bureau d'étude ou encore les employés chargés des achats. Ils doivent donc s'adapter aux différents niveaux techniques de leurs interlocuteurs pour bien comprendre leurs besoins et leurs enjeux.

Dans certains cas, lorsque le projet d'un client nécessite des capacités bien spécifiques dans une technologie particulière, les Généralistes ont recours aux **Spécialistes** (SFAE) qui sont donc des ingénieurs spécialisés dans l'une des **verticales** proposées par Future Electroncis.

Au niveau Européen, les verticales proposées sont :

- **Connectivity (Wireless and Interconnect)**
- **High-end (Display, Processor and FPGA)**
- **Sensor**
- **Power and Analog**
- **Lighting**

Connectivity : il s'agit des composants liés à la connectique et le wireless, on y retrouve entre autres le Bluetooth ainsi que le BLE (Bluetooth Low Energy), les modules Wi-Fi ou encore l'Ultra Wideband.

High-end : le High-end regroupe tous les composants de type écrans, processeurs, logique programmables, etc...

Sensor : dans cette verticale, nous retrouvons tous les types de capteurs, il en existe de plus en plus et deviennent extrêmement précis. Par exemple les capteurs de fréquence cardiaque, de position, de température, de qualité d'air, de taux d'humidité, ...

Power and Analog : la partie power fait référence à tout les composants liés aux technologies de chargeurs, de batteries, de convertisseurs, etc... Ce domaine est devenu très convoité ces dernières années avec l'arrivée (entre autres) des véhicules électriques sur le marché et leur démocratisation ainsi que toutes les technologies basées sur l'énergie électrique qui nécessite nombre de ces composants.

Lighting : l'éclairage fait également parti des enjeux contemporains avec la recherche de très basses consommations et de designs de plus en plus sophistiqués.

On retrouve donc (au niveau de l'Europe) au moins un spécialiste par verticale. Ces ingénieurs sont des experts dans leurs domaines.

Généralement issus de bureaux d'études ou de grosses entreprises ils ont su se forger un bagage technique extrêmement pointu dans l'une des verticales vues précédemment ainsi qu'une aptitude de communication très développée, nécessaire dans leur travail de tous les jours qui consiste surtout à vendre les produits des fabricants.

Comme pour le Généraliste, le rôle du Spécialiste est de guider le client vers les technologies les plus adaptées à ses besoins et à ses attentes (que ce soit en terme de qualité ou de prix car le portfolio de Future Electronics regorge de références diverses et variées).

L'intérêt de travailler avec un distributeur comme Future Electronics réside également dans le fait que ces experts (Généralistes et Spécialistes) sont au fait des dernières technologies, souvent même des technologies qui ne sont pas encore disponibles sur le marché (ces informations sont souvent issues des fabricants qui cherchent également à promouvoir leurs produits en avant-première). De ce fait, les clients peuvent trouver un avantage concurrentiel en collaborant avec un entreprise comme Future Electronics.

N.B Chaque Généraliste possède une liste de clients qui lui sont attitrés. En moyenne, un Généraliste possède 30 à 40 clients sur lesquels il est mesuré. Ces performances sont évaluées en fonction de la quantité de pièces vendues et du chiffre d'affaires qui en découle.

CENTER OF EXCELLENCE

Jusqu'à maintenant, nous avons vu la partie commerciale et marketing de Future Electronics ainsi qu'une partie technique apportée par les ingénieurs (GFAE et SFAE) au sein de l'entreprise.

Néanmoins, ces ingénieurs sont contraints (par manque temps) à ne pas "rentrer en profondeur" dans les projets des clients. Cela peut être parfois frustrant pour eux car ils sont passionnés par leurs technologies et le fait d'avoir simplement un aperçu du projet sans réellement suivre toutes les étapes de développement peut être déroutant pour des profils ayant travaillé dans des bureaux d'études ou des entreprises dans lesquelles ils ont été amenés à résoudre des problèmes dans leur globalité.



Future Electronics a donc trouvé une autre manière de promouvoir les produits des fabricants avec lesquelles l'entreprise travaille, c'est à travers le COE ou Center Of Excellence. Il s'agit de bureaux d'études où travaillent des ingénieurs mais cette fois-ci uniquement sur la partie technique et non-plus commerciale.

Ces ingénieurs répondent aux problématiques que rencontrent les ingénieurs terrains (GFAE et SFAE) en proposant des solutions qui leurs permettent de mettre en avant et promouvoir certains composants à travers des produits qui sont propres à Future Electronics. A titre d'exemple, en faisant fonctionner un nouveau module Bluetooth (BLE) alimenté par une batterie fournie par un autre fabricant, Future Electronics peut proposer cette nouvelle solution à ses clients, dans le but de vendre non pas un seul, mais deux composants (dispositif Bluetooth + batterie).

De cette manière, la combinaison de plusieurs composants peut permettre à l'entreprise de promouvoir plus de produits et donc d'en vendre plus à ses clients en ayant des démonstrations fonctionnelles; ce qui est toujours plus parlant pour le client.

Il s'agit de la branche à laquelle j'ai été rattaché durant mon stage car le COE est également sous la direction d'Etienne LANOY.

III – CONTEXTE DU STAGE

PROJET "SYSTEM SOLUTION DESIGNER"

Comme nous l'avons vu en introduction, le logiciel clé dans la gestion de la demand creation était jusqu'à maintenant le logiciel "BlockDiagram".

Il s'agit d'une logiciel de bureau proposant une interface graphique à travers laquelle les utilisateurs peuvent :

- Créer un projet (le lier à un client)
- Ajouter des blocks, contenant chacun un "Part Number", la quantité de pièces souhaitée, etc...
- Lier ces blocks ensembles, selon leurs liens électroniques
- Partager un diagram avec les personnes concernées par le projet (vendeur/ FAE/ ...)
- Enregistrer des demandes sur des produits "registrables"

Les fabricants travaillent généralement avec plusieurs distributeurs. Lorsqu'un client passe par Future Electronics par exemple et qu'il souhaite une pièce en particulier, si cette pièces se trouve parmi les pièces "registrables" dans le portfolio de Future, le FAE ou le vendeur en charge du projet va alors faire une demande auprès du fabricant pour obtenir une sorte "d'avantage concurrentiel" sur ses concurrents (les autres distributeurs) en bénéficiant d'un rabais (de l'ordre d'une dizaine de pourcent). Le fabricant peut alors choisir de valider cette demande ou bien de donner cet avantage à un autre distributeur (s'il s'est positionné sur cette offre en premier par exemple). Il se peut également que le choix du fabricant soit basé sur les relations qu'il entretient avec le distributeur. C'est pour cela qu'il est important d'entretenir de bonnes relations et d'essayer de développer du business avec plusieurs fabricants en parallèle.

Tous ces éléments peuvent ensuite être exportés par le créateur du block-diagram vers le CRM de Future Electronics qui est le logiciel regroupant la totalité des données relatives à la partie business de l'entreprise. Et c'est donc à partir de ces données que les performances de chaque employé sont mesurées sur une année.

Ce logiciel, BlockDiagram, se devait d'être une grande aide pour les ingénieurs/ vendeurs ainsi que pour les managers pour visualiser et mesurer les performances de chacun des employés. Cependant, les besoins de l'entreprise ayant beaucoup évoluées depuis la création du logiciel , ce dernier est devenu petit à petit vieillissant et de moins en moins utilisé par les employés.

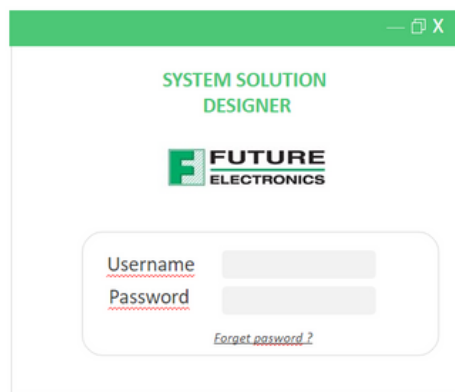
Toujours dans l'optique de développer le business au sein de l'entreprise, Etienne Lanoy a décidé, au niveau Européen, d'entamer une refonte d'une grande partie de logiciels et outils de l'entreprise. Avec la montée en puissance des navigateurs webs qui ont atteint des niveaux de performances très intéressants (nous en parlerons plus en détail dans le chapitre suivant), c'est à travers une application web que mon manager souhaitait faire interagir les utilisateurs avec l'interface du nouveau "System Solution Designer", qui aura pour vocation de remplacer BlockDiagram.

C'est donc ce projet dans sa globalité qui m'a été confié dans le contexte de ce stage.

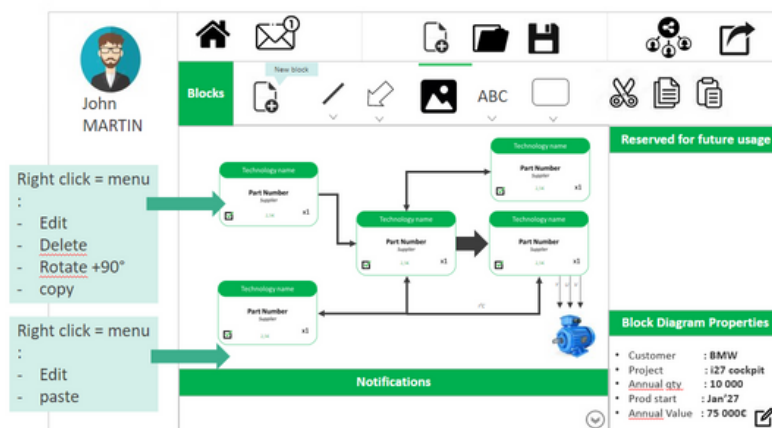
DÉCOUVERTE DU PROJET

Tout a commencé par une réunion, la première semaine de mon stage, que mon manager a organisé dans les bureaux de Future Electronics à Paris. Il a commencé par me présenter les nouveaux besoins de l'entreprise ainsi que ses attentes pour ce nouveau logiciel. Il m'a ensuite présenté une première maquette qu'il avait pu réaliser dans laquelle se trouvaient les éléments clés du logiciel ainsi que l'interface utilisateur qu'il souhaitait retrouver. Ainsi qu'une liste des fonctionnalités à implémenter.

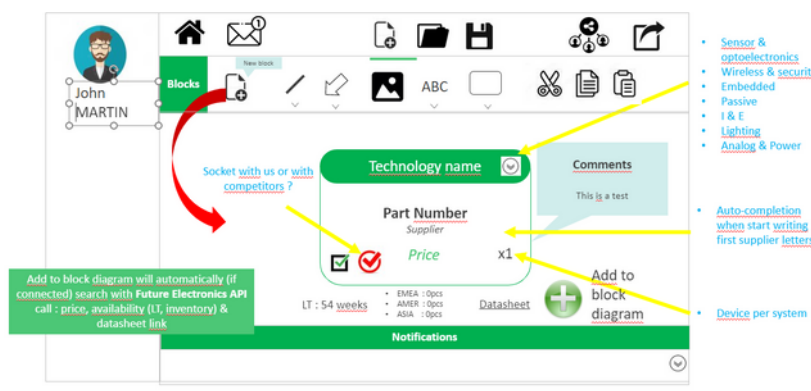
Voici les maquettes présentées :



Page de connexion



Design Center



Éléments attendus dans un block



MÉTHODOLOGIES DE TRAVAIL

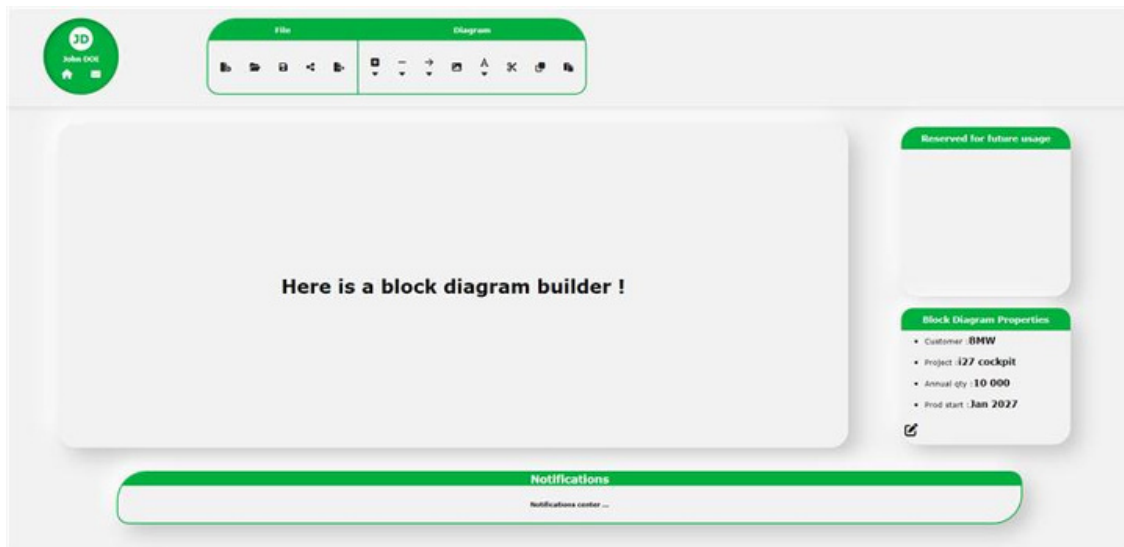
Nous avons ensuite mis en place avec mon manager différentes méthodologies de travail. Le premier point consistait à lui donner un retour hebdomadaire sur mon avancée. En général, chaque vendredi en fin de journée j'effectuais un récapitulatif de toutes les tâches accomplies au cours de la semaine passée que j'envoyais par mail à Etienne.

Dans le cadre d'un projet de grande ampleur comme celui-ci, un des outils les plus utiles pour un développeur est Git. Il s'agit d'un système de contrôle de version décentralisé qui permet de suivre et de gérer les changements apportés à un ensemble de fichiers, en particulier le code source d'une application. Il offre aux développeurs la capacité de travailler de manière collaborative et parallèle, tout en assurant une traçabilité et une réversibilité des modifications. Dans le cadre du développement de mon application web, j'ai été amené(e) à utiliser GitHub, une plateforme en ligne basée sur Git. GitHub offre non seulement toutes les fonctionnalités de versionnage de Git, mais ajoute également des outils de collaboration comme la gestion de projets, le suivi des problèmes et la possibilité de contribuer à des projets open source. Ainsi, l'utilisation de GitHub m'a permis d'organiser efficacement mon code et de bénéficier d'une sauvegarde en ligne de l'intégralité de mon travail. Cela a considérablement augmenté l'efficacité de mon processus de développement et m'a offert un moyen sûr de conserver et de partager mes progrès tout au long du stage.

GitHub ne se limite pas à être un simple outil de versionnage ; il offre également des solutions pour l'intégration continue (CI) et le déploiement continu (CD). L'intégration continue permet d'automatiser la vérification de chaque modification apportée au code, garantissant ainsi que les nouvelles contributions n'introduisent pas d'erreurs. Cela se traduit souvent par la mise en place de tests automatiques qui s'exécutent à chaque fois qu'un nouveau code est poussé sur le dépôt. Le déploiement continu, quant à lui, vise à automatiser la livraison de ce code dans un environnement de production, garantissant une mise à jour fluide et régulière de l'application sans nécessiter d'interventions manuelles répétées. GitHub Actions est l'un des outils de GitHub qui facilite la mise en œuvre de ces processus CI/CD. Grâce à cet outil, j'ai pu automatiser une suite de tâches, allant de l'exécution des tests à la mise en ligne de l'application, garantissant ainsi la qualité et l'efficacité du processus de développement jusqu'au déploiement. Cela m'a donc permis de rendre disponible sur internet (par le biais d'AWS) une première version fonctionnelle du projet afin de le faire tester à quelques utilisateurs (mon manager notamment).

Le second point important que je souhaiterais apporter est celui de certaines deadlines qui m'ont été données au cours de l'évolution de l'application. Nous avons pu ainsi définir différentes features que l'on devait retrouver dans l'application, ce qui donnerait lieu à différentes versions du projet.

Basé sur les maquettes vues précédemment, j'ai réalisé une première version de l'interface utilisateur dans les deux semaines qui ont suivis notre entretien à Paris :



UI datant du 22/11/2022

Le second outil qui a considérablement aidé la gestion et l'avancement de ce projet est Teams. Au sein de Future Electronics, la suite Microsoft est massivement utilisée notamment pour les présentations (PowerPoint), pour les bilans (PowerBI) ainsi que Excel, Outlook pour la messagerie et Word pour les comptes rendus.

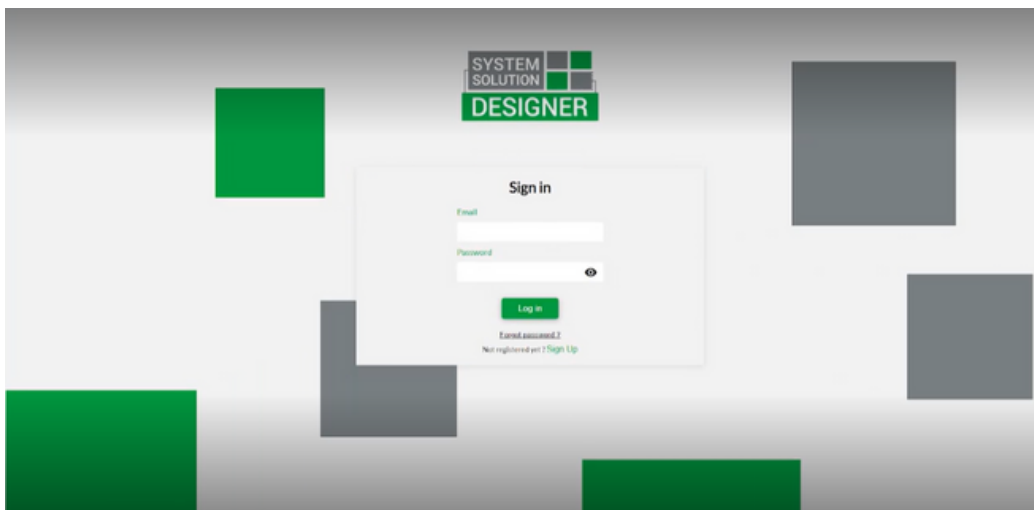
Mon manager étant basé dans l'Ouest de la France et moi à Lyon, la plupart de nos échanges se sont donc fait via Teams. Je me suis également servi d'Outlook pour les comptes rendus hebdomadaires que je faisais à Etienne, pour les questions plus importantes et nécessitant un retour rapide, teams était la solution qu'Etienne préconisait.

A chaque nouveau rendu que je proposais, Etienne me donnait des feedbacks écrits, soit par mail soit via des documents Words (dans le cas de feedbacks plus détaillés). C'est comme cela que nous avons pu avancer de versions en versions, en apportant les modifications nécessaires après de nombreux tests d'essais/ erreurs qui nous ont permis de voir les éléments nécessaires à implémenter dans le logiciel.

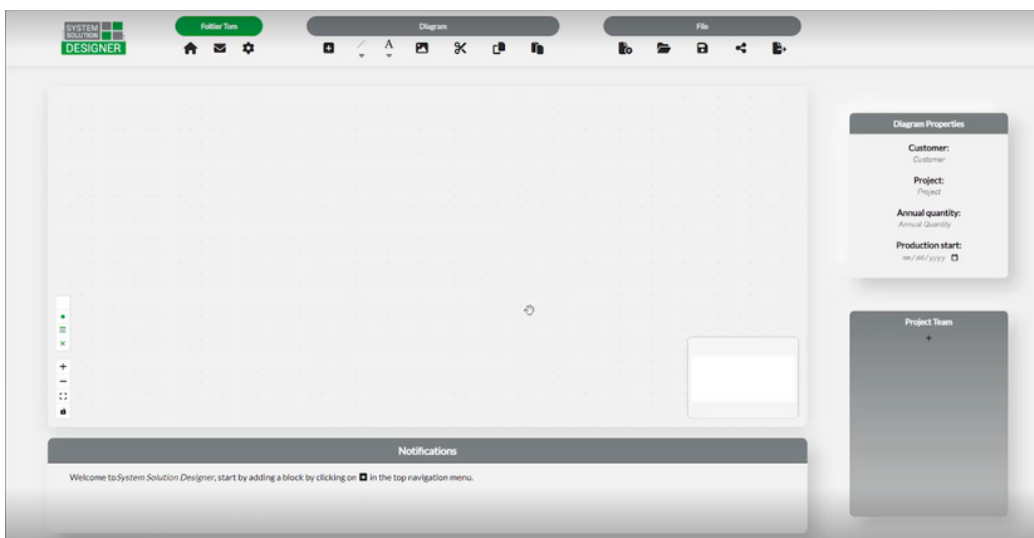
Etienne me proposait également des réunions Teams toutes les deux semaines environ pour faire un point et échanger de vive voix.

La seconde version du projet devait servir de démonstration lors du meeting européen annuel de Future Electronics (l'AEU) qui se tenait cette année à Malaga en Espagne et qui s'est déroulé sur une semaine (en Avril 2023) et auquel j'ai pu participer. Lors de la conférence d'ouverture de ce séminaire, Etienne (qui était le maître de cérémonie) a présenté System Solution Designer à près de 200 ingénieurs et vendeurs européens au sein de Future Electronics à travers quelques slides ainsi qu'une vidéo de l'application que j'ai pu réaliser en amont.

Voici ce à quoi ressemblait l'application à ce moment précis :



Page de connexion



Design Center

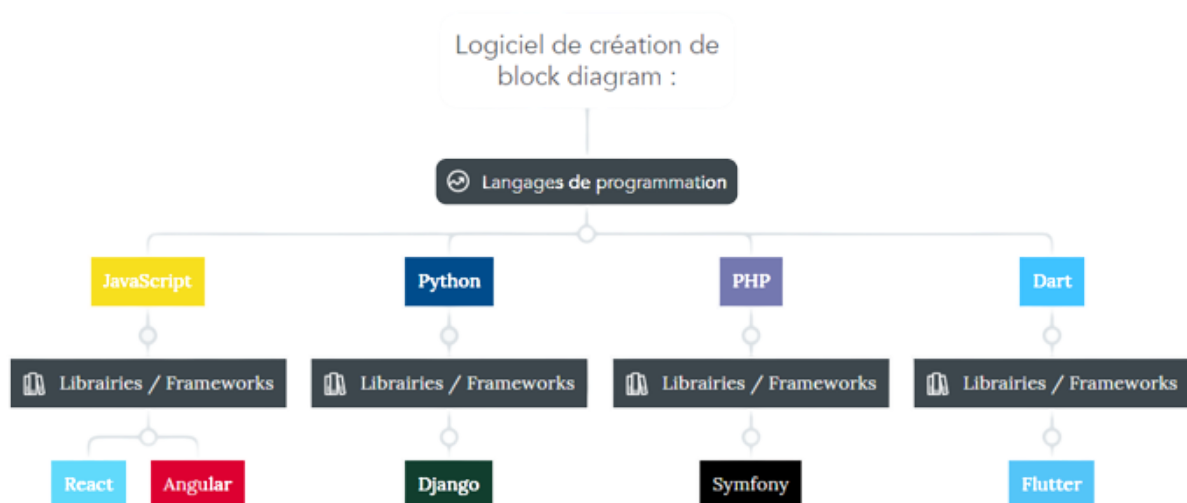
IV – CONTENU DU STAGE

TECHNOLOGIES UTILISÉES ET DÉVELOPPEMENT

PRÉSENTATION DES TECHNOLOGIES

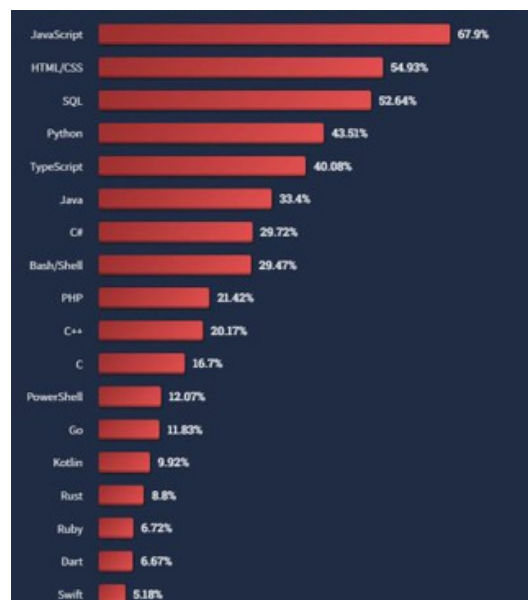
Le choix des technologies a été la première étape dans la création de cette nouvelle application. Suite à mon entrevue à Paris avec Etienne, j'ai effectué de nombreuses recherches sur les technologies qui pourraient nous permettre de développer System Solution Designer. La première version que j'ai présentée (page 19), a été réalisée en pure HTML/CSS pour donner un premier aperçu à mon manager de ce qui était possible d'obtenir dans le cadre d'une application web. Ces langages étant les langages de base des navigateurs webs, il était quasiment certains que je réutiliserais cette base pour l'application finale, peu importe les technologies que je choisirais par la suite pour ajouter toutes les features attendues.

J'ai donc commencé par me documenter sur les différentes technologies web existantes de manière générale (sans entrer vraiment dans le détail de ce que j'aurais besoin pour la suite), cela m'a permis de découvrir certains frameworks que je ne connaissais pas comme Symfony et Laravel qui sont des frameworks du langage PHP. J'ai, suite à cela, préparé une présentation pour mon manager à travers laquelle je lui ai proposé les différents langages/ frameworks qui pourraient nous être utiles. Voici un aperçu des technologies abordées lors de cette présentation :



Je me suis servi premièrement de mes connaissances acquise lors de ma formation à l'ISEN pour proposer ces technologies. En effet, nous avons été amenés à travailler avec différents langages ce qui m'a permis d'avoir une vision globale de : à quoi sert tel ou tel langage (ce qu'il permet de faire), pourquoi utiliser tel framework plutôt qu'un autre, etc... Ce qui m'a permis d'avoir un regard critique sur ces derniers.

Deuxièmement, je me suis servi des ressources du web traitants des dernières technologies en vogue et des comparatifs sur des sites extrêmement reconnus dans le monde du développement de manière général comme StackOverflow sur lequel j'ai pu trouver ce graphique :



StackOverflow 2022: les langages les plus utilisés

Pour finir, je me suis servi de mes contacts travaillant dans le domaine du développement web (famille, amis, collègues, camarades de classe) pour avoir des avis complémentaires et des points de vue différents sur des technologies que je ne connaissais pas forcément.

Tout ce processus m'a permis, après près d'un mois de recherche et questionnements, d'opter pour le langage **Javascript**.

Présentation de JavaScript :

JavaScript est un langage de programmation interprété, principalement connu pour sa capacité à ajouter de l'interactivité et des fonctionnalités dynamiques aux sites web. Il s'exécute côté client, directement dans le navigateur de l'utilisateur, ce qui le rend idéal pour créer des expériences utilisateur réactives. Né à la mi-1995, JavaScript a évolué bien au-delà de ses origines modestes pour devenir l'un des piliers du développement web, aux côtés des technologies HTML et CSS. Aujourd'hui, grâce à des environnements comme Node.js, JavaScript peut également être exécuté côté serveur, ce qui le rend encore plus polyvalent.

Pourquoi avoir choisi JavaScript pour mon application web ?

J'ai opté pour JavaScript en raison de sa flexibilité et de sa popularité dans le monde du développement web (98% des sites webs utilisent Javascript). Ses vastes bibliothèques et frameworks, tel que React notamment m'a permis d'accélérer le développement et de bénéficier de fonctionnalités avancées sans avoir à les construire à partir de zéro. L'énorme communauté de développeurs JavaScript offre également une richesse de ressources et d'outils, ce qui m'a aidé à résoudre rapidement les problèmes et à adopter les meilleures pratiques du secteur.

Une fois le langage de programmation déterminé et approuvé par mon manager, j'ai poursuivi le développement de mon interface graphique entamé précédemment. Je me suis tourné vers React, qui est une librairie Javascript pour le développement de la partie front-end de l'application.

Présentation de React :

React est une bibliothèque JavaScript développée par Facebook pour construire des interfaces utilisateur dynamiques. Grâce à son modèle basé sur des composants, React permet de créer des UIs complexes qui peuvent facilement évoluer en fonction des données, tout en optimisant les performances grâce à un algorithme de réconciliation sophistiqué.

Pourquoi avoir choisi React pour mon front-end ?

React s'est imposé comme un choix évident en raison de sa flexibilité, de son écosystème riche et de sa capacité à créer des applications web réactives avec une excellente expérience utilisateur. La réutilisabilité des composants permet d'accélérer le processus de développement, et l'approche basée sur l'état de React m'a permis de gérer facilement les interactions utilisateur dynamiques.

React dispose également d'une vaste communauté de programmeurs, il existe donc de nombreuses librairies développées pour React, c'est principalement pour cette raison que je me suis tourné vers React plutôt que vers Angular ou VueJS qui sont ses deux concurrents directs. En effet, après de nombreuses recherches sur les librairies proposant la création et la gestion de diagrammes, j'ai fini par découvrir GoJS qui avait l'inconvénient d'être une librairie payante pour la phase de production (déploiement de l'application à grande échelle, sur le web). Puis en affinant mes recherches j'ai découvert la librairie ReactFlow qui avait l'avantage d'être totalement gratuite (sauf pour bénéficier de fonctionnalités avancées).

Présentation de React Flow :

React Flow est une bibliothèque pour React conçue pour construire des interfaces de diagrammes, de graphiques et de flux de manière intuitive et flexible. Elle offre des fonctionnalités telles que le déplacement des nœuds, le zoom, le panning, des liaisons courbées, des points d'ancrage personnalisables, et bien plus. React Flow est idéal pour créer des éditeurs de diagrammes visuels, allant des diagrammes simples aux applications plus complexes comme celles pour l'édition de flux de travail ou la visualisation de données.

Pourquoi avoir choisi React Flow ?

Envisageant d'intégrer une fonctionnalité de diagramme dans une application React, React Flow était l'option la plus attrayante en raison de sa flexibilité et de sa facilité d'intégration. Elle est hautement personnalisable, ce qui permet de créer une expérience utilisateur adaptée à des besoins spécifiques comme ceux que nous avons pour l'entreprise. La bibliothèque prend en charge une variété de nœuds (blocks) et de connecteurs, et elle est optimisée pour garantir de bonnes performances même avec de grands diagrammes. De plus, sa nature modulaire et extensible signifie qu'il est possible d'ajouter des fonctionnalités supplémentaires si nécessaire, comme des interactions plus avancées, des styles personnalisés ou des comportements spécifiques à un domaine.

C'est donc en utilisant ces différentes technologies que j'ai pu réaliser la version plus sophistiquée présentée lors de l'AEU ([page 20](#)). Comme nous l'avons vu, tout cela m'a permis de designer la partie front-end de l'application, à savoir la partie visible par l'utilisateur, et avec laquelle il va pouvoir interagir. Cependant, pour que notre application soit complète et fonctionnelle, il est nécessaire de stocker des données, liées à l'utilisateur (création d'un compte, identifiants, mot de passe, etc..) ainsi qu'à ses créations (les diagrammes qu'il va créer). Il me fallait donc un back-end.

Ma formation et mes expériences personnelles m'ont toujours poussées à travailler seulement sur la partie front-end des applications que j'ai développées et je n'étais jusqu'à ce moment pas familière avec les technologies back-end. Je me suis donc à nouveau servi des toutes les ressources que j'avais à disposition (contacts, web, connaissances) pour déterminer la manière dont j'allais m'y prendre pour gérer tout cela.

Tout a commencé par le stockage des données, jusqu'à présent, je faisais tourner mon application en local (seulement accessible depuis ma machine), je pouvais y créer des blocks-diagrams et remplir les blocks mais impossible de les sauvegarder, il me fallait une base de données.

La première question que je me suis posée est l'utilisation d'une base de données SQL (que je connaissais bien grâce à mon parcours scolaire) ou bien d'une base de données NoSQL (que je ne connaissais pas encore). Je me suis donc à nouveau documenté sur ce sujet, j'ai également contacté mon ancien professeur de l'ISEN qui, durant 3 années, nous a donné des cours sur les bases de données et suite à ces échanges je me suis tourné vers l'option NoSQL, qui correspondait le mieux aux besoins que nous avions, à savoir des données non-structurées et pas ou peu de relation entre les données. Et c'est vers la base de données MongoDB que je me suis orienté.

Présentation de MongoDB :

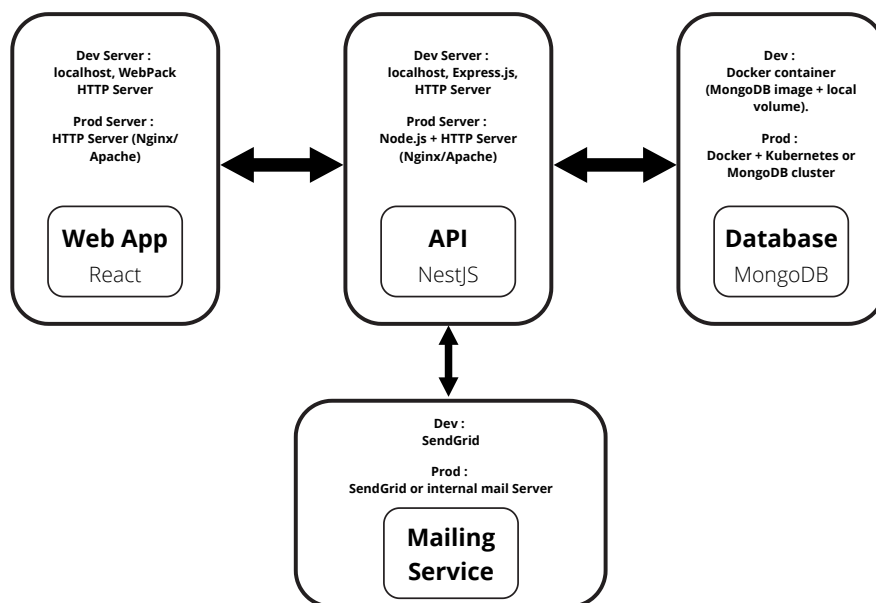
MongoDB est une base de données orientée documents qui offre une haute performance, une haute disponibilité et une facilité de mise à l'échelle. Contrairement aux bases de données relationnelles traditionnelles qui stockent les données sous forme de tables, MongoDB utilise un format semblable à JSON appelé BSON (Binary JSON) pour stocker des enregistrements sous forme de documents. Ces documents sont organisés en collections, qui sont l'équivalent des tables dans les bases de données relationnelles, mais sans schéma fixe. Cela signifie que chaque document dans une collection peut avoir des champs différents.

Pourquoi avoir choisi MongoDB ?

L'absence de schéma fixe de MongoDB offre une grande flexibilité pour gérer les données, ce qui est particulièrement avantageux lorsqu'on travaille avec des applications en évolution rapide où les besoins en matière de données peuvent changer. Sa nature orientée document permet une représentation intuitive des objets métier et facilite le stockage de structures de données complexes comme les listes ou les objets imbriqués, très utile pour les données issues des diagrams.

Avec des fonctionnalités comme la réplication ou la mise à l'échelle horizontale. MongoDB est également conçu pour la haute disponibilité et la distribution à grande échelle. Son écosystème riche, comprenant des outils pour le monitoring, la sauvegarde et la restauration, ainsi que son driver pour plusieurs langages de programmation, en font une solution complète pour les besoins modernes en matière de gestion de données.

Cependant, nativement, React ne peut pas "échanger" directement avec une base de données car le code de React est exécuté côté navigateur et cela poserait de nombreux problèmes (notamment en matière de sécurité) d'effectuer des requêtes directes depuis React vers une base de données. La solution à cela est de mettre en place une API. Pour comprendre le fonctionnement, voici l'architecture que j'ai réalisé et sur laquelle je me suis basée pour poursuivre le développement de l'application (l'application React communique avec l'API via des requêtes HTTP) :



System Solution Designer Architecture

Il ne me manquait plus qu'un langage de script côté serveur pour pouvoir développer cette API. Le langage Javascript ayant pris une telle ampleur, un environnement nommé Node.js a été développé pour permettre aux développeurs Javascript d'écrire du code dans leur langage de prédilection, compréhensible par un serveur (Node.js en l'occurrence).

Présentation de Node.js :

Node.js est un environnement d'exécution JavaScript construit sur le moteur JavaScript V8 de Chrome. Contrairement à JavaScript traditionnel qui s'exécute dans le navigateur, Node.js permet d'exécuter JavaScript côté serveur, ce qui ouvre la voie à la création d'applications web complètes avec une seule technologie. Grâce à son système de gestion de packages, npm, Node.js possède une vaste bibliothèque de modules qui facilitent le développement rapide d'applications.

Pourquoi avoir choisi Node.js pour mon backend ?

J'ai choisi Node.js pour sa performance et sa capacité à gérer de nombreuses connexions simultanées avec une faible latence, grâce à son architecture non bloquante et orientée événements. La cohérence de travailler avec JavaScript à la fois sur le front-end et le back-end a également simplifié le processus de développement, favorisant une meilleure compréhension et une intégration plus fluide entre les deux.

De la même manière que dans le Javascript traditionnel, on retrouve des frameworks/ bibliothèques dans le monde Node.js, l'un d'eux NestJS permet la création de backends sécurisés, modulables et extrêmement performants.

Présentation de NestJS :

NestJS est un framework pour construire des applications server-side efficaces, fiables et évolutives en Node.js. Il utilise TypeScript par défaut comme langage de programmation (il s'agit d'une surcouche Javascript) et combine des éléments de la programmation orientée objet, de la programmation fonctionnelle et de la programmation fonctionnelle réactive.

Pourquoi avoir choisi NestJS pour mon backend ?

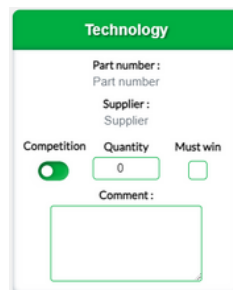
Opter pour NestJS était une décision naturelle étant donné sa structure modulaire et son architecture fortement inspirée d'Angular (framework Javascript que j'ai eu l'occasion de découvrir durant mes études à l'ISEN), facilitant la création d'une application maintenable et organisée. Son support natif pour TypeScript m'a offert des avantages en matière de typage statique, améliorant la qualité du code et simplifiant le débogage.

Structure de données et interface graphique

Fonctionnement de System Solution Designer :

- L'utilisateur se connecte à l'application, il peut alors choisir de **créer** un nouveau diagram ou bien d'en **éditer** un existant.
- Dans le cas de la création d'un nouveau diagram, l'utilisateur peut choisir entre : créer un diagram **"from scratch"** (vide) ou bien de se baser sur un **template**.
- Il doit ensuite renseigner les **informations** relatives au diagram (client, nom du projet, date de lancement du projet, etc...), données qui correspondent à l'objet *diagramInformation* que l'on a vu précédemment en base de données.
- L'utilisateur est alors redirigé vers le playground, le "Design Center" dans lequel il peut commencer à **dessiner** son diagram.
- Au moment de la rédaction de ce rapport, il existe trois types de **nodes** (blocks) que l'utilisateur peut ajouter dans son diagram :

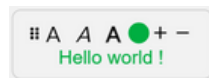
- BlockNode (ajouter un composant au projet) :



- ImageNode (importer une image, ici le logo de Future) :



- TextNode (écrire du texte et le styliser) :



- Chaque node peut ensuite être relié à un autre par le biais des **edges**, il en existe actuellement 6 différents :



- L'utilisateur peut également ajouter des **collaborateurs** au projet afin de leur partager le block diagram et de leur donner l'accès pour l'éditer à leurs tours s'ils le souhaitent.

N.B Un template est un diagram créé dans le but d'être réutilisé comme base pour la création d'un nouveau diagram. En effet, il existe de nombreux projets qui réutilisent les mêmes composants. Les templates sont donc créés d'une part pour faire gagner du temps aux créateurs de nouveaux diagrams et d'autre part pour éviter les oublis.

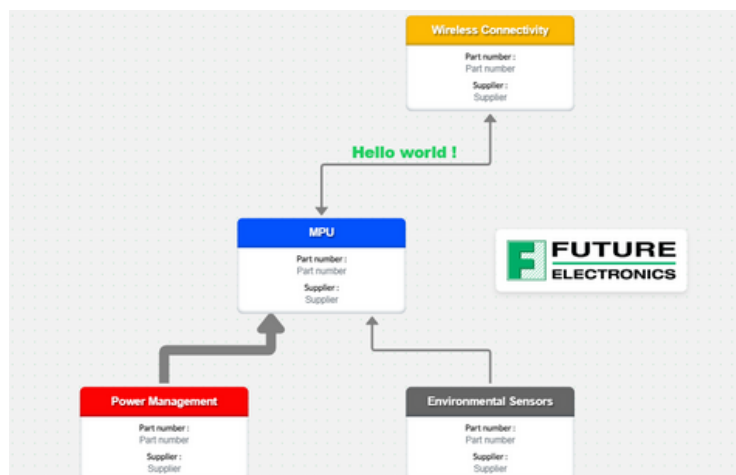
```

    _id: ObjectId('64d0cbb1a10c21c77ac62ee2')
    ▾ diagramInformations: Object
      id: "64d0cbb1a10c21c77ac62ee2"
      customer: "Mercedes"
      project: "C63S"
      annualQuantity: "50000"
      productionStart: "2024-01-01"
      fromTemplate: ""
    ▸ nodePositions: Array
    ▸ nodeContents: Array
    ▸ edges: Array
    ▾ creator: Object
      _id: ObjectId('64943b4b620c87a5268ebcbe')
      email: "tom.foltier@futureelectronics.com"
      firstName: "Tom"
      lastName: "Foltier"
      role: "administrator"
      status: "F.A.E"
    ▸ collaborators: Array
    isTemplate: false
    ▾ templateInformations: Object
      createdAt: 2023-08-07T10:47:13.656+00:00
      updatedAt: 2023-08-08T15:13:42.745+00:00
  
```

Structure de données d'un diagram dans MongoDB

- **_id** : identifiant unique associé au diagram lors de sa création.
- **diagramInformations** : informations relative au diagram (client, projet, etc...).
- **nodePositions** : coordonnées de chaque block dans le flux (position dans le diagram).
- **nodeContents** : contenu de chaque block (rempli par l'utilisateur via l'interface graphique (image de droite)).
- **edges** : fait référence aux liens (flèches et lignes) entre chaque block, tracés par l'utilisateur pour relier les blocks entre eux.
- **creator** : objet contenant les informations du créateur du diagram (son nom, son email, etc...).
- **collaborators** : liste des collaborateurs ajoutés au projet (y ayant donc accès) par le créateur.
- **isTemplate** : booléen indiquant si le diagram est un template ou non.
- **templateInformations** : si le diagram est un template, informations à propos de ce template.
- **createdAt / updatedAt** : dates de création et de mise à jour du diagram .

Voici un exemple de diagram pouvant être créé à ce stade dans System Solution Designer :

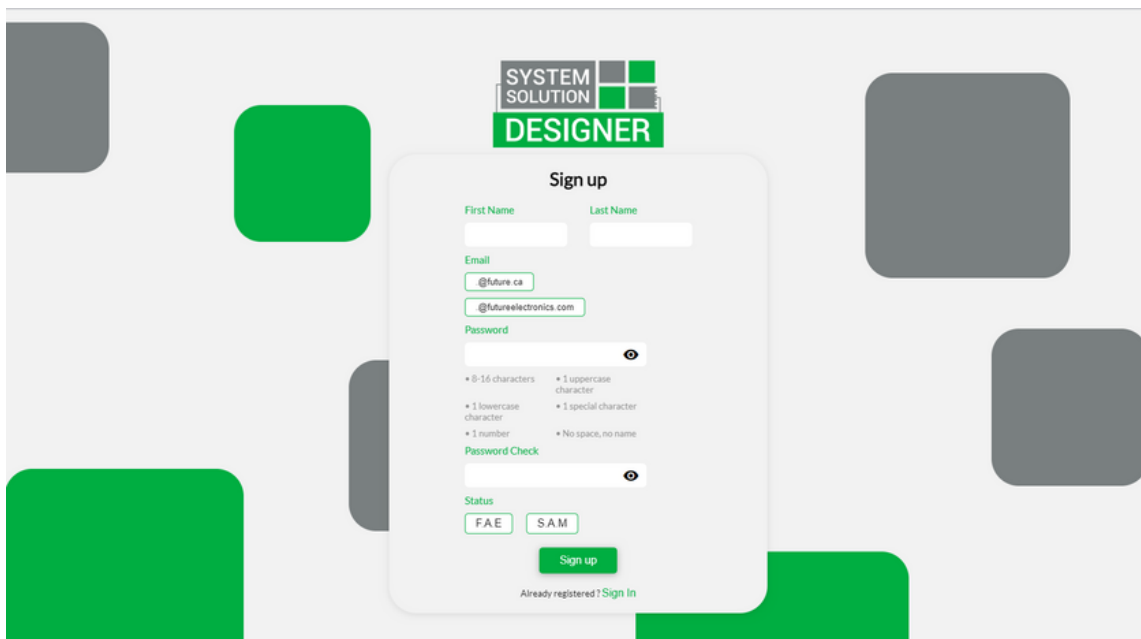


```
_id: ObjectId('64943b4b620c87a5268ebcce')
firstName: "Tom"
lastName: "Foltier"
email: "tom.foltier@futureelectronics.com"
password: [REDACTED]
status: "F.A.E"
verifiedEmail: true
role: "administrator"
manage: Array
createdAt: 2023-06-22T12:15:07.477+00:00
updatedAt: 2023-08-08T15:27:01.847+00:00
```

Structure de données d'un utilisateur dans MongoDB

- **_id** : identifiant unique associé à l'utilisateur lors de sa création.
- **firstName** : prénom de l'utilisateur.
- **lastName** : nom de famille de l'utilisateur.
- **email** : email de l'utilisateur.
- **password** : mot de passe haché (masqué ici pour des raisons de sécurité).
- **status** : statut de l'utilisateur dans l'entreprise.
- **verifiedEmail**: booléen servant à vérifier si l'email a bien été confirmé.
- **role**: role de l'utilisateur dans l'application (il en existe 3: creator, administrator, user).
- **manage**: si le diagram est un template, informations à propos de ce template.
- **createdAt / updatedAt** : dates de création et de mise à jour de l'utilisateur.

Voici la page de création de compte dans System Solution Designer :



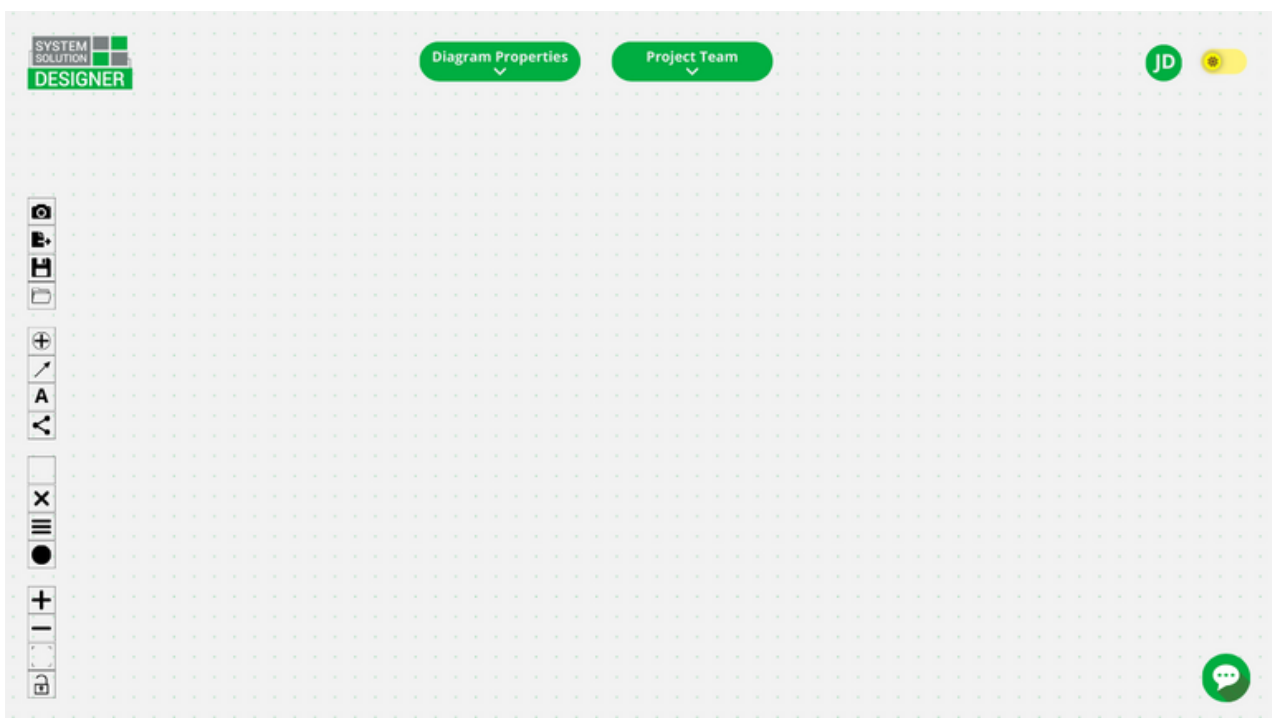
Page d'inscription à System Solution Designer

Comme nous avons pu le voir dans les différentes images traitant de l'interface utilisateur, au cours du développement et des réflexions de la part d'Etienne et de moi-même, cette interface a été amenée à changer.

D'abord pour des raisons de style, la première interface que j'ai réalisée (page 19) avait seulement pour but de servir de "fondations" à l'application, positionner les briques de bases pour pouvoir commencer à développer quelque chose de fonctionnel, avec chaque espace dédié à sa fonctionnalité et les éléments allant servir à l'utilisateur d'interagir avec le logiciel (boutons, formulaires, champ de texte, etc...). Cependant le style de l'application n'était (à mon goût) pas assez moderne et pas assez professionnel.

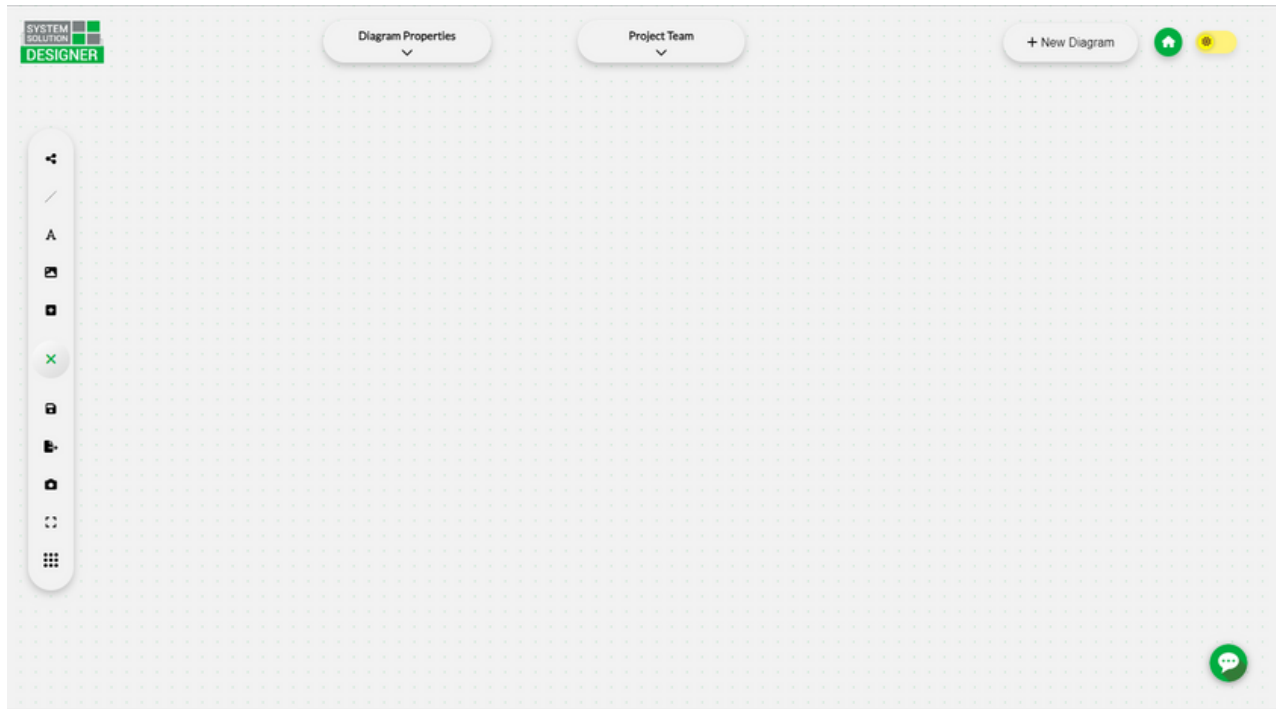
D'autre part, en travaillant tous les jours sur cette interface, je me suis vite rendu compte que certains éléments n'étaient pas nécessaires, que d'autres manquaient ou encore que la logique initialement prévue n'était ni optimale pour le code (la logique de code dépend aussi de l'interface utilisateur que l'on souhaite implémenter) ni pour le parcours utilisateur.

Après de nombreuses itérations sur le style et la logique de l'interface et de nombreuses réflexions sur le parcours utilisateur. J'ai fini par réaliser la maquette d'une toute nouvelle interface utilisateur que j'ai présenté à mon manager :



Maquette UI System Solution Designer, Design Center

Mon manager ayant validé ce design, j'ai alors entamé un processus de refonte graphique, ce qui a donné cette UI finale (qui est la version actuelle de System Solution Designer) :



UI System Solution Designer, Design Center

Toutes ces recherches et itérations à propos de l'UX/UI, m'ont permis d'une part d'appliquer toutes les connaissances que j'ai acquises au cours des dernières années que ce soit à l'école ou dans des projets personnels et d'autre part de développer de nouvelles compétences ainsi que ma créativité.

Je n'avais jamais eu la chance d'aller aussi loin dans le travail de recherche UX/UI dans le cadre d'un projet professionnel, rien ne doit être laissé au hasard. Il est important de tester toutes les fonctionnalités implémentées dans le détail, de penser à tout ce que pourrait faire l'utilisateur et tout ce dont il aurait besoin dans le but de proposer la meilleure expérience possible et de réduire au maximum les frictions et autres bugs pouvant altérer cette expérience.

Aspect sécuritaire

Nous avons jusqu'à présent étudié les données liées à un diagram dans sa globalité. La deuxième grosse partie du développement de cette application fut la gestion de l'**authentification** de l'utilisateur avec : la création d'un compte, la connexion, le changement de mot de passe, ainsi que l'accès aux données de cet utilisateur. L'un des défis majeurs du développement d'applications modernes est la garantie de la sécurité, tant du point de vue de l'infrastructure que des données, c'est en cela que réside la difficulté de cette partie.

En effet, les données stockées en bases de données et transitant dans l'application System Solution Designer sont des données extrêmement confidentielles pour l'entreprise. Les projets créés dans l'application contiennent tous les éléments qui sont à la base du business de l'entreprise, ces données doivent donc rester totalement innaccessibles pour quiconque ne faisant pas partie de Future Electronics ou du projet.

À cet égard, NestJS m'a offert un ensemble d'outils et de bonnes pratiques qui ont grandement simplifié ce processus. D'abord, grâce à l'intégration native de NestJS avec **Passport.js**, j'ai pu mettre en place une authentification robuste, permettant de vérifier l'identité des utilisateurs avant qu'ils n'accèdent à certaines ressources. De plus, NestJS encourage le principe de la séparation des préoccupations, me permettant de définir clairement des "**guards**" et des "**interceptors**" pour contrôler finement l'accès aux routes et aux méthodes de l'API. En matière de sécurisation des données, le système de validation intégré à NestJS, associé à l'utilisation de TypeScript, m'a assuré que les données entrantes sont du type attendu, prévenant ainsi d'éventuelles injections ou manipulations malveillantes. Enfin, la documentation exhaustive de NestJS propose des recommandations sur la gestion sécurisée des secrets, la prévention des attaques CSRF, et d'autres aspects cruciaux de la sécurité web. Grâce à toutes ces fonctionnalités, j'ai pu développer mon application en ayant confiance dans sa capacité à résister aux menaces et à protéger les données sensibles.

L'un des aspects essentiels de la sécurisation des applications est la manière dont les mots de passe des utilisateurs sont stockés. Enregistrer des mots de passe en clair dans une base de données est une pratique risquée, car en cas de fuite ou de compromission de la base, les informations sensibles des utilisateurs sont directement exposées. C'est là qu'intervient **bcrypt**, une bibliothèque de hachage de mots de passe largement reconnue pour sa robustesse. Lors de l'inscription d'un utilisateur, au lieu de stocker son mot de passe directement, je l'ai d'abord passé à travers bcrypt pour en obtenir une version hachée. Bcrypt utilise un "sel" (une série de caractères aléatoires) qu'il ajoute au mot de passe avant le hachage, rendant chaque hachage unique, même pour des mots de passe identiques.

Ainsi, même si deux utilisateurs ont le même mot de passe, les hachages stockés dans la base de données seront différents.

Lors de la connexion, le mot de passe fourni par l'utilisateur est de nouveau haché avec le sel original et comparé au hachage stocké dans la base de données. Si les deux correspondent, l'authentification est réussie. Sinon, l'accès est refusé. Ce mécanisme, en plus de masquer le mot de passe réel, offre une protection contre les attaques par force brute. En effet, bcrypt est conçu pour être intentionnellement lent, rendant inefficaces les tentatives d'attaquants de deviner rapidement de nombreux mots de passe.

Grâce à la combinaison de NestJS et bcrypt, j'ai pu assurer une gestion sécurisée des mots de passe, renforçant ainsi la confiance de nos utilisateurs et la robustesse globale de notre application.

FORMATIONS

Après une prépa intégrée MPSI à l'ISEN, je me suis naturellement orienté vers la spécialité "Développement Logiciel, Big Data et Cloud Computing". En effet, en parallèle de mes études, je profitais de mon temps libre pour développer des sites webs et me former sur des nouvelles technologies web, il s'agissait pour moi de l'option la plus adéquat pour poursuivre ma formation dans ce domaine. Au cours de ce cursus de deux ans, j'ai été formé sur de nombreuses technologies diverses et variées. Nous avons étudié des langages de bas niveaux comme le C, des langages intermédiaires : C++ / Java / Python, des SGBD comme Oracle ou Postgres, des frameworks webs comme Angular, nous avons également suivis plusieurs formations sur l'Intelligence Artificielle et les réseaux de neurones artificiels ainsi que sur les infrastructures et le déploiement comme Docker, AWS, IBM Cloud Computing ou encore Kubernetes. En parallèle de cela je me suis formé sur des technologies complémentaires comme PHP et React Native. Tout cela m'a permis d'avoir une vue d'ensemble sur les technologies existantes et de leur utilité ainsi qu'une base solide pour pouvoir continuer à construire dessus.

Lorsque j'ai intégré l'entreprise, je ne connaissais pas encore la mission qui me serait attribuée dans le détail, je connaissais les grandes lignes mais je n'avais aucune idée des technologies que j'allais devoir aborder durant ce stage. Comme nous l'avons vu dans le chapitre précédent, le choix des technologies que j'allais utiliser s'est fait en fonction des besoins pour le projet, j'ai donc dû m'adapter à cela et me former sur les technologies que j'allais utiliser et que je ne connaissais pas.

Comme dit précédemment, je disposais de certaines bases sur diverses technologies mais je ne maîtrisais pas encore celles que j'ai été amené à utiliser dans le cadre de ce projet. En effet, après avoir choisis les langages de programmation et frameworks que j'allais utiliser à l'aide de mon manager, j'ai immédiatement commencé à développer l'application. Commenant par la partie front-end, j'ai dû apprendre à maîtriser le framework React. Etant légèrement familier avec le Javascript, j'avais les bases pour comprendre cet écosystème. Cependant, React regorge de principes et méthodes qui lui sont propres. J'ai alors commencé à me documenter dans la documentation officielle de react, puis à visionner des tutoriels sur Youtube puis je me suis aperçu qu'il me fallait une formation plus complète et plus détaillée pour vraiment comprendre les tenants et aboutissants de ce framework. J'ai donc opté pour une formation de la platform Udemy qui m'a été conseillée par un ancien professeur de l'ISEN.

J'ai alors commencé à décomposer mes journées de travail en deux parties, je suivais la formation le matin puis l'après-midi, j'appliquais ce que j'avais appris directement dans l'application System Solution Designer. De cette manière, j'ai pu mettre en oeuvre directement ce que j'apprenais au fur et à mesure, cette méthode a été très efficace dans mon chemin vers la maîtrise de React.

Comme nous l'avons vu précédemment, j'ai rapidement été amené à travailler sur la partie back-end en parallèle de mon avancée sur la partie front-end. Ayant une formation et des projets personnels surtout orientés front-end, le milieu du back-end était tout à fait nouveau pour moi. Bien que les technologies et les langages choisis pour le back-end et le front-end soient similaires dans le cas de notre application, les concepts fondamentaux et les frameworks utilisés sont bien différents à plusieurs égards. J'ai donc du de la même manière que pour React me former sur la partie back-end. J'ai commencé par choisir une formation sur Node.js pour avoir les bases d'un serveur Node.js et bien comprendre son fonctionnement, j'ai ensuite continué sur Express qui est st un framework d'application web back-end pour la création d'API RESTful avec Node.js. Puis j'ai fait la découverte de NestJS qui est un framework basé lui-même sur Express pour la construction de mon API.

J'ai donc utilisé la même méthodologie que pour l'apprentissage de React pour me forme sur le back-end, ce qui m'a permis de mettre en place mon API de manière efficace et professionnelle.

Nous verrons dans la prochaine partie que j'ai également du me former sur certains services d'AWS, je me suis pour cela servi d'un livre officiel, recommandé par un intervenant rencontré à l'ISEN, expert AWS.

J'ai gardé cette habitude de me former au quotidien (au minimum une heure par jour) que ce soit par le biais de tutoriels/ formations (je suis actuellement une formation sur la création de réseaux de neurones artificiels)/ labs ou encore grâce à de la veille dans le but de rester au fait des actualités qui se passent dans le milieu du développement et de la tech de manière générale afin de pouvoir proposer des systèmes toujours plus performants, basé sur des technologies de pointe.

DÉPLOIEMENT

Comme tout projet développé dans le but d'être rendu accessible à des utilisateurs via leur propre machine, j'ai dû me confronter au déploiement de mon application. En effet, jusqu'à maintenant, que ce soit l'application React ou mon serveur Node.js qui servait mon application back-end, tout l'écosystème que j'avais créé était exécuté depuis ma machine et n'était donc accessible que depuis mon poste. Cependant, mon manager souhaitait commencer à travailler sur l'application afin de vérifier si tout fonctionnait comme il le souhaitait et dans le but de pouvoir me faire quelques premiers feedbacks.

Pour mieux comprendre, en local, mon application React est rendue accessible à mon navigateur par le biais d'un serveur de développement géré par Webpack, serveur configuré directement à la création d'un projet React. L'application NestJS (l'API) quant à elle nécessite un serveur Node.js pour fonctionner, j'ai donc configuré ce serveur pour rendre l'API fonctionnelle. Pour finir, la base de données MongoDB est exécutée directement dans un conteneur Docker que j'ai rendu accessible à mon serveur Node.js afin que le serveur puisse communiquer avec le conteneur Docker.

Dans le cadre du déploiement, les choses se compliquent : il faut rendre l'application disponible sur internet, potentiellement à des dizaines voir des centaines d'utilisateurs en même temps sans que cela ne nuise au bon fonctionnement de l'application. Pour le moment, le but n'est pas de "scaler" l'application (mettre l'application à l'échelle, la rendre disponible à autant d'utilisateurs que nécessaires), nous verrons plus tard pourquoi. J'ai donc à nouveau effectué de nombreuses recherches et j'ai fini par me tourner vers les services d'AWS pour héberger mon application React. En effet, j'ai suivi une formation complète sur les différents services d'AWS à l'ISEN et la transparence de l'entreprise (Amazon) sur la gestion et le stockage des données ainsi que l'expérience utilisateur quelle fournit me convenaient parfaitement dans le cadre de ce premier déploiement. Un service d'AWS nommé : AWS Amplify, permet d'héberger un projet React et de le rendre disponible sur internet, de plus ce service propose un système d'intégration et de déploiement continu (IC, DC) qui permet de redéployer l'application à chaque fois que le dépôt Git sur lequel se trouve mon application React change. C'est à dire que lorsque l'on effectue un 'git push' sur la branche principale de l'application, celle-ci est automatiquement redéployée, donnant accès aux utilisateurs aux dernières mises à jour de manière totalement automatique.

Malheureusement AWS Amplify ne permet pas la même chose pour la partie back-end et donc l'API basée sur NestJS. J'ai donc dû trouver une solution alternative: Vercel. Vercel est une plateforme de déploiement basé sur le cloud, axée sur les applications front-end et les fonctions serverless mais elle peut également être utilisée pour déployer des back-ends comme ceux créés avec NestJS.

Nativement, Vercel ne permet pas de déployer des application NestJS comme la notre avec un fonctionnement convenable (certaines fonctionnalités ne sont pas prise en compte). J'ai du utiliser des méthodes particulières pour contourner ces problèmes et faire fonctionner notre API avec les mêmes fonctionnalités qu'en local.

```
You, last month | 1 author (You)
1 {
2   "version": 2,
3   "builds": [
4     {
5       "src": "src/main.ts",
6       "use": "@vercel/node"
7     }
8   ],
9   "routes": [
10    {
11      "src": "/(.*)",
12      "dest": "src/main.ts",
13      "methods": ["GET", "POST", "PUT", "DELETE", "OPTIONS"],
14      "headers": {
15        "Access-Control-Allow-Origin": "https://main...amplifyapp.com",
16        "Access-Control-Allow-Methods": "GET, POST, PUT, DELETE, OPTIONS",
17        "Access-Control-Allow-Headers": "Origin, X-Requested-With, Content-Type, Accept, Authorization"
18      }
19    }
20  ]
21 }
```

Fichier de configuration vercel.json

La méthode consiste à créer un fichier de configuration vercel.json (voir ci-dessus), ce fichier donne (entre autres) accès à notre API au domaine :

"https://main...amplifyapp.com" qui est le domaine sur lequel notre application React est hébergée (nom de domaine attribué aléatoirement par AWS), ce qui autorise ce nom de domaine à requêter l'API via des requêtes HTTP.

L'utilisation de Vercel CLI m'a ensuite permis, à l'aide de la commande : 'vercel deploy', de déployer mon application NestJS sur la plateforme Vercel.

La dernière étape consistait à rendre la base de données disponible en tout temps et accessible depuis notre API. Il existe de multiples manières de faire cela comme déployer le conteneur Docker que j'utilisais en local à l'aide de Kubernetes, système open-source qui automatise le déploiement, la mise à l'échelle et la gestion des applications conteneurisées, il est également possible d'utiliser des services de déploiement de bases de données d'AWS comme une instance EC2 ou bien EKS (Elastic Kubernetes Service). Cependant, il existe MongoDB Atlas, un service cloud officiel proposé par MongoDB, Inc. pour héberger des bases de données MongoDB. Il s'agit d'une solution Database-as-a-Service (DBaaS) qui élimine la complexité inhérente à la configuration, à la sécurisation, à l'évolutivité et à la gestion des bases de données. Avec MongoDB Atlas, il est facile de déployer des clusters MongoDB dans divers environnements cloud, tels qu'AWS, Google Cloud et Azure. Le service intègre des fonctionnalités essentielles telles que les sauvegardes automatiques, le monitoring, l'équilibrage de charge, et la mise à l'échelle automatique. De plus, la sécurité est renforcée avec des fonctionnalités comme l'authentification, le chiffrement au repos et en transit, et la mise en réseau VPC peering. J'ai donc opté pour cette stratégie qui offrait tous les services dont nous avons besoin.

Il est important de noter que toute cette configuration de déploiement n'est que temporaire. En effet, à terme, lorsque l'application sera prête à être utilisée par le grand public (à savoir tous les employés de Future Electronics), l'objectif est de déployer l'intégralité de l'application (application React, application NestJS et base de données) directement sur les serveurs de l'entreprise (dans les datacenters de Future Electronics) pour des raisons de sécurité afin d'éviter les fuites de données et dans le but d'avoir un visuel sur tout ce qui pourrait se passer au sein de l'application comme des tentatives d'attaques par exemple.

V – BILANS

BILAN PERSONNEL

Premièrement, ce stage au sein de Future Electronics représente une étape marquante dans ma formation professionnelle, car il s'agit du plus long stage en entreprise que j'ai réalisé jusqu'à présent. Cette durée m'a permis de m'immerger pleinement dans la culture de l'entreprise et d'appréhender de manière approfondie son fonctionnement.

Déjà familiarisé avec Future Electronics grâce à mon précédent stage effectué entre Juillet et Octobre 2022 (dans le cadre du stage SAI de 4ème année de l'ISEN), cette nouvelle expérience m'a offert l'opportunité de consolider mes connaissances préalables et de les étendre bien au-delà. Si initialement j'avais une vue générale de la hiérarchie de l'entreprise, cette fois-ci, j'ai pu saisir la dynamique entre les différents départements, comprendre les intrications entre eux et le rôle essentiel que jouent les ressources humaines dans l'orchestration globale.

Un des points forts de ce stage a été le développement de mes compétences relationnelles. J'ai appris l'importance de l'écoute active : chaque échange, qu'il soit avec un collègue, un supérieur ou un client, m'a montré que comprendre véritablement les besoins de l'autre est fondamental pour apporter une réponse adéquate et construire des relations de travail harmonieuses. Par exemple, lors de réunions, prendre le temps de décrypter les retours (de mon manager) m'a aidé à adapter mes propositions et à mieux appréhender certains problèmes.

En outre, ce stage a renforcé ma capacité à synthétiser l'information. Dans un monde professionnel où l'information est omniprésente et les sollicitations nombreuses, savoir aller à l'essentiel est un atout majeur. J'ai pris conscience de la nécessité de formuler des requêtes efficaces, notamment dans la rédaction de mails. Par exemple, lors de la préparation de rapports pour ma hiérarchie, j'ai appris à structurer mes écrits, à mettre en avant les points clés et à poser des questions pertinentes et ciblées.

En conclusion, ce stage a été une opportunité exceptionnelle d'apprentissage et de développement personnel. Il m'a préparé à aborder les défis du monde professionnel avec une meilleure compréhension des dynamiques d'entreprise et des compétences renforcées en matière de communication.

BILAN PROFESSIONNEL

Comme nous l'avons vu dans ce rapport, au cours de mon stage chez Future Electronics, j'ai été chargé de concevoir et de développer un logiciel en partant de zéro, un défi stimulant et éducatif qui a été central dans ma formation professionnelle.

Dès le début du stage, il m'était clair que la maîtrise des technologies web serait cruciale. L'utilisation de React pour le développement front-end a été une révélation. Cette bibliothèque m'a permis de créer des interfaces utilisateurs dynamiques, réactives et modernes. Grâce à ses composants modulaires et à son architecture unidirectionnelle des données, j'ai appris à structurer mon code de manière cohérente et à optimiser la performance de l'application.

Avant ce stage, je n'avais qu'une connaissance superficielle du back-end. Cependant, la nécessité de construire une application robuste et sécurisée m'a plongé dans cet univers fascinant. J'ai appris à configurer et à gérer un serveur, à traiter les requêtes HTTP, à mettre en place des API et à assurer la sécurité et la cohérence des données.

Une application n'est efficace que si ses données sont bien structurées et accessibles. J'ai donc été initié à la conception et à la gestion de bases de données. Cela m'a amené à comprendre l'importance de la normalisation des données, de l'indexation pour optimiser les requêtes, et de la mise en place de procédures pour garantir la sécurité des informations.

Développer un logiciel depuis le début m'a permis de comprendre comment les différentes pièces du puzzle technologique s'emboîtent les unes dans les autres. Passer de la théorie à la pratique, de l'idée à la mise en œuvre concrète, a renforcé ma confiance en mes compétences techniques. et m'a permis d'enfin mettre les connaissances que j'ai acquies durant ces cinq dernières années au profit d'un projet concret.

Comme tout projet d'envergure, j'ai été confronté à des défis techniques et des problèmes inattendus. Ces moments, bien que parfois frustrants, ont été d'une valeur inestimable pour mon apprentissage. Ils m'ont appris l'importance de la persévérance, de la recherche autonome et de la collaboration pour trouver des solutions.

En conclusion, ce stage a été un véritable tremplin pour ma carrière dans le développement de logiciels. Les compétences techniques acquises, couplées à l'expérience pratique, m'ont préparé à affronter des projets futurs avec plus de confiance.

VI – CONCLUSION

Pour conclure ce rapport, ce stage chez Future Electronics a représenté pour moi une immersion profonde et enrichissante tant sur le plan personnel que professionnel.

Il m'a offert la possibilité d'évoluer dans un environnement exigeant, de découvrir le monde professionnel sous ses multiples facettes et de forger des compétences techniques cruciales pour ma future carrière.

Sur le plan personnel, j'ai pu m'immerger dans la dynamique et la culture d'une grande entreprise, en comprenant les rouages de sa hiérarchie et de son fonctionnement, ainsi que l'importance du relationnel en milieu professionnel.

J'ai appris à écouter, à collaborer efficacement avec mes collègues et à communiquer de manière concise et pertinente, des compétences qui sont essentielles pour évoluer dans le monde professionnel.

D'un point de vue professionnel, la responsabilité qui m'a été confiée et la confiance que m'a donné mon manager dans la gestion quasi-totale de ce projet m'a permis d'aller au-delà de mes capacités et de ce que je pensais être capable de produire. Cela m'a donné l'occasion de me familiariser avec des technologies web de pointe et de comprendre comment lier les différents éléments d'une application pour que tout fonctionne correctement.

Chaque défi technique rencontré a renforcé ma résilience, mon ingéniosité et ma passion pour le développement de logiciels.

En somme, cette expérience chez Future Electronics a transformé ma vision des choses, en apprenant à découper les gros problèmes en de plus petits problèmes plus faciles à résoudre, j'ai su parvenir à mes fins. Cette expérience a consolidé mon désir de m'engager dans le secteur technologique.

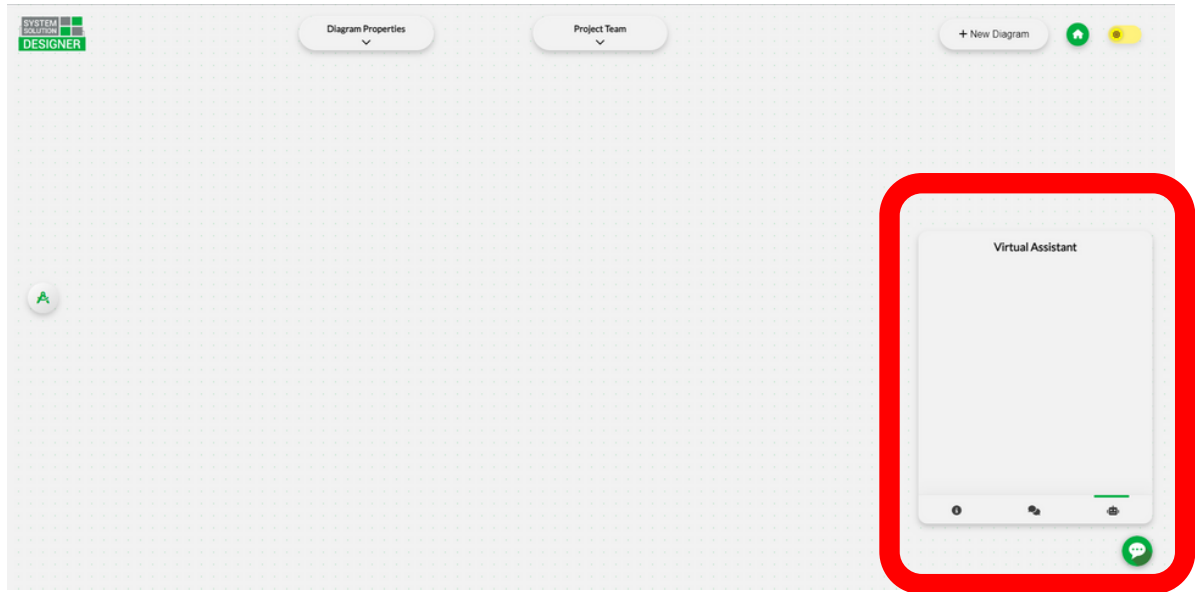
Je suis profondément reconnaissant pour les opportunités d'apprentissage qui m'ont été offertes et je suis impatient de mettre en pratique les connaissances et compétences acquises au service de ma future carrière. Car ce stage m'a permis de décrocher un contrat de travail chez Future Electronics, j'y poursuivrais le développement de ce logiciel en tant que Demand Creation Software Developer, en parallèle de cela, j'occuperais un poste de BDM pour développer le domaine de l'intelligence artificielle chez les clients de Future Electronics.

VI – OUVERTURE

La poursuite de mon stage vers mon contrat d'embauche à temps plein dans l'entreprise va me permettre de continuer le développement de System Solution Designer. En effet, à ce stade, le logiciel permet "seulement" de créer des block-diagrams, de les enregistrer et de les partager. Cependant, l'objectif principal derrière ce logiciel serait d'augmenter la productivité des ingénieurs et vendeurs de Future Electronics. Pour cela, l'idée principale de mon manager est d'implémenter un système de **recommandations** pour les utilisateurs. Ce système serait potentiellement basé sur une intelligence artificielle étant capable d'accompagner les utilisateurs de System Solution Designer lors de la conception de leurs diagrams. Il s'agirait principalement d'être capable de fournir aux utilisateurs des ressources techniques puisées dans les données que Future possède. Par exemple, lors de l'ajout d'un composant dans un diagram, à l'aide de son part-number par exemple, le système pourrait proposer un ou plusieurs composants (en indiquant le/les part-number) pouvant être compatible avec celui ajouté par l'utilisateur. L'évitant ainsi de se référer à la documentation ou de devoir aller chercher l'information dans un autre système.

Avec la démocratisation de l'IA ces derniers temps, la conception de modèles pouvant réaliser ce type de tâches est devenue plus accessible et leurs performances deviennent petit à petit très intéressantes. Nous le savons, le problème qui se pose généralement dans le développement d'un tel modèle est le manque de données. Heureusement, comme nous l'avons vu précédemment, Future Electronics dispose d'immenses bases de données et d'API pouvant fournir ces données.

Il sera donc de mon devoir, par la suite, de tenter de résoudre ce problème et de proposer un modèle d'IA pouvant accompagner les utilisateurs de System Solution Designer en temps réel, lors de leur parcours dans l'application. Ce système serait accessible via un assistant virtuel prenant la forme d'un chatbot (visible en page suivante). Celui-ci alerterait l'utilisateur en temps réel par le biais de notifications pour lui fournir des recommandations et serait capable de répondre aux demandes de l'utilisateur par l'intermédiaire d'un input (champs de texte) dans lequel l'utilisateur formulerait ses requêtes.



Interface ChatBot dans le design center

Le second point à développer sera l'**export** d'un diagram, probablement au format CSV. En effet, un logiciel existe déjà au sein de l'entreprise (un CRM) dans lequel toutes les données liées à l'activité commerciale de l'entreprise sont enregistrées. L'objectif serait donc d'exporter (**en un clic**) les données tirées d'un block-diagram (client, nom du projet, pièces utilisées, nombre de pièces, part-numbers, etc...) afin de les importer dans ce CRM (évitant ainsi aux créateurs de block-diagrams d'entrer à nouveau dans le CRM toutes le données du projet).

VII – LISTE DES RÉFÉRENCES

React : <https://legacy.reactjs.org/> (documentation officielle)

NestJS : <https://docs.nestjs.com/> (documentation officielle)

ReactFlow : <https://reactflow.dev/> (documentation officielle)

MongoDB Atlas : <https://www.mongodb.com/atlas/database> (pour le déploiement d'une base de données)

Docker : <https://www.docker.com/> (pour la conteneurisation d'applications)

AWS : <https://docs.aws.amazon.com/> (pour le déploiement d'applications et autres services, visiter la documentation pour plus d'informations)

w3schools : <https://www.w3schools.com/> (pour des tutoriels)

MDN : <https://developer.mozilla.org/fr/> (pour des tutoriels)

Stack Overflow : <https://stackoverflow.com/> (pour des conseils, poser des questions à des experts)

Medium : <https://medium.com/> (pour des tutoriels)

Vercel : <https://vercel.com/> (pour le déploiement d'applications)

Udemy : <https://www.udemy.com/> (pour des formations complètes)