

BUT Informatique

Tom DIEZ

Rapport de stage

12 semaines de stage en entreprise

Sujet : Conception d'une application de cartographies des systèmes d'information

Organisme d'accueil
Cogex Développement

Département Informatique
site d'Arles

Maître de stage :
Alain SANNA

Tuteur :
Bruno COLOMBEL

Je souhaite exprimer ma sincère gratitude à toute l'équipe de Cogex Développement pour m'avoir accueilli durant mon stage de deuxième année en BUT informatique. Cette expérience a été incroyablement enrichissante et a grandement contribué à mon développement personnel et professionnel.

Un merci tout particulier à mon maître de stage, Alain SANNA, pour son encadrement, son soutien et la confiance qu'il m'a accordée. Ses conseils avisés m'ont permis de me familiariser avec des technologies modernes et de renforcer mes compétences académiques et professionnelles.

Je remercie également Bruno COLOMBEL, mon tuteur de stage, pour son accompagnement quotidien, ainsi que David LAGARDE, le directeur de l'entreprise, pour son accueil chaleureux et son leadership exemplaire.

Je tiens aussi à exprimer ma reconnaissance envers tous les autres membres du personnel qui ont contribué à rendre cette expérience aussi agréable qu'instructive.

Enfin, je suis reconnaissant envers Laurent CARMIGNAC, mon professeur, pour son soutien et ses conseils précieux face aux défis techniques rencontrés.

Je vous adresse à tous mes remerciements les plus sincères pour avoir fait de ce stage une période si bénéfique et mémorable.

Sincèrement,

Tom DIEZ

Table des matières

| | |
|---|----|
| REMERCIEMENTS | 2 |
| TABLE DES MATIERES | 3 |
| TABLE DES FIGURES | 5 |
| GLOSSAIRE | 6 |
| LEXIQUE..... | 7 |
| INTRODUCTION..... | 10 |
| PRESENTATION DE L'ENTREPRISE | 11 |
| I. Histoire de l'entreprise | 11 |
| 1. Implantation géographique | 11 |
| 2. Historique de l'entreprise | 12 |
| 3. Modèle économique de l'entreprise | 12 |
| 4. Situation économique | 12 |
| 5. Secteurs d'activité | 12 |
| 6. Priorités d'investissement | 12 |
| 7. Organisation interne | 13 |
| II Fonctions et moyens informatiques..... | 13 |
| 1. Matériel | 13 |
| 2. Architecture réseau | 14 |
| 3. Langages de programmation..... | 14 |
| 4. Logiciels utilisés | 14 |
| SUJET DU STAGE | 15 |
| I. Contexte | 15 |
| II. Sujet | 15 |
| III. Objectifs derrière le projet | 16 |
| PRESENTATION DU TRAVAIL ACCOMPLI..... | 17 |
| I. Approche et Stratégie | 17 |
| II. Mise en œuvre du projet | 18 |
| 1. La structure | 18 |
| 2. Développement de l'interface utilisateur | 20 |
| 3. Fonctionnalités supplémentaires | 21 |
| 4. Problèmes Techniques Rencontrés | 23 |

| | |
|--|----|
| III. Analyse des Résultats Obtenus | 25 |
| 1. Aspect Technique | 25 |
| 2. Aspect Économique..... | 25 |
| 3. Aspect Humain | 26 |
| CONCLUSION | 27 |
| FEUILLE DE TEMPS..... | 28 |
| I. Emploi du temps de la semaine | 28 |
| II. Résumé des semaines..... | 28 |
| RESUME..... | 29 |
| MOTS CLES | 30 |
| REFERENCES..... | 31 |
| ANNEXES | 32 |
| I. Annexe Illustrative..... | 32 |
| II. Annexes Techniques | 33 |
| ATTESTATION DE VALIDATION DE STAGE | 36 |

TABLE DES FIGURES

| | |
|---|----|
| Figure 1. Photo aérienne du site Cogex Développement et Cogex Sud | 11 |
| Figure 2. Partie du modèle conceptuel de données du projet de cartographie..... | 19 |
| Figure 3. Extrait de la documentation développeur | 20 |
| Figure 4. Interface utilisateur avec le menu de gauche, le menu de droite, la fonctionnalité du zoom et les liens entre les objets..... | 21 |
| Figure 5. Interface administrateur | 22 |
| Figure 6. Interface montrant l'importance des points intermédiaires dans les liaisons | 24 |
| Figure 7. Interface principale de l'application | 32 |
| Figure 8. Jour 02/06/2024, Feuille de suivi de projet | 33 |
| Figure 9. Documentation, structure du projet..... | 33 |
| Figure 10. Documentation, Objet Relationnel Mapping..... | 34 |
| Figure 11. Documentation, seeders | 34 |
| Figure 12. Documentation, Identity Framework | 34 |
| Figure 13. Documentation, Images..... | 34 |
| Figure 14. Documentation, controllers | 35 |
| Figure 15. Documentation, Composants | 35 |
| Figure 16. Documentation, Components Class..... | 35 |
| Figure 17. Documentation, Components Service..... | 35 |

IUT (Institut Universitaire de Technologie) : Établissement d'enseignement supérieur proposant des formations technologiques en deux ou trois ans.

BUT (Bachelor Universitaire de Technologie) : Diplôme de l'enseignement supérieur en France, préparé dans les IUT

DSI (Directeur des Systèmes d'Information) : Est un cadre supérieur chargé de superviser l'ensemble des activités liées aux technologies de l'information au sein d'une organisation.

NAS (Network Attached Storage) : Le NAS est un dispositif de stockage connecté à un réseau, permettant à plusieurs utilisateurs et appareils de stocker et de récupérer des données centralisées.

MVC (Modèle-Vue-Contrôleur) : Architecture logicielle pour le développement web et d'interfaces utilisateur, séparant les préoccupations en Modèle, Vue et Contrôleur pour faciliter la gestion du code.

ORM (Object-Relational Mapping) : Technique de programmation simplifiant l'interaction avec les bases de données relationnelles en convertissant automatiquement les données entre les types de données orientés objet et relationnels.

Cartographie (d'un Système d'Information) : La cartographie d'un Système d'Information est une représentation visuelle et/ou descriptive détaillée des éléments constitutifs d'un système d'information au sein d'une organisation. Elle inclut la documentation des composants matériels et logiciels, des flux de données, des interactions entre les différents systèmes et des processus métiers soutenus par le SI.

Système d'Information (SI) : Un Système d'Information est un ensemble structuré de ressources, de procédures et de technologies conçu pour collecter, stocker, traiter, diffuser et gérer des informations au sein d'une organisation. Il est au cœur de la gestion de l'information et soutient les processus métiers, la prise de décision et la communication interne et externe.

Cybersécurité : La cybersécurité est l'ensemble des techniques, des pratiques et des mesures conçues pour protéger les systèmes informatiques, les réseaux et les données contre les attaques, les dommages ou l'accès non autorisé. Elle vise à assurer la confidentialité, l'intégrité et la disponibilité des informations et des systèmes.

Société Holding : Une société holding est une entreprise qui détient les actions d'autres entreprises pour les gérer et contrôler leur activité

Conditionnement : Dans le contexte industriel ou commercial, le conditionnement désigne l'ensemble des techniques et des méthodes utilisées pour emballer des produits, afin de les protéger, de faciliter leur transport et leur vente, ou de les rendre plus attrayants pour les consommateurs.

Serveur : Un serveur est un système informatique conçu pour fournir des services, des données ou des ressources à d'autres ordinateurs, appelés clients, via un réseau.

Pare-feu : Un pare-feu est une solution de sécurité qui surveille et contrôle le trafic réseau entrant et sortant basé sur des règles de sécurité prédéfinies. Il sert de barrière entre un réseau interne sécurisé et des réseaux externes non sécurisés, comme Internet, afin de protéger les systèmes informatiques contre les menaces potentielles.

Framework : Infrastructure logicielle facilitant le développement et le déploiement en fournissant une structure et des outils prédéfinis

C# : Langage de programmation moderne, orienté objet, développé par Microsoft pour sa plateforme .NET, conçu pour construire une variété d'applications allant des systèmes embarqués aux applications web et mobiles.

VBA (Visual Basic for Applications) : Langage de programmation événementiel développé par Microsoft, principalement utilisé pour l'automatisation des tâches et le développement d'applications personnalisées au sein des applications Microsoft Office, comme Excel, Word, Access, et Outlook.

SQL : Langage informatique normalisé servant à exploiter des bases de données relationnelles.

.NET : est une plateforme de développement gratuite, open source et polyvalente créée par Microsoft, qui permet de développer et exécuter des applications sur de multiples systèmes d'exploitation, y compris Windows, Linux et macOS.

Blazor : est un framework web de pointe qui permet de développer des interfaces utilisateur en utilisant C# à la place de JavaScript, intégrant des concepts modernes tels que des modèles de composants, inspirés de frameworks comme Vue.js, et des outils de migration avancés.

Razor : Razor est un moteur de balisage côté serveur développé par Microsoft pour créer des pages web dynamiques avec ASP.NET. Il permet aux développeurs de mélanger du code C# avec du code HTML pour construire des pages web de manière efficace et propre.

Vue : Dans le contexte du développement web et du modèle MVC (Modèle-Vue-Contrôleur), une vue est une représentation visuelle de données, c'est-à-dire tout ce que l'utilisateur peut voir sur l'écran (interface utilisateur).

Visual Studio : est un environnement de développement intégré (IDE) de Microsoft pour le développement d'applications, de sites web, et de services web sur plusieurs langages de programmation.

Rider : est un IDE (Environnement de Développement Intégré) de JetBrains conçu principalement pour le développement .NET et cross-platform, offrant une prise en charge étendue des technologies .NET, des fonctionnalités avancées de débogage et d'intégration avec d'autres outils JetBrains.

L'Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI) : est une agence gouvernementale française créée en 2009, placée sous l'autorité du Secrétariat général de la défense et de la sécurité nationale (SGDSN) et du ministère des Armées. Son rôle principal est de protéger les systèmes d'information sensibles de l'État français et des opérateurs d'importance vitale (OIV) contre les cybermenaces.

Jeux de tests : Un jeu de tests est un ensemble organisé de scénarios de test conçu pour valider le bon fonctionnement d'un logiciel ou d'un système informatique.

SOLID : est un acronyme qui représente cinq principes de conception de logiciels visant à créer des systèmes plus maintenables, extensibles et faciles à comprendre. Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion.

INTRODUCTION

Mon stage chez Cogex Développement a constitué une immersion fascinante au cœur du développement informatique. Entre le 2 avril et le 21 juin 2024, durant douze semaines, j'ai eu l'opportunité de rejoindre l'équipe du service informatique de l'entreprise, située à Fos-sur-Mer en France.

L'objectif principal de ma mission durant ce stage chez Cogex Développement, était de concevoir une application de gestion cartographique pour le système d'information de l'entreprise, un projet qui contribue directement aux efforts de l'entreprise pour renforcer sa cybersécurité. Cela m'a personnellement engagé à diversifier et renforcer mes compétences en C#, un langage de programmation robuste largement utilisé dans le secteur des entreprises. En parallèle, j'ai eu l'opportunité d'introduire de nouvelles méthodes de développement, apportant ainsi une touche de fraîcheur et d'innovation au projet lui-même.

Dans ce rapport, je présenterai d'abord Cogex Développement et le rôle spécifique de mon stage au sein de cette structure. Je détaillerai ensuite les tâches réalisées, les compétences acquises, ainsi que les défis rencontrés. Je partagerai également mon ressenti global sur cette expérience enrichissante. Enfin, je conclurai sur ce que ce stage m'a apporté tant professionnellement que personnellement.

I. Histoire de l'entreprise

1. Implantation géographique

Cogex Développement est une société holding qui supervise deux filiales spécialisées dans le conditionnement de produits chimiques : Cogex Sud et Limpidex. Cogex Sud est implantée à Fos-sur-Mer, tout comme Cogex Développement, un site stratégique sur la côte méditerranéenne, tandis que Limpidex se trouve à Fontainebleau, en région Île-de-France.



Figure 1. Photo aérienne du site Cogex Développement et Cogex Sud

Cogex Développement a acquis Limpidex, en 2006, une entreprise spécialisée dans le conditionnement de petits contenants (de quelques grammes à 5kg), comme des cartouches ou encore des flacons, élargissant ainsi le spectre des services offerts par le groupe.

Cette acquisition a permis de compléter les capacités de Cogex Sud, focalisée sur le conditionnement de gros contenants (60L à 235L), tels que des fûts, et de renforcer la présence de la holding dans des régions stratégiques : Fos-sur-Mer en Provence pour tirer parti de son accès maritime et Fontainebleau en Île-de-France pour sa proximité avec les principaux circuits commerciaux et industriels.

2. Historique de l'entreprise

1990 : Création de l'entreprise Cogex, par Cyrille Sellon

2000 : Création de la holding Cogex Développement propriétaire de ses filiales

2006 : Acquisition de Limpidex par Cogex Développement

2022 : Acquisition de Cogex Développement par DL Conseil, présidé par David LAGARDE

3. Modèle économique de l'entreprise

L'entreprise est une PME (petite ou moyenne entreprise) de part son nombre d'employé.

Le modèle économique de l'entreprise est structuré autour d'une Société par Actions Simplifiée (SAS). DL Conseil qui est la holding principale contenant Cogex Développement qui est la société mère de Limpidex et Cogex Sud.

4. Situation économique

Cogex Développement : 6 personnes

Reste de l'effectif Cogex Sud avec 10 personnes et 10 sur le site de Limpidex

La situation économique de l'entreprise se caractérise par :

* Un effectif de 26 employés.

* Un chiffre d'affaires annuel de 5.3 millions d'euros.

5. Secteurs d'activité

Service lié au conditionnement de produits chimiques et services d'ingénierie de solutions.

6. Priorités d'investissement

Les priorités d'investissement de Cogex sont multiples. Tout d'abord, ils investissent dans un confort de travail, notamment avec les récents achats d'engins électriques diminuant la sonorité au travail. Mais aussi, la cartérisation des appareils bruyant, ou encore l'ergonomie des poses de travail pour lutter contre les TMS (trouble musculo squelettique).

Le groupe cherche à investir dans d'autres entreprises pour élargir ses offres, permettant de faire face aux difficultés rencontrées dans le secteur de la chimie lié à la crise sanitaire du COVID 19.

7. Organisation interne

Le groupe est dirigé par le président directeur général et plusieurs chefs de services notamment commerciaux, comptables et informatique.

Chaque filiale opérationnelle a un responsable de site qui gère ses équipes.

II Fonctions et moyens informatiques

1. Matériel

Cogex possède une infrastructure hybride : le serveur web et le serveur de messagerie sont hébergés dans le cloud, le reste de l'infrastructure est répartie majoritairement sur le site de Fos sur mer et chez Limpidex.

Un serveur Windows sert de serveur d'application et de services réseau. Des NAS servent au partage de fichiers et aux sauvegardes. Les 2 sites sont protégés par des pare-feu.

Le parc PC est sous Windows 10/11. Soit fixe soit portable en fonction des besoins des personnes.

En ce qui me concerne, j'ai utilisé mon Surface Pro 9 de Microsoft, un appareil suffisamment puissant pour faire tourner aisément l'application lourde développée durant cette période. Cette continuité dans l'utilisation de mon propre matériel m'a permis de rester rapide et efficace dans mon travail.

2. Architecture réseau

Les deux sites ont une architecture réseau identique : fibre, pare-feu, switch qui dessert le réseau local. Le réseau est uniquement filaire (pas de Wifi) : câbles Ethernet qui supportent des échanges TCP/IP.

3. Langages de programmation

Le langage de programmation prédominant au sein de l'entreprise et choisi pour ce projet, sous la directive du Directeur des Systèmes d'Information, est le C#. L'entreprise maintient également une application écrite en VBA.

C#, développé par Microsoft, est largement utilisé pour sa robustesse et sa flexibilité dans la création d'applications Windows et de services web.

En parallèle, ce travail a marqué la première utilisation de Blazor, un framework web de pointe développé par Microsoft, au sein de l'entreprise.

4. Logiciels utilisés

Pour ce projet, j'ai employé Microsoft Visual Studio et Rider de JetBrains IDE. Cette combinaison d'outils m'a offert une transition fluide entre la compatibilité requise avec les systèmes de l'entreprise et l'efficacité personnelle que je tire de l'utilisation habituelle de ces environnements. J'ai également utilisé SQL Management Studio de Microsoft : un environnement intégré pour la gestion des infrastructures SQL.

I. Contexte

À la suite d'une formation en cybersécurité suivie par le Directeur des Systèmes d'Information (DSI), d'une recommandation d'un audit Cyber sécurité effectué l'année dernière et des témoignages d'autres entreprises ayant subi des cyberattaques (lesquelles ont causées de lourdes pertes), une initiative a été prise pour renforcer la sécurité des systèmes d'information de l'entreprise.

Dans ce contexte, ma mission consiste à concevoir un logiciel dédié à la création de cartographies des systèmes d'information des entreprises.

II. Sujet

Mon projet porte sur le développement d'un logiciel de cartographie dynamique des systèmes d'information. Il s'agit d'un outil informatique qui permet de visualiser et d'analyser l'architecture d'un système d'information.

Un système d'information est un ensemble de ressources (personnes, matériel, logiciels, données) qui collectent, traitent et distribuent des informations dans une organisation.

Contrairement à une solution telle que PowerPoint, très populaire dans la création de cartographie, ce logiciel permet aux utilisateurs d'interagir avec les données simultanément et en temps réel.

Cela signifie que l'utilisateur peut cliquer, faire glisser, zoomer, naviguer, etc., pour explorer les données de différentes manières. De plus, le logiciel impose une structure à la modélisation en se basant sur l'analyse proposée dans le « guide d'élaboration d'une cartographie des systèmes d'information » de l'ANSSI, dont le lien est en référence.

Le logiciel est conçu pour être facile à utiliser, même pour les personnes novices. Il dispose de menus prédéfinis qui permettent aux utilisateurs de faire des modifications rapidement et

intuitivement. En outre, le logiciel est complètement dédié à la gestion de cartographie le rendant le plus efficace possible.

Le développement du logiciel suit une méthodologie structurée pour garantir sa qualité et sa sécurité.

III. Objectifs derrière le projet

En examinant l'architecture d'un système d'information à travers une cartographie, il devient plus aisé d'identifier les vulnérabilités potentielles et de mettre en place des mesures pour renforcer la sécurité.

En cas d'attaque, cette cartographie aide les professionnels de l'informatique et le personnel à localiser rapidement la zone touchée et à prendre des décisions appropriées pour minimiser l'impact de l'attaque et l'arrêter.

Ce genre de logiciel étant peu répandu et très peu accessible pour de petites et moyennes entreprises.

Un objectif de mon maître de stage est de le mettre gratuitement à disposition pour que d'autres entreprises renforcent leur cybersécurité.

I. Approche et Stratégie

Dès les premiers jours, ma compréhension du projet s'est approfondie grâce à une étude minutieuse du cahier des charges et à des discussions détaillées avec mon maitre de stage.

Cette phase préparatoire ne s'est pas limitée à une simple réception d'instructions ; j'ai activement contribué à la modification du cahier des charges en y intégrant mes idées et suggestions.

Tout en tenant compte des contraintes techniques et temporelles. Cette implication directe dans la conception du cahier des charges était cruciale, car le développement de l'application m'était entièrement confié.

La collaboration avec mon maitre de stage s'est concentrée sur la conception structurelle, notamment sur la base de données ainsi que sur les étapes et méthodes de développement de l'application.

Pour développer une solution adéquate, j'ai dû me familiariser avec le langage C# et le Framework Blazor.

Bien que Blazor fût nouveau pour moi, j'ai rapidement trouvé des points communs avec des technologies que je maîtrisais déjà, comme Laravel, grâce à leur modèle vue-composant et à l'objet-relationnel mapping.

Blazor intègre également l'utilisation de JavaScript ainsi que des concepts de React.js que je connais bien ; ma confiance dans la faisabilité du projet était renforcée.

La stratégie que j'ai choisie pour mener à bien le projet consistait à aborder les aspects les plus simples afin de me familiariser progressivement avec le langage C# et sa syntaxe.

J'ai ensuite suivi scrupuleusement le cahier des charges en respectant l'ordre de priorité des différentes fonctionnalités à implémenter, lesquelles ont été définies en accord avec mon maître de stage.

Pour assurer un suivi régulier et efficace, un bilan rapide de 10 minutes était réalisé chaque matin et chaque soir, complété par une réunion plus approfondie d'une heure en début et en fin de semaine.

Ces points réguliers me permettaient de planifier mes activités pour la semaine suivante et d'ajuster mes priorités de codage en fonction de l'avancement du projet.

De plus, ces réunions offraient l'opportunité d'ajuster le cahier des charges si nécessaire, aligné avec les principes de la méthodologie agile qui permettent l'adaptabilité et la réactivité aux changements pour mieux répondre aux exigences du projet.

II. Mise en œuvre du projet

1. La structure

Dans le cadre du projet, il était essentiel de comprendre comment cartographier l'écosystème d'une entreprise, conformément aux recommandations du guide de l'ANSSI sur la cartographie des systèmes d'information.

Ce guide détaille les diverses entités d'une organisation et la manière de les représenter entre elles. Les entités sont par exemple les fournisseurs, les bâtiments, les postes de travail.

Ainsi, le guide de l'ANSSI, dont le lien figure dans les références, représente une grande part dans le cahier des charges.

Pour appliquer ces principes, mon maître de stage et moi avons travaillé étroitement et de manière commune pour concevoir le modèle conceptuel de données (MCD), qui organisait toutes ces entités et équipements ainsi que leurs interrelations.

Ce schéma a bénéficié d'ajustements à la suite de conseils d'un de mes professeurs en raison de contraintes techniques spécifiques telles que l'intégrité des données.

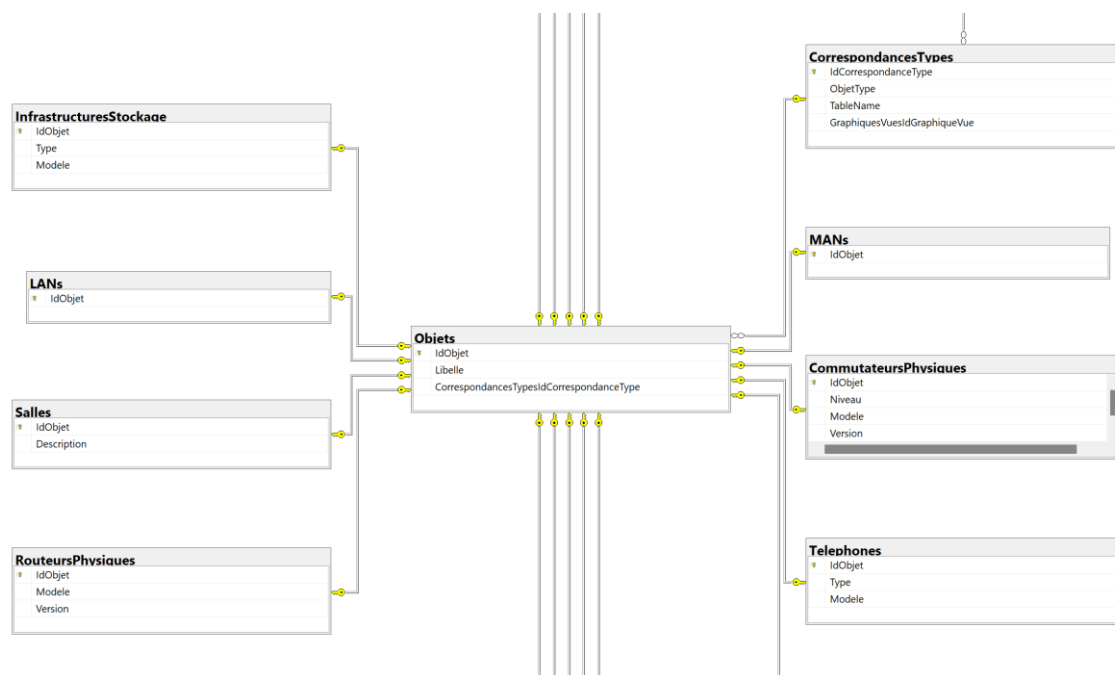


Figure 2. Partie du modèle conceptuel de données du projet de cartographie

Une fois la structure de la base de données définie, je suis passé à la structure du code.

Blazor est un framework pour la construction d'applications web interactives côté client en .NET et C#. Il ne suit pas strictement le modèle MVC (Modèle-Vue-Contrôleur). Blazor utilise un modèle de composants pour la construction d'interfaces utilisateur riches et réactives.

Cependant, dans un sens plus large, il y a toujours une logique de "modèle", "vue" et "contrôleur" qui est appliquée.

Ainsi dans un projet Blazor, la structure est en effet bien délimitée. Elle comprend différentes parties telles que l'interface utilisateur (la "vue"), des services, des modèles, des migrations, et bien d'autres.

Chaque composant joue un rôle distinct, ce qui assure une organisation et une hiérarchie claire dans le projet.

Pour respecter les principes SOLID, j'ai découpé les contrôleurs en classes séparées, en injectant les dépendances nécessaires.

Par exemple, les méthodes qui font des opérations sur la base de données peuvent être placées dans des classes de services distinctes, qui sont ensuite injectées dans le contrôleur.

Cette approche favorise une meilleure maintenabilité, une plus grande modularité et rend le développement plus efficace et rapide.

Structure du projet

Areas: Contient les fichiers Identity (login, register, logout)

Components: Contient les composants (vue et contrôleur)

ComponentsClass: Contient les classes utiles aux composants notamment des structures, et des classes pour les méthodes liés à la base de données.

ComponentsServices: Contient les services permettant la communication entre les composants

Controllers: Contient les contrôleurs externes

Database: Contient la configuration du DbContext

Factories: Contient la configuration des factories

Images: Contient les logos des objets

Migrations: Contient les migrations

Models: Contient les modèles

Seeders: Contient les seeders

Figure 3. Extrait de la documentation développeur

(Vous trouverez la documentation complète en annexe.)

2. Développement de l'interface utilisateur

La phase suivante a consisté à développer l'interface utilisateur de l'application, en particulier les menus de navigation.

Mon maitre de stage souhaitait une distinction claire entre les fonctionnalités accessibles via le menu de gauche, dédié à l'ajout de nouveaux éléments au schéma central, et le menu de droite, réservé à la gestion des propriétés des éléments existants.

Cette distinction avait pour but de faciliter l'utilisation de l'application tout en respectant les conventions d'interface utilisateur modernes.

Ensuite, vient s'ajouter la région centrale, où se trouvent tous les éléments de notre cartographie.

Pour construire la cartographie, il suffit de glisser-déposer les éléments (bâtiment, serveur Physique, ...) du menu de gauche vers le schéma du milieu.

Pour une représentation plus sophistiquée, nous pouvons créer des liens entre les éléments, personnaliser ces liens en ajoutant des nœuds et déplacer les nœuds ainsi que les objets pour les personnaliser. De plus, nous pouvons zoomer et dézoomer pour une navigation plus fluide.

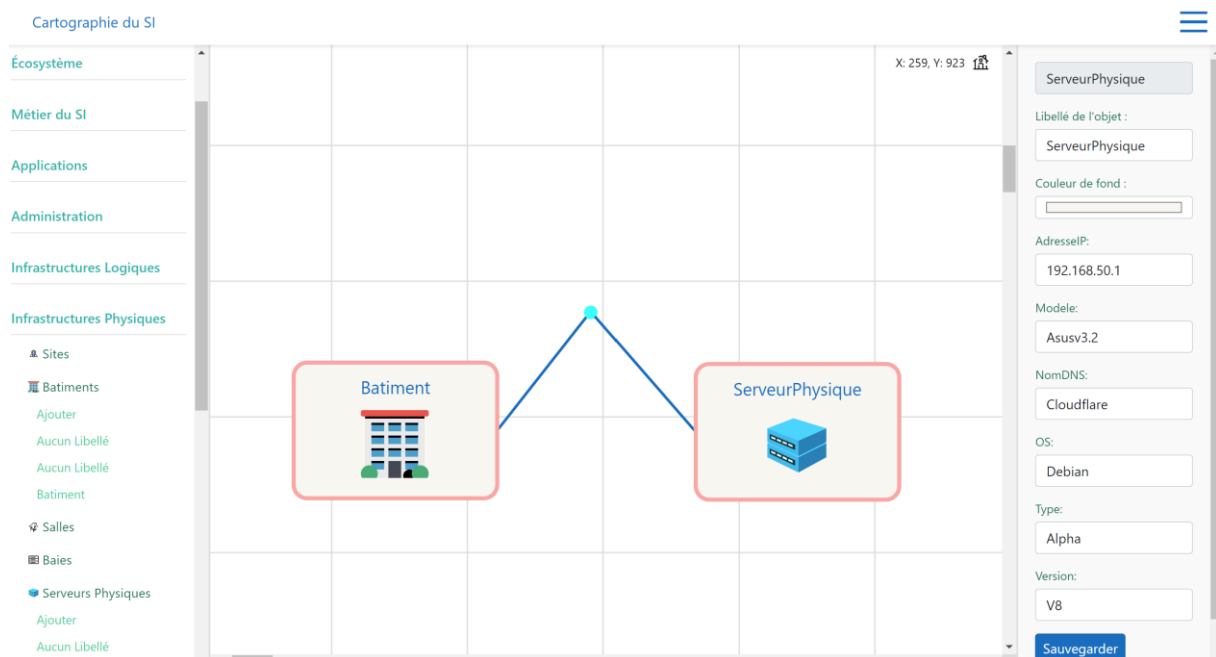


Figure 4. Interface utilisateur avec le menu de gauche, le menu de droite, la fonctionnalité du zoom et les liens entre les objets

3. Fonctionnalités supplémentaires

J'ai aussi ajouté une fonctionnalité de connexion, permettant aux différents personnels de l'entreprise de visualiser la cartographie, avec des rôles particuliers ajustés par l'administrateur, tels que read, write, et admin.

Il est également possible de créer plusieurs schémas, permettant de hiérarchiser la cartographie selon les recommandations de l'ANSII et d'améliorer la lisibilité des schémas.

Ces schémas sont imprimables de deux façons différentes. Soit l'utilisateur peut transformer son schéma en un PDF éditable avec des logiciels comme Acrobat Reader pour personnaliser son impression, soit l'utilisateur peut imprimer tous les objets qu'il aura créé ainsi que leurs paramètres.

De ce fait, on peut imprimer le côté graphique et le côté technique de la cartographie.

Des jeux de tests côté client permettent de s'assurer que tout fonctionne correctement lors de l'ajout de nouvelles fonctionnalités sans avoir à tout retester.

Pour cela, j'ai utilisé la bibliothèque de navigateurs automatisés Selenium, un outil que je maîtrisais déjà avec Python.

Ensuite, j'ai effectué une refonte bien épurée de toute l'application pour de meilleures conditions d'utilisation.

J'ai également rédigé une documentation pour les utilisateurs et les développeurs en approfondissant la structure du projet avec des commentaires dans le code pour une maintenabilité accrue.

Enfin, j'ai implémenté une fonctionnalité permettant d'aider les utilisateurs lors d'erreur de configuration du projet avec une FAQ ainsi que des options prédéfinies permettant la modification des paramètres de configurations bien plus aisément.

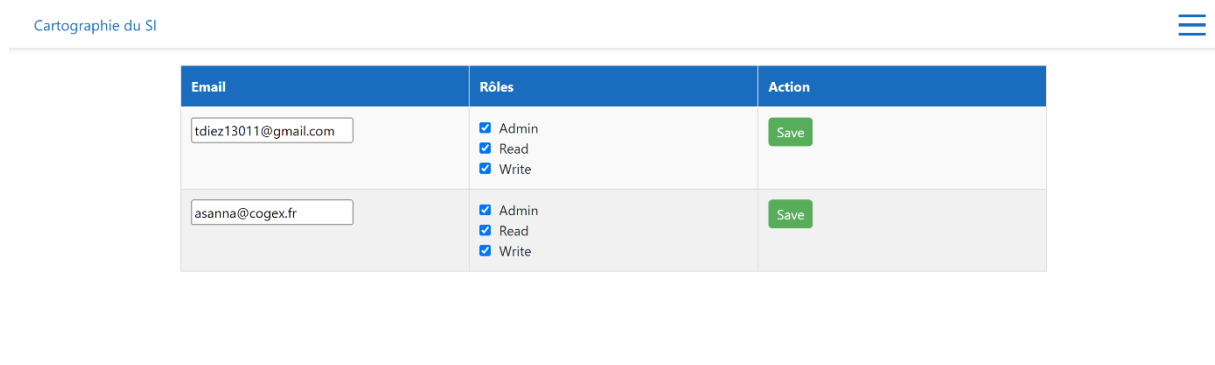


Figure 5. Interface administrateur

4. Problèmes Techniques Rencontrés

Deux fonctionnalités en particulier m'ont causé des difficultés techniques majeures, nécessitant la gestion de divers outils mathématiques, logiques et informatiques.

Premièrement, les liaisons entre les objets, bien que simples en apparence, se sont avérées complexes à implémenter.

Calculer les coordonnées de départ et d'arrivée des liaisons était relativement facile. Cependant, il fallait que ces coordonnées soient dynamiques, c'est-à-dire qu'elles devaient se mettre à jour automatiquement lorsqu'un élément était déplacé.

Cela impliquait une gestion complexe de la logique et des enregistrements des liaisons.

La difficulté s'est accentuée avec l'introduction des nœuds. Ces points intermédiaires ajoutaient une couche de complexité supplémentaire, nécessitant une gestion précise pour éviter toute confusion dans les calculs et les mises à jour des coordonnées.

Deuxièmement, la fonctionnalité de zoom a également été un défi technique. À première vue, un simple zoom peut sembler facile à réaliser, mais ce n'était pas le cas dans ce contexte.

Le problème résidait dans l'utilisation de deux composants distincts : la page web et le composant de schéma.

Chacun ayant son propre repère orthonormé, les coordonnées n'étaient pas les mêmes entre les deux et encore moins lors de l'application du zoom.

Des calculs mathématiques ont été nécessaires pour synchroniser ces coordonnées. De plus, le schéma étant très grand et ajustable, des barres de défilement étaient disponibles.

Le zoom devait donc permettre de rendre toute la carte accessible, même les parties non visibles.

Il fallait également zoomer à l'endroit où la souris pointait, ce qui impliquait de changer de repère et de déplacer automatiquement le schéma pour centrer l'élément zoomé, tout en évitant de rendre le schéma inaccessible en le déplaçant hors des limites du conteneur.

L'aide de mon maitre de stage a été précieuse, en particulier pour la stratégie de développement et la manière d'aborder ces problèmes complexes. Ses conseils ont été déterminants pour surmonter ces défis et avancer efficacement dans le projet.

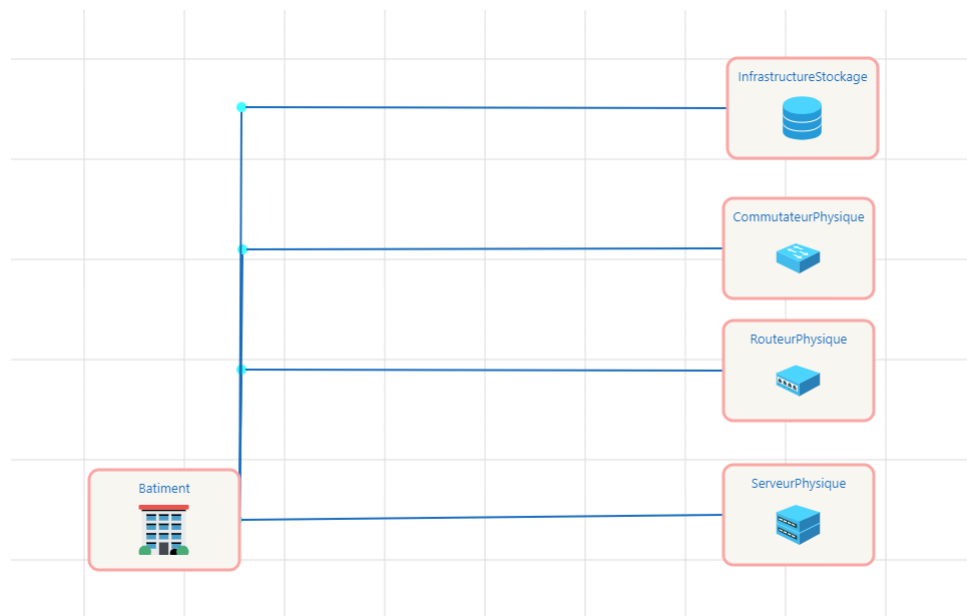


Figure 6. Interface montrant l'importance des points intermédiaires dans les liaisons

Cette application permet désormais à l'entreprise de concevoir et de maintenir une cartographie de leur système d'information claire. Elle est accessible par le personnel avec la possibilité d'interagir à plusieurs, tout en étant facile d'utilisation et grandement personnalisable.

Cela renforce notamment leur sécurité puisque ça permet une [approche globale de gestion des risques au sein de l'organisation](#). Cela peut s'avérer très utile pour les assurances.

Ce projet répond à la problématique de la protection de l'entreprise dans un monde où les cyberattaques sont de plus en plus puissantes et régulières.

III. Analyse des Résultats Obtenus

1. Aspect Technique

Le projet a été une formidable opportunité d'apprentissage et de développement technique.

Il m'a permis de découvrir Blazor, ainsi que de m'améliorer dans plusieurs techniques de développement, notamment les concepts SOLID, le modèle MVC et les ORM. Mais aussi, dans la conception de bases de données puisque le projet a nécessité une structure complexe et optimale.

Ces technologies ont enrichi mon bagage technique et ont consolidé mes compétences.

J'ai également apporté une contribution significative en proposant l'utilisation des ORM, une technologie qui n'était pas utilisée par mon maître de stage.

Cette initiative a introduit des méthodes modernes et a apporté un souffle de nouveauté, ce qui a également enrichi les compétences de mon maître de stage.

Techniquement, les attentes ont été pleinement satisfaites. Elles ont été validées par mon maître de stage, responsable du projet. De plus, le directeur et les autres membres de l'équipe, ont pu juger du design ainsi que de l'ergonomie avec pour tous, un retour très positif sur la qualité du travail que j'ai réalisé.

J'ai dépassé les attentes du projet, ce qui a été très gratifiant.

2. Aspect Économique

D'un point de vue économique, le projet permet à l'entreprise de réaliser un gain de temps considérable lors de la création et de la maintenance des cartographies.

La réduction des efforts se traduit par une diminution des coûts. De plus, ce projet représente un investissement à long terme en matière de sécurité informatique. En permettant une gestion plus efficace et préventive, il aide à éviter de potentielles cyberattaques.

En cas d'attaque, il réduit considérablement la zone d'impact, limitant ainsi les dégâts.

Par ailleurs, mon maître de stage envisage de faire profiter ses réseaux professionnels de ce travail en publiant l'application sur des portails OpenSource.

3. Aspect Humain

Sur le plan humain, ma relation avec mon maitre de stage a été très positive et constructive.

Son ouverture et sa disponibilité ont été constantes, renforçant notre collaboration dès le début du stage.

Nos réunions régulières ont permis de renforcer ce lien. Nos sorties, comme la conférence de la société SCC France, intéressante pour ses solutions IA, ont également contribué à consolider notre relation professionnelle.

Ma participation à un comité d'entreprise m'a donné un aperçu du management d'une entreprise. J'ai noué des liens avec les autres employés lors des déjeuners partagés et les discussions informelles qui ont favorisées mon intégration au sein de l'équipe.

Mon maitre de stage et moi avons établi des liens à la fois professionnels et amicaux. Grâce au covoiturage, nous avons eu l'occasion de discuter de divers sujets, tels que, politiques, d'actualité, d'informatiques ou de la vie quotidienne.

Nous avons également partagé des activités sportives, deux sorties d'escalade après les heures de bureau, ce qui a renforcé nos liens. Bien que la sortie d'escalade prévue avec le reste des employés n'ait pas eu lieu pendant mon stage, elle témoigne du bon esprit de groupe au sein de l'entreprise.

En conclusion, je me suis très bien intégré au sein de l'équipe et cette expérience a été formidable.

Bien que mon stage touche à sa fin, je garderai de très bons souvenirs de cette période enrichissante et pleine de collaboration.

CONCLUSION

Au terme de ces douze semaines de stage chez Cogex Développement, j'ai non seulement consolidé les connaissances théoriques acquises lors de mon BUT Informatique, mais j'ai également élargi mes compétences pratiques, particulièrement en développement d'applications avec Blazor.

Ce projet a renforcé ma capacité à collaborer efficacement au sein d'une équipe ainsi qu'à maîtriser la méthodologie agile lors du projet.

L'une des principales difficultés du stage a été l'adaptation aux outils technologiques spécifiques de l'entreprise, tels que le framework Blazor et le langage C#.

Cependant, j'ai transformé ces défis en opportunités d'apprentissage enrichissantes.

Un des acquis les plus significatifs a été de découvrir à quel point le monde de l'entreprise est proche de ce que nous expérimentons à l'école.

Cette proximité a facilité mon adaptation et a rendu l'expérience plus agréable.

Mais surtout, elle m'a prouvé l'importance des études dans le milieu professionnel. En acquis technique, l'utilisation de Blazor a été très riche puisque c'est un framework très complet et avancé.

En regardant vers l'avenir, cette expérience a renforcé mon désir de poursuivre mes études en informatique pour acquérir un bagage encore plus solide.

Elle a aussi motivé mon envie de continuer à travailler en parallèle dans le monde de l'entreprise. Ce qui s'avère être une source riche d'apprentissage professionnel et personnel.

FEUILLE DE TEMPS

I. Emploi du temps de la semaine

| | Lundi | Mardi | Mercredi | Jeudi | Vendredi | Total |
|---------|---------|---------|----------|---------|----------|-------|
| Arrivée | 9h | 9h | 9h | 9h | 9h | |
| Pause | 12h-13h | 12h-13h | 12h-13h | 12h-13h | | |
| Départ | 18h | 18h | 18h | 18h | 12h | |
| Total | 8h | 8h | 8h | 8h | 3h | 35h |

II. Résumé des semaines

Semaine 1 : Initiation au projet par une compréhension approfondie du cahier des charges et des technologies à utiliser. Exploration des motivations sous-jacentes au projet, de ses objectifs et de son utilité.

Semaines 2-3 : Développement de la structure de la base de données et des modèles dans le code. Cette phase, exigeante en temps, a été essentielle compte tenu de la complexité des données. Début de l'élaboration de l'interface graphique.

Semaines 4-9 : Achèvement des fonctionnalités essentielles du projet, assurant le bon fonctionnement de l'ensemble des composants clés.

Semaine 10 : Mise en production du projet sur le serveur de l'entreprise. Mon maître de stage commence alors à utiliser l'application pour identifier les bugs et élaborer sa cartographie.

Semaine 11 : Intégration des dernières fonctionnalités pratiques, telles que la documentation pour les développeurs et les utilisateurs, les ensembles de tests et les procédures de déploiement.

Semaine 12 : Finalisation des corrections de bugs et des fonctionnalités. Amélioration de l'expérience utilisateur avec l'ajout d'animations et d'éléments de style CSS.

RESUME

Ce rapport de stage détaille mon expérience chez Cogex Développement, où j'ai collaboré avec le Directeur des Systèmes d'Information (DSI) pendant douze semaines, du 2 avril au 21 juin 2024.

Mon principal objectif était de développer une application de gestion de cartographie des systèmes d'information.

Au cours de cette période, j'ai travaillé en étroite collaboration avec le DSI pour réaliser ce projet, en utilisant des technologies modernes comme Blazor afin de développer une application au goût du jour.

Ce stage a été pour moi une opportunité de découvrir de nouvelles technologies, de renforcer mes compétences et de gagner en confiance pour mes futures expériences professionnelles.

J'ai également constaté combien notre préparation à l'école est adaptée à l'intégration dans le monde professionnel.

Le rapport contient des détails sur mes activités, les défis rencontrés et les leçons apprises.

Cette expérience m'a encouragé à poursuivre mes études pour enrichir mes compétences techniques, non pas parce que je me sentais limité durant le stage, mais plutôt parce que je trouve captivant de travailler sur des projets de plus en plus complexes.

MOTS CLES

Informatique

Application

Développement

Stage

Blazor

Cogex

Cartographie

Cybersécurité

Collaboration

REFERENCES

- Guide de l'ANSSI pour la cartographie du système d'information – ANSSI

<https://cyber.gouv.fr/publications/cartographie-du-systeme-dinformation>

- Présentation de Cogex – Cogex

<https://www.cogex.fr/presentation-cogex/>

- Marchés Cibles – Cogex

<https://www.cogex.fr/marches-cibles/>

- Métiers – Cogex

<https://www.cogex.fr/metiers/>

- Conseil, Assistance, Développement – Cogex

<https://www.cogex.fr/conseil-assistance-developpement/>

I. Annexe Illustrative

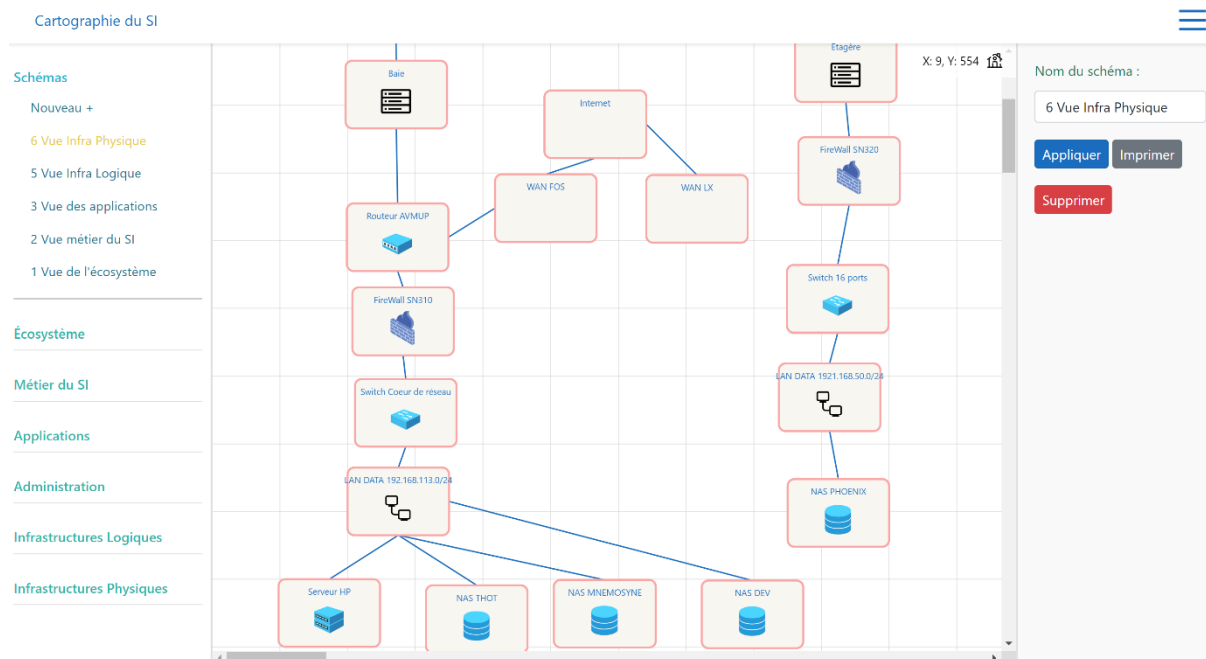


Figure 7. Interface principale de l'application

II. Annexes Techniques

| | A | B | C | D | E |
|----|---------------------|---|----------|----------|--------------|
| 1 | Thème | Fonctionnalité | priorité | état | Avancement % |
| 2 | Base de données | Structure de la base de donnée | Haute | en cours | 95% |
| 3 | La Map | Dessiner | Haute | terminé | 100% |
| 4 | | déplacement dans la map (ascenseurs) | Moyenne | terminé | 100% |
| 5 | | zoom : agrandissement, réduction de la zone affichée | Faible | terminé | 100% |
| 6 | | agrandissement, réduction limitée | Faible | terminé | 100% |
| 7 | Les objets | Création, modification | Haute | terminé | 100% |
| 8 | | Suppression, cascade uniquement sur les liaisons | Moyenne | terminé | 100% |
| 9 | | Dessiner | Haute | terminé | 100% |
| 10 | Les liaisons | Création, suppression | Haute | terminé | 100% |
| 11 | | Dessiner | Haute | terminé | 100% |
| 12 | | pouvoir ajouter des points d'inflexions | Moyenne | terminé | 100% |
| 13 | Les schémas | Création | Basse | terminé | 100% |
| 14 | | Suppression | Basse | terminé | 100% |
| 15 | | Sélection pour affichage | Basse | terminé | 100% |
| 16 | Menu administration | Création, modification, suppression utilisateur | Basse | terminé | 100% |
| 17 | | Login, forme modif mot de passe | Basse | terminé | 100% |
| 18 | Edition | Pouvoir imprimer un schéma | Basse | terminé | 100% |
| 19 | | Pouvoir imprimer tous les objets avec toutes leurs propriétés | Basse | terminé | 100% |
| 20 | Maintenance | Installation du logiciel | Moyenne | à faire | 90% |
| 21 | | Sauvegarde de la base de données | Basse | à faire | |
| 22 | | Log qui tracent les modifications de données | Basse | à faire | |
| 23 | Autres | Jeux de tests | Basse | à faire | |
| 24 | | Documentation utilisateurs et développeur | Moyenne | en cours | 10% |
| 25 | | | | | |

Figure 8. Jour 02/06/2024, Feuille de suivi de projet

Structure du projet

Areas: Contient les fichiers Identity (login, register, logout)

Components: Contient les composants (vue et contrôleur)

ComponentsClass: Contient les classes utiles aux composants notamment des structures, et des classes pour les méthodes liés à la base de données.

ComponentsServices: Contient les services permettant la communication entre les composants

Controllers: Contient les contrôleurs externes

Database: Contient la configuration du DbContext

Factories: Contient la configuration des factories

Images: Contient les logos des objets

Migrations: Contient les migrations

Models: Contient les modèles

Seeders: Contient les seeders

Figure 9. Documentation, structure du projet

Objet Relationnel Mapping (ORM)

L'application Cartographie SI est basée sur l'utilisation de ORM. Par conséquent, vous pouvez trouver tous les modèles dans le dossier Models.

Vous verrez des dossiers qui correspondent aux vues du guide de l'ANSII. Ces dossiers contiennent tous les éléments du guide. Vous pouvez en rajouter à votre guise.

De même, vous pouvez ajouter d'autres modèles si vous voulez pousser la cartographie plus loin. Pensez à mettre à jour le DbContext dans le dossier Database. Il permet de dire au ModelBuilder que d'autres modèles ont été ajoutés et de modifier leur propriété, tels que des valeurs par défaut, des relations en cascades, etc.

Referrez vous au README pour toutes les commandes utiles aux migrations. Notez cependant que le projet est conçu pour mettre automatiquement à jour les nouvelles migrations ajoutées.

Voici la [documentation](#) à propos des modèles.

Figure 10. Documentation, Objet Relationnel Mapping

Seeders - Données de tests

Les seeders se trouvent dans le dossier Seeders. Vous pouvez les activer en modifiant la variable seeder dans le fichier appsetting.json. Ils sont utiles à des fins de tests.

Vous pouvez les compléter selon votre utilité.

Figure 11. Documentation, seeders

AREAS - Identity Framework

Identity Framework est un cadre conçu par Microsoft pour les applications Blazor Server. Il permet de gérer l'authentification (login, register et logout) des utilisateurs sur votre application. Un middleware est mis à disposition pour pouvoir gérer les utilisateurs dans l'application.

Veuillez trouver la [documentation d'Identity Framework](#) ici.

Figure 12. Documentation, Identity Framework

Images

Le dossier Images contient tous les logos au format SVG des objets. Vous pouvez les modifier à votre guise, n'oubliez pas d'en refléter les changements dans le DbContext se trouvant dans le dossier Database. Il faudra spécifier le chemin d'accès de votre fichier.

Les images sont stockées en bytes dans la base de données pour de meilleures performances et sont rendues au démarrage de l'application.

Figure 13. Documentation, Images

Controllers

Les contrôleurs sont des composants côté backend server, indépendants de la vue et de son composant. Pour communiquer avec eux, il faut utiliser des appels API.

Il y en a par défaut deux. Le premier transforme le style de votre SVG pour le rendre adéquat à l'application (il prend en charge presque tous les types de SVG). Le second aide à la configuration du projet en cas d'échec au premier démarrage.

Si vous souhaitez ajouter d'autres contrôleurs, n'oubliez pas de les ajouter dans le fichier `program.cs` pour qu'ils soient correctement initialisés.

Figure 14. Documentation, controllers

Composants

Les composants regroupent les vues et leur composants associés. La vue est codée en Razor, et le composant en C#. Vous pouvez utiliser du JavaScript en vous référant à [cette page](#).

Dans la structure actuelle, vous trouverez le schéma de gauche, le schéma de droite, la navbar, et le plan. La page est le regroupement de ces quatre éléments.

Notez que chaque vue possède sa propre couche, dans le sens où elle est isolée des autres à un certain niveau. Cela implique des complications au niveau des coordonnées du schéma. Le calcul effectué consiste à prendre la coordonnée du client sur le plan et à soustraire l'écart avec le bord de gauche (le leftmenu). C'est simple, mais il ne faut pas oublier.

Figure 15. Documentation, Composants

Components Class

C'est avant tout un choix de développer selon les concepts SOLID, séparant ainsi un maximum le code. Components Class regroupe des classes associées à chaque composant dans lesquels se trouvent tous les appels à la base de données.

Figure 16. Documentation, Components Class

Components Services

Les services de composants permettent la communication entre les composants. En effet, pour communiquer avec les composants, il faut utiliser les services, qui sont séparés en fonction de leur utilité, toujours en respectant les concepts SOLID.

Si vous ajoutez des services, pensez à les ajouter au fichier `program.cs`, sinon ils ne seront pas initialisés.

Il y a déjà beaucoup d'événements définis, notamment la gestion des clics, que ce soit global, sur des composants en particulier. Ainsi, il suffit d'écouter l'événement pour y avoir accès. Je vous laisse examiner le code et naviguer à travers les usages via votre IDE.

Figure 17. Documentation, Components Service

ATTESTATION DE VALIDATION DE STAGE

Nom de l'entreprise : Cogex Développement

Adresse : Route d'Arles, 13270 Fos sur Mer

Validation du rapport de stage en entreprise :

Je soussigné, M. SANNA Alain, agissant en qualité de DSI, certifie qu'en tant que maître de stage de M. DIEZ Tom, avoir bien lu et approuvé le contenu du rapport de stage. Cette attestation est délivrée à l'intéressé afin de servir et valoir ce que de droit.

Fait à Fos-sur-Mer, le 12/06/2024.

Signature :

A handwritten signature in black ink, appearing to read 'Sanna', with a stylized flourish underneath.