

Mémoire de fin d'études

"Passer d'une application client lourd à une application web : exemple de transition d'un logiciel logistique"

Anthony PINEAU
IR2023



Stage effectué du
13 mars 2023 au 22 septembre 2023

Maître de stage : Monsieur Thierry NEROT
Tuteur pédagogique : Docteur Sofiane HAMRIOUI

Remerciements

En premier lieu, je tiens à remercier Monsieur Laurent GOURDON, directeur général France de l'entreprise SSI SCHÄFER, de m'avoir permis de rejoindre à nouveau son équipe et m'avoir offert la possibilité de mettre en pratique mes connaissances et mes compétences acquises pendant mes précédentes années d'études ainsi que ces trois années du cycle ingénieur du numérique à l'ESAIP.

Je tiens ensuite à remercier Monsieur Thierry NEROT, qui, en tant que maître de stage, s'est rendu disponible pour moi, et a su m'accompagner et partager ses connaissances avec moi. Il a aussi participé à la rédaction de ce mémoire en m'a aidant dans la relecture et en me donnant ses conseils les plus précieux

J'en profite pour remercier aussi tous les membres de l'équipe avec qui j'ai pu travailler, Messieurs Sébastien NICOD, Thibaut LUCAS et Pascal MARCELOT, qui ont su se rendre disponible pour moi si besoin. Il a d'ailleurs été très facile pour moi de pouvoir échanger avec eux et j'ai beaucoup appris auprès d'eux.

Par ailleurs, je remercie également tous mes autres collègues qui ont été très accueillants tout au long du stage, Mesdames Charlotte BARGMAN, Océane SACHOT, Pauline CHAPRON, Anita BABONNEAU, Nathalie DELARUE, Roukia ATHOUMANI ainsi que Messieurs Quentin VOLLEAU, Robin BALLON et Nicolas PERDRIAU.

Je saisir cette occasion pour adresser mes profonds remerciements aux responsables et au personnel de l'ESAIP d'avoir fait le nécessaire en ce qui concerne la gestion des conventions de stage ainsi que de nous soutenir au quotidien dans notre projet pédagogique et professionnel.

Je tiens ainsi à remercier tout particulièrement Monsieur Sofiane HAMRIOUI. En tant que tuteur pédagogique, nous avons pu échanger tout au long de mon stage, parler de l'avancement du projet sur lequel j'ai travaillé ainsi que travailler sur les différentes parties de ce présent mémoire.

Ensuite je tiens à adresser un grand merci à ma famille, sans qui tout cela ne serait pas possible. Merci pour leurs conseils ainsi que leur soutien inconditionnel, à la fois moral et économique. Ils m'accompagnent et me soutiennent au quotidien aussi bien dans mon projet pédagogique que dans mon projet professionnel et sont une source d'inspiration pour moi.

Enfin je tiens à adresser un profond remerciement à ma compagne, Madame Antonia HAMMER, pour son soutien tout au long de mon stage, et pour son aide pour la rédaction des parties en allemand de ce présent mémoire ainsi que pour la préparation de la partie allemande de la soutenance. Vielen Dank meine Liebe.

Table des matières

Remerciements	1
Table des figures	12
Liste des tableaux	12
Glossaire	19
Einleitung (Introduction)	20
1 Contexte global	23
1.1 L'entreprise : SSI SCHÄFER	23
1.1.1 Histoire	24
1.1.2 SSI SCHÄFER en France	25
1.1.3 Solutions intralogistiques	25
1.1.4 Solutions logicielles	27
1.2 MORPHEUS	27
1.2.1 Morpheus WMS	27
1.2.2 Morpheus WCS	28
2 Cadre théorique	29
2.1 Application client lourd	29
2.2 Application web	29
2.3 Frontend et backend	30
2.4 Framework	30
2.5 API : Application Programming Interface	31

2.6 REST : Respresentational State Transfer	31
3 Partie scientifique : analyse comparative des applications client lourd et des applications web	32
3.1 Introduction	32
3.1.1 Présentation du sujet	32
3.1.2 Contexte de l'analyse	33
3.1.3 Énoncé de la problématique	33
3.1.4 Résumé	33
3.2 Définitions et concepts de base	34
3.2.1 Explication des applications client lourd	34
3.2.2 Explication des applications web	34
3.2.3 Discussion sur la connectivité internet et son rôle dans les applications web	34
3.3 Avantages des applications client lourd	35
3.3.1 Performances	35
3.3.1.1 Performances du serveur	36
3.3.1.2 Performances de l'ordinateur client	36
3.3.2 Expérience utilisateur	36
3.3.2.1 Interface utilisateur graphique riche	36
3.3.2.2 Possibilité de personnalisation	37
3.3.3 Fonctionnalités avancées	37
3.3.4 Disponibilité hors ligne	37
3.4 Inconvénients des applications client lourd	38
3.4.1 Installation et mise à jour	38
3.4.2 Compatibilité	38
3.4.3 Coût	39
3.5 Avantages des applications web	39
3.5.1 Accessibilité	39
3.5.2 Installation et mises à jour simplifiées	40
3.5.3 Coût	40
3.5.4 Collaboration	41
3.5.4.1 Travail en équipe	41
3.6 Inconvénients des applications web	41
3.6.1 Performances	41

3.6.2	Expérience utilisateur	42
3.6.3	Dépendance internet	42
3.6.4	Sécurité	42
3.7	Cas d'utilisations appropriés	43
3.7.1	Identifier les scénarios où les applications client lourd sont préférables	43
3.7.2	Identifier les scénarios où les applications web offrent des avantages significatifs	44
3.8	Conclusion	45
4	Präsentation des während des Praktikums durchgeführten Projekts (Présentation du projet réalisé au cours du stage)	46
4.1	Gesamtprojekt	46
4.2	Praktikumsprojekt	47
4.3	Bestehende Anwendung	47
5	Enjeux et importances	49
5.1	Enjeux du projet global	49
5.1.1	Enjeux financiers	50
5.1.2	Enjeux stratégiques	50
5.1.3	Enjeux humains	50
5.2	Objectifs globaux	50
5.3	Enjeux du projet de stage	51
5.4	Objectifs du projet de stage	51
5.5	Parties prenantes et intervenants	51
5.6	Contraintes et limitations du projet de stage	52
5.6.1	Contraintes de délai	52
5.6.1.1	Durée fixe du stage	52
5.6.1.2	Complexité du projet	52

5.6.1.3	Apprentissage et conception initiale	53
5.6.1.4	Solutions	53
5.6.1.4.1	Planification avancée	53
5.6.1.4.2	Priorisation des tâches	53
5.6.2	Limitations en ressources humaines	54
5.6.2.1	Effectif limité et surcharge de travail	54
5.6.2.2	Solutions	54
5.6.3	Conclusion	54
5.7	Synthèse des enjeux et de l'importance	55
6	Méthodologie et outils	57
6.1	Méthodologie adoptée	57
6.1.1	Choix de la méthodologie : méthode en V	58
6.1.2	Étapes de la méthodologie	59
6.2	Planification et gestion de projet	59
6.2.1	Planification	60
6.2.2	Diagramme de GANTT	60
6.2.3	Budgétisation	61
6.3	Confluence : outil de gestion de la documentation	61
6.4	Collaboration et communication	62
6.4.1	Teams : outil de messagerie instantanée	63
6.4.2	Outlook : outil de messagerie	63
6.4.3	"Point dev" : réunion hebdomadaire	64
6.4.4	Réunions ponctuelles	64
6.4.5	Communication avec les collègues	64
6.5	Outils technologiques	64
6.5.1	Delphi	65
6.5.1.1	TMS Software et TMS Web	65
6.5.1.1.1	TMS Software	65
6.5.1.1.2	TMS Web	65
6.5.2	HTML	66
6.5.3	CSS	66

6.5.3.1	Sass	66
6.5.4	JavaScript	66
6.5.4.1	TypeScript	67
6.5.4.2	React	67
6.5.4.3	DevExtreme par DevExpress	68
6.5.4.3.1	DexExpress	68
6.5.4.3.2	DevExtreme	68
6.5.5	Git	68
6.5.6	Insomnia	68
6.5.7	Microsoft SQL Server & Microsoft SQL Server Management Studio	69
6.5.7.1	Microsoft SQL Server	69
6.5.7.2	Microsoft SQL Server Management Studio	69
6.5.8	Visual Studio Code	69
6.5.9	Modelio	70
6.5.10	Latex	70
7	Project progress (Déroulement du projet)	71
7.1	Objectives et initial planning	71
7.1.1	Detailed objectives	72
7.1.2	Initial planning	74
7.2	Project stages	74
7.2.1	Step #1 : Choosing the programming language for the frontend	74
7.2.1.1	Possible solutions	75
7.2.1.1.1	Delphi and TMS Software	75
7.2.1.1.2	JavaScript	75
7.2.1.1.3	React and DevExtreme	75
7.2.1.2	Chosen solution	76
7.2.1.2.1	Financial impact	76
7.2.2	Step #2 : Development of the first version of the frontend	77

7.2.3	Step #3 : Development of a backend to interact with the database	78
7.3	Monitoring and control	78
7.3.1	Weekly meeting :"Point dev"	79
7.3.2	Ad hoc follow-up meetings	79
8	Cyberattaque au sein de l'entreprise	80
8.1	Impacts et conséquences	80
8.1.1	Télétravail impossible	80
8.1.2	Chômage partiel	81
8.1.3	Impossibilité d'aider le client	81
8.2	Solutions	81
8.2.1	Venue des employés sur le site de Cholet	81
8.2.2	Utilisation d'autres outils	81
8.3	Conclusion	82
9	Réalisations et résultats	83
9.1	Objectifs atteints	83
9.1.1	Rappel des objectifs du projet de stage	83
9.1.2	Tester les outils de développement	84
9.1.3	Créer une maquette	84
9.1.4	Créer une première version de l'application web	84
9.2	Réalisations	84
9.3	Conclusion	86
10	Compétences mises en œuvre	87
10.1	Compétences techniques	87
10.2	Compétences en gestion	88
10.2.1	Gestion du temps	88
10.2.2	Planification	88

10.3 Compétences en communication	88
10.3.1 Rédaction d'une documentation technique	89
10.4 Compétences en leadership	89
10.4.1 Développement de l'API REST	89
10.5 Compétences en adaptabilité	89
10.5.1 Choix du changement de langage de programmation . .	90
10.6 Leçons apprises et développement professionnel	90
 Abschluss (Conclusion)	 92
 Bibliographie	 99
 Annexes	 100
A Traduction de la partie Introduction	102
B Dates importantes de l'histoire SSI SCHÄFER	104
C Carte des filiales SSI SCHÄFER	105
D Liste complète des fonctionnalités de MORPHEUS WMS	106
E Liste complète des fonctionnalités de MORPHEUS WCS	106
F Traduction de la partie Présentation du projet réalisé au cours du stage	107
F.1 Projet global	107
F.2 Projet du stage	108
F.3 Application existante	108
G Traduction de la partie Déroulement du projet	110
G.1 Objectifs et planification initiale	110
G.1.1 Objectifs détaillés	111
G.1.2 Planification initiale	113
G.2 Étapes du projet	113
G.2.1 Étape n°1 : Choix du langage de programmation pour le frontend	114

G.2.1.1	Solutions possibles	114
G.2.1.1.1	Delphi et TMS Software	114
G.2.1.1.2	JavaScript	114
G.2.1.1.3	React et DevExtreme	114
G.2.1.2	Solution retenue	115
G.2.1.2.1	Impact financier	115
G.2.2	Étape n°2 : Développement de la première version du frontend	116
G.2.3	Étape n°3 : Développement d'un backend pour interagir avec la base de données	117
G.3	Suivi et contrôle	117
G.3.1	Réunion hebdomadaire :"Point dev"	118
G.3.2	Réunions ponctuelles de suivi	118
H	Résultat obtenu avec Delphi	119
I	Code produit en Delphi	120
J	Code produit en React et DevExtreme	121
K	Comparatif technique entre les solutions TMS Web et DevExtreme	122
L	Traduction de la partie Conclusion	125
Mots clés		127
Mots clés		127
Keywords		127
Schlüsselwörter		127

Résumés	128
Résumé	128
Summary	128
Zusammenfassung	129

Nombre de caractères totaux (hors glossaire, annexes et résumés ; inclus les parties en langues étrangères) : 131 525 caractères

Nombre de caractères en anglais (hors annexes et résumé) : 11 479 caractères

Nombre de caractères en allemand (hors annexes et résumé) : 13 236 caractères

Table des figures

1.1.1	Logo de l'entreprise SSI SCHÄFER	23
1.1.2	Exemple de solution intralogistique	26
4.3.1	Teil des Menüs der bestehenden Morpheus-Anwendung	47
4.3.2	Beispiel eines vorhandenen Morpheus-Anwendungsfensters	48
6.1.1	Étapes de la méthodologie en V	59
6.2.1	Diagramme de GANTT initial du projet réalisé au cours du cours	60
6.3.1	Extrait de la documentation émise sur confluence	61
6.3.2	Extrait de la documentation émise sur confluence	62
7.2.1	Class diagram for storing information from a grid window	77
9.2.1	Capture d'écran du menu de l'application web morpheus	84
9.2.2	Capture d'écran de l'application web morpheus	85
9.2.3	Capture d'écran de l'application web morpheus	85
B.0.1	Frise historique de l'histoire de SSI SCHÄFER	104
C.0.1	Carte des filiales SSI SCHÄFER dans le monde	105
D.0.1	Fonctionnalités de Morpheus WMS	106
E.0.1	Fonctionnalités de Morpheus WCS	106
F.3.1	Partie du menu de l'application existante Morpheus	108
F.3.2	Exemple d'une fenêtre de l'application existante Morpheus	109
G.2.1	Diagramme de classe pour le stockage des informations d'une fenêtre type grille	116
H.0.1	Menu développé en Delphi	119
H.0.2	Fenêtre de test développé en Delphi	119
I.0.1	Interface de création de la fenêtre en Delphi	120
I.0.2	Code associé en Delphi	120
J.0.1	Exemple de code produit en react	121

Liste des tableaux

K.0.1	Comparatif technique entre les solutions TMS Web et DevExtreme	122
-------	--	-----

Glossaire

API Application Programming Interface. 2, 8, 13, 21, 31, 54, 68, 77, 78, 89, 97, 98, 111, 112, 123, 124

Application Programming Interface Retrouvez une définition approfondie de ce qu'est une API en section 2.5 API : Application Programming Interface. 2, 13, 23, 31

application web Retrouvez une définition approfondie de ce qu'est une application web en section 2.2 Application web. 1–4, 7, 12, 13, 16, 29, 32–35, 38–45, 50, 51, 54–56, 61, 65–68, 83–85, 87, 88, 90, 97, 98, 102–104, 109, 111–113, 118, 120, 121, 123

backend Retrouvez une définition approfondie de ce qu'est un backend en section 2.3 Frontend et backend. 2, 7, 9, 13, 15, 20, 30, 40, 78, 96, 97, 112, 123, 124

bibliothèque logicielle En informatique, une bibliothèque logicielle est un ensemble de routines qui peuvent être préalablement compilées et prêtées à être utilisées par des programmes. Ces bibliothèques sont généralement stockées sous forme de fichiers similaires, voire identiques, aux fichiers de programmes. Elles se composent d'une collection de fichiers de code objet regroupés, accompagnés d'un index facilitant l'accès à chaque routine¹. 30, 31, 67, 117

client lourd Retrouvez une définition approfondie de ce qu'est un client lourd en section 2.1 Application client lourd. 1–4, 13, 29, 32–38, 41–45, 50, 51, 97, 98, 102–104, 110, 120, 121, 123

client léger Sur le plan matériel, un client léger est un ordinateur qui, dans une architecture client-serveur, présente une logique d'application minimale, reposant principalement sur le serveur central pour le traitement. Un exemple de client léger du point de vue logiciel serait un navigateur web, qui est considéré comme un client universel. En général, une application en client léger nécessite uniquement un navigateur Web de l'utilisateur, bien qu'il puisse nécessiter l'installation de logiciels tiers. Dans le contexte d'une application web, le terme "client léger" est souvent associé au navigateur web. Cependant, il peut également être utilisé pour décrire des terminaux tels que Terminal Services, Secure Shell, Apple Remote Desktop, Citrix XenApp, TeamViewer, etc². 29, 34, 36, 37, 40, 42

-
1. Référence utilisée pour la définition du terme bibliothèque logicielle : [1].
 2. Référence utilisée pour la définition du terme client léger : [2].

CRUD L'acronyme informatique anglais CRUD (pour Create, Read, Update, Delete), parfois élargi à SCRUD en ajoutant un "S" pour Search, représente les quatre opérations fondamentales de la persistance des données, notamment lorsqu'il s'agit de stocker des informations dans une base de données. Ces opérations sont les suivantes : Create : Créer, Read : Lire, Update : Mettre à jour, Delete : Supprimer. De manière plus générale, cet acronyme englobe les opérations essentielles pour la gestion d'une collection d'éléments.³ . 78, 112

CSS Retrouvez une définition approfondie de ce qu'est CSS en section 6.5.3 CSS. 5, 14, 66, 87

Delphi Retrouvez une définition approfondie de ce qu'est Delphi en section 6.5.1 Delphi. 5, 6, 9, 12, 14, 46, 65, 68, 75, 87, 90, 95, 96, 102, 109, 114, 115, 117, 118

DevExpress Retrouvez une définition approfondie de ce qu'est DevExpress en section 6.5.4.3 DevExpress par DevExpress. 6, 14, 68, 75, 110, 119

DevExtreme Retrouvez une définition approfondie de ce qu'est DevExtreme en section 6.5.4.3 DevExtreme par DevExpress. 6, 9, 12, 14, 68, 75, 76, 88, 90, 95, 96, 109, 110, 116–119

Document Object Model Le Document Object Model (DOM) est une interface de programmation standardisée par le W3C, qui permet aux scripts d'explorer et de modifier le contenu d'un navigateur web. Grâce au DOM, la structure d'un document HTML ou XML est représentée sous forme d'un ensemble d'objets, chacun pouvant représenter des éléments tels qu'une fenêtre, une phrase ou un style, et ils sont organisés selon une structure arborescente. En utilisant le DOM, un script peut apporter des modifications au document affiché dans le navigateur, notamment en ajoutant ou en supprimant des noeuds de cet arbre⁴ . 14

DOM Document Object Model. 67

EDI Environnement de développement intégré. 65

enterprise resource planning Un progiciel de gestion intégré ou PGI (en anglais, enterprise resource planning system ou ERP system) est un progiciel qui permet « de gérer l'ensemble des processus d'une

3. Référence utilisée pour la définition du terme CRUD : [3].

4. Référence utilisée pour la définition du terme DOM : [4].

entreprise en intégrant l'ensemble de ses fonctions, dont la gestion des ressources humaines, la gestion comptable et financière, l'aide à la décision, mais aussi la vente, la distribution, l'approvisionnement et le commerce électronique »⁵. 15

environnement de développement intégré En programmation informatique, un environnement de développement représente un ensemble d'outils conçu pour accroître l'efficacité des développeurs de logiciels. Cet ensemble comprend généralement un éditeur de texte destiné à la programmation, des fonctions permettant d'initier rapidement des étapes telles que la compilation ou l'édition des liens en appuyant simplement sur un bouton, ainsi qu'un débogueur pour exécuter le programme en cours de création ligne par ligne. Certains environnements sont spécifiquement dédiés à un langage de programmation particulier⁶. 14, 65

ERP Enterprise resource planning. 80, 81

framework Retrouvez une définition approfondie de ce qu'est un framework en section 2.4 Framework. 2, 15, 21, 30, 65, 67, 68, 75, 76, 88, 90, 98, 109, 110, 117, 119, 123, 124

frontend Retrouvez une définition approfondie de ce qu'est un frontend en section 2.3 Frontend et backend. 2, 6, 8, 9, 13, 15, 20, 30, 40, 54, 74, 77, 95–97, 109, 111, 123, 124

HTML Retrouvez une définition approfondie de ce qu'est HTML en section 6.5.2 HTML. 5, 15, 60, 66–68, 72, 74, 87, 106, 108

hypertexte Un hypertexte est un document ou un ensemble de documents informatiques qui permet de passer d'une information à l'autre grâce à un système de renvois appelés hyperliens, ou liens hypertextes. Ceux-ci prennent la plupart du temps la forme d'un texte souligné en bleu, bien qu'ils puissent également être placés sur une photo, une image, un bouton, etc⁷. 66

JavaScript Retrouvez une définition approfondie de ce qu'est JavaScript en section 6.5.4 JavaScript. 6, 9, 15, 66–68, 75, 87, 95, 109, 110, 117–119

5. Référence utilisée pour la définition du terme enterprise resource planning : [5].

6. Référence utilisée pour la définition du terme environnement de développement intégré : [6].

7. Référence utilisée pour la définition du terme hypertexte : [7].

Langage de Modélisation Unifié Le Langage de Modélisation Unifié, plus connu sous l'acronyme UML (Unified Modeling Language), représente un langage graphique de modélisation. Il utilise des pictogrammes pour faciliter la visualisation et sert de méthode normalisée dans les domaines du développement logiciel et de la conception orientée objet⁸ . 18

langage de programmation Un langage de programmation est une langue informatique conçue pour exprimer des algorithmes et créer des programmes informatiques qui les mettent en œuvre. De manière analogue à une langue naturelle, un langage de programmation est construit avec un alphabet, un ensemble de mots, des règles grammaticales et des significations spécifiques. De plus, il est associé à un environnement de traduction visant à rendre sa syntaxe intelligible pour la machine⁹ . 8, 65–67, 90, 95, 98, 109, 111, 117

logiciel de gestion de version Un système de gestion de versions (VCS en anglais pour Version Control System) est un programme informatique qui permet de sauvegarder un ensemble de fichiers tout en enregistrant l'historique de toutes les modifications qui ont été apportées à ces fichiers. Il offre notamment la possibilité de récupérer différentes versions d'un groupe de fichiers liés¹⁰ . 68

Modèle-Vue-Contrôleur Le Modèle-Vue-Contrôleur (MVC) est un motif d'architecture logicielle utilisé pour les interfaces graphiques, il a été introduit en 1978 et est largement adopté dans le domaine des applications web. Ce motif est constitué de trois types de modules, chacun ayant des responsabilités distinctes : Le modèle (Model) : Il stocke les données à afficher., La vue (View) : Elle gère la présentation de l'interface graphique., Le contrôleur (Controller) : Il contient la logique relative aux actions entreprises par l'utilisateur¹¹ . 16, 67

MVC Modèle-Vue-Contrôleur. 67

open source L'expression open source, également connue sous le nom de code source ouvert, est utilisée pour décrire des logiciels (et désormais des œuvres intellectuelles) dont la licence est conforme à des critères rigoureux définis par l'Open Source Initiative. Ces critères comprennent la possibilité de redistribuer librement le logiciel,

-
8. Référence utilisée pour la définition du terme Langage de Modélisation Unifié : [8].
 9. Référence utilisée pour la définition du terme langage de programmation : [9].
 10. Référence utilisée pour la définition du terme logiciel de gestion de version : [10].
 11. Référence utilisée pour la définition du terme Modèle-Vue-Contrôleur : [11].

d'accéder à son code source et de créer des versions modifiées. Le code source, rendu accessible au public, est généralement le fruit d'une collaboration entre plusieurs programmeurs¹². 67, 68

orienté objet La programmation orientée objet (POO) est un modèle de programmation informatique basé sur l'utilisation d'entités logicielles appelées objets. Chaque objet représente un concept, une entité ou une idée, qu'elle soit abstraite ou physique, telle qu'une personne, une voiture ou même une page d'un livre. Chaque objet possède sa propre structure interne et son comportement, et il peut interagir avec d'autres objets de manière organisée. Ainsi, la POO implique la création, la représentation et la gestion de ces objets, ainsi que la définition de leurs relations. La modélisation joue un rôle crucial dans la POO, car elle permet de traduire les éléments du monde réel en entités virtuelles, en concevant des objets et en définissant leurs interactions. Cette approche facilite la résolution de problèmes et la mise en œuvre des fonctionnalités requises dans le développement logiciel¹³. 65, 117

React Retrouvez une définition approfondie de ce qu'est React en section 6.5.4.2 React. 6, 9, 17, 67, 68, 75, 76, 88, 90, 95, 96, 109, 110, 116–118

Respresentational State Transfer Retrouvez une définition approfondie de ce qu'est le REST en section 2.6 REST : Respresentational State Transfer. 3, 17, 31

REST Respresentational State Transfer. 3, 8, 17, 21, 31, 78, 89, 97, 98, 112, 123, 124

r  seau virtuel priv   En informatique, un r  seau priv   virtuel, également appel   VPN (Virtual Private Network en anglais), est un dispositif qui ´tablit une connexion directe entre des ordinateurs distants. Cette connexion isole leurs ´changes du reste du trafic circulant sur les r  seaux de t  l  communication publics¹⁴. 18

SGBD Syst  me de gestion de base de donn  es. 69

syst  me de gestion de base de donn  es Un syst  me de gestion de base de donn  es (SGBD), ou database management system

12. R  f  rence utilis  e pour la d  finition du terme open source : [12].

13. R  f  rence utilis  e pour la d  finition du terme programmation orient  e objet : [13].

14. R  f  rence utilis  e pour la d  finition du terme r  seau virtuel priv   : [14].

(DBMS) en anglais, est un logiciel système conçu pour stocker, manipuler, gérer et partager des données au sein d'une base de données. Son rôle principal est d'assurer la qualité, la durabilité et la confidentialité des informations tout en simplifiant les opérations associées. Le SGBD, ou DBMS, permet d'ajouter, récupérer, modifier, trier, transformer et imprimer les données stockées dans la base de données. Il facilite la création de rapports à partir des informations enregistrées et intègre des mécanismes pour garantir la cohérence des données, prévenir les pertes en cas de pannes, assurer la confidentialité et faciliter l'intégration avec d'autres logiciels. En fonction du modèle, le SGBD peut présenter une interface utilisateur graphique simple ou intégrer des langages de programmation sophistiqués¹⁵. 17, 69

TMS Software Retrouvez une définition approfondie de ce qu'est TMS Software en section 6.5.1.1 TMS Software et TMS Web. 5, 6, 9, 18, 61, 65, 75, 84, 95, 109, 118

TMS Web Retrouvez une définition approfondie de ce qu'est TMS Web en section 6.5.1.1 TMS Software et TMS Web. 5, 9, 12, 18, 65, 72, 76, 87, 96, 106, 110, 117–119

UML Langage de Modélisation Unifié. 70

VPN Réseau virtuel privé. 80, 81

Warehouse Control System ou Système de Pilotage des Activités représente un outil d'optimisation du flux logistique. Un système ou logiciel WCS pilote et synchronise les différents des éléments mécanisés (Robots, automates, convoyeurs, dépose étiquettes , balances) de l'entrepôt¹⁶. 27

Warehouse Management System ou Système de Gestion d'Entrepôt est un logiciel informatique dédié à l'optimisation de la gestion des stocks au sein des entrepôts¹⁷. 27

WCS Warehouse Control System. 2, 28

WMS Warehouse Management System. 2, 27

15. Référence utilisée pour la définition du terme Système de gestion de base de données : [15].

16. Référence utilisée pour la définition du terme Warehouse Control System : [16].

17. Référence utilisée pour la définition du terme Warehouse Management System : [17].

XML L'Extensible Markup Language, couramment désigné sous l'acronyme XML, est un métalangage informatique de balisage générique. Il constitue un sous-ensemble du Standard Generalized Markup Language (SGML). L'originalité de sa syntaxe réside dans son caractère « extensible », ce qui signifie qu'il permet de définir des langages spécifiques avec leurs propres vocabulaires et règles grammaticales. Des exemples de langages basés sur XML incluent XHTML, XSLT, RSS, SVG, entre autres. L'une de ses caractéristiques distinctives est l'utilisation de chevrons (<, >) pour encadrer les noms des balises. À l'origine, XML a été conçu pour simplifier l'échange automatisé de contenus complexes, tels que des structures arborescentes ou du texte enrichi, entre des systèmes d'information divers et hétérogènes, dans le but de favoriser l'interopérabilité¹⁸. 14, 66, 73, 108

18. Référence utilisée pour la définition du terme XML : [18].

Einleitung¹⁹ (Introduction)

In der sich ständig verändernden, digitalen Landschaft passen sich Softwareentwicklungsansätze kontinuierlich an die, sich ständig ändernden, Benutzeranforderungen und technologischen Fortschritte an. Mit dem Übergang von traditionellen Thick-Client-Anwendungen zu webbasierten Lösungen hat ein großer Wandel stattgefunden. Der Aufstieg des Internets und die Verbreitung von Webtechnologien haben Entwickler dazu ermutigt, die Möglichkeiten und Vorteile der Erstellung von Webanwendungen aus vorhandenen Thick-Client-Anwendungen zu erkunden.

Diese Arbeit untersucht die Hauptbestandteile des komplexen Prozesses der Entwicklung einer Webanwendung aus einem bereits vorhandenen Thick Client. Ihr Ziel ist es, Entwicklern ein Beispiel für den Übergang von einer umfangreichen Client-Anwendung zu einer Webanwendung zu geben, indem ich das Projekt beschreibe, das ich während meines Abschlusspraktikums durchgeführt habe, sowie den Transformationsprozess genauer erläutere. Die Diskussion befasst sich mit den Herausforderungen, Methoden und geeigneten Vorgehensweisen, die mit dieser Transformation verbunden sind, und ermöglicht es Softwareentwicklern, die Feinheiten des Entwicklungsprozesses erfolgreich zu meistern.

Während meines Abschlussprojekts hatte ich die Möglichkeit, an der Gestaltung einer Webanwendung mitzuwirken, die aus einer Thick-Client-Anwendung entstand. Dieser Ansatz zielt darauf ab, die Bereitstellung zu beschleunigen und die Ästhetik der Benutzeroberfläche zu modernisieren. Diese Erfahrung motivierte mich, mein Verständnis für den Prozess der Entwicklung von Webanwendungen zu vertiefen, die aus Thick-Client-Anwendungen entstehen.

Typischerweise besteht eine Webanwendung im Wesentlichen aus zwei Hauptkomponenten : dem Frontend, das sich auf die, für den Benutzer wahrnehmbare grafische Oberfläche bezieht, und dem Backend, das den Zugriff auf die Datenbank und die Verarbeitung von Datenanfragen

19. Retrouver la traduction en français de la section Introduction en Annexe A Traduction de la partie Introduction.

Références utilisées pour l'aide à la rédaction de certaines parties du mémoire [19] [20] [21] [22].

verwaltet. In diesem speziellen Szenario wird unsere Anwendung erstellt, indem ein bestimmter Teil für die Benutzeroberfläche abgetrennt wird, die über HTTP-Anfragen mit einer REST-API interagiert. Diese API fungiert als Kommunikationsverbindung mit der Datenbank.

Dieses Dokument beschreibt die Überlegungen zum Design einer Webanwendung, die von einer Thick-Client-Anwendung abgeleitet wird. Wir werden mehrere Elemente abdecken, darunter die Wahl der Programmiersprache und des Frameworks sowie die vollständige Implementierung der Anwendung, einschließlich der Erstellung der grafischen Oberflächen.

Um sich mit dem Thema auseinanderzusetzen und Antworten auf die aufgeworfenen Fragen zu geben, wurde ein Forschungsplan entwickelt. Zunächst galt es, eine fundierte Entscheidung bezüglich der Programmiersprache zu treffen und so eine solide Grundlage zu schaffen. Als nächstes wurde an der Erstellung einer neuen grafischen Oberfläche gearbeitet. Zum Abschluss des Prozesses wurde die Integration mit der Datenbank über eine REST-API implementiert.

Das Ziel der Studie besteht darin, den Prozess der Erstellung einer Webanwendung aus einem umfangreichen Kundenstamm zu verstehen und gleichzeitig die Hauptunterschiede zwischen diesen beiden Ansätzen zu identifizieren.

Zu Beginn dieser Dissertation beginnen wir mit einer Präsentation des Unternehmens, das mich willkommen geheißen hat, nämlich SSI SCHÄFER. Als nächstes werden wir den theoretischen Rahmen festlegen, der für das Verständnis der, in diesem Dokument beschrieben, Schlüsselkonzepte unerlässlich ist. Anschließend wenden wir uns dem wissenschaftlichen Teil zu, der sich auf eine vergleichende Analyse zwischen Thick-Client-Anwendungen und Webanwendungen konzentriert.

Danach werden wir uns mit dem Kern des Projekts befassen, es im Detail vorstellen und gleichzeitig seine Probleme und seine Bedeutung hervorheben. Es folgt ein ausführlicher Überblick über die Methodik und die Tools, die zum Erfolg des Projekts beigetragen haben. Anschließend beschreiben wir den Verlauf des Projekts, stellen die verschiedenen durchgeführten Phasen dar und besprechen die aufgetretenen Herausforderungen, die erzielten Erfolge sowie die während dieser Praktikumszeit erzielten Ergebnisse.

Bevor wir zum Abschluss kommen, ziehen wir eine Bilanz der in den letzten sechs Monaten erworbenen und entwickelten Fähigkeiten. Abschließend finden Sie im Anhang zusätzliche Informationen, gefolgt von einer Zusammenfassung der Dissertation in drei Sprachen : Französisch, Englisch und Deutsch.

1**Contexte global²⁰**

Dans le cadre de ce mémoire, il est essentiel de comprendre le contexte global dans lequel l'entreprise SSI SCHÄFER évolue. Ainsi dans cette section, nous allons découvrir l'histoire de l'entreprise ainsi que les produits et services qu'elle propose. Nous reviendrons aussi suite la suite logicielle Morpheus qui sera au cœur de ce document. Application Programming Interface

1.1**L'entreprise : SSI SCHÄFER**

L'entreprise SSI SCHÄFER est aujourd'hui l'un des premiers fournisseurs mondiaux de produits et systèmes logistiques pour le stockage, la préparation des commandes et la gestion des déchets. Ces solutions intra-logistiques, partiellement ou entièrement automatisées, permettent d'optimiser les flux logistiques et l'aménagement des entrepôts et centres logistiques de nos clients, peu importe leur surface de stockage et leur secteur d'activité.



FIGURE 1.1.1 – Logo de l'entreprise SSI SCHÄFER

Source de l'image : <https://www.ssi-schaefer.com/resource/crblob/480/a14c9a665a8272cc2b80687168a2e3d7/logo-ssi-schaefer-svg-data.svg>

20. Références utilisées dans la partie 1 Contexte global : [23] [24] [25] [26].

1.1.1 Histoire

C'est en 1935 que Fritz SCHÄFER (1893-1951) commença à construire ses premiers conteneurs de transport, pendant son temps libre, dans l'espace minuscule de sa propre buanderie. Deux ans plus tard, le plombier et soudeur de formation réalise son rêve et fonde son entreprise pour produire des « produits et articles en tôle ». L'entreprise a été officiellement enregistrée à Burbach, en Allemagne, le 16 janvier 1937. Elle connaît un succès immédiat et doit dès 1939 construire son premier grand hall de production.

La société s'internationalise au début des années 1960 avec la création de filiale en Suisse et en Angleterre. En 1965, Fritz SCHÄFER commence désormais la production de rayonnages à étagères et à palettes. L'entreprise se diversifie à l'aube du deuxième millénaire avec la création de SSI SCHÄFER Noell et SSI SCHÄFER Peem qui proposent des prestations avec des solutions d'automatisation. C'est en 2005 que les trois entreprises se regroupent sous la marque SSI SCHÄFER.

En 2008, le spécialiste logiciel, Salomon Automation, rejoint l'entreprise. Puis SSI SCHÄFER rachète l'entreprise danoise Handler A/S, expert en tours de stockage, en 2010. 2017 est un nouveau tournant pour l'entreprise avec la création de SSI SCHÄFER IT SOLUTIONS GMBH et une nouvelle dénomination sociale SSI SCHÄFER AUTOMATION GMBH.

Vous pourrez par ailleurs trouver en annexes B Dates importantes de l'histoire SSI SCHÄFER et C Carte des filiales SSI SCHÄFER une frise historique avec les dates importantes de l'histoire SSI SCHÄFER ainsi que la carte de toutes les filiales de l'entreprise.

1.1.2 SSI SCHÄFER en France

Le groupe SSI SCHÄFER est représenté par deux filiales en France, qui couvrent l'ensemble du territoire français et des besoins intralogistiques.

Créée en 1963, SSI SCHAEFER SAS a son siège en Moselle, on y retrouve notamment toutes les fonctions administratives et de soutien stratégique ainsi que la conception. Les commerciaux et techniciens de maintenance sont déployés sur l'ensemble du territoire français, ce qui garantit une proximité avec les clients et des temps de réaction brefs dans toute la France.

En automne 2017, SSI SCHÄFER fait l'acquisition de GRN Logistic, basé à Cholet, qui vient compléter, par son expertise dans le domaine des logiciels logistiques, l'expérience nationale et internationale du groupe dans la vente de solutions intralogistiques. Ce qui vient améliorer la compétence globale de SSI SCHÄFER en France.

1.1.3 Solutions intralogistiques

SSI SCHÄFER élabore, conçoit et fabrique des systèmes intralogistiques modulaires, destinés à tous les besoins en stockage, préparation de commandes et convoyage :

- Solutions de stockage
 - Rayonnage tous types
 - Bacs et conteneurs
 - Tours de stockage
 - Trans-stockeurs
 - Navettes
 - Orbiters
- Solutions de convoyage
 - Convoyeur de bacs, cartons, palettes et plateaux
 - Convoyeur aérien
 - Système de transport sans conducteurs (AGVs)
- Systèmes de préparation de commandes

- "Man to products" : pick by light, pick by voice, RF picking
 - "Products to Man" : carroussel, trieur, A-Frame, pick to tote
 - Tours de stockage et rayonnages dynamiques
- Solutions clefs en main
 - Planification logistique globale et conseil
 - Construction des bâtiments et installations
 - Prestations de service et de maintenance sur mesure

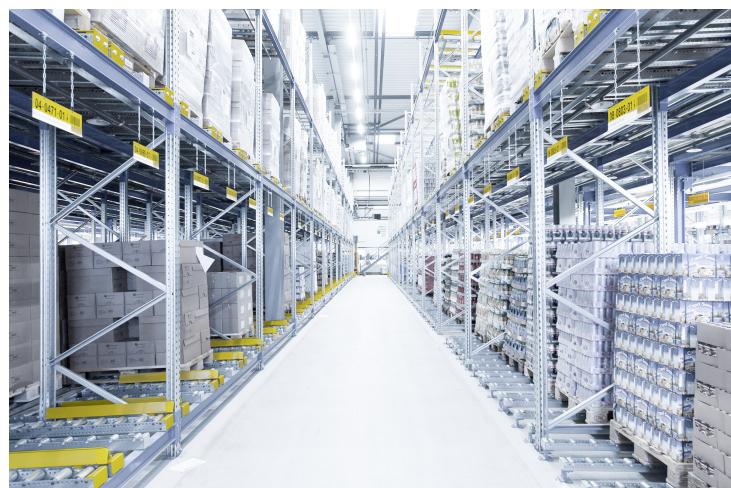


FIGURE 1.1.2 – Exemple de solution intralogistique

Source de l'image : base de données d'images interne de l'entreprise

1.1.4 Solutions logicielles

Les processus intralogistiques sont de plus en plus complexes. Dans les entrepôts manuels, automatisés et entièrement automatisés, d'innombrables processus doivent être contrôlés, visualisés et optimisés. Cela exige de l'expertise. En tant que premier fournisseur mondial de systèmes logistiques, SSI SCHÄFER ne se contente pas de proposer tout ce dont on a besoin pour des entrepôts, mais propose également l'expertise informatique.

Exemples de logiciels logistiques développés par SSI SCHÄFER :

- WAMAS
- MORPHEUS

Nous nous intéresserons ici à MORPHEUS, logiciel développé en France par GRN Logistics, avec lequel j'ai travaillé durant mon stage.

1.2 MORPHEUS

La suite logicielle MORPHEUS est une gamme de logiciels dédiée à la gestion et à l'optimisation de la performance de la logistique. Elle est composée de différents modules :

- Warehouse Management System Gestion des entrepôts
- Warehouse Control System Pilotage de stockeurs et de préparation de commandes automatisés

1.2.1 Morpheus WMS

Les objectifs du WMS :

- Augmenter votre productivité
- Réduire les stocks
- Optimiser les surfaces de stockages
- Diminuer les erreurs

Quelques fonctionnalités du WMS :

- Moteur de règles pour la gestion des stratégies métiers
- Planification et pré-réception
- Création et affectation automatique des emplacements
- Ordonnancement manuel et automatique
- Pilotage des ressources humaines

1.2.2 **Morpheus WCS**

Les objectifs du WCS :

- Mécaniser et automatiser les flux
- Intégrer et synchroniser les WMS & WCS
- Séquencier l'activité des équipements
- Superviser et optimiser les process

Quelques fonctionnalités du WCS :

- Pilotage de convoyeurs pour le transport de cartons, bacs...
- Pilotage du stockage et de la préparation de commandes
- Pilotage des trieurs / robots / meubles de tri

Retrouvez plus d'informations sur le WMS et le WCS en annexe D Liste complète des fonctionnalités de MORPHEUS WMS et E Liste complète des fonctionnalités de MORPHEUS WCS

2

Cadre théorique

Le cadre théorique est un pilier fondamental de tout projet ou étude, car il offre la base conceptuelle nécessaire à la compréhension et à la contextualisation de tout travail. Dans cette section, nous allons explorer les théories et les concepts les plus importants pour bien comprendre le projet.

Vous pourrez également trouver de nombreuses définitions dans le glossaire, la plupart d'entre elles étant des termes techniques, mais moins importantes.

2.1

Application client lourd²¹

Un logiciel client lourd offre des fonctionnalités complexes avec une capacité de traitement autonome. Le terme "client" est lié à l'architecture client-serveur. Contrairement au client léger, le client lourd dépend principalement du serveur pour l'échange de données, mais il assume habituellement la totalité du traitement.

2.2

Application web²²

Dans le domaine informatique, une application web (également désignée sous le terme web application) est une application qui peut être utilisée directement via un navigateur web²³, sans nécessiter une installation sur les dispositifs utilisateurs, contrairement aux applications client lourd. De manière similaire aux sites web, une application web est généralement hébergée sur un serveur et peut être contrôlée en interagissant avec des éléments interactifs via un navigateur web, en exploitant un réseau informatique (comme Internet, un intranet ou un réseau local).

21. Référence utilisée dans la section 2.1 Application client lourd : [27].

22. Référence utilisée dans la section 2.2 Application web : [28].

23. Le navigateur web est un client léger

2.3**Frontend et backend²⁴**

En génie logiciel, les termes "frontend" et "backend" font référence à la séparation des préoccupations entre la couche de présentation (frontend) et la couche d'accès aux données (backend) d'un logiciel. Dans le modèle client-serveur, le client est généralement considéré comme le frontend et le serveur comme le backend, même lorsque certaines tâches de présentation sont effectivement réalisées sur le serveur lui-même.

2.4**Framework²⁵**

En programmation informatique, un framework représente un ensemble cohérent de composants logiciels structurels qui servent à établir les bases ainsi que les contours généraux d'une partie ou de l'ensemble d'un logiciel, en d'autres termes, son architecture. Ce qui distingue un framework d'une simple bibliothèque logicielle, c'est principalement, d'une part, sa nature générique et sa faible spécialisation, contrairement à certaines bibliothèques logicielles. Un framework peut être composé de plusieurs bibliothèques logicielles, chacune ayant une spécialisation dans un domaine particulier. Néanmoins, un framework peut aussi être spécialisé dans un langage spécifique, une plateforme particulière, ou un domaine spécifique, comme la communication de données ou le mappage de données. D'autre part, il impose un cadre de travail inhérent à sa propre structure, ce qui guide l'architecture logicielle voire encourage le développeur à suivre certains modèles de conception. Les bibliothèques logicielles à l'intérieur du framework sont organisées selon le même paradigme.

24. Référence utilisée dans la section 2.3 Frontend et backend : [29].

25. Référence utilisée dans la section 2.4 Framework : [30].

2.5 API : Application Programming Interface²⁶

En informatique, une interface de programmation d'application, également connue sous le nom d'interface de programmation applicative (API), souvent abrégée en API pour "application programming interface", constitue un ensemble normalisé de classes, de méthodes, de fonctions et de constantes servant de façade par laquelle un logiciel propose des services à d'autres logiciels. Celle-ci est fournie par une bibliothèque logicielle ou un service web, généralement accompagnée d'une description qui détaille comment les programmes "consommateurs" peuvent tirer parti des fonctionnalités du programme "fournisseur".

2.6 REST : Respresentational State Transfer²⁷

REST (Respresentational State Transfer) représente un style d'architecture logicielle qui énonce un ensemble de directives à suivre pour la création de services web. Les services web qui adhèrent à l'architecture REST, également appelés services web RESTful, établissent une communication interopérable entre les ordinateurs connectés sur Internet. Ces services web REST autorisent les systèmes qui effectuent des requêtes à interagir avec des ressources en ligne par le biais de leurs représentations textuelles, en utilisant un ensemble uniforme et prédéfini d'opérations, sans état. En opposition, d'autres types de services web tels que les services web SOAP présentent leurs propres ensembles d'opérations variées.

Grâce à l'utilisation d'un protocole sans état et d'opérations standardisées, les systèmes REST visent la réactivité, la fiabilité et l'extensibilité, en permettant la réutilisation de composants qui peuvent être gérés et mis à jour sans perturber le fonctionnement global du système, même pendant son exécution.

26. Référence utilisée dans la section 2.5 API : Application Programming Interface : [31].

27. Référence utilisée dans la section 2.6 REST : Respresentational State Transfer : [32].

3

Partie scientifique : analyse comparative des applications client lourd et des applications web²⁸

Afin d'établir des bases solides pour l'étude, cette section réalisera une analyse comparative approfondie des caractéristiques et des fonctionnalités des applications client lourd par rapport aux applications web. En comprenant les forces et les faiblesses des deux approches, les développeurs acquerront des informations cruciales sur les défis et les opportunités qui les attendent lors du processus de transformation.

3.1

Introduction

Au cours de cette analyse, nous allons explorer en profondeur les différences fondamentales entre les applications client lourd et les applications web. Nous examinerons leurs performances, leurs capacités d'accessibilité, leur sécurité ainsi que leur convivialité. En mettant en évidence les forces et les faiblesses de chaque approche, nous serons en mesure d'évaluer les cas d'utilisation optimaux pour chaque type d'application.

3.1.1

Présentation du sujet

L'essor de la technologie a donné lieu à une diversification des plateformes logicielles, dont les applications client lourd et les applications web, qui offrent des expériences utilisateur distinctes. Cette analyse comparative se penche sur ces deux approches de développement logiciel, mettant en évidence leurs avantages, inconvénients et domaines d'application respectifs. En examinant les caractéristiques techniques, les performances, la convivialité et d'autres aspects clés, cette étude vise à fournir un aperçu approfondi des forces et des faiblesses des applications client lourd et des applications web.

28. Références utilisées dans la partie 3 Partie scientifique : analyse comparative des applications client lourd et des applications web : [2] [27] [28] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55].

3.1.2 Contexte de l'analyse

L'analyse comparative des applications client lourd et des applications web concerne l'évaluation des avantages et des inconvénients de ces deux types d'applications informatiques. Les applications client lourd et les applications web sont deux approches différentes pour fournir des logiciels et des services aux utilisateurs. Ici l'analyse n'a pas pour but de dire quelle méthode est la meilleure mais de comparer deux façons de faire et de voir dans quels scénarios il peut être plus intéressant d'utiliser l'un ou l'autre.

3.1.3 Énoncé de la problématique

Face à la diversité croissante des besoins en matière de logiciels interactifs, comment choisir de manière éclairée entre le développement d'applications client lourd et d'applications web ? Quels sont les facteurs décisifs, tels que les performances, la sécurité, l'accessibilité et l'expérience utilisateur, qui devraient guider cette décision pour répondre au mieux aux besoins des utilisateurs et aux objectifs des développeurs ? Cette analyse comparative vise à éclairer les avantages et les inconvénients de chaque approche, afin de fournir des recommandations éclairées pour le choix optimal entre applications client lourd et applications web en fonction de différents contextes et exigences.

3.1.4 Résumé

L'objectif ici n'est donc pas de déterminer la supériorité entre les applications client lourd et les applications web, mais plutôt d'analyser les diverses situations dans lesquelles l'une pourrait s'avérer plus avantageuse que l'autre. Cette démarche nous permettra également de saisir les raisons qui nous ont poussés à développer une application web dérivée du client Morpheus.

3.2 Définitions et concepts de base

3.2.1 Explication des applications client lourd

Un logiciel client lourd offre des fonctionnalités complexes avec une capacité de traitement autonome. Le terme "client" est lié à l'architecture client-serveur. Contrairement au client léger, le client lourd dépend principalement du serveur uniquement pour l'échange de données et assume habituellement la totalité du traitement. Les applications client lourd sont des logiciels installés et exécutés localement sur l'ordinateur de l'utilisateur. Elles offrent généralement une expérience riche en fonctionnalités et en performances, car elles peuvent tirer parti des ressources locales de l'ordinateur, telles que la puissance de traitement et la mémoire.

3.2.2 Explication des applications web

Dans le domaine informatique, une application web (également désignée sous le terme web application) est une application qui peut être utilisée directement via un navigateur web, sans nécessiter une installation sur les dispositifs utilisateurs, contrairement aux applications mobiles. De manière similaire aux sites web, une application web est généralement hébergée sur un serveur et peut être contrôlée en interagissant avec des éléments interactifs via un navigateur web, en exploitant un réseau informatique (comme Internet, un intranet ou un réseau local). Elles ne nécessitent donc pas d'installation locale et offrent une approche plus flexible pour l'accès aux logiciels et aux services.

3.2.3 Discussion sur la connectivité internet et son rôle dans les applications web

Internet est un réseau informatique planétaire accessible au grand public. C'est un assemblage de réseaux, fonctionnant par commutation de paquets, dépourvu de point central, qui rassemble des millions de réseaux

variés, qu'ils soient publics ou privés, universitaires, commerciaux ou gouvernementaux. Tous ces réseaux sont regroupés en entités autonomes. L'accès à Internet fait référence aux moyens fournis afin de se connecter à la toile mondiale, Internet.

La conception et l'utilisation d'applications directement sur le Web offrent la possibilité de reproduire l'expérience d'un logiciel traditionnel depuis le navigateur, en mettant à disposition des fonctionnalités autrefois exclusives aux environnements des ordinateurs individuels. Cela permet aux utilisateurs de bénéficier d'un accès uniforme aux mêmes fonctionnalités à travers différents dispositifs.

3.3 Avantages des applications client lourd

Dans le paysage numérique en constante évolution, les applications client lourd occupent une place importante en offrant une expérience utilisateur exceptionnelle et des fonctionnalités avancées. Contrairement aux applications web qui fonctionnent dans un navigateur, les applications client lourd sont installées localement sur l'ordinateur de l'utilisateur. Cette approche présente plusieurs avantages significatifs qui en font le choix privilégié pour de nombreuses entreprises et utilisateurs exigeants. Dans cette section, nous explorerons en détail les atouts majeurs des applications client lourd, allant des performances optimisées à la disponibilité hors ligne, en passant par la flexibilité de personnalisation.

3.3.1 Performances

L'application de bureau vous offre la possibilité d'implémenter toute sorte de fonctionnalité. Jusqu'à présent, aucun équivalent en ligne complet de logiciels tels que Photoshop ou Sony Vegas n'a été développé. Les utilitaires système demeurent principalement dans le domaine du développement de bureau. En plus des programmes destinés à fonctionner en arrière-plan sur de longues périodes, comme les chats ou les clients torrent, utiliser ces programmes via un navigateur serait tout simplement inconfortable. De plus, ces logiciels sont souvent utilisés pour des projets spécifiques, avec des interfaces ou des fonctions non conventionnelles.

3.3.1.1 Performances du serveur

Une architecture client-serveur qui repose sur des clients lourds n'exige pas des serveurs hautement performants. En effet, le traitement et d'autres fonctionnalités matérielles se produisent au niveau local ou individuel plutôt qu'à un niveau centralisé. De plus, cet atout implique que le serveur peut prendre en charge un nombre accru d'utilisateurs, ce qui se traduit par une capacité de serveur supérieure.

3.3.1.2 Performances de l'ordinateur client

Toute application exigeante en termes de ressources ou de bande passante doit pouvoir fonctionner de manière optimale, étant donné que les ressources proviennent des ordinateurs individuels et ne sont pas allouées par un serveur central. Les clients lourds sont en mesure de faire fonctionner des applications qui requièrent une quantité considérable de ressources. Par exemple, les jeux vidéo nécessitent une grande bande passante et une puissance de traitement élevée de l'ordinateur, ce qui garantit une expérience fluide, sans interruptions ni ralentissements.

3.3.2 Expérience utilisateur

3.3.2.1 Interface utilisateur graphique riche

L'un des aspects positifs marquants des clients lourds réside dans leur aptitude à offrir une interface utilisateur graphique de haute qualité. Des exemples d'interfaces de ce genre englobent des systèmes d'exploitation complets, des programmes ou applications informatiques immersives, ainsi que des jeux vidéo à la forte intensité graphique. Il est important de noter que la majorité des clients légers ne sont pas en mesure de reproduire des graphismes sophistiqués en raison des restrictions liées aux capacités de traitement, de calcul et d'espace de stockage disponibles.

3.3.2.2 Possibilité de personnalisation

En règle générale, les clients légers sont administrés à distance avec une participation limitée de la part de l'utilisateur final. En revanche, les clients lourds peuvent être adaptés par les employés individuels grâce à l'installation des logiciels et applications requis directement au niveau local.

3.3.3 Fonctionnalités avancées

Un inconvénient majeur des clients légers réside dans leur incapacité à effectuer localement le traitement de leurs propres données et/ou programmes. En contraste,似ilairement à leur capacité à offrir une interface utilisateur graphique sophistiquée, les clients lourds peuvent accomplir des traitements de données ou de programmes qui requièrent des ressources importantes. Des exemples de ces traitements incluent l'exécution d'applications pour éditer du contenu vidéo ou audio, jouer à des jeux vidéo, traiter des données et simuler un environnement informatique, entre autres.

3.3.4 Disponibilité hors ligne

L'autonomie vis-à-vis des serveurs ou d'un environnement en réseau constitue un autre avantage offert par les clients lourds. Il est important de noter que des dispositifs comme les ordinateurs personnels entièrement fonctionnels restent utilisables et restent opérationnels. Les clients lourds n'exigent pas une connexion réseau constante, en opposition aux clients légers qui dépendent étroitement d'une connexion permanente avec leurs serveurs. Bien sûr, les clients lourds nécessitent toujours une interaction avec leurs serveurs, particulièrement pour le partage ou la synchronisation des données à travers l'ensemble du réseau.

Les clients lourds possèdent des exigences en termes de matériel et de logiciel qui leur permettent d'accomplir des tâches même lorsqu'ils ne sont pas connectés aux serveurs centraux. Cela implique également que l'utilisation d'une connexion internet n'est pas requise.

3.4

Inconvénients des applications client lourd

Alors que les applications client lourd offrent une gamme d'avantages indéniables, comme on vient de le voir précédemment, il est également important de reconnaître les défis auxquels elles peuvent faire face. Contrairement aux applications web plus flexibles, les applications client lourd sont installées localement sur les ordinateurs des utilisateurs, ce qui peut entraîner certaines limitations et contraintes. Dans cette section, nous explorerons en détail les aspects moins favorables des applications client lourd, notamment la gestion des mises à jour, la dépendance vis-à-vis du matériel local, et les défis liés au coût.

3.4.1

Installation et mise à jour

Une application de bureau nécessite d'être installée sur un ordinateur ou un appareil mobile, et mise à jour chaque fois qu'une nouvelle version est disponible. Bien que le processus soit souvent automatisé, il exige néanmoins du temps et des ressources de la part des utilisateurs et de l'appareil. De plus, il faut gérer les versions sur chaque ordinateur, smartphone et tablette.

Dans le contexte d'une entreprise avec un grand nombre d'utilisateurs, l'installation manuelle de l'application de bureau sur chaque appareil peut être chronophage. Heureusement, il n'est pas nécessaire de sélectionner un serveur ni de rechercher des ressources à publier, à moins que nous ne parlions d'une solution client-serveur.

3.4.2

Compatibilité

L'application de bureau est tributaire du système d'exploitation, du processeur, de la carte vidéo, ainsi que d'autres paramètres multiples. Il est essentiel de considérer les particularités de chaque environnement, y compris lors de la détection et de la résolution d'erreurs. Vous devrez rédiger du code en prenant en compte les diverses possibilités, engager des développeurs individuels voire des équipes complètes pour élaborer des versions compatibles avec différentes plates-formes d'exploitation.

3.4.3 Coût

Chaque client nécessite un investissement dédié. Le matériel et le logiciel entraîneront un coût initial plus élevé, suivi de coûts continus liés à la maintenance et aux mises à jour. La mise en place peut être plus onéreuse étant donné que cela exige davantage de puissance de traitement et de mémoire. L'achat de matériel engendre des coûts élevés. Les opérations engendrent des coûts élevés en raison des demandes énergétiques substantielles.

3.5 Avantages des applications web

Dans le paysage numérique moderne, les applications web occupent une place prépondérante en offrant une accessibilité et une flexibilité inégalées. Contrairement aux applications client lourd qui nécessitent une installation locale, les applications web fonctionnent directement à travers les navigateurs, offrant ainsi une expérience utilisateur sans contraintes d'appareil ou de système d'exploitation. Dans cette section, nous explorerons en détail les avantages clés des applications web, allant de la facilité d'accès à la collaboration en passant par la facilité de mise à jour.

3.5.1 Accessibilité

L'application web est accessible de n'importe où dans le monde, depuis n'importe quel appareil, et les fichiers des utilisateurs sont toujours à portée de main, même sur un réseau local accessible de n'importe quel ordinateur.. Les applications web établissant des systèmes d'entreprise basés sur le web, elles sont accessibles 24 heures sur 24, 7 jours sur 7, à condition que vous disposiez d'une connexion internet et des identifiants de connexion nécessaires. En outre, elles sont totalement adaptables, permettant un accès depuis pratiquement n'importe quel appareil ou navigateur. Cela permet aux individus d'accéder à des données critiques lorsqu'ils ne sont pas au bureau. En outre, cela permet aux employés de travailler à distance.

3.5.2 Installation et mises à jour simplifiées

L'application web n'exige aucune installation ; toutes les mises à jour sont réalisées sur le serveur et sont immédiatement accessibles pour les utilisateurs. Il suffit de recharger la page ou de se déconnecter puis de se reconnecter à son compte pour les appliquer. Cependant, il peut être nécessaire d'installer des bibliothèques logicielles supplémentaires ou de recourir à des protocoles réseau sécurisés pour que cela fonctionne correctement.

L'application web est hébergée sur un serveur local ou dans le cloud, et le processus de mise à jour s'y effectue. Dans ce contexte, le serveur est incontournable, même si la solution est relativement simple. En effet, en plus du frontend, avec lequel les utilisateurs interagissent via le navigateur, il est nécessaire de trouver un emplacement pour le backend.

3.5.3 Coût

Les clients légers se distinguent par leur accessibilité économique. Ils reposent sur des serveurs distants pour le traitement des données, évitant ainsi le besoin de dispendieux équipements matériels locaux. Étant donné qu'ils exécutent peu, voire pas du tout, d'applications en local, les clients légers présentent une consommation énergétique réduite. Dans une grande entreprise, l'adoption d'une configuration basée sur les clients légers peut considérablement réduire la consommation d'énergie et l'impact environnemental. En équipant des milliers d'employés avec des appareils clients légers plus économiques, l'entreprise peut notablement abaisser ses coûts généraux. De même, la diminution de la consommation énergétique individuelle des clients légers engendre une réduction des dépenses ainsi que de l'empreinte carbone globale de l'entreprise.

3.5.4 Collaboration

3.5.4.1 Travail en équipe

Au sein d'une entreprise, les applications web facilitent la collaboration en équipe. Elles offrent la possibilité de mettre en place des environnements collaboratifs où le partage de travail avec toute une équipe devient extrêmement simple. Toutes les données sont hébergées en ligne, et il suffit de partager un lien ou d'accorder l'accès pour accéder aux espaces collaboratifs. Il n'est plus nécessaire de travailler en local et d'envoyer les documents par e-mail à chaque modification.

3.6 Inconvénients des applications web

Bien qu'elles offrent une accessibilité et une souplesse remarquables, il est important de reconnaître que les applications web ne sont pas exemptes de défis. Contrairement aux applications client lourd qui peuvent offrir des performances optimisées et une expérience utilisateur riche, les applications web doivent jongler avec des contraintes liées à la dépendance à l'égard de la connectivité, aux limitations de fonctionnalités et à la sécurité en ligne. Dans cette section, nous explorerons en profondeur les aspects moins favorables des applications web, tels que la vulnérabilité aux pannes de réseau, la question de la sécurité des données, l'expérience utilisateur et les performances.

3.6.1 Performances

L'application web repose entièrement sur le navigateur et sa technologie, ce qui entraîne diverses limitations, notamment en ce qui concerne l'accès aux ressources matérielles de votre appareil. Certains de ces obstacles, au moins pour l'instant, ne peuvent pas être contournés. Cependant, plusieurs tâches peuvent être résolues en appliquant le principe selon lequel "ce qui ne peut pas être contourné peut être construit ou développé". Des éditeurs de documents, d'images, d'audio, de vidéo, de graphiques 3D, des systèmes de gestion de projets, des espaces de stockage de fichiers et des créateurs sans code fonctionnent efficacement dans les navigateurs. Les outils d'intégration rapide de services et les bibliothèques logicielles frontales étendent encore davantage les capacités

existentes.

En ce qui concerne la vitesse de fonctionnement, tout n'est pas aussi évident qu'il n'y paraît. Même si le client du navigateur échange constamment des données avec le serveur, les performances seront grandement influencées par la qualité de la conception, la "pureté" du code, les capacités de l'équipement et la stabilité du canal de communication.

3.6.2 Expérience utilisateur

Bien que cela constitue un atout des applications web personnalisées, l'expérience utilisateur peut également se transformer en un inconvénient. En effet, cela requiert une expertise qui n'est pas nécessairement disponible en interne ou au sein d'une agence. Il est assez fréquent de se retrouver dans une situation où l'objectif est de fournir une expérience utilisateur adaptée à travers une application web sur mesure, mais où le résultat obtenu est de qualité moyenne. Par conséquent, il est essentiel de faire preuve de prudence à cet égard et de s'assurer que si l'adaptation de l'expérience utilisateur est un critère de choix primordial pour le sur mesure, il sera crucial de trouver des individus compétents dans ce domaine.

3.6.3 Dépendance internet

Mais seulement s'il existe une connexion Internet ou la possibilité de travailler hors ligne et de télécharger et télécharger des données est mise en œuvre. Compte tenu de ce qui précède, les clients légers nécessitent une connexion réseau stable. À défaut, ils perdent toute utilité. Les clients lourds, en revanche, peuvent tout à fait fonctionner hors ligne avec leurs propres composants matériels et logiciels.

3.6.4 Sécurité

Une application web élaborée au moyen de protocoles et d'outils de sécurité modernes a la capacité de garantir pleinement la sécurité des données. Cependant, certains aspects échappent au contrôle des développeurs : le navigateur, le serveur cloud et le canal de communication. Bien que les développeurs puissent améliorer le niveau de sécurité

en ajoutant des vérifications supplémentaires, les vulnérabilités présentes peuvent également en diminuer l'efficacité. Un avantage indéniable pour les utilisateurs est que ce type de logiciel est plus aisé à surveiller. Les limitations de l'environnement réduisent les probabilités qu'il puisse accéder en cachette aux fichiers ou initier des processus sans autorisation.

3.7 Cas d'utilisations appropriés

Lors de la réalisation d'une analyse comparative entre les applications web et les applications client lourd, il est essentiel d'examiner les cas d'utilisation appropriés pour chacune de ces approches. Cette exploration permet de mieux comprendre dans quels contextes chaque type d'application excelle et répond le mieux aux besoins spécifiques. En effet, la décision entre développer une application web ou une application client lourd dépend en grande partie des caractéristiques et des exigences particulières du projet. Dans cette section, nous allons mettre en lumière les scénarios où l'utilisation d'une application web ou d'une application client lourd se révèle la plus pertinente et bénéfique.

3.7.1 Identifier les scénarios où les applications client lourd sont préférables

Dans l'identification des scénarios où les applications client lourd sont préférables, il est essentiel de reconnaître les contextes où les performances optimales, la richesse des fonctionnalités natives et l'absence de dépendance internet sont cruciales. Les applications client lourd sont souvent la meilleure option dans les environnements professionnels où la vitesse de traitement des données est critique. Par exemple, dans les applications de modélisation 3D, la conception assistée par ordinateur (CAO) ou la retouche d'images haute résolution, l'utilisation des ressources locales permet une réactivité sans compromis. De plus, dans des secteurs tels que la finance, où la confidentialité et la sécurité sont primordiales, les applications client lourd permettent un meilleur contrôle sur les données sensibles. Enfin, lorsque l'absence de connexion internet est courante, comme dans des environnements industriels ou des régions éloignées, les applications client lourd assurent une continuité d'utilisation sans interruption. En somme, les applications client lourd sont préférables dans des cas où les performances, les fonctionnalités avancées et la

stabilité sont prioritaires par rapport à l'accessibilité web.

Ainsi voici une liste, non-exhaustive, d'exemples de type d'applications où il est préférable d'utiliser un client lourd.

- Application où l'on veut une expérience utilisateur riche
- Toute application hors ligne
- Jeux vidéo
- Logiciels de conception
- Logiciels scientifiques

3.7.2 Identifier les scénarios où les applications web offrent des avantages significatifs

L'identification des scénarios où les applications web offrent des avantages significatifs repose sur la reconnaissance des besoins en mobilité, en collaboration et en facilité d'accès à l'information. Les applications web sont particulièrement avantageuses dans un environnement professionnel en constante évolution, où la mobilité est essentielle. Les utilisateurs peuvent accéder à leurs données et applications depuis n'importe quel appareil connecté à Internet, ce qui favorise la flexibilité et la productivité en déplacement.

De plus, les applications web sont idéales pour les collaborations à distance. Dans un monde de plus en plus globalisé, où les équipes sont dispersées géographiquement, les applications web permettent un partage instantané de l'information et la collaboration en temps réel, facilitant ainsi le travail d'équipe. Les outils de communication intégrés dans les applications web, tels que les chats en ligne et les visioconférences, renforcent cette capacité collaborative.

Ainsi voici une liste, non-exhaustive, d'exemples de type d'applications où il est préférable d'utiliser une application web.

- Applications de collaboration en ligne
- Plateformes d'e-commerce
- Applications d'information

- Plateformes sociales
- Outils de productivité

3.8 Conclusion

Dans la conclusion de cette analyse comparative entre les applications client lourd et les applications web, nous pouvons clairement discerner que chaque approche présente ses avantages et ses inconvénients distincts, et leur préférence dépend fortement du contexte spécifique d'utilisation ainsi que des besoins particuliers des utilisateurs et des entreprises. En examinant attentivement ces deux catégories d'applications, nous avons mis en évidence les critères qui guident le choix entre elles et les situations où l'une l'emporte sur l'autre.

Cependant, il est important de noter que l'évolution rapide de la technologie peut transformer la dynamique entre ces deux types d'applications. Les progrès dans la vitesse d'Internet, les capacités des navigateurs web et les technologies de virtualisation peuvent influencer la pertinence des applications web par rapport aux applications client lourd, et vice versa.

À mesure que les besoins des utilisateurs et des entreprises continuent d'évoluer, le choix entre les applications client lourd et les applications web peut devenir de plus en plus contextuel. Il peut également être modulé par des facteurs tels que la sécurité, la conformité réglementaire, les coûts de développement et de maintenance, et la rapidité de mise sur le marché. En fin de compte, il n'y a pas de réponse universelle à la question de savoir quel type d'application est le meilleur. L'adéquation dépendra toujours de la situation particulière à laquelle une organisation est confrontée.

En conclusion, le choix entre les applications client lourd et les applications web est un équilibre délicat entre performances, accessibilité, coûts et fonctionnalités. Les décideurs doivent évaluer soigneusement les besoins spécifiques de leur entreprise et les préférences de leurs utilisateurs pour prendre une décision éclairée sur la meilleure approche à adopter. Dans un paysage technologique en constante évolution, la flexibilité et la capacité d'adaptation seront essentielles pour garantir des solutions logicielles efficaces et compétitives.

4**Präsentation des während des Praktikums durchgeführten Projekts (Présentation du projet réalisé au cours du stage)²⁹**

Während meines Praktikums bei SSI SCHÄFER hatte ich die außergewöhnliche Gelegenheit, an einem Projekt mitzuarbeiten, das nicht nur meine beruflichen Fähigkeiten stärkte, sondern auch einen erheblichen Einfluss auf das Unternehmen selbst hatte. In den folgenden Abschnitten werde ich das, von mir durchgeführte, Projekt im Detail vorstellen und mich dabei auf seinen Kontext, seine Ziele, die aufgetretenen Herausforderungen sowie die umgesetzten Lösungen konzentrieren. Im Rahmen dieser Dissertation lade ich Sie ein, die Schritte zu entdecken, die ich während dieses Praktikums unternommen habe, sowie die konkreten Ergebnisse, die ich erzielt habe. Dieses Projekt stellt für mich nicht nur eine bereichernde Erfahrung dar, sondern zeigt auch die Fähigkeit zur Zusammenarbeit und Innovation innerhalb des SSI SCHÄFER-Entwicklerteams.

4.1**Gesamtprojekt**

Derzeit ist Morpheus eine Software-Suite, die in einem, in Delphi entwickelten, Thick-Client gruppiert ist. Das heißt, wenn wir Morpheus verwenden möchten, müssen wir es auf dem Computer installieren, auf dem wir es benötigen, was manchmal einschränkend sein kann, insbesondere wenn wir nur Verwaltungsvorgänge ausführen müssen. Um den Zugriff auf Morpheus zu vereinfachen, ohne einen umfangreichen Client installieren zu müssen, wurde beschlossen, eine Webanwendung zu entwickeln, die den Zugriff auf Morpheus von jedem Arbeitsplatzrechner mit einem Internetbrowser aus ermöglicht. Darüber hinaus wurde parallel zum Projekt zur Entwicklung einer Webversion des Morpheus-Clients auch ein Projekt zur Neugestaltung des Heavy-Clients gestartet. Das Ziel des Projekts besteht daher nicht darin, einen umfangreichen Client durch eine Webanwendung zu ersetzen, sondern den Benutzern mehrere Möglichkeiten, ents-

29. Retrouver la traduction en français de la section Présentation du projet réalisé au cours du stage en Annexe F Traduction de la partie Présentation du projet réalisé au cours du stage

prechend den individuellen Bedürfnissen, anzubieten. Ein Manager benötigt nicht Zugriff auf die gleichen Funktionalitäten wie ein Techniker und umgekehrt.

4.2 Praktikumsprojekt

Natürlich ist es unmöglich, in nur sechs Monaten Praktikum eine komplette Webanwendung zu entwickeln, die das Verhalten von Morpheus reproduzieren würde. Aus diesem Grund habe ich während meines Praktikums nur einen Teil des Projekts zur Erstellung der Webanwendung durchgeführt. Das Praktikumsprojekt bestand also darin, mehrere Tools und Programmiersprachen zu testen, herauszufinden, welche in unserem Fall die Besten wären und eine erste Version der Webanwendung zu entwickeln.

4.3 Bestehende Anwendung

Wir haben in der vergleichenden Analyse gesehen, dass Thick Clients in den meisten Fällen nicht auf Netzwerkkonnektivität angewiesen sind. Es gibt jedoch auch einige Ausnahmen wie beispielsweise die Anwendung Morpheus. Da es sich bei der Logistik um eine sehr präzise Domäne handelt, müssen Vorgänge in Echtzeit ausgeführt werden, sodass die Anwendung ständig Zugriff auf die Datenbank benötigt. Es ist wichtig zu verstehen, dass viele Informationen in der Datenbank gespeichert sind und für die ordnungsgemäße Funktion von Morpheus erforderlich sind.

Hier sind einige Screenshots der aktuellen Morpheus-Software.

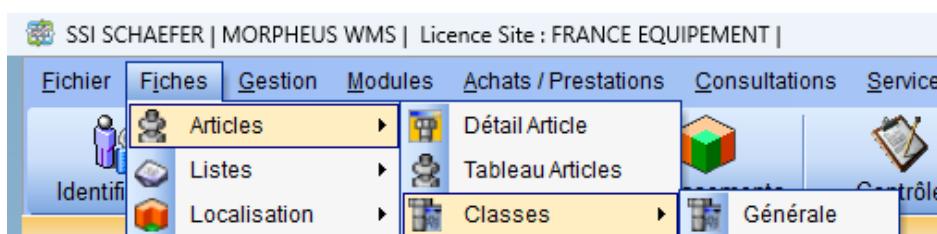


FIGURE 4.3.1 – Teil des Menüs der bestehenden Morpheus-Anwendung

Bildquelle : Screenshot

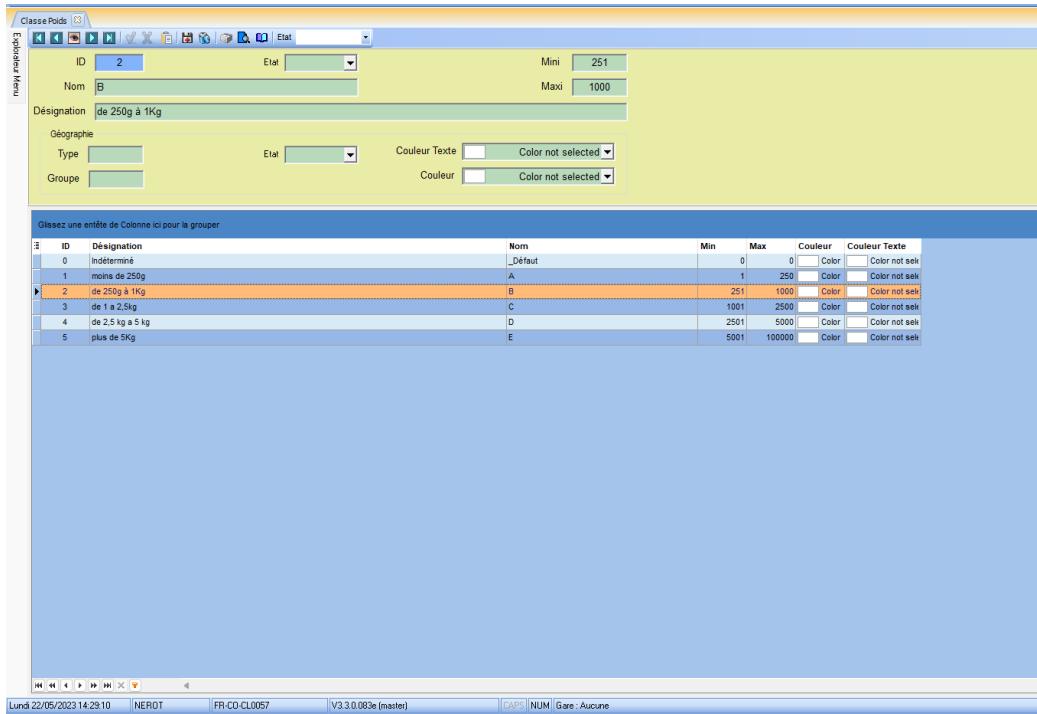


FIGURE 4.3.2 – Beispiel eines vorhandenen Morpheus-Anwendungsfensters

Bildquelle : Screenshot

Obwohl der Morpheus-Client über viele Funktionen verfügt, ist er langsam in die Jahre gekommen. Somit verleiht das Webanwendungsentwicklungsprojekt ihm mit erneuerten grafischen Oberflächen auch ein zweites Leben. Darüber hinaus können Manager ihre Verwaltungsvorgänge direkt über ihren Browser ausführen und müssen den umfangreichen Client nicht mehr installieren.

5

Enjeux et importances

Au cours de mon stage au sein de SSI SCHÄFER, j'ai eu l'opportunité de travailler sur un projet d'une importance stratégique considérable. Ce projet n'était pas simplement une tâche parmi d'autres, mais il incarnait un défi clé que l'entreprise cherchait à relever pour atteindre ses objectifs à long terme. Dans cette section, je vais mettre en lumière les enjeux majeurs auxquels ce projet répondait et expliquer pourquoi sa réalisation revêtait une signification cruciale pour SSI SCHÄFER.

Au-delà des aspects techniques, ce projet avait un impact direct sur la capacité de l'entreprise à prospérer dans un environnement concurrentiel en constante évolution. Il s'agissait d'une opportunité de consolider sa position sur le marché et de répondre aux attentes changeantes de ses clients. Tout au long de cette section, je vais détailler les aspects stratégiques et opérationnels qui ont rendu ce projet essentiel, ainsi que les résultats tangibles que j'ai pu obtenir grâce à mon travail acharné.

En comprenant les enjeux et l'importance de ce projet, vous pourrez mieux apprécier la contribution significative que j'ai apportée à l'entreprise pendant mon stage, ainsi que la manière dont mes actions ont eu un impact direct sur ses objectifs globaux.

5.1

Enjeux du projet global

Chaque projet, qu'il soit grand ou petit, naît d'une nécessité, d'un défi ou d'un besoin spécifique. Mon expérience de stage au sein de SSI SCHÄFER m'a permis de plonger au cœur d'un projet passionnant, dont la genèse repose sur des enjeux initiaux essentiels. Cette section est dédiée à l'exploration de ces enjeux, qui ont été le point de départ de cette aventure.

Au cours de cette section, je vais détailler ces enjeux initiaux, identifier les contraintes qui se posaient, et montrer pourquoi il était impératif de mettre en place ce projet. Comprendre ces enjeux est la clé pour appréhender l'évolution du projet, les décisions prises en cours de route et les résultats obtenus. En saisissant les défis initiaux, vous serez en

mesure de mesurer l'impact positif de mon travail dans la résolution de ces enjeux, et comment cela a contribué à l'entreprise.

5.1.1 Enjeux financiers

Le développement d'une application web Morpheus n'a pas pour premier enjeu un enjeu financier. En effet le projet a pour but de mener à une nouvelle communication sur le logiciel et offrir aux clients de nouvelles fonctionnalités. Néanmoins on peut imaginer qu'à plus long terme, la création d'une application web Morpheus peut avoir pour enjeu d'attirer de nouveaux clients.

5.1.2 Enjeux stratégiques

Le projet a pour enjeux stratégiques de permettre à l'entreprise de communiquer sur la suite de logiciels Morpheus et promouvoir de nouvelles fonctionnalités et un accès facilité à Morpheus auprès des clients.

5.1.3 Enjeux humains

Les enjeux humains du projet incluent la possibilité pour les clients d'accéder à Morpheus pour effectuer des opérations de gestion depuis leurs bureaux, sans nécessiter une présence dans l'entrepôt ni l'installation d'un client lourd, en utilisant simplement un navigateur web. Le projet vise également à réduire le temps que les techniciens passent à installer le nouveau client Morpheus. Puisqu'il suffirait simplement de déployer l'application sur un serveur web et non à installer un client sur chaque poste.

5.2 Objectifs globaux

Dans cette section, nous nous pencherons sur les objectifs globaux de notre projet, une étape fondamentale pour éclairer mon chemin vers la réalisation de mes aspirations. Ces objectifs constituent les piliers qui soutiendront mon initiative du début à la fin, et ils définissent la vision que je cherche à concrétiser. Nous parlerons ici des objectifs du projet global,

nous reviendrons sur les objectifs du projet réalisé au cours du stage juste après dans la prochaine sous-section.

Voici donc les objectifs globaux du projet :

- Développer un client web morpheus (à partir du client lourd existant)
- Rendre morpheus plus accessible
- Créer un nouveau design ergonomique pour morpheus
- Faire une application web responsive accessible depuis n'importe quel terminal

5.3 Enjeux du projet de stage

Le principal enjeu du projet de stage consistait à explorer diverses possibilités de développement pour le projet et à établir des fondations solides en vue de sa poursuite à long terme.

5.4 Objectifs du projet de stage

Voici une présentation sommaire des objectifs du projet de stage sur lesquels nous reviendrons par la suite.

- Tester les outils de développement
- Créer une maquette
- Créer une première version de l'application web

5.5 Parties prenantes et intervenants

Voici les principales parties prenantes et intervenants qui ont pris part activement durant la réalisation de ce projet.

- Monsieur Laurent GOURDON : directeur général France
- Monsieur Thierry NEROT : chef de projet et développeur du nouveau client lourd mopheus

5.6 Contraintes et limitations du projet de stage

Tout projet, qu'il soit de petite ou de grande envergure, est confronté à des contraintes et à des limitations qui peuvent influencer sa planification, son exécution et ses résultats. Dans cette section, nous allons explorer les défis et les restrictions spécifiques auxquels j'ai dû faire face tout au long de mon projet.

La reconnaissance et la gestion des contraintes et des limitations sont essentielles pour une gestion de projet efficace. Ces facteurs peuvent être de nature diverse, qu'il s'agisse de contraintes budgétaires, de délais serrés, de ressources limitées, de contraintes techniques, ou d'autres obstacles potentiels. Dans cette partie, je vais détailler les contraintes et les limitations qui ont marqué mon projet et expliquer comment je les ai abordées.

5.6.1 Contraintes de délai

L'une des contraintes cruciales auxquelles j'ai été confronté pendant ce stage était la durée limitée qui m'a été accordée pour accomplir mes missions et atteindre les objectifs du projet. Le stage était fixé à une période de 27 semaines, ce qui représentait un défi en termes de planification et d'exécution.

5.6.1.1 Durée fixe du stage

Conformément aux exigences de l'établissement d'enseignement et de l'entreprise d'accueil, le stage était prévu pour une durée fixe de 27 semaines. Cette contrainte temporelle était immuable et ne pouvait pas être prolongée.

5.6.1.2 Complexité du projet

Le projet sur lequel j'ai travaillé était d'une certaine complexité, avec des objectifs ambitieux à atteindre. Cette complexité a rendu la gestion du

temps d'autant plus cruciale.

5.6.1.3 Apprentissage et conception initiale

Une période initiale était nécessaire pour me familiariser avec l'entreprise, ses processus, et les spécificités du projet. Cela signifiait que je disposais en réalité de moins de temps pour la phase active de développement.

5.6.1.4 Solutions

Face à cette contrainte de durée, j'ai adopté une approche stratégique pour maximiser mon efficacité.

5.6.1.4.1 Planification avancée

Dès le début du stage, avec Monsieur NEROT, chef du projet, nous avons élaboré un plan détaillé, en identifiant les tâches essentielles à accomplir et en les répartissant sur la période disponible.

5.6.1.4.2 Priorisation des tâches

Nous avons établi des priorités claires en fonction des objectifs du projet et de leur importance stratégique, afin de nous concentrer sur les éléments les plus critiques en premier.

Malgré la contrainte de durée du stage, j'ai réussi à atteindre mes objectifs en me concentrant sur l'efficacité, la planification minutieuse et la gestion rigoureuse du temps. Cette expérience m'a permis d'acquérir des compétences précieuses en gestion du temps et en prise de décision stratégique.

5.6.2 Limitations en ressources humaines

5.6.2.1 Effectif limité et surcharge de travail

Tout d'abord, il est important de rappeler, que j'ai principalement travailler seul sur le projet de création de l'application web Morpheus. Paradoxalement, une des principales limitations auxquelles j'ai été confronté tout au long de ce stage était liée aux ressources humaines. En effet l'équipe de développeurs est composée de quatres personnes, en plus de moi en tant que stagiaire, qui ont toutes déjà un emploi du temps très chargé. Il est donc apparu certaines problématiques lorsque j'avais besoin de l'aide ou de la participation de mes collègues au projet. Il me revient en tête un exemple assez significatif. Alors que j'avancais sur la partie frontend de l'application, Monsieur NEROT devait développer l'API qui me permettrait de faire des requêtes à la base de données. Cependant étant très surchargé il n'a pas pu me fournir l'API dans les temps souhaités.

5.6.2.2 Solutions

Pour pallier à cette limitation en ressources humaines, nous avons mis en place des méthodes de communication efficaces afin de ne pas perdre de temps lorsque nous avions à échanger à propos du projet. De plus j'ai pris la responsabilité de développer moi-même une API afin de pouvoir tester la connexion à la base de données.

5.6.3 Conclusion

Au cours de ce projet, j'ai été confronté à un ensemble varié de contraintes et de limitations, chacune apportant son propre défi. Cependant, au lieu de voir ces contraintes comme des obstacles insurmontables, je les ai abordées comme des opportunités d'apprentissage et de croissance.

La gestion des contraintes, des délais serrés, des ressources humaines limitées, et d'autres défis m'a permis de développer des compétences essentielles en gestion de projet, en résolution de problèmes, et en collaboration. Ces expériences m'ont montré que, même lorsque les ressources sont limitées, une planification minutieuse, une communication

efficace, et une approche stratégique peuvent conduire à des résultats réussis.

En fin de compte, la gestion des contraintes et des limitations a été une expérience formatrice qui a renforcé ma compréhension de la réalisation de projets dans des conditions du monde réel. Je suis fier des résultats que j'ai obtenus malgré ces défis, et je suis convaincu que ces compétences acquises me seront bénéfiques dans mes futures entreprises professionnelles.

5.7 Synthèse des enjeux et de l'importance

Pendant mon stage chez SSI SCHÄFER, j'ai travaillé sur un projet stratégique de grande importance, allant au-delà d'une simple tâche. Ce projet visait à relever des défis clés pour l'entreprise à long terme. Dans cette synthèse, je vais mettre en lumière les enjeux majeurs du projet, expliquer son importance pour SSI SCHÄFER, détailler les objectifs globaux, évoquer les contraintes rencontrées, et mettre en avant mon approche pour les surmonter.

Le projet avait un impact direct sur la compétitivité de l'entreprise dans un marché en constante évolution. Il visait à consolider sa position et à répondre aux attentes changeantes des clients. Le projet répondait à des enjeux financiers, en améliorant la communication sur le logiciel Morpheus et en offrant de nouvelles fonctionnalités. Il avait également des enjeux stratégiques, favorisant la promotion de Morpheus et son accessibilité. Les enjeux humains permettront aux clients de gérer Morpheus depuis leurs bureaux via un navigateur web.

Les objectifs globaux du projet comprenaient le développement d'un client web Morpheus, le rendant plus accessible et ergonomique, ainsi que la création d'une application web responsive.

Le principal enjeu du projet de stage était d'explorer les possibilités de développement et de poser des bases solides pour l'avenir.

Les objectifs du projet de stage incluaient le test des outils de développement, la création d'une maquette et d'une première version de

l'application web.

Les parties prenantes comprenaient le directeur général de SSI SCHÄFER France ainsi que le chef de projet et développeur du nouveau client Morpheus.

Le projet était soumis à des contraintes de délai, avec une période de stage fixe de 27 semaines. La complexité du projet et la nécessité d'une phase d'apprentissage initiale ont renforcé cette contrainte. Malgré cela, une planification avancée et une priorisation des tâches ont permis d'atteindre les objectifs.

Les limitations en ressources humaines, avec une équipe restreinte, ont également influencé le projet, mais malgré cela, des compétences essentielles en gestion de projet ont été développées.

En conclusion, ce projet a été une opportunité d'apprentissage et de croissance malgré les contraintes et les limitations. La gestion efficace de ces défis a renforcé mes compétences et me servira dans ma future carrière professionnelle.

6

Méthodologie et outils

La réussite d'un projet repose souvent sur une méthodologie solide et l'utilisation adéquate d'outils appropriés. Au cours de mon stage au sein de SSI SCHÄFER, j'ai eu l'opportunité de travailler sur un projet stimulant qui nécessitait une approche réfléchie et une utilisation judicieuse des ressources technologiques. Dans cette section, je vais vous présenter la méthodologie que nous avons adoptée pour mener à bien ce projet ainsi que les outils technologiques qui ont été essentiels à sa réalisation.

La méthodologie que nous avons choisie pour aborder ce projet a joué un rôle déterminant dans ma capacité à atteindre mes objectifs de manière efficace et efficiente. De plus, les outils que j'ai sélectionnés ont facilité la gestion, la collaboration et la mise en œuvre de ce projet complexe. En comprenant la méthodologie et les outils que j'ai utilisés, vous aurez une vision plus précise de la manière dont j'ai planifié et exécuté ce projet avec succès.

6.1

Méthodologie adoptée

La réalisation de tout projet réussi repose sur une méthodologie solide et des approches bien définies. Dans cette section, nous allons plonger dans les détails de la méthodologie que j'ai adoptée pour concevoir, développer et mettre en œuvre mon projet.

La méthodologie adoptée a été un élément clé de mon succès, en m'a aidant à maintenir la structure, la cohérence et la qualité tout au long du processus. Cette section jettera les bases pour la suite de la présentation, où nous explorerons les résultats obtenus grâce à l'application de cette méthodologie.

6.1.1**Choix de la méthodologie : méthode en V³⁰**

Le modèle en V est un cadre d'organisation des activités de développement d'un produit. Il se caractérise par deux flux d'activités : l'un descendant qui décompose le produit jusqu'à sa réalisation, et l'autre ascendant qui assemble le produit tout en vérifiant sa qualité. Ce modèle découle de l'approche séquentielle et linéaire du modèle en cascade, mais il l'enrichit en intégrant des activités d'intégration système à partir de composants plus élémentaires. De plus, il associe chaque phase de production à sa phase de validation correspondante, créant ainsi une structure en forme de V.

Dérivé de l'ingénierie système, le modèle en V est souvent perçu comme un cycle de projet, bien que l'ingénierie système et la gestion de projet soient complémentaires. L'ingénierie système se concentre sur le développement du produit, tandis que la gestion de projet vise à atteindre les avantages escomptés par le client ou l'utilisateur. Il est donc important de noter que le modèle en V ne constitue pas en lui-même un cycle de projet.

Comme nous le verrons tout au long de cette section, même si le projet n'a pas suivi parfaitement les étapes d'une méthodologie en V, on peut y retrouver beaucoup de similitudes.

30. Référence utilisée dans la section 6.1.1 Choix de la méthodologie : méthode en V : [56].

6.1.2 Étapes de la méthodologie

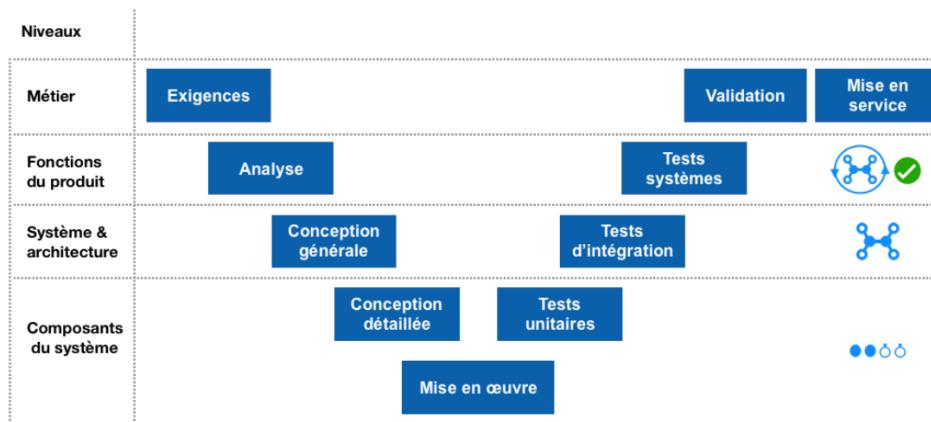


FIGURE 6.1.1 – Étapes de la méthodologie en V

Source de l'image : https://upload.wikimedia.org/wikipedia/commons/9/9d/Cycle_en_V_g%C3%A9n%C3%A9ral.png?uselang=fr

6.2 Planification et gestion de projet

Au cœur de tout projet réussi réside une planification minutieuse et une gestion efficace. Dans cette section, nous allons plonger dans les détails de la manière dont mon projet a été structuré, planifié et géré pour atteindre ses objectifs.

La planification et la gestion de projet sont des éléments cruciaux qui guident chaque étape de notre travail. Elles nous permettent d'établir des fondations solides, de définir des priorités, de gérer les ressources et de maintenir le cap tout au long du parcours. Dans cette partie, nous allons explorer les principes et les pratiques que j'ai mis en place pour assurer le succès de mon projet.

Chaque aspect de la planification et de la gestion de projet a contribué à ma capacité à atteindre mes objectifs, à respecter les délais et à

maintenir la qualité de mon travail. En comprenant ces processus, vous aurez une vue d'ensemble claire de la façon dont mon projet a été dirigé du début à la fin.

6.2.1 Planification

Phases initiales du projet :

- Installation et configuration des outils
- Expérimentation des outils
- Rédaction détaillée des besoins
- Développement du nouveau client HTML5
- Test et recettage du nouveau client
- Livraison

6.2.2 Diagramme de GANTT

Voici le diagramme de GANTT de mon projet :

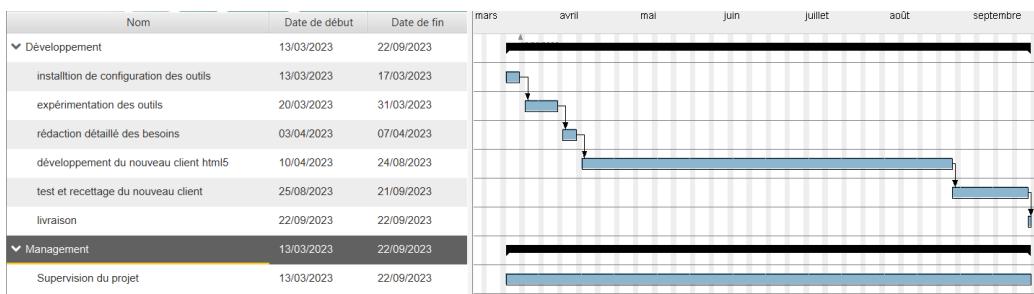


FIGURE 6.2.1 – Diagramme de GANTT initial du projet réalisé au cours du cours

Source de l'image : capture d'écran

Malheureusement nous n'avons pas eu le temps de faire le recettage, néanmoins des tests ont été faits et j'ai rendu mon travail à mon entreprise d'accueil.

6.2.3 Budgétisation

Le budget du projet est composé des éléments suivants :

- Investissement des outils TMS Software : 1800€
- 25 semaines * 5 jours * 750€ : budget temps passé au développement
- 7 semaines * 1 jour * 950€ : budget temps passé au management

6.3 Confluence : outil de gestion de la documentation³¹

Confluence est un logiciel de wiki, utilisé comme logiciel de travail collaboratif. Dans le cadre du projet de développement de l'application web morpheus, confluence m'a permis de documenter le projet et ainsi de permettre une reprise du projet simplifiée par un tiers.

FIGURE 6.3.1 – Extrait de la documentation émise sur confluence

Source de l'image : capture d'écran

31. Référence utilisée dans la section 6.3 Confluence : outil de gestion de la documentation : [57].

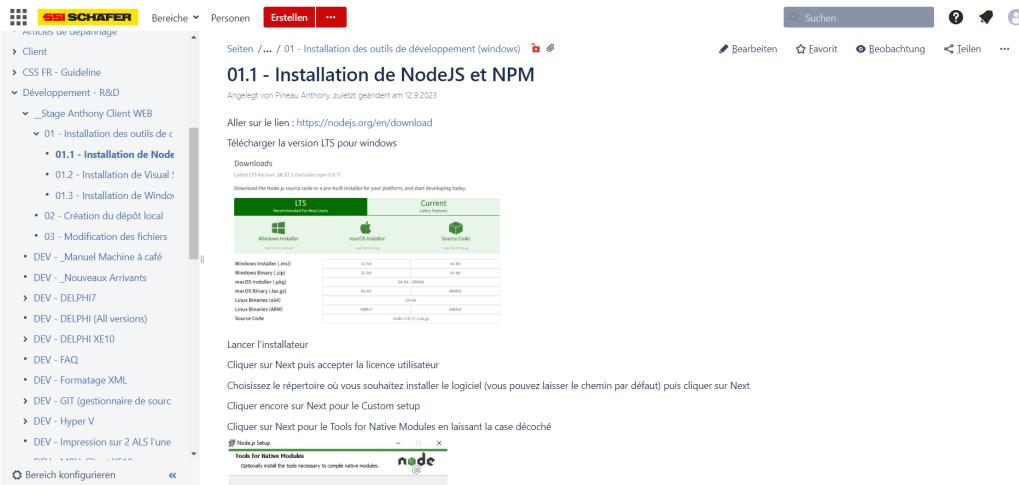


FIGURE 6.3.2 – Extrait de la documentation émise sur confluence

Source de l'image : capture d'écran

6.4

Collaboration et communication

La réussite d'un projet ne repose pas seulement sur la qualité de la planification et de l'exécution, mais également sur la manière dont les membres de l'équipe collaborent et communiquent. Dans cette section, nous allons explorer comment la collaboration efficace et la communication transparente ont été des piliers de ma réussite tout au long de ce projet.

Dans cette partie, nous allons détailler les méthodes, les outils et les pratiques que nous avons utilisés pour favoriser la collaboration au sein de l'équipe. Nous examinerons également comment la communication a été structurée pour assurer la circulation fluide de l'information.

6.4.1**Teams : outil de messagerie instantanée³²**

Microsoft Teams est une plateforme de collaboration et de communication professionnelle développée par Microsoft. Elle permet aux équipes de travail de collaborer en temps réel, que ce soit au sein d'une même entreprise ou entre différentes organisations. Microsoft Teams offre une gamme de fonctionnalités, notamment la messagerie instantanée, la vidéoconférence, le partage de fichiers, la gestion de tâches, l'intégration d'applications tierces, et bien plus encore.

Teams s'avère extrêmement pratique pour faciliter la communication avec les collègues, particulièrement lorsqu'ils travaillent à distance. De plus, il sert de manière très pratique pour la tenue de réunions en ligne. Pendant mon stage, j'ai principalement eu recours à Teams pour exploiter ces deux fonctionnalités.

6.4.2**Outlook : outil de messagerie³³**

Outlook est un client de messagerie électronique qui permet aux utilisateurs de gérer leurs e-mails, leurs calendriers, leurs tâches, leurs contacts et leurs notes en un seul endroit. Il offre un ensemble de fonctionnalités qui facilitent la communication, la planification et l'organisation de tâches professionnelles et personnelles. Les principales caractéristiques d'Outlook comprennent la réception et l'envoi d'e-mails, la gestion des calendriers pour planifier des rendez-vous et des réunions, la création de tâches pour suivre les activités à accomplir, la gestion des contacts pour organiser les informations de contact, et la possibilité de prendre des notes.

Pendant mon stage, j'ai principalement utilisé Outlook pour l'envoi de courriels à mes collègues et à Monsieur Hamrioui. Mais j'ai aussi profité de ses autres fonctionnalités pour planifier diverses tâches, rendez-vous et réunions grâce au calendrier intégré dans l'application.

32. Référence utilisée dans la section 6.4.1 Teams : outil de messagerie instantanée : [58].

33. Référence utilisée dans la section 6.4.2 Outlook : outil de messagerie : [59].

6.4.3 "Point dev" : réunion hebdomadaire

Toutes les semaines, le vendredi à 14H, nous effectuons une réunion, appelée "Point dev", qui réunissait tous les développeurs de l'entreprise. Durant cette réunion, chacun indiquait ce qu'il avait fait pendant la semaine écoulée et pouvait parler des problèmes rencontrés. Cela permettait aussi de trouver des solutions aux problèmes évoqués.

6.4.4 Réunions ponctuelles

Parfois nous faisions des réunions avec Monsieur GOURDON et Monsieur NEROT afin de parler de l'avancement du projet et vérifier qu'il avançait toujours dans la bonne direction.

6.4.5 Communication avec les collègues

Bien que j'étais principalement seul à travailler sur le projet, je pouvais, si besoin, discuter avec mes collègues des problèmes rencontrés et chercher ensemble une solution. La communication avec eux a toujours été facile et fluide et a participé à la bonne réussite du projet.

6.5 Outils technologiques

Les avancées technologiques ont transformé la manière dont nous concevons, développons et mettons en œuvre des projets. Dans cette section, nous allons explorer les outils technologiques que j'ai exploités pour donner vie à mon projet et atteindre mes objectifs.

Dans cette partie, je vais présenter les outils technologiques spécifiques que j'ai utilisés, en expliquant pourquoi je les ai choisis. Nous allons détailler comment ces outils ont contribué à la réalisation de mes objectifs et comment ils ont amélioré mon efficacité tout au long du projet.

6.5.1 Delphi³⁴

Delphi est à la fois un langage de programmation orienté objet et un environnement de développement intégré (EDI) pour ce langage. Delphi est un EDI propriétaire fonctionnant sous Windows créé en 1995 par Borland. Delphi embarque une version orientée objet du langage Pascal : le Pascal Objet, renommé Langage de programmation Delphi au fil des modifications apportées par Borland.

Dans le cadre de mon projet, Delphi a été le premier langage de programmation envisagé pour réaliser l'application web Morpheus. Avec l'utilisation du framework TMS Web de TMS Software, j'ai pu créer la première version de l'application web

6.5.1.1 TMS Software et TMS Web

6.5.1.1.1 TMS Software

La société TMS Software est une entreprise de développement de logiciels spécialisée dans :

- Le développement de composants VCL, FMX, LCL, FNC, ASP.NET, .NET, IntraWeb
- Des projets de développement pour Windows, le Web, Android, iOS, macOS, Linux, et Node.js
- La formation, le conseil et le développement de projets personnalisés.

6.5.1.1.2 TMS Web

TMS Web est un framework permettant de créer des applications web en Delphi.

D'abord envisagé parce que déjà utilisé par l'entreprise, cet outil technologique m'a permis de me rendre compte qu'il n'était pas le plus adapté pour le projet et a participé au choix d'une technologie plus adaptée.

34. Références utilisées dans la section 6.5.1 Delphi : [60] [61] [62] .

6.5.2**HTML** ³⁵

Le HyperText Markup Language, couramment abrégé en HTML, représente le langage de balisage conçu pour la création de pages web. Ce langage permet la rédaction d'hypertexte, la structuration sémantique des pages web, la mise en forme de contenu, la création de formulaires interactifs et l'intégration de ressources multimédias telles que des images, des vidéos et des scripts informatiques. L'HTML offre également la possibilité de créer des documents compatibles avec une grande variété de dispositifs, en respectant les normes d'accessibilité du web. HTML est donc le langage que j'ai utilisé pour la création de toutes mes pages web.

6.5.3**CSS** ³⁶

Les feuilles de style en cascade, souvent abrégées en CSS à partir de l'anglais Cascading Style Sheets, représentent un langage informatique utilisé pour définir la mise en forme des documents HTML et XML.

6.5.3.1**Sass**

Sass (Syntactically awesome stylesheets) est un langage de script préprocesseur³⁷ qui est compilé ou interprété en CSS (Cascading Style Sheets).

CSS et Sass ont été les principaux outils que j'ai utilisés pour mettre en forme mes pages web.

6.5.4**JavaScript** ³⁸

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript

35. Référence utilisée dans la section 6.5.2 HTML : [63].

36. Références utilisées dans la section 6.5.3 CSS : [64] [65] .

37. Retrouver en annexe plus d'informations sur ce qu'est un langage de script préprocesseur

38. Références utilisées dans la section 6.5.4 JavaScript : [66] [67] [68] [69] .

est au cœur des langages utilisés par les développeurs web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript pour l'interpréter.

6.5.4.1 Typescript

TypeScript représente un langage de programmation open source créé par Microsoft dans l'optique d'améliorer et de renforcer la qualité de la production de code JavaScript. Il s'érige comme un sur-ensemble syntaxique rigoureux de JavaScript, ce qui signifie que tout code JavaScript valide est également compatible avec TypeScript. Le code TypeScript est transformé en JavaScript via une opération de transcompilation, permettant ainsi son interprétation par n'importe quel navigateur web ou moteur JavaScript.

6.5.4.2 React

React (également connu sous le nom de React.js ou ReactJS) est une bibliothèque logicielle JavaScript open source développée par Facebook (aujourd'hui Meta) à partir de 2013. L'objectif principal de cette bibliothèque logicielle est de simplifier la création d'applications web monopages en permettant la création de composants dépendants de l'état, qui génèrent une page (ou une portion de page) HTML à chaque modification de cet état.

React se concentre exclusivement sur la gestion de l'interface utilisateur de l'application, considérée comme la vue dans le modèle MVC (Modèle-Vue-Contrôleur). Par conséquent, elle peut être utilisée en combinaison avec d'autres bibliothèques logicielles ou frameworks MVC tels qu'AngularJS. Ce qui distingue cette bibliothèque logicielle de ses concurrentes, c'est sa flexibilité et ses performances. Elle travaille avec un DOM virtuel et met à jour le rendu dans le navigateur uniquement lorsque cela est nécessaire, ce qui contribue à améliorer les performances de l'application.

6.5.4.3 DevExtreme par DevExpress

6.5.4.3.1 DexExpress

Developer Express Inc. est une entreprise de développement de logiciels. Au départ, DevExpress produisait des contrôles d'interface utilisateur pour Borland Delphi/C++Builder et des contrôles ActiveX pour Microsoft Visual Studio. À l'heure actuelle, DevExpress propose des produits destinés aux développeurs qui utilisent Delphi/C++Builder, Visual Studio, ainsi que les technologies HTML5/JavaScript.

6.5.4.3.2 DevExtreme

De Angular et React à Vue, DevExtreme comprend une collection complète de composants d'interface utilisateur haute performance et réactifs pour une utilisation dans des applications web traditionnelles et des applications mobiles de nouvelle génération.

Tous ces outils font partie des langages de programmation/framework que j'ai utilisé pour développer l'application web Morpehus.

6.5.5 Git³⁹

Git est un logiciel de gestion de version décentralisé. Dans le projet, il m'a principalement servi à faire une sauvegarde régulière du code produit. Par ailleurs dans un projet où plusieurs développeurs travaillent en ensemble il est possible de créer différentes branches du projet, où chacun travaille sur sa partie, puis de fusionner les résultats obtenus.

6.5.6 Insomnia⁴⁰

Insomnia, est une plateforme open source de Kong Inc. (en) pour tester des API. Cet outil m'a permis de tester des requêtes sur l'API que nous avons créé sans avoir besoin de développer quoi que ce soit.

39. Référence utilisée dans la section 6.5.5 Git : [70].

40. Référence utilisée dans la section 6.5.6 Insomnia : [71].

6.5.7**Microsoft SQL Server & Microsoft SQL Server Management Studio⁴¹**

6.5.7.1**Microsoft SQL Server**

Microsoft SQL Server est un système de gestion de base de données (SGBD) en langage SQL incorporant entre autres un SGBDR (SGBD "relationnel").

6.5.7.2**Microsoft SQL Server Management Studio**

Microsoft SQL Server Management Studio (SSMS) est une application logicielle développée par Microsoft qui est utilisée pour configurer, gérer et administrer tous les composants au sein de Microsoft SQL Server.

Ces outils technologiques sont les outils utilisés par l'entreprise pour la gestion de la base de données de Morpheus. Ainsi ils ont donc été la base pour moi de la base de données que j'ai utilisée au cours du projet.

6.5.8**Visual Studio Code⁴²**

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code (IntelliSense.), les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

Visual Studio Code est l'éditeur de code que j'utilise de nombreuses années, je sais donc assez bien comment l'utiliser. Il a donc été évident pour moi de l'utiliser dans ce projet afin d'être le plus productif possible et de ne pas perdre de temps à prendre en main un nouvel éditeur de code.

41. Références utilisées dans la section 6.5.7 Microsoft SQL Server & Microsoft SQL Server Management Studio : [72] [73] .

42. Référence utilisée dans la section 6.5.8 Visual Studio Code : [74].

6.5.9 Modelio⁴³

Modelio est un outil de modélisation UML. Il m'a notamment servi à créer le diagramme de classe des nouvelles tables de la base de données.

6.5.10 Latex⁴⁴

LaTeX est un langage et un système de composition de documents. Il s'agit d'une collection de macrocommandes destinées à faciliter l'utilisation du « processeur de texte » TeX de Donald Knuth. J'ai notamment utilisé Latex pour la rédaction de ce mémoire ainsi que les rapports d'avancements durant tout le stage.

43. Référence utilisée dans la section 6.5.9 Modelio : [75].

44. Référence utilisée dans la section 6.5.10 Latex : [76].

7

Project progress (Déroulement du projet)⁴⁵

The success of a project depends on its objective and tools, but also on how it is planned, executed and managed throughout its life cycle. During my internship at SSI SCHAEFER, I had the opportunity to participate in a large-scale project that required careful planning and rigorous execution. In this section, I will immerse you in the complete progress of this project, describing the different steps that I followed.

7.1

Objectives et initial planning

Clarity of objectives and methodical planning are the foundations of any successful project. In this section, we'll explore the goals I set at the start of my project and the initial planning that guided my actions.

The process of defining objectives is a fundamental step in carrying out any project. In this part, we will detail these objectives, explain why they were chosen, and how they were aligned with the issues and needs that we identified previously.

Initial planning is just as essential. It allows us to define the trajectory of the project, identify key stages, necessary resources, and set realistic deadlines. In this section, I will present the initial plan that I had in place.

By understanding these elements, you will have a clear overview of how my project was designed and structured to achieve specific objectives, while taking into account the constraints and realities of the environment. This part will lay the foundations for the rest of the dissertation, where I will detail the implementation and results in relation to these initial objectives.

45. Retrouver la traduction de la partie Déroulement du projet en Annexe G Traduction de la partie Déroulement du projet

7.1.1 Detailed objectives

The initial objectives of the project were :

- Develop a morpheus client in full HTML 5 and responsive for some windows of the existing Mopheus client
- Experiment the development tools of TMS Web
- Make a UI mockup
 - Main window
 - Menu
 - Icons
 - Style (as windows' style : colors)
 - X configurable levels and sub levels
 - ◆ From a database : display or not some menu entries
 - Function buttons
 - Style
 - Icons
 - Size of buttons (16, 32 or 48)
 - Make buttons visible or not
 - Buttons layout depending of the number : alignements..
 - Treeview
 - Style
 - Icons (in front of the label with touching triangle)
 - Display a list over a level following a request
 - Dynamically load of trees
 - ◆ Configuration to make the trees visible or not
 - Style management
 - MDI multi-window management (tabs)
 - Icons
 - Can be closed or not with a cross
 - Modal window management
 - Cross to close the window visible or not
 - Position memory
 - Login

- Database connection
- Write a description of the client's operating mode at the ergonomic level
- Develop a first version with basic windows
 - Main window
 - Menu
 - Icons
 - Style (as windows' style : colors)
 - X configurable levels and sub levels
 - ◆ From a database : display or not some menu entries
 - Function buttons
 - Style
 - Icons
 - Size of buttons (16, 32 or 48)
 - Make buttons visible or not
 - Buttons layout depending of the number : alignements..
 - Treeview
 - Style
 - Icons (in front of the label with touching triangle)
 - Display a list over a level following a request
 - Dynamically load of trees
 - ◆ Configuration to make the trees visible or not
 - Style management
 - MDI multi-window management (tabs)
 - Icons
 - Can be closed or not with a cross
 - Modal window management
 - Cross to close the window visible or not
 - Position memory
 - Login
 - Database connection
 - Access rights management
 - Windows configuration management in XML
 - Window type
 - Associated grid
 - SQL query

7.1.2 Initial planning

We've already talked about it a little but here's the planning of the initial tasks

- Installation and configuration of tools
- Experimenting with tools
- Detailed drafting of requirements
- Development of the new HTML5 client
- Testing and acceptance of the new client
- Delivery

7.2 Project stages

Completing a successful project requires a solid plan and methodical execution. In this section, we will explore the key stages that marked my journey, from the start to the end of the project. This timeline will allow us to understand the logical sequence of my work and the accomplishments that marked each phase.

Managing the stages of a project is essential to ensure consistency, efficiency and control of deadlines. Each milestone represents a specific set of tasks, objectives, and deliverables that move us closer to achieving our overall mission. During this section, we will detail these steps and explain how they aligned with the initial objectives.

7.2.1 Step #1 : Choosing the programming language for the frontend

The front-end application of the Morpheus web application is undoubtedly the most important of the project. Indeed, it represents the graphical interface part and will therefore be the one with which the user will interact. It therefore seems logical that the first step of the project is to choose the most suitable programming language to develop this application.

7.2.1.1 Possible solutions

Three main solutions were considered to suit the objectives of this project. We will see what these solutions are and why they were considered.

7.2.1.1.1 Delphi and TMS Software

Delphi and TMS Software were the first solution considered in this project. Indeed, the Delphi language is already used within the company SSI Schäfer, it was logical for them to ask me to carry out the project using this tool. However, I very quickly realized that this was not going to be the most suitable tool for our web application project.

Find a presentation of the result obtained with Delphi in the Appendix H Résultat obtenu avec Delphi.

Find an example of code produced in Delphi in the Appendix I Code produit en Delphi.

7.2.1.1.2 JavaScript

JavaScript was the second option I thought of for doing this project. Indeed, seeing that Delphi would not be the most suitable solution, I started to develop a web application in vanilla JavaScript. I was quite satisfied with the first results but I quickly realized that if I had to redevelop all the components I needed it would take way too much time. So I needed to find a JavaScript framework that would give me access to the components I needed.

7.2.1.1.3 React and DevExtreme

After thinking and discussion with Mr. NEROT, we thought that React could be a good solution for developing the Morpheus web application. However, we also had to find a framework capable of providing us with the graphic components we needed. That's when we thought of DevExtreme. In fact, the company already uses DevExpress frameworks for the development of the Morpheus heavy client. Following an exchange with Mr. GOURDON, he told us that there was a framework for JavaScript. So we looked into it and decided to test out what we could do with it.

Find an example of code produced in React and DevExtreme in the Appendix J Code produit en React et DevExtreme⁴⁶.

46. You can have an overview of the results produced in React and DevExtreme in the

7.2.1.2 Chosen solution

Thus in light of the tests carried out in the different languages/frameworks as well as the technical comparison between TMS Web and DevExtreme⁴⁷, we chose to retain the React + DevExtreme solution. In fact, we established that the React + DevExtreme solution was more suitable for our project due to its increased flexibility for multiplatforms as well as a wide range of modern widgets.

7.2.1.2.1 Financial impact

The advantage of the DevExtreme framework, compared to other frameworks, is that you can use it for development without having to pay. This is an advantage because if we realize that the framework is not the right one, we do not have to pay for a framework for nothing. But it remains very complicated to quantify the money saved that can be obtained from the use of one framework compared to another, because it is not the only variant to take into account. Indeed, the experience of the developer or even the complexity of the project can be important factors.

section 9 Conclusion

47. Find the technical comparison between the TMS Web and DevExtreme solutions in the Appendix K Comparatif technique entre les solutions TMS Web et DevExtreme

7.2.2

Step #2 : Development of the first version of the frontend

After choosing the programming language we wanted to work with, I started developing the first version of the web application frontend. So the first step was to create a menu to be able to navigate through the different windows of the Morpheus web client. Then I developed a typical window with a grid.

Here is the class diagram that I created to store the information of a grid type window. This part is linked to the creation of the API which interacts with the database. However, it was best to start thinking about the information I want to store in the database when creating the frontend.

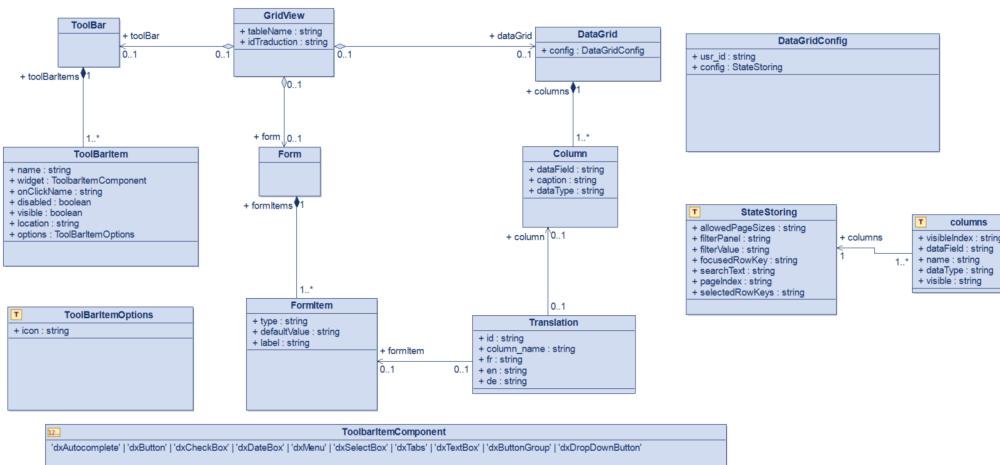


FIGURE 7.2.1 – Class diagram for storing information from a grid window

Image source : screenshot

We will return to the achievements and results of the project in the following part.

7.2.3**Step #3 : Development of a backend
to interact with the database**

Currently a lot of information useful to Morpheus is stored in a database, and not only the data displayed to users but also many management parameters of the graphical interfaces. There are also many stored procedures that allow you to perform certain actions. At the moment, we do not want to change this system, so in order to be able to access the information contained in the database, we have decided to set up a REST API serving as a solution backend to the web application.

As mentioned previously, Mr. NEROT had to develop a REST API in order to be able to communicate with the database. However, being very overloaded with his work, he did not have time to create this API. So I took the initiative to develop the API myself in order to test the web application with data coming from the database. This API works on the CRUD principle.

7.3**Monitoring and control**

Monitoring and control are the pillars that ensure the consistency, quality and control of a project throughout its progress. In this section, we'll dive into the details of how we monitored and controlled my project to ensure it remained aligned with my goals and met quality standards.

Monitoring and controlling are essential elements of project management because they allow you to stay on course, resolve problems in real time, and make informed decisions to ensure the success of the project. In this part, we will detail the methods, tools and practices we used to monitor progress, evaluate results and maintain the quality of my work.

7.3.1 Weekly meeting :"Point dev"

We talked about it earlier in the 6 Methodology part, the "Point dev" is a weekly meeting where each developer talks about what they did during the week. It was therefore an opportunity for me to present my work, which ensured the monitoring of the project and made it possible to check that the project was still going in the right direction.

7.3.2 Ad hoc follow-up meetings

As Managing Director France of SSI SCHAEFER, Mr GOURDON has a very busy schedule. However, it was important to him to follow the evolution of the web application development project. This is why we sometimes had occasional meetings, most often with Mr. NEROT, who allowed me to present my work to him and allowed him, if necessary, to give me his instructions on the procedure to follow to continue the project. on the right track.

8

Cyberattaque au sein de l'entreprise

Tout d'abord il est important de rappeler que l'entreprise SSI SCHÄFER est un groupe mondial et possède de nombreuses filiales dans plusieurs pays différents. Ainsi l'envergure du système informatique est conséquent.

En avril 2023, l'entreprise SSI SCHÄFER a subi une cyberattaque qui a eu pour principale conséquence une coupure du système informatique complet du groupe.

Bien que je n'aie personnellement pas été impacté⁴⁸, ou très peu, j'ai pu assister aux conséquences qu'une telle crise peut avoir au sein d'une grande entreprise. J'ai notamment vu des collègues dans l'impossibilité complète de pouvoir travailler.

Il est aussi important de noter que le réseau informatique était coupé vers l'extérieur, et donc vers les autres sites de SSI SCHÄFER, mais nous pouvions toujours accéder aux ressources sur le réseau local de Cholet.

8.1

Impacts et conséquences

Les répercussions de cette cyberattaque ont été nombreuses et ont touché l'ensemble de l'entreprise, y compris sa filiale française. Dans ce contexte, je vais brièvement exposer les principales conséquences que j'ai identifiées en France.

8.1.1

Télétravail impossible

En premier lieu, tous les collaborateurs qui dépendent de l'accès à l'ERP de l'entreprise, particulièrement via un VPN, se sont retrouvés dans l'incapacité totale d'y accéder, étant donné que la connectivité du réseau

48. En effet je travaillais surtout sur mon poste en local et n'avait besoin que d'une connexion internet pour effectuer des recherches. Durant la cyberattaque j'ai donc tout simplement utilisé mon téléphone comme point d'accès avec mes données mobiles (4G et 5G)

informatique de l'entreprise vers l'extérieur a été interrompue. Par conséquent, le télétravail était impossible pour eux. De plus, le serveur de messagerie était également hors service.

8.1.2 Chômage partiel

Effectivement, comme je l'ai mentionné précédemment, certains de mes collègues se sont malheureusement retrouvés dans l'incapacité de travailler. Ils avaient alors deux alternatives : prendre des congés pour ceux qui le pouvaient ou opter pour le chômage partiel.

8.1.3 Impossibilité d'aider le client

Étant donné que la plupart des outils d'assistance client sont normalement accessibles via le réseau informatique, il était donc impossible de fournir de l'aide aux clients de l'entreprise pendant la cyberattaque.

8.2 Solutions

8.2.1 Venue des employés sur le site de Cholet

Bien que cette solution aurait eu sa place dans la catégorie conséquences de la cyberattaque, je tiens à la présenter ici car elle représente l'une des rares alternatives viables qui ont été envisagées. Face à l'incapacité de télétravailler, les collègues qui nécessitaient un accès à l'ERP se sont rendus sur le site de Cholet pour poursuivre leur travail.

8.2.2 Utilisation d'autres outils

Après la reprise de la connexion Internet et la réouverture de l'accès aux clients, en attendant la réactivation des VPN indispensables pour se connecter, quelques solutions temporaires ont été déployées. L'utilisation de TeamViewer et Teams s'est avérée très utile pour maintenir l'assistance aux clients. Cependant, ces méthodes n'étaient pas les plus efficaces et ont rapidement été abandonnées une fois les VPN rétablis.

8.3 Conclusion

La cyberattaque subie par SSI SCHÄFER m'a permis de prendre conscience des répercussions potentiellement dévastatrices qu'une crise majeure peut avoir au sein d'une entreprise de cette envergure. J'ai également pu constater la nécessité de faire preuve d'ingéniosité pour mettre en place des solutions, même si elles s'avèrent souvent temporaires.

9

Réalisations et résultats

Au cœur de tout projet se trouvent les réalisations et les résultats qui attestent du succès et de la valeur d'une initiative. Au cours de mon stage au sein de SSI SCHÄFER, j'ai eu l'opportunité de contribuer activement à un projet qui a généré des réalisations significatives et des résultats tangibles. Dans cette section, je vais vous dévoiler ces réalisations et résultats, qui illustrent l'impact concret de notre travail sur le projet.

Chacune des actions entreprises au cours de ce projet visait à atteindre des objectifs spécifiques, et ces objectifs étaient étroitement liés à la réussite de l'entreprise. Au-delà des défis et des obstacles rencontrés en chemin, ces réalisations et résultats reflètent notre engagement envers la réussite et notre capacité à transformer des idées en actions concrètes.

Au fil de cette section, je vais mettre en avant les réalisations clés qui ont émergé de notre travail acharné, les résultats obtenus grâce à nos efforts collectifs, et l'impact positif de ces réalisations sur l'entreprise ou l'organisation. Je vais également illustrer comment ces réalisations ont contribué à l'atteinte des objectifs du projet.

En comprenant les réalisations et les résultats de ce projet, vous aurez une vue complète de la valeur ajoutée que j'ai apportée en tant que stagiaire, ainsi que de l'impact concret de notre collaboration sur les objectifs globaux de l'entreprise. Ces réalisations témoignent de notre engagement envers l'excellence et de notre contribution à la réussite de l'entreprise.

9.1

Objectifs atteints

9.1.1

Rappel des objectifs du projet de stage

Pour rappel les objectifs du projet étaient les suivants :

- Tester les outils de développement
- Créer une maquette
- Créer une première version de l'application web

9.1.2**Tester les outils de développement**

L'objectif initial était de tester les outils de développement TMS Software. Je pense que cet objectif a été parfaitement réalisé. En effet au-delà d'avoir tester les outils de développement. Nous avons su prouver que ce n'était pas forcément la meilleure solution et changer pour une solution plus appropriée à notre projet.

9.1.3**Créer une maquette**

Pour cet objectif, je n'ai pas créé une maquette au sens traditionnel du terme, néanmoins à la suite du test des outils de développement nous avions une toute première version de l'application web qui faisait office de maquette.

9.1.4**Créer une première version de l'application web**

Au vu des résultats du projet, je pense que nous pouvons dire que l'objectif de créer une première version de l'application web a été rempli.

9.2**Réalisations**

FIGURE 9.2.1 – Capture d'écran du menu de l'application web morpheus

Source de l'image : capture d'écran

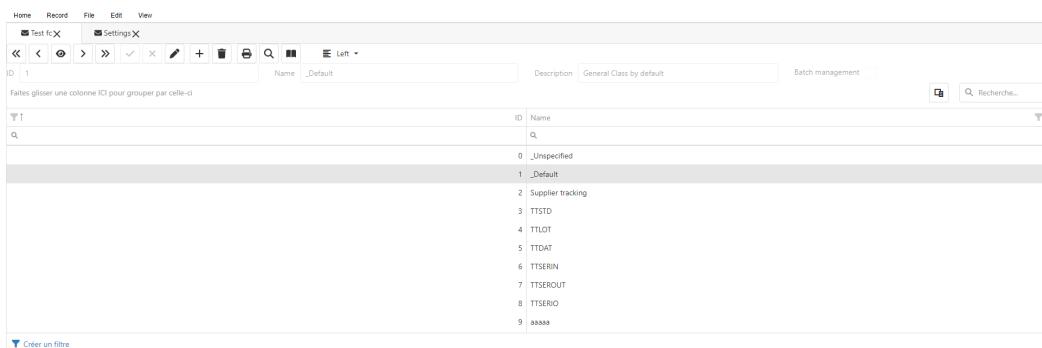


FIGURE 9.2.2 – Capture d'écran de l'application web morpheus

Source de l'image : capture d'écran

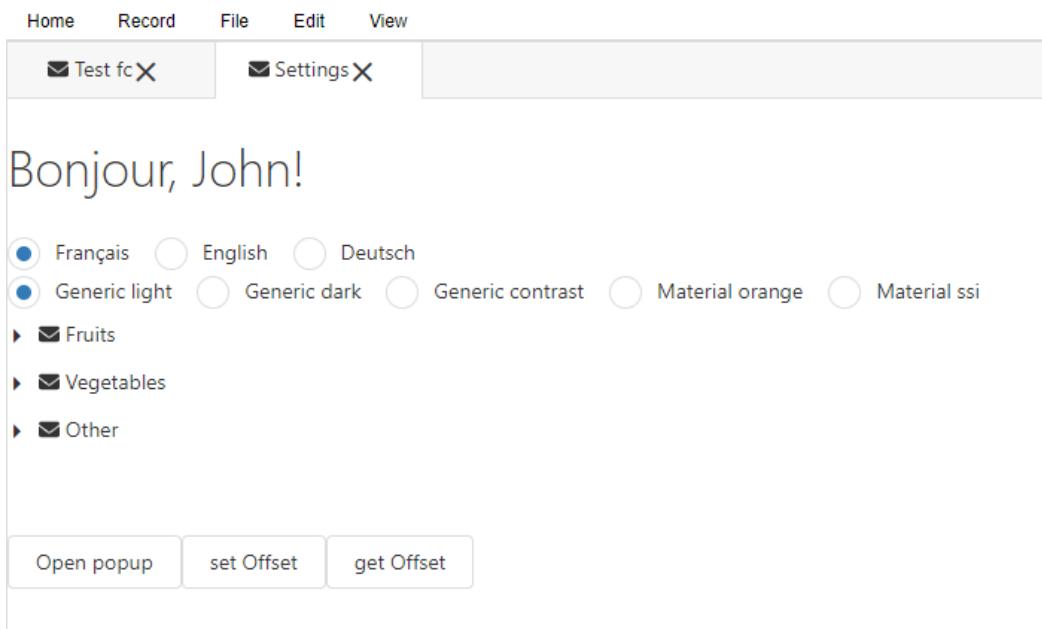


FIGURE 9.2.3 – Capture d'écran de l'application web morpheus

Source de l'image : capture d'écran

9.3 Conclusion

En conclusion, les résultats de ce projet témoignent clairement de sa belle réussite, démontrant que j'ai atteint avec succès chacun des objectifs rigoureusement fixés. Ces objectifs, initialement tracés comme des jalons à franchir, sont désormais des réalisations concrètes. Les réalisations effectuées et les retours positifs des parties prenantes confirment non seulement la réalisation de ces objectifs, mais également la création de valeurs tangibles pour l'entreprise. Cela témoigne de l'efficacité de la planification, de la gestion du projet et de l'engagement envers l'excellence qui ont été au cœur de ce projet. Les résultats obtenus fournissent une base solide pour une future reprise du projet et soulignent l'importance de la persévérance et de l'engagement à atteindre l'excellence dans la réalisation des objectifs professionnels.

10

Compétences mises en œuvre

Chaque expérience professionnelle, notamment un stage, est une opportunité précieuse pour développer et mettre en œuvre des compétences essentielles. Au cours de mon stage au sein de SSI SCHÄFER, j'ai eu la chance de participer activement à un projet qui m'a permis d'acquérir et de mettre en œuvre diverses compétences professionnelles de manière concrète. Cette section est dédiée à ces compétences, qui sont le socle de ma croissance en tant que professionnel.

Lorsque nous évoluons dans un environnement professionnel, il est essentiel de savoir identifier, adapter et appliquer nos compétences en réponse aux défis spécifiques rencontrés. Ce projet a été le terrain d'entraînement idéal pour mettre en pratique mes compétences, tout en les renforçant et en les perfectionnant. En partageant ces compétences mises en œuvre, je souhaite démontrer comment mon stage a été bien plus qu'une simple expérience d'observation, mais une réelle occasion d'apprentissage et de croissance professionnelle.

Au fil de cette section, je vais mettre en avant les compétences clés que j'ai mobilisées et développées tout au long du projet, expliquer comment elles ont été appliquées dans un contexte professionnel concret, et montrer en quoi elles ont contribué aux réalisations et au succès du projet. Ces compétences, acquises au cours de mon stage, seront sans aucun doute un atout précieux dans ma future carrière professionnelle, et je suis enthousiaste à l'idée de les développer davantage dans des contextes similaires.

10.1

Compétences techniques

Les compétences techniques jouent un rôle central dans notre capacité à concevoir, développer, mettre en œuvre et optimiser l'application web morpheus. Tout au long de cette partie, je vais mettre en lumière les compétences spécifiques que j'ai utilisées pour atteindre mes objectifs.

- Développement d'une application web en Delphi avec l'utilisation de TMS Web
- Développement d'une application web en HTML, CSS et JavaScript

- Développement d'une application web en React avec l'utilisation de typescript et du framework DevExtreme
- Réalisation d'un diagramme de classe
- Rédaction de diverses requête SQL
- Rédaction du présent mémoire avec Latex

10.2 Compétences en gestion

La réussite de tout projet ou mission professionnelle repose non seulement sur des compétences techniques solides, mais également sur une expertise en gestion efficace. Dans cette section, nous explorerons les compétences en gestion que j'ai mises en œuvre tout au long de mon stage et comment elles ont contribué à la réalisation de mes objectifs.

10.2.1 Gestion du temps

Au cours de mon stage au sein de SSI SCHÄFER, j'ai pu faire preuve d'une bonne gestion de mon temps, en répartissant les tâches que j'avais en faire pour une semaine donnée.

10.2.2 Planification

Comme tout projet d'envergure, il paraît évident de proposer une planification des différents jalons qui rythmeront l'avancée des travaux. Ainsi j'ai pu mettre en pratique mes compétences en planification en établissant un diagramme de GANTT des différentes étapes du projet de stage.

10.3 Compétences en communication

La communication efficace est le cœur de toute entreprise ou projet réussi. Dans cette section, nous allons explorer en détail les compétences en communication que j'ai utilisées et développées tout au long de mon stage et comment elles ont été essentielles pour atteindre mes objectifs.

10.3.1**Rédaction d'une documentation technique**

Un projet ne dure pas nécessaire seulement le temps d'un stage. J'ai donc dû, en parallèle de l'avancée du projet, écrire une documentation technique de celui-ci, afin qu'une personne tierce puisse reprendre et continuer le projet même après la fin de mon stage et mon départ de l'entreprise.

10.4**Compétences en leadership**

Le leadership est une compétence fondamentale qui transcende les rôles et les industries. Dans cette section, nous allons explorer en détail comment ma capacité à exercer un leadership efficace a été déterminante tout au long de mon stage et comment cela a eu un impact significatif sur sa réussite.

10.4.1**Développement de l'API REST**

Comme je l'ai déjà évoqué plusieurs fois au cours de ce mémoire, cela n'était, à la base, pas prévu que je développe l'API REST qui servirait à communiquer avec la base de données. Néanmoins, lorsque j'ai vu que la personne qui devait s'en charge, n'en avait pas le temps, j'ai alors pris l'initiative de développer ma propre version de l'API afin de pouvoir continuer à avancer dans le projet.

10.5**Compétences en adaptabilité**

L'adaptabilité est une compétence clé dans le monde professionnel en rapide évolution d'aujourd'hui. Dans cette section, nous allons explorer comment ma capacité à m'adapter aux changements, aux défis et aux nouvelles situations a été mise en évidence tout au long de mon stage et comment cela a contribué à sa réussite.

10.5.1**Choix du changement de langage de programmation**

Au départ du projet, l'entreprise souhaitait utiliser le langage de programmation Delphi pour développer l'application web. Néanmoins je me suis vite aperçu que ce n'était pas la technologie la plus adaptée pour ce projet et ai donc suggéré que l'on utilise des technologies plus modernes qui conviendrait mieux. Après avoir présenté mes arguments auprès de monsieur NEROT et monsieur GOURDON, nous avons donc décidé que l'application serait développée en React avec l'utilisation du framework DevExtreme.

10.6**Leçons apprises et développement professionnel**

Au cours de mon stage, j'ai été constamment confrontés à une série de défis exigeants, mais aussi à des moments de réussite gratifiants. Chacun de ces moments a été une occasion d'apprentissage précieuse, contribuant à mon développement professionnel de manière significative.

Une des leçons clés que j'ai tirées de cette expérience est la valeur de la résilience face à l'adversité. Lorsque nous avons été confrontés à des obstacles apparemment insurmontables, j'ai découvert une détermination et une créativité que je ne me connaissais pas. Ces moments difficiles m'ont appris que la persévérance est souvent la clé pour surmonter les défis et atteindre nos objectifs.

D'autre part, mon travail m'a également montré l'importance de la communication efficace. Travailler au sein d'une équipe multidisciplinaire exigeait une communication transparente et ouverte. Apprendre à écouter activement, à exprimer mes idées de manière claire et à résoudre les conflits de manière constructive sont des compétences que j'ai affinées et que je continuerai à cultiver dans mon développement professionnel.

Une autre leçon cruciale a été la valeur de la planification stratégique. Les moments où j'ai pris le temps de planifier soigneusement nos actions et de définir des objectifs clairs ont généralement été les moments où j'ai réalisé mes meilleurs résultats. Cela m'a montré que le temps consacré

à une planification minutieuse est un investissement précieux pour la réussite.

En réfléchissant à ces leçons et à d'autres, nous pouvons voir comment mon expérience au sein de SSI SCHÄFER a profondément influencé mon développement professionnel. Je suis convaincu que les compétences et les connaissances que j'ai acquises ici seront des atouts précieux dans mon parcours professionnel futur, me permettant de relever de nouveaux défis avec confiance et détermination. Cette expérience m'a également renforcé dans mon engagement envers l'apprentissage continu et le développement de compétences pour rester compétitifs et pertinents dans mon domaine.

Abschluß⁴⁹ (Conclusion)

Zusammenfassend hat diese Dissertation die Hauptbestandteile des komplexen und bedeutenden Prozesses des Übergangs von einer Thick-Client-Anwendung zu einer Webanwendung untersucht. Im Rahmen unserer Studie haben wir uns mit verschiedenen technischen, organisatorischen und strategischen Aspekten dieses Übergangs befasst und die mit dieser Entwicklung verbundenen Herausforderungen und Chancen hervorgehoben.

Wir untersuchten zunächst den Kontext dieses Übergangs, stellten das Unternehmen SSI SCHÄFER vor und definierten die Gründe und die Motivationen, die das Unternehmen zu dieser großen Veränderung bewegten. Anschließend legten wir den theoretischen Rahmen der Dissertation fest, der es ermöglichte, die Schlüsselkonzepte zu definieren, die im gesamten Dokument Anwendung fanden. Wir haben außerdem eine vergleichende Analyse zwischen Webanwendungen und Thick-Client-Anwendungen durchgeführt, um die wesentlichen Unterschiede zwischen den beiden Ansätzen besser zu verstehen.

Danach stellten wir das Projekt vor und untersuchten die Herausforderungen und Bedeutung, die es für das Unternehmen hatte. Anschließend kehrten wir zu den, während des Projekts verwendeten, Methoden und Werkzeugen zurück. Damit haben wir die Bedeutung von Planung und Projektmanagement sowie Zusammenarbeit und Kommunikation hervorgehoben. Dies ermöglichte uns auch, die eingesetzten technologischen Werkzeuge vorzustellen.

Anschließend haben wir die wichtigsten Phasen des Migrationsprozesses detailliert beschrieben und dabei den Schwerpunkt auf die strategische Planung, die Auswahl der Technologien und die Entwicklung einer ersten Version der Webanwendung gelegt. Wir haben die entscheidende Bedeutung der interdisziplinären Zusammenarbeit zwischen den Beteiligten hervorgehoben, um die Projektsteuerung und -überwachung sicherzustellen und einen erfolgreichen Übergang zu ermöglichen. Durch den Cyberangriff, dem das Unternehmen ausgesetzt war, konnten wir

49. Retrouver la traduction en français de la section Conclusion en Annexe L Traduction de la partie Conclusion

eine Vision für das Krisenmanagement entwickeln.

Abschließend untersuchten wir die konkreten Ergebnisse des Übergangs und hoben funktionale Verbesserungen hervor. Wir haben aus dieser Erfahrung auch wichtige Lehren gezogen, darunter die Bedeutung sorgfältiger Planung, transparenter Kommunikation und Anpassungsfähigkeit angesichts unvorhergesehener Herausforderungen. Außerdem konnte ich alle Fähigkeiten hervorheben, die ich während meines Praktikums anwenden, erwerben und weiterentwickeln konnte und die als Grundpfeiler für meine zukünftige berufliche Tätigkeit dienen werden.

Kurz gesagt, diese Dissertation hat gezeigt, dass der Übergang von einer Thick-Client-Anwendung zu einer Webanwendung ein komplexer, aber wesentlicher Prozess ist, um in einem sich ständig weiterentwickelnden digitalen Umfeld wettbewerbsfähig zu bleiben. Er betonte auch die Bedeutung der strategischen Ausrichtung, des Teamengagements und einer eingehenden Kosten-Nutzen-Analyse. Diese Erfahrung hat uns gezeigt, dass ein Unternehmen mit sorgfältiger Planung und effektiver Umsetzung diesen Meilenstein erfolgreich meistern und sich an neue Marktanforderungen anpassen kann.

Letztendlich war diese Dissertation viel mehr als nur eine technische Fallstudie, sie war ein tiefer Einblick in die, sich ständig verändernde, Welt der Technologie, des Projektmanagements, der interdisziplinären Kommunikation und der Anpassungsfähigkeit an Veränderungen. Darüber hinaus beschränkt sich diese Dissertation nicht nur auf die Dokumentation eines technologischen Wandels, sie erinnert uns auch daran, dass in der sich ständig verändernden Welt von heute Anpassungsfähigkeit, strategische Planung und Ausdauer für jedes Unternehmen von unschätzbarem Wert sind.

Wie SSI SCHÄFER stehen auch wir alle vor der Notwendigkeit, uns anzupassen und innovativ zu sein, um erfolgreich zu sein. Die aus dieser Erfahrung gewonnenen Erkenntnisse sind Kompassse, die uns auf unserem eigenen Weg in die digitale Zukunft leiten können.

Wenn wir die letzte Seite dieser Memoiren umblättern, sollten wir bedenken, dass unsere Erkundung gerade erst begonnen hat. Die Welt der Technologie entwickelt sich weiter, Herausforderungen tauchen auf unerwartete Weise auf und Chancen eröffnen sich für diejenigen, die bereit sind, sie zu ergreifen. Der Übergang von einer Thick-Client-Anwendung

zu einer Webanwendung ist nur ein Kapitel in der sich ständig weiterentwickelnden Geschichte von Innovation und Anpassung.

Bibliographie

- [1] Wikipedia, “Bibliothèque logicielle.” https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle.
- [2] Wikipedia, “Client léger.” https://fr.wikipedia.org/wiki/Client_l%C3%A9ger.
- [3] Wikipedia, “CRUD.” <https://fr.wikipedia.org/wiki/CRUD>.
- [4] Wikipedia, “Document Object Model.” https://fr.wikipedia.org/wiki/Document_Object_Model.
- [5] Wikipedia, “Progiciel de gestion intégré.” https://fr.wikipedia.org/wiki/Progiciel_de_gestion_int%C3%A9gr%C3%A9.
- [6] Wikipedia, “Environnement de développement.” https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement.
- [7] Wikipedia, “Hypertexte.” <https://fr.wikipedia.org/wiki/LaTeX>.
- [8] Wikipedia, “UML (informatique).” [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique)).
- [9] Wikipedia, “Langage de programmation.” https://fr.wikipedia.org/wiki/Langage_de_programmation.
- [10] Wikipedia, “LaTeX.” <https://fr.wikipedia.org/wiki/LaTeX>.
- [11] Wikipedia, “Modèle-vue-contrôleur.” <https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur>.
- [12] Wikipedia, “Open source.” https://fr.wikipedia.org/wiki/Open_source.
- [13] Wikipedia, “Programmation orientée objet.” https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet.
- [14] Wikipedia, “Réseau privé virtuel.” https://fr.wikipedia.org/wiki/R%C3%A9seau_priv%C3%A9_virtuel.
- [15] Wikipedia, “Système de gestion de base de données.” https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es.
- [16] SSI SCHÄFER, “Définition WCS.” <https://wamas.fr/morpheus-wcs/definition-wcs-quest-ce-quun-wcs/>.
- [17] SSI SCHÄFER, “Définition WMS.” <https://wamas.fr/morpheus-wms/definition-wms-quest-ce-quun-wms/>.

- [18] Wikipedia, “Extensible Markup Language.” https://fr.wikipedia.org/wiki/Extensible_Markup_Language.
- [19] Justine DEBRET - Scribbr, “Remerciements de votre mémoire - Explications et exemples.” <https://www.scribbr.fr/plan-memoire/remerciements-memoire/>.
- [20] Justine DEBRET - Scribbr, “Introduction de votre mémoire.” <https://www.scribbr.fr/plan-memoire/introduction/>.
- [21] Scribbr, “Le plan d'un mémoire : quels éléments à intégrer ?” <https://www.scribbr.fr/category/plan-memoire/>.
- [22] Bas SWAEN - Scribbr, “Cadre Théorique d'un Mémoire - Contenu et Exemple.” <https://www.scribbr.fr/plan-memoire/cadre-theorique-dun-memoire/>.
- [23] SSI SCHÄFER, “SSI SCHÄFER en France.” <https://www.ssi-schaefer.com/fr-fr>.
- [24] SSI SCHÄFER, “Histoire de l'entreprise SSI SCHÄFER.” <https://www.ssi-schaefer.com/fr-fr/a-propos-de-nous/histoire>.
- [25] SSI SCHÄFER, *SSI SCHÄFER en France*. Document interne.
- [26] SSI SCHÄFER, *SSI SCHÄFER vous propose sa suite logicielle MOPHEUS*. Document interne.
- [27] Wikipedia, “Client lourd.” https://fr.wikipedia.org/wiki/Client_lourd.
- [28] Wikipedia, “Application web.” https://fr.wikipedia.org/wiki/Application_web.
- [29] Wikipedia, “Frontend and backend.” https://en.wikipedia.org/wiki/Frontend_and_backend.
- [30] Wikipedia, “Framework.” <https://fr.wikipedia.org/wiki/Framework>.
- [31] Wikipedia, “Interface de programmation.” https://fr.wikipedia.org/wiki/Interface_de_programmation.
- [32] Wikipedia, “Representational state transfer.” https://fr.wikipedia.org/wiki/Representational_state_transfer.
- [33] Wikipedia, “Accès à Internet.” https://fr.wikipedia.org/wiki/Acc%C3%A8s_%C3%A0_Internet.
- [34] Wikipedia, “Internet.” <https://fr.wikipedia.org/wiki/Internet>.
- [35] Wikipedia, “Plateforme client riche.” https://fr.wikipedia.org/wiki/Plateforme_client_riche.

- [36] Wikipedia, “Client (informatique).” [https://fr.wikipedia.org/wiki/Client_\(informatique\)](https://fr.wikipedia.org/wiki/Client_(informatique)).
- [37] AppMaster, “Application de bureau ou application Web : avantages et inconvénients.” <https://appmaster.io/fr/blog/application-de-bureau-ou-application-web-avantages-et-inconvenients>.
- [38] Eposaudio, “Client léger ou client lourd ? Quelle est la différence ?.” <https://www.eposaudio.com/fr/fr/enterprise/insights/articles/thin-client-or-thick-client-whats-the-difference>.
- [39] Ronny Guillaume - Journal du net, “Osez le client lourd !.” <https://www.journaldunet.com/web-tech/developpeur/1142976-osez-le-client-lourd/>.
- [40] Alex Ivankov - Profolus, “Thick Client vs. Thin Client : Advantages and Disadvantages.” <https://www.profolus.com/topics/thick-client-vs-thin-client-advantages-and-disadvantages/>.
- [41] AudvikLabs, “Thin Client vs. Thick Client.” <https://audviklabs.com/thin-client-vs-thick-client/>.
- [42] MinnaLearn, “Les implications de la connectivité et de l’accessibilité.” <https://courses.minnalearn.com/fr/courses/digital-revolution/the-internet-and-connectivity-revolution/the-implications-of-connectivity-and-accessibility/>.
- [43] CyberPreventys, “Quelle est la différence entre Internet et Web ?.” <https://www.cyberpreventys.com/conseils-informatique-cybersecurite/difference-internet-web/>.
- [44] Alexander S. Gillis - Techtarget, “What is a thick client ?.” <https://www.techtarget.com/whatis/definition/fat-client-thick-client>.
- [45] Ali Qamar - Techgenix, “Thick Client vs. Thin Client : Choose The Best for Your Workflow.” <https://techgenix.com/thick-client-thin-client/>.
- [46] Georgios Nanos - Baeldung, “Distributed Systems : Thin and Thick Clients.” <https://www.baeldung.com/cs/distributed-systems-thin-vs-thick-clients>.
- [47] Ionos, “What is a thick client ?.” <https://www.ionos.com/digitalguide/server/know-how/what-is-a-thick-client/>.

- [48] Agence Scroll, “Quels sont les avantages et les inconvénients d'une application web ?.” <https://agence-scroll.com/blog/avantages-et-inconvenients-dune-application-web>.
- [49] Yeeply, “Pourquoi choisir le développement d'application web pour votre site ?.” <https://fr.yeeply.com/blog/developpement-d-application-web-pour-votre-site/>.
- [50] Digitiz, “7 avantages à créer une application web pour votre entreprise.” <https://digitiz.fr/blog/avantages-application-web-entreprise/>.
- [51] Hicones, “Avantages et inconvénients d'une application web pour votre entreprise.” <https://www.hicones.fr/avantages-et-inconvenients-dune-application-web-pour-votre-entreprise/>.
- [52] Arun Goyal - Your story , “TOP 5 PROS AND CONS OF A CUSTOM WEB APPLICATION.” <https://yourstory.com/mystory/top-five-pros-cons-custom-web-application>.
- [53] Ionos, “Qu'est-ce qu'un client léger ?.” <https://www.ionos.fr/digitalguide/serveur/know-how/quest-ce-quun-client-leger/>.
- [54] Proofpoint, “Qu'est-ce qu'un thin client ou client léger ?.” <https://www.proofpoint.com/fr/threat-reference/thin-client>.
- [55] Leslie Saladin - Bluemind, “Webmail ou client lourd : faut-il vraiment choisir ?.” <https://blog.bluemind.net/fr/webmail-ou-client-lourd-faut-il-vraiment-choisir/>.
- [56] Wikipedia, “Cycle en V.” https://fr.wikipedia.org/wiki/Cycle_en_V.
- [57] Wikipedia, “Confluence (logiciel).” [https://fr.wikipedia.org/wiki/Confluence_\(logiciel\)](https://fr.wikipedia.org/wiki/Confluence_(logiciel)).
- [58] Wikipedia, “Microsoft Teams.” https://fr.wikipedia.org/wiki/Microsoft_Teams.
- [59] Wikipedia, “Microsoft Outlook.” https://fr.wikipedia.org/wiki/Microsoft_Outlook.
- [60] Wikipedia, “Delphi (langage).” [https://fr.wikipedia.org/wiki/Delphi_\(langage\)](https://fr.wikipedia.org/wiki/Delphi_(langage)).
- [61] TMS Software, “TMS Software | About.” <https://www.tmssoftware.com/site/about.asp>.
- [62] TMS Software, “WEB framework & components for Delphi & Visual Studio Code | TMS Software.” <https://www.tmssoftware.com/site/tmswebcoreintro.asp>.

- [63] Wikipedia, “Hypertext Markup Language.” https://fr.wikipedia.org/wiki/Hypertext_Markup_Language.
- [64] Wikipedia, “Feuilles de style en cascade.” https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade.
- [65] Wikipedia, “Sass (langage).” [https://fr.wikipedia.org/wiki/Sass_\(langage\)](https://fr.wikipedia.org/wiki/Sass_(langage)).
- [66] Wikipedia, “JavaScript.” <https://fr.wikipedia.org/wiki/JavaScript>.
- [67] Wikipedia, “TypeScript.” <https://fr.wikipedia.org/wiki/TypeScript>.
- [68] Wikipedia, “React.” <https://fr.wikipedia.org/wiki/React>.
- [69] DevExpress, “DevExtreme - JavaScript UI Components for Angular, React, Vue and jQuery by DevExpress.” <https://js.devexpress.com/>.
- [70] Wikipedia, “Git.” <https://fr.wikipedia.org/wiki/Git>.
- [71] Insomnia, “The Collaborative API Development Platform - Insomnia.” <https://insomnia.rest/>.
- [72] Wikipedia, “Microsoft SQL Server.” https://fr.wikipedia.org/wiki/Microsoft_SQL_Server.
- [73] Wikipedia, “SQL Server Management Studio.” https://en.wikipedia.org/wiki/SQL_Server_Management_Studio.
- [74] Wikipedia, “Visual Studio Code.” https://fr.wikipedia.org/wiki/Visual_Studio_Code.
- [75] Wikipedia, “Modelio.” <https://en.wikipedia.org/wiki/Modelio>.
- [76] Wikipedia, “LaTeX.” <https://fr.wikipedia.org/wiki/LaTeX>.

Annexes

Liste des annexes

A	Traduction de la partie Introduction	102
B	Dates importantes de l'histoire SSI SCHÄFER	104
C	Carte des filiales SSI SCHÄFER	105
D	Liste complète des fonctionnalités de MORPHEUS WMS	106
E	Liste complète des fonctionnalités de MORPHEUS WCS	106
F	Traduction de la partie Présentation du projet réalisé au cours du stage	107
F.1	Projet global	107
F.2	Projet du stage	108
F.3	Application existante	108
G	Traduction de la partie Déroulement du projet	110
G.1	Objectifs et planification initiale	110
G.1.1	Objectifs détaillés	111
G.1.2	Planification initiale	113
G.2	Étapes du projet	113
G.2.1	Étape n°1 : Choix du langage de programmation pour le frontend	114
G.2.1.1	Solutions possibles	114
G.2.1.1.1	Delphi et TMS Software	114
G.2.1.1.2	JavaScript	114
G.2.1.1.3	React et DevExtreme	114
G.2.1.2	Solution retenue	115

G.2.1.2.1	Impact financier	115
G.2.2	Étape n°2 : Développement de la première version du frontend	116
G.2.3	Étape n°3 : Développement d'un backend pour interagir avec la base de données	117
G.3	Suivi et contrôle	117
G.3.1	Réunion hebdomadaire :"Point dev"	118
G.3.2	Réunions ponctuelles de suivi	118
H	Résultat obtenu avec Delphi	119
I	Code produit en Delphi	120
J	Code produit en React et DevExtreme	121
K	Comparatif technique entre les solutions TMS Web et DevExtreme	122
L	Traduction de la partie Conclusion	125

A

Traduction de la partie Introduction

Dans le paysage numérique en constante évolution, les approches de développement logiciel s'adaptent continuellement pour répondre aux exigences en perpétuelle évolution des utilisateurs et aux avancées technologiques. Une transformation majeure s'est opérée avec le passage des applications traditionnelles en client lourd à des solutions basées sur le web. La montée en puissance d'Internet et la prédominance des technologies web ont incité les développeurs à explorer les possibilités et les avantages de créer des applications web à partir d'applications client lourd existantes.

Ce mémoire examine en partie le processus complexe de développement d'une application web à partir d'un client lourd préexistant. Son objectif est de fournir aux développeurs un exemple de transition d'une application client lourd en application web au travers d'une étude du projet que j'ai réalisé lors de mon stage de fin d'études et des informations sur ce cheminement de transformation. La discussion abordera les défis, les méthodologies et les meilleures pratiques impliquées dans cette transformation, permettant aux ingénieurs en logiciel de naviguer avec succès dans les subtilités du processus de développement.

Au cours de mon projet de fin d'études, j'ai eu la chance de participer à la conception d'une application web, lors d'une transition depuis une application client lourd. Cette démarche a pour but de rendre le déploiement plus rapide et de rafraîchir l'esthétique de l'interface. Cette expérience m'a motivé à approfondir ma compréhension du processus de développement d'applications web qui évoluent à partir d'applications client lourd.

De manière habituelle, une application web se compose essentiellement de deux composants majeurs : le frontend, qui se rapporte à l'interface graphique perceptible par l'utilisateur, et le backend, qui gère l'accès à la base de données et le traitement des demandes. Dans ce scénario précis, notre application sera élaborée en séparant une partie spécifique destinée à l'interface utilisateur, qui interagira au moyen de requêtes HTTP avec une API REST. Cette API agira comme le lien de communication avec la base de données.

Ce document traitera de la réflexion concernant la conception d'une application web dérivée d'une application client lourd. Nous aborderons

plusieurs éléments, notamment le choix du langage de programmation et du framework, ainsi que la mise en œuvre complète de l'application, englobant la création des interfaces graphiques.

Pour aborder le sujet et fournir des réponses aux questions soulevées, un plan de recherche a été élaboré. En premier lieu, il était essentiel de prendre une décision éclairée concernant le langage de programmation, posant ainsi des fondations solides. Ensuite, la création d'une nouvelle interface graphique a été entreprise. Enfin, l'intégration à la base de données via une API REST a été mise en œuvre pour conclure le processus.

L'objectif d'étude réside donc dans la compréhension du processus de création d'une application web à partir d'une base client lourd, tout en identifiant les distinctions majeures entre ces deux approches.

Pour entamer ce mémoire, nous débuterons par une présentation de l'entreprise qui m'a accueilli, à savoir SSI SCHÄFER. Ensuite, nous établirons le cadre théorique, essentiel pour appréhender les concepts clés abordés dans ce document. Par la suite, nous aborderons la section scientifique, qui se concentrera sur une analyse comparative entre les applications client lourd et les applications web.

Nous plongerons ensuite au cœur du projet, en le présentant de manière détaillée, tout en mettant en lumière ses enjeux et son importance. Un examen approfondi de la méthodologie et des outils qui ont contribué au succès du projet suivra. Nous décrirons ensuite le déroulement du projet, en exposant les différentes étapes entreprises, et nous discuterons des défis rencontrés, des réalisations accomplies, ainsi que des résultats obtenus au cours de cette période de stage.

Avant de conclure, nous dresserons un bilan des compétences acquises et développées au cours des six derniers mois. Enfin, en annexe, vous trouverez des informations complémentaires, suivies d'un résumé du mémoire rédigé en trois langues : français, anglais et allemand.

B

Dates importantes de l'histoire SSI SCHÄFER

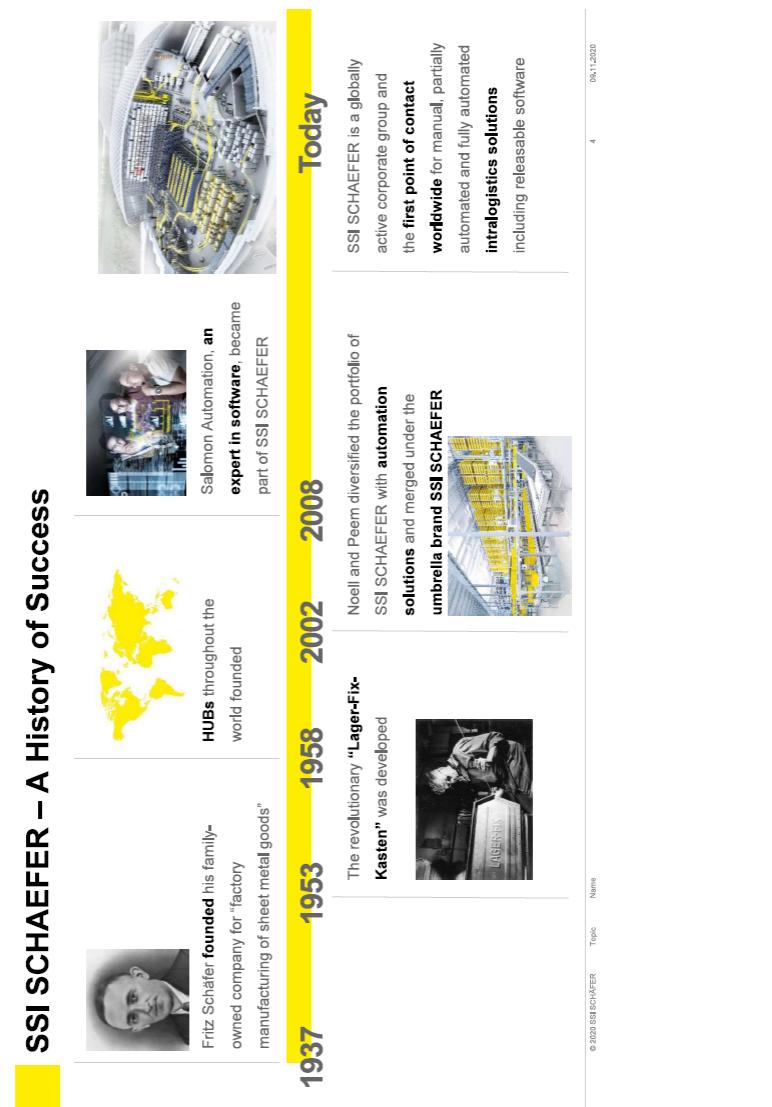
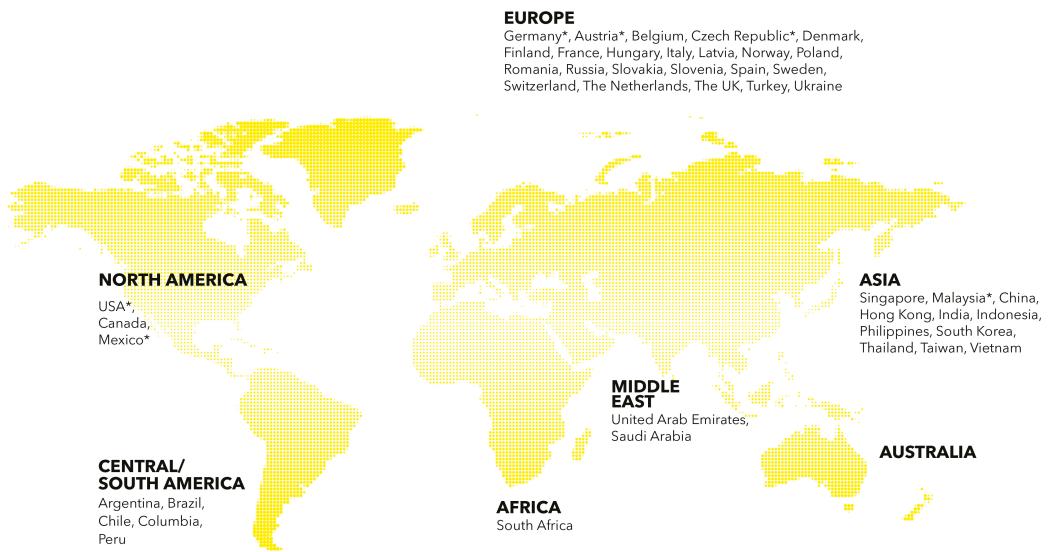


FIGURE B.0.1 – Frise historique de l'histoire de SSI SCHÄFER

Source de l'image : document interne

C**Carte des filiales SSI SCHÄFER****FIGURE C.0.1 – Carte des filiales SSI SCHÄFER dans le monde**

Source de l'image : document interne

D Liste complète des fonctionnalités de MORPHEUS WMS

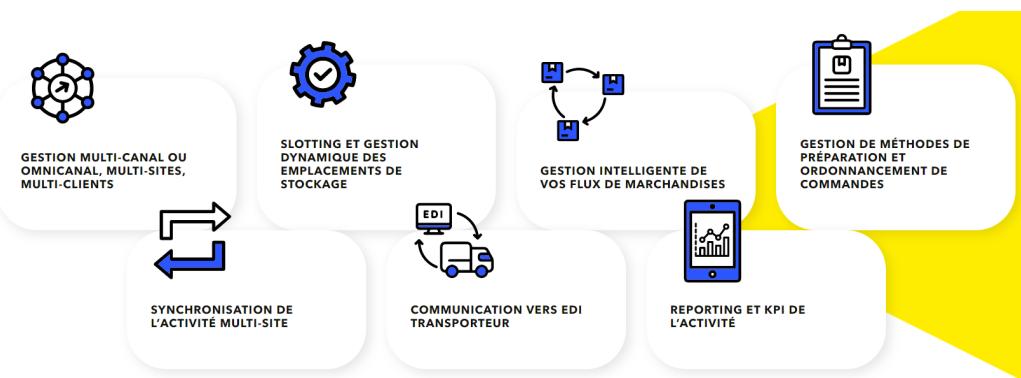


FIGURE D.0.1 – Fonctionnalités de Morpheus WMS

Source de l'image : document interne

E Liste complète des fonctionnalités de MORPHEUS WCS

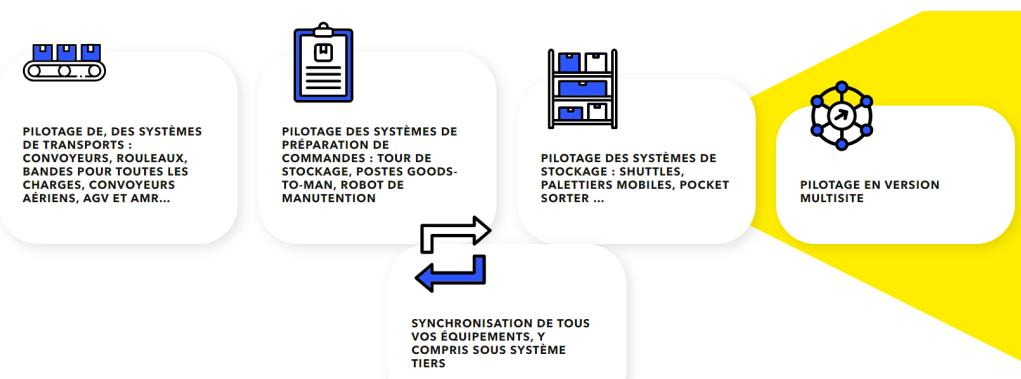


FIGURE E.0.1 – Fonctionnalités de Morpheus WCS

Source de l'image : document interne

F

Traduction de la partie Présentation du projet réalisé au cours du stage

Pendant la durée de mon stage au sein de l'entreprise SSI SCHÄFER, j'ai eu l'opportunité exceptionnelle de travailler sur un projet qui a non seulement renforcé mes compétences professionnelles, mais qui a également eu un impact significatif sur l'entreprise elle-même. Dans les sections suivantes, je vais vous présenter en détail le projet que j'ai entrepris, en mettant l'accent sur son contexte, ses objectifs, les défis rencontrés et les solutions mises en place. Au fil de ce mémoire, je vous invite à découvrir les étapes que j'ai parcouru au cours de ce stage, ainsi que les résultats concrets que j'ai obtenus. Ce projet représente non seulement une expérience enrichissante pour moi, mais il témoigne également de la capacité de collaboration et d'innovation au sein de l'équipe de développeurs de SSI SCHÄFER.

F.1

Projet global

À l'heure actuelle, Morpheus est une suite de logiciels regroupés au sein d'un client lourd développé en Delphi. Cela implique que si nous voulons utiliser Morpheus, nous devons l'installer sur la machine sur laquelle nous en avons besoin ce qui peut être parfois contraignant, surtout lorsque l'on a que des opérations de gestion à faire. Ainsi afin de simplifier l'accès à Morpheus, et ce sans avoir à installer de client lourd, il a été décidé de développer une application web qui permettra d'accéder à Morpheus depuis n'importe quel poste ayant un navigateur internet. Par ailleurs, en parallèle du projet de développer une version web du client Morpheus, un projet de refonte du client lourd a aussi été lancé. Le but du projet n'est donc pas de remplacer un client lourd par une application web mais d'offrir plusieurs possibilités aux utilisateurs en fonction des besoins de chacun. Un manager n'a pas besoin d'accéder aux mêmes fonctionnalités qu'un technicien et vice versa.

F.2 Projet du stage

Bien entendu il est impossible de développer une application web entière qui reproduirait le comportement de Morpheus en seulement six mois de stage. C'est pour cela que durant mon stage j'ai seulement réalisé une partie du projet de création de l'application web. Ainsi le projet du stage était donc de tester plusieurs outils et langages de programmation, déterminer lequel serait le mieux dans notre cas et développer une première version de l'application web.

F.3 Application existante

Nous avons vu dans l'analyse comparative que dans la majorité des cas les clients lourds n'ont pas de dépendance à une connectivité réseau, néanmoins cela n'est pas une vérité générale et c'est le cas pour Morpheus. En effet la logistique étant un milieu très précis, les opérations doivent être effectuées en temps réel, ainsi l'application a besoin en permanence d'accéder à la base de données. Il est important de comprendre que de nombreuses informations sont stockées dans la base de données et sont nécessaires au bon fonctionnement de Morpheus.

Voici quelques captures d'écran du logiciel Morpheus à l'heure actuelle.

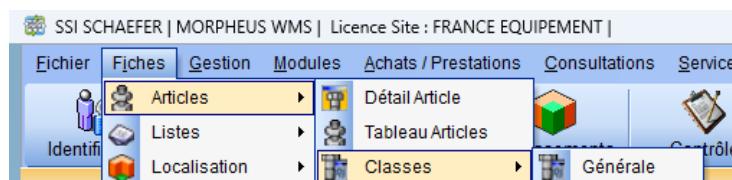


FIGURE F.3.1 – Partie du menu de l'application existante Morpheus

Source de l'image : capture d'écran

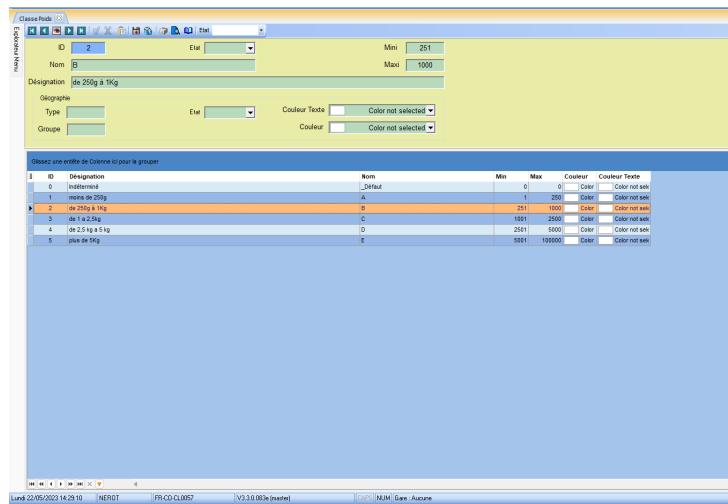


FIGURE F.3.2 – Exemple d'une fenêtre de l'application existante Morpheus

Source de l'image : capture d'écran

Bien que le client Morpheus regroupe de nombreuses fonctionnalités, il commence à devenir ancien. Ainsi le projet de développement de l'application web lui offrir aussi une seconde jeunesse avec des interfaces graphiques renouvelées. De plus les managers pourront effectuer leurs opérations de gestion directement depuis leur navigateur et n'auront plus besoin d'installer le client lourd.

G

Traduction de la partie Déroulement du projet

Le succès d'un projet repose non seulement sur son objectif et ses outils, mais également sur la manière dont il est planifié, exécuté et géré tout au long de son cycle de vie. Au cours de mon stage au sein de SSI SCHÄFER, j'ai eu l'opportunité de participer à un projet d'envergure qui a nécessité une planification minutieuse et une exécution rigoureuse. Dans cette section, je vais vous plonger dans le déroulement complet de ce projet, en décrivant les différentes étapes que j'ai suivies.

G.1

Objectifs et planification initiale

La clarté des objectifs et la planification méthodique sont les fondements de tout projet réussi. Dans cette section, nous allons explorer les objectifs que je me suis fixés au début de mon projet et la planification initiale qui a guidé mes actions.

Le processus de définition des objectifs est une étape fondamentale dans la réalisation de tout projet. Dans cette partie, nous allons détailler ces objectifs, expliquer pourquoi ils ont été choisis, et comment ils étaient alignés sur les enjeux et les besoins que nous avons identifiés précédemment.

La planification initiale est tout aussi essentielle. Elle nous permet de définir la trajectoire du projet, d'identifier les étapes clés, les ressources nécessaires, et de fixer des échéances现实. Dans cette section, je vais présenter le plan initial que j'avais mis en place.

En comprenant ces éléments, vous aurez une vue d'ensemble claire de la manière dont mon projet a été conçu et structuré pour atteindre des objectifs spécifiques, tout en tenant compte des contraintes et des réalités du terrain. Cette partie jettera les bases pour la suite du mémoire, où je détaillerai la mise en œuvre et les résultats par rapport à ces objectifs initiaux.

G.1.1 Objectifs détaillés

Les objectifs initiaux du projet étaient les suivants :

- Développer un client morpheus full HTML5 responsive pour une partie des fenêtres du client morpheus existant
- Expérimenter les outils de développement TMS Web
- Faire une maquette d'interface utilisateur
 - Fenêtre principale
 - Menu
 - Icônes
 - Style (comme style windows : couleurs)
 - X niveaux et sous niveaux paramétrables
 - ◆ Par rapport à une bdd afficher ou pas des points d'entrées du menu
 - Boutons de fonction
 - Style
 - Icônes
 - Taille des boutons (16, 32 or 48)
 - Rendre visible ou non les boutons
 - Mise en page des boutons en fonction du nombre : alignements...
 - Treeview
 - Style
 - Icônes (devant nom sans toucher au petit triangle)
 - Afficher une liste sur un niveau suivant une requête
 - Charger les arborescences dynamiques (si on clique sur site afficher les sites et pas avant)
 - ◆ Paramétrage visible ou non des arborescences
 - Gestion des styles
 - Gestion multi-fenêtres mdi (tabs)
 - Icônes
 - Fermable ou non par une croix
 - Gestion fenêtres modales

- Rendre visible la croix pour fermer la fenêtre ou non
- Mémorisation de la position
- Login
- Connexion à la base de données
- Rédiger un descriptif du mode de fonctionnement du client au niveau ergonomique
- Développer une première version avec les fenêtres de base
 - Fenêtre principale
 - Menu
 - Icônes
 - Style (comme style windows : couleurs)
 - X niveaux et sous niveaux paramétrables
 - ◆ Par rapport à une bdd afficher ou pas des points d'entrées du menu
 - Boutons de fonction
 - Style
 - Icônes
 - Taille des boutons (16, 32 or 48)
 - Rendre visible ou non les boutons
 - Mise en page des boutons en fonction du nombre : alignements...
 - Treeview
 - Style
 - Icônes (devant nom sans toucher au petit triangle)
 - Afficher une liste sur un niveau suivant une requête
 - Charger les arborescences dynamiques (si on clique sur site afficher les sites et pas avant)
 - ◆ Paramétrage visible ou non des arborescences
 - Gestion des styles
 - Gestion multi-fenêtres mdi (tabs)
 - Icônes
 - Fermable ou non par une croix
 - Gestion fenêtres modales

- Rendre visible la croix pour fermer la fenêtre ou non
- Mémorisation de la position
- Login
- Connexion à la base de données
- Gestion des droits d'accès
- Gestion paramétrage en XML des fenêtres
 - Type de fenêtre
 - Grid associé
 - Requête sql

G.1.2 Planification initiale

Nous en avons déjà parlé un peu plutôt voici la planification des tâches initiales :

- Installation et configuration des outils
- Expérimentation des outils
- Rédaction détaillée des besoins
- Développement du nouveau client HTML5
- Test et recettage du nouveau client
- Livraison

G.2 Étapes du projet

La réalisation d'un projet réussi nécessite un plan solide et une exécution méthodique. Dans cette section, nous allons explorer les étapes clés qui ont jalonné mon parcours, du début à la fin du projet. Cette chronologie nous permettra de comprendre la séquence logique de mon travail et les accomplissements qui ont marqué chaque phase.

La gestion des étapes d'un projet est essentielle pour assurer la cohérence, l'efficacité et la maîtrise des délais. Chaque étape représente un ensemble spécifique de tâches, d'objectifs et de livrables qui nous rapprochent de la réalisation de notre mission globale. Au cours de cette section, nous allons détailler ces étapes et expliquer comment elles étaient alignées sur les objectifs initiaux.

G.2.1

Étape n°1 : Choix du langage de programmation pour le frontend

L'application front-end de l'application web Morpheus est sans doute la plus importante du projet. En effet, elle représente la partie interface graphique et sera donc celle avec laquelle l'utilisateur va interagir. Il paraît donc logique que la première étape du projet soit de choisir le langage de programmation le plus adapté pour développer cette application.

G.2.1.1 Solutions possibles

Trois principales solutions ont été envisagées pour convenir aux objectifs de ce projet. Nous allons voir quelles sont ces solutions et pourquoi elles ont été envisagées.

G.2.1.1.1 Delphi et TMS Software

Delphi et TMS Software ont été la première solution envisagée dans ce projet. En effet le langage Delphi étant déjà utilisé au sein de l'entreprise SSI Schäfer, il a été logique pour eux de me proposer de réaliser le projet en utilisant cet outil. Néanmoins, je me suis très vite rendu compte que cela n'allait pas être l'outil le plus adapté pour notre projet d'application web.

Retrouvez une présentation du résultat obtenu avec Delphi en Annexe H Résultat obtenu avec Delphi.

Retrouvez un exemple de code produit en Delphi en Annexe I Code produit en Delphi.

G.2.1.1.2 JavaScript

JavaScript a été la deuxième option à laquelle j'ai pensé pour faire ce projet. En effet, voyant que Delphi ne serait pas la solution la plus adaptée, j'ai commencé à développer une application web en JavaScript vanilla. Les premiers résultats m'ont plutôt satisfait mais j'ai vite compris que si je devais redévelopper tous les composants dont j'avais besoin cela prendrait beaucoup trop de temps. Il me fallait donc trouver un framework JavaScript qui me donnerait accès aux composants dont j'avais besoin.

G.2.1.1.3 React et DevExtreme

Après réflexion et échange avec Monsieur NEROT, nous avons pensé que React pourrait être une bonne solution pour développer l'application

web Morpheus. Néanmoins il nous fallait aussi trouver un framework capable de nous fournir les composants graphiques donc nous avions besoin. C'est là que nous avons pensé à DevExtreme. En effet l'entreprise utilise déjà des frameworks DevExpress pour le développement du client lourd morpheus. Suite à un échange avec Monsieur GOURDON, il nous a indiqué qu'il existait un framework pour JavaScript. Nous nous sommes donc renseignés sur celui et avons décidé de tester ce que nous pourrions faire avec celui-ci.

Retrouvez un exemple de code produit en React et DevExtreme en Annexe J Code produit en React et DevExtreme⁵⁰.

G.2.1.2 Solution retenue

Ainsi à la lumière des tests réalisés dans les différents langages/frameworks ainsi que le comparatif technique entre TMS Web et DevExtreme⁵¹, nous avons choisi de retenir la solution React + DevExtreme. En effet nous avons établi que la solution React + DevExtreme était plus adapté à notre projet du fait de sa flexibilité accrue pour le multi-plateformes ainsi qu'une large gamme de widgets modernes.

G.2.1.2.1 Impact financier

L'avantage du framework DevExtreme, comparé à d'autres frameworks, est qu'on peut l'utiliser pour développer sans avoir à payer. Cela est un avantage car si on se rend compte que le framework n'est pas le bon, nous n'avons pas payer un framework pour rien. Mais cela reste très compliqué de chiffrer le gain d'argent que peut procurer l'utilisation d'un framework par rapport à un autre, car ce n'est pas la seule variante à prendre en compte. En effet l'expérience du développeur ou même la complexité du projet peuvent être des facteurs importants.

50. Vous pourrez avoir un aperçu des résultats produits en react et DevExtreme dans la section 9 Conclusion

51. Retrouvez le comparatif technique entre les solutions TMS Web et DevExtreme en Annexe K Comparatif technique entre les solutions TMS Web et DevExtreme

G.2.2**Étape n°2 : Développement de la première version du frontend**

Après avoir choisi le langage de programmation avec lequel nous voulions travailler, j'ai commencé à développer la première version du frontend de l'application web. Ainsi la première étape était de créer un menu pour pouvoir naviguer dans les différentes fenêtres du client web morpheus. Puis j'ai développé une fenêtre type avec une grille.

Voici le diagramme de classe que j'ai réalisé afin de stocker les informations d'une fenêtre type grille. Cette partie est liée à la création de l'API qui interagit avec la base de données. Néanmoins il était préférable de commencer à réfléchir aux informations que l'on veut stocker dans la base de données dès la création du frontend.

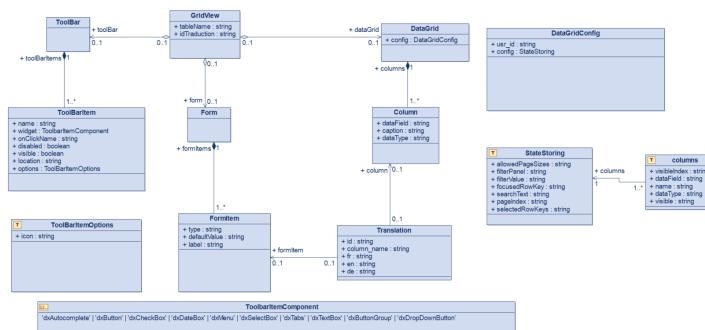


FIGURE G.2.1 – Diagramme de classe pour le stockage des informations d'une fenêtre type grille

Source de l'image : capture d'écran

Nous reviendrons sur les réalisations et les résultats du projet dans la section suivante.

G.2.3**Étape n°3 : Développement d'un backend pour intéragir avec la base de données**

Actuellement beaucoup d'informations utiles à Morpheus sont stockés dans une base de données, et pas seulement les données affichées aux utilisateurs mais aussi de nombreux paramètres de gestion des interfaces graphiques. Il existe aussi de nombreuses procédures stockées permettant d'effectuer certaines actions. À l'heure actuelle, nous ne souhaitons pas changer ce système, c'est ainsi qu'afin de pouvoir accéder aux informations contenues dans la base de données, nous avons décidé de mettre en place une API REST servant de solution backend à l'application web.

Comme évoqué précédemment, Monsieur NEROT devait développer une API REST afin de pouvoir communiquer avec la base de données. Néanmoins, étant très surchargé par son travail, il n'a pas eu le temps de réaliser cette API. Ainsi j'ai pris l'initiative de moi-même développer l'API afin de tester l'application web avec des données venant de la base de données. Cette API fonctionne sur le principe CRUD.

G.3**Suivi et contrôle**

Le suivi et le contrôle sont les piliers qui assurent la cohérence, la qualité et la maîtrise d'un projet tout au long de son déroulement. Dans cette section, nous allons plonger dans les détails de la manière dont nous avons suivi et contrôlé mon projet pour nous assurer qu'il restait aligné sur mes objectifs et respectait les normes de qualité.

Le suivi et le contrôle sont des éléments essentiels de la gestion de projet, car ils permettent de maintenir le cap, de résoudre les problèmes en temps réel, et de prendre des décisions éclairées pour assurer le succès du projet. Dans cette partie, nous allons détailler les méthodes, les outils et les pratiques que nous avons utilisés pour suivre la progression, évaluer les résultats et maintenir la qualité de mon travail.

G.3.1 Réunion hebdomadaire :"Point dev"

Nous en avons parlé plutôt dans la section 6 Synthèse des enjeux et de l'importance, le point dev est une réunion hebdomadaire où chaque développeur parle de ce qu'il a fait pendant la semaine. C'était donc l'occasion pour moi de présenter mon travail, ce qui assurait le suivi du projet et permettait de contrôler que le projet allait toujours dans la bonne direction.

G.3.2 Réunions ponctuelles de suivi

En tant que directeur général France de SSI SCHÄFER, Monsieur GOURDON a un emploi du temps très chargé. Néanmoins cela lui tenait à cœur de suivre l'évolution du projet de développement de l'application web. C'est donc pour cela que nous faisions parfois des réunions ponctuelles, le plus souvent avec Monsieur NEROT, qui me permettait de lui présenter mon travail et lui permettait, si besoin, de me donner ses indications sur la marche à suivre pour continuer le projet sur la bonne voie.

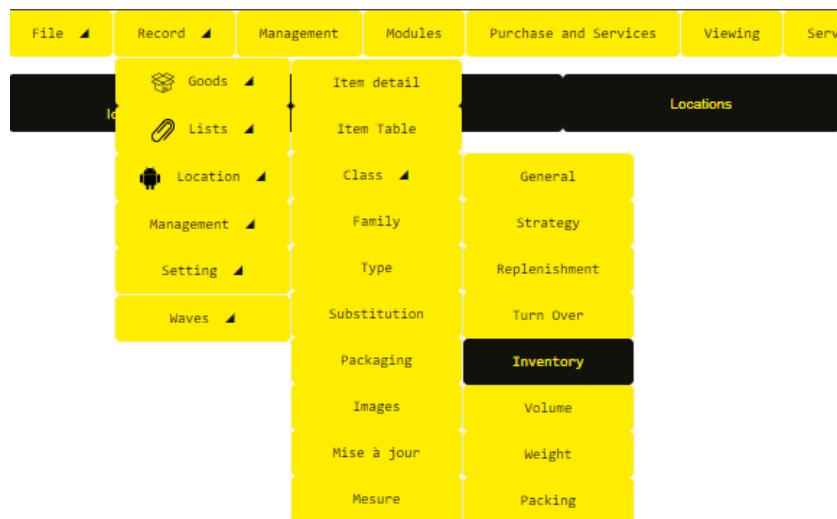
H**Résultat obtenu avec Delphi**

FIGURE H.0.1 – Menu développé en Delphi

Source de l'image : capture d'écran

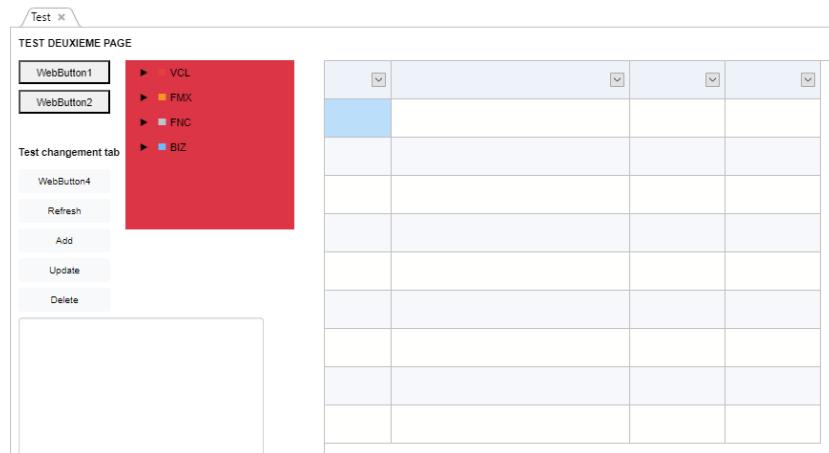


FIGURE H.0.2 – Fenêtre de test développée en Delphi

Source de l'image : capture d'écran

I Code produit en Delphi

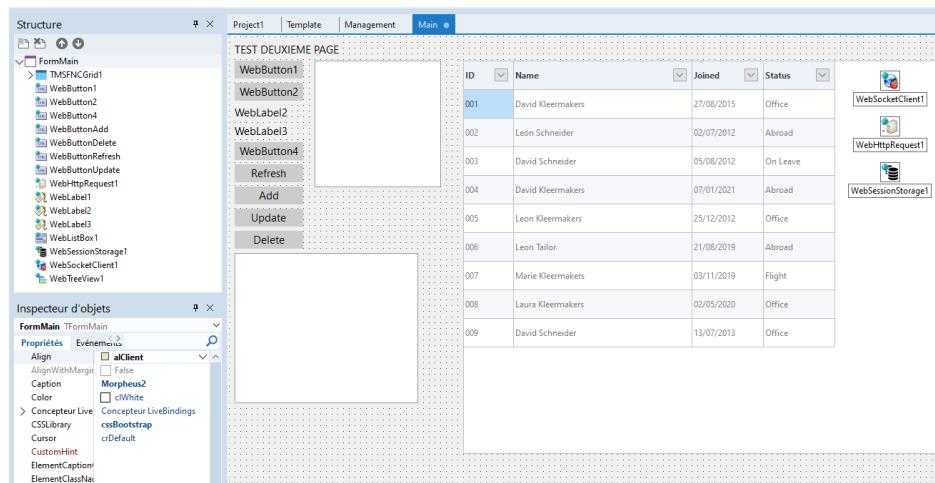


FIGURE I.0.1 – Interface de création de la fenêtre en Delphi

Source de l'image : capture d'écran

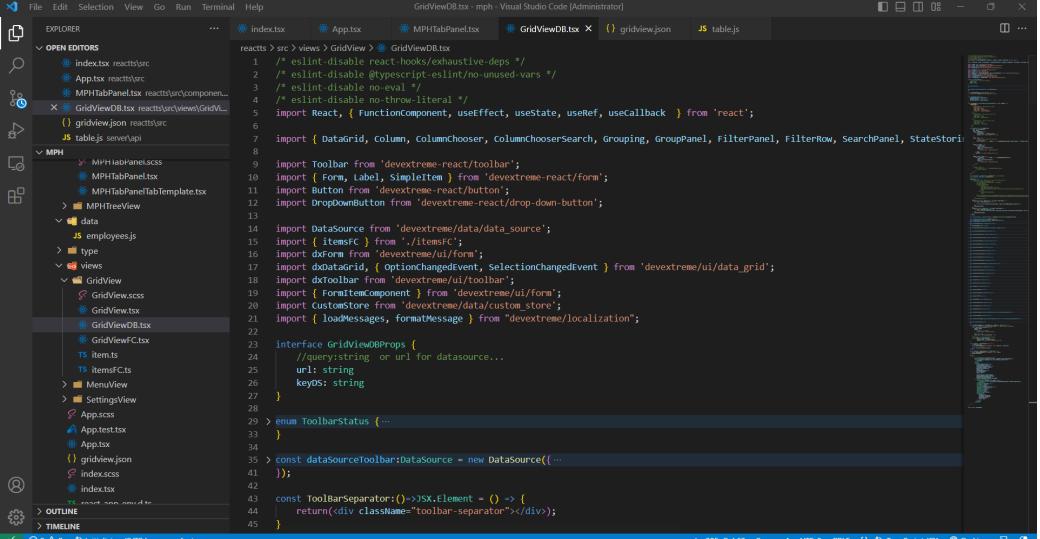
```
Project1 | Templates | Management | Main.x  
File Edit View Insert Tools Components Help  
FormMain.UidlFormCreate  
  
end;  
  
var  
  newform: TFormMain;  
  
begin  
  //newform := TFormMain.Create(Self);  
  //newform.Caption := 'Child form';  
  //newform.Popup := true;  
  //newform.Load();  
  newform := TFormMain.CreateNew();  
  newform.Popup := true;  
  newform.Border := fbDialog; // <-- defines to use a popup form with caption  
  newform.ShowModal(@AfterShowModal);  
end;  
  
procedure TFormMain.WebButton4Click(Sender: TObject);  
begin  
  WebLabel3.Caption := 'Test changement tab';  
end;  
  
procedure TFormMain.WebButtonRefreshClick(Sender: TObject);  
begin  
  displayData();  
end;  
  
procedure TFormMain.WebButtonAddClick(Sender: TObject);  
begin  
  WebHttpRequest1.URL := 'http://localhost:3000/test';  
  WebHttpRequest1.Command := httpPOST;  
  WebHttpRequest1.PostData := '["MOA_I_ID": 100, "MOA_C_NAME": "Indéfinie Test", "MOA_C_DESC": "Mode allocation indéterminé", "MOA_S_ETAT": 0,";  
  WebHttpRequest1.Execute();  
  displayData();  
end;  
  
procedure TFormMain.WebButtonUpdateClick(Sender: TObject);  
begin  
  WebHttpRequest1.URL := 'http://localhost:3000/test/100';  
  WebHttpRequest1.Command := httpPUT;
```

FIGURE I.0.2 – Code associé en Delphi

Source de l'image : capture d'écran

J

Code produit en React et DevExtreme



```

File Edit Selection View Go Run Terminal Help GridviewDB.tsx - mph - Visual Studio Code [Administrator]
EXPLORER OPEN EDITORS ...
index.tsx reacttsrc ...
App.tsx reacttsrc ...
MPHTabPanel.tsx reactsrc/components/MPH...
GridviewDB.tsx reactsrc/views/Gridv...
JS table.js server/api ...
MPH ...
MPHTabPanel.scss ...
MPHTabPanel.tsx ...
MPHTabPanelTemplate.tsx ...
MPHTreeView ...
data ...
JS employees.js ...
type ...
views ...
GridView ...
Gridview.scss ...
Gridview.tsx ...
GridviewDB.tsx ...
GridviewC.tsx ...
Item.ts ...
ItemsC.ts ...
MenuView ...
SettingsView ...
App.scss ...
App.test.tsx ...
App.tsx ...
gridview.json ...
index.scss ...
index.tsx ...
OUTLINE ...
TIMELINE ...
Initialization JS/TS language features
Ln 395, Col 63 Spaces: 4 UTF-8 CR LF () Q TypeScript JSX ⌂ Go Live ⌂

```

FIGURE J.0.1 – Exemple de code produit en react

Source de l'image : capture d'écran

K Comparatif technique entre les solutions TMS Web et DevExtreme⁵²

TABLE K.0.1: Comparatif technique entre les solutions
TMS Web et DevExtreme

Critères	TMS Web	DevExtreme
langage de programmation	TMS Web se base sur le Pascal Objet, un langage de programmation orienté objet. Il est particulièrement adapté aux développeurs familiers avec Delphi ou Free Pascal. L'utilisation de Pascal Objet dans TMS Web offre une courbe d'apprentissage plus facile pour les développeurs Delphi.	DevExtreme se base sur JavaScript, un langage incontournable pour le développement web. Cela signifie que les développeurs travaillent directement avec JavaScript et peuvent choisir d'utiliser des frameworks supplémentaires tels qu'Angular, React ou Vue.js pour développer l'application.
Composants et widgets	TMS Web propose des composants et des widgets basés sur les principes de la bibliothèque logicielle VCL (Visual Component Library) de Delphi. Cela permet de créer des interfaces riches avec des fonctionnalités avancées.	DevExtreme offre une large gamme de widgets prêts à l'emploi, allant des tableaux de bord interactifs aux graphiques, aux grilles, aux calendriers, etc. Ces widgets sont optimisés pour les performances et l'expérience utilisateur.

52. Référence utilisée dans la section K Comparatif technique entre les solutions TMS Web et DevExtreme : [62, 69].

Critères	TMS Web	DevExtreme
Intégration et compatibilité	Le TMS Web est étroitement intégré à l'environnement Delphi. Il est conçu pour être utilisé principalement avec Delphi, ce qui offre une expérience de développement cohérente pour les utilisateurs existants de Delphi. Cependant, cela peut limiter la portabilité vers d'autres plates-formes.	DevExtreme est indépendant de la plate-forme et peut être utilisé avec différents environnements de développement tels que Visual Studio Code, Angular, React, etc. Cela permet une plus grande flexibilité en termes de choix de la technologie et de la plate-forme.
Licence et coûts	Les coûts de TMS Web sont souvent associés aux licences Delphi de TMS Software. Les prix varient en fonction des modules nécessaires et des licences Delphi choisies.	DevExtreme propose différentes options de licence, y compris une version gratuite pour une utilisation non commerciale. Pour une utilisation commerciale ou l'accès à des fonctionnalités avancées, une licence payante peut être nécessaire.
Performance	Les applications TMS Web peuvent bénéficier des optimisations et des performances liées à Delphi. Cependant, étant donné qu'il s'appuie sur le Pascal Objet, il peut y avoir des limites en termes de performances JavaScript (le Pascal Object étant traduit en JavaScript) par rapport aux concurrents JavaScript natifs.	DevExtreme est optimisé pour les performances JavaScript et offre une expérience utilisateur réactive et fluide, en particulier dans les applications web modernes.

Critères	TMS Web	DevExtreme
Documentation et communauté	TMS Web dispose d'une documentation et d'une communauté pour soutenir les développeurs, bien que la communauté soit potentiellement plus restreinte que celle des grands frameworks JavaScript.	DevExtreme bénéficie du soutien de DevExpress, une entreprise bien établie, et propose une documentation complète et une communauté de développeurs JavaScript active et étendue.

L

Traduction de la partie Conclusion

En conclusion, ce mémoire a exploré en partie le processus complexe et significatif de la transition d'une application client lourd vers une application web. À travers notre étude, nous avons abordé divers aspects techniques, organisationnels et stratégiques de ce passage, mettant en évidence les défis et les opportunités inhérents à cette évolution.

Nous avons d'abord examiné le contexte de cette transition, en présentant l'entreprise SSI SCHÄFER et en définissant les raisons et les motivations qui l'ont poussée à entreprendre cette transformation majeure. Ensuite nous avons posé le cadre théorique du mémoire, ce qui a permis de définir les concepts clés qui ont été présents tout au long du document. Nous avons également effectué une analyse comparative entre les applications web et les applications client lourd afin de mieux comprendre les différences significatives entre les deux approches.

Nous avons ensuite présenté le projet et avons exploré les enjeux et importances qu'il avait pour l'entreprise. Nous sommes ensuite revenus sur la méthodologie et les outils utilisés au cours du projet. Nous avons ainsi mis en évidence l'importance de la planification et de la gestion de projet ainsi que la collaboration et la communication. Cela nous a aussi permis de présenter les outils technologiques utilisés.

Par la suite, nous avons détaillé les étapes clés du processus de migration, en insistant sur la planification stratégique, le choix des technologies, et le développement d'une première version de l'application web. Nous avons mis en évidence l'importance cruciale de la collaboration interdisciplinaire entre les parties prenantes pour assurer le contrôle et le suivi du projet afin de garantir le succès de la transition. Nous avons pu, à travers la cyberattaque qu'a subi l'entreprise, avoir une vision de la gestion d'une crise.

Enfin, nous avons examiné les résultats concrets de la transition, en mettant en évidence les améliorations fonctionnelles. Nous avons également identifié les enseignements clés tirés de cette expérience, notamment l'importance de la planification minutieuse, de la communication transparente et de l'adaptabilité face aux défis imprévus. J'ai aussi pu mettre en avant toutes les compétences que j'ai pu utiliser, acquérir et développer au cours de mon stage et qui me serviront de piliers pour mon

futur professionnel.

En somme, ce mémoire a démontré que la transition d'une application client lourd à une application web est un processus complexe mais essentiel pour rester compétitif dans un environnement numérique en constante évolution. Il a également souligné l'importance de l'alignement stratégique, de l'engagement des équipes, et de l'analyse approfondie des coûts et des avantages. Cette expérience nous a montré que, avec une planification soignée et une exécution efficace, une entreprise peut réussir à franchir ce cap avec succès et à s'adapter aux nouvelles exigences du marché.

En fin de compte, ce mémoire a été bien plus qu'une simple étude de cas technique, il a été une plongée profonde dans le monde en constante évolution de la technologie, de la gestion de projet, de la communication interdisciplinaire, et de l'adaptabilité aux changements. De plus ce mémoire ne se limite pas à documenter une transition technologique, il nous rappelle que dans le monde en constante évolution d'aujourd'hui, l'adaptabilité, la planification stratégique et la persévérance sont des atouts inestimables pour toute entreprise.

À l'instar de SSI SCHÄFER, nous sommes tous confrontés à la nécessité de nous adapter et d'innover pour prospérer. Les leçons tirées de cette expérience sont autant de boussoles qui peuvent nous guider dans notre propre voyage vers l'avenir numérique.

Alors que nous tournons la dernière page de ce mémoire, gardons à l'esprit que notre exploration ne fait que commencer. Le monde de la technologie continue d'évoluer, les défis émergent de manière inattendue, et les opportunités se présentent à ceux qui sont prêts à les saisir. La transition d'une application client lourd à une application web n'est qu'un chapitre dans l'histoire en constante évolution de l'innovation et de l'adaptation.

Mots clés

Mots clés

Liste des mots clés : delphi, développement web, logistique, Cholet, WMS, WCS, MORPHEUS, tms, react, nodejs, devexpress

Keywords

List of keywords : delphi, web development, logistic, Cholet, WMS, WCS, MORPHEUS, tms, react, nodejs, devexpress

Schlüsselwörter

Liste der Schlüsselwörter : delphi, web enwicklung, logistik, Cholet, WMS, WCS, MORPHEUS, tms, react, nodejs, devexpress

Résumés

Résumé

Ce mémoire explore la transition des applications client lourd vers des applications web, en se basant sur mon expérience personnelle lors de mon stage chez SSI SCHÄFER. Après avoir présenté l'entreprise SSI SCHÄFER, acteur mondial dans la logistique et entreprise d'accueil du stage, nous avons pu poser le cadre théorique qui couvre des concepts clés tels que les applications client lourd, les applications web, les frontend et backend, les frameworks, les API, et REST.

La partie scientifique analyse les avantages et inconvénients de ces deux types d'applications, en mettant en évidence que le choix dépend du contexte et des besoins de l'entreprise. Le projet principal avait pour objectif de développer une application web pour rendre le logiciel Morpheus accessible via un navigateur web.

La méthodologie et les outils ont joué un rôle crucial dans le projet, avec l'utilisation d'un modèle en V, de diagrammes de GANTT, de Confluence, et de nombreux outils technologiques. Le projet a suivi diverses étapes, avec un suivi et un contrôle réguliers. Une cyberattaque en avril 2023 a également été observée.

Cela m'a aussi permis de mettre en avant les compétences techniques, en gestion, en communication, en leadership, et en adaptabilité développées pendant le stage, qui sont précieuses pour ma future carrière. Le mémoire conclut en encourageant l'adaptabilité et l'innovation dans un monde technologique en constante évolution.

Summary

This dissertation explores the transition from thick client applications to web applications, based on my personal experience during my internship at SSI SCHAEFER. After presenting the company SSI SCHAEFER, a global player in logistics and host company of the internship, we were able to establish the theoretical framework which covers key concepts such

as heavy client applications, web applications, frontend and backend, frameworks, APIs, and REST.

The scientific part analyzes the advantages and disadvantages of these two types of applications, highlighting that the choice depends on the context and the needs of the company. The main project aimed to develop a web application to make the Morpheus software accessible via a web browser.

The methodology and tools played a crucial role in the project, with the use of a V-model, GANTT charts, Confluence, and numerous technological tools. The project followed various stages, with regular monitoring and monitoring. A cyberattack in April 2023 was also observed.

It also allowed me to highlight the technical, management, communication, leadership, and adaptability skills developed during the internship, which are valuable for my future career. The dissertation concludes by encouraging adaptability and innovation in an ever-changing technological world.

Zusammenfassung

Diese Dissertation untersucht den Übergang von Thick-Client-Anwendungen zu Webanwendungen, basierend auf meiner persönlichen Erfahrung während meines Praktikums bei SSI SCHÄFER. Nach der Vorstellung des Unternehmens SSI SCHÄFER, einem Global Player in der Logistik und Gastgeberunternehmen des Praktikums, konnten wir den theoretischen Rahmen festlegen, der Schlüsselkonzepte wie Heavy-Client-Anwendungen, Webanwendungen, Frontend und Backend, Frameworks, APIs und REST umfasst.

Der wissenschaftliche Teil analysiert die Vor- und Nachteile dieser beiden Arten von Anwendungen und betont, dass die Wahl vom Kontext und den Bedürfnissen des Unternehmens abhängt. Das Hauptprojekt zielte darauf ab, eine Webanwendung zu entwickeln, um die Morpheus-Software über einen Webbrower zugänglich zu machen.

Die Methodik und die Tools spielten mit der Verwendung eines V-Modells, GANTT-Diagrammen, Confluence sowie zahlreichen technologischen Tools eine wichtige Rolle im Projekt.

logischen Tools im Projekt eine entscheidende Rolle. Das Projekt verlief in verschiedenen Phasen mit regelmäßiger Überwachung und Kontrolle. Auch ein Cyberangriff im April 2023 wurde beobachtet.

Dadurch konnte ich auch die technischen, Management-, Kommunikations-, Führungs- und Anpassungsfähigkeiten hervorheben, die ich während des Praktikums entwickelt habe und die für meine zukünftige Karriere wertvoll sind. Die Dissertation schließt mit der Förderung von Anpassungsfähigkeit und Innovation in einer sich ständig verändernden technologischen Welt.