



Section : bachelier en informatique de gestion

Epreuve intégrée

WinDom Beaconing Project

Olivier CAMUS

Année académique 2021 – 2022

Remerciement

Je tiens à remercier toutes les personnes qui m'ont aidée lors de la rédaction de ce mémoire.

Je remercie également toute l'équipe pédagogique de l'IEPSCF Namur Cadets et les intervenants professionnels responsables de ma formation, pour avoir assuré la partie théorique de celle-ci.

Page | 1

Je tiens à témoigner toute ma reconnaissance aux personnes suivantes, pour leur aide dans la réalisation de ce mémoire :

La hiérarchie, l'équipe technique de l'AIR C2 Sp Sqn et le personnel du détachement eVA pour m'avoir supporté durant la rédaction de ce TFE.

Aux équipes ATC de Beauvechain Airport et infra Ops – électricité pour avoir eu la patience de répondre à mes innombrables questions.

Mesdames CAMUS Marina et BECK Adeline, pour avoir relus et corrigés mon mémoire. Leurs conseils de rédaction ont été très précieux.

La lieutenant d'aviation THIENPONT Laura, mission supervisor au CRC Beauvechain, pour son soutien et son aide ainsi que pour la documentation fournie.

Les étudiants du bac info de l'IEPSCF Namur Cadets, pour les conseils, l'ambiance et les réunions qualités.

Enfin, je remercie madame ANCEAU Dominique qui a toujours été là pour moi. Son soutien inconditionnel et ses encouragements ont été d'une grande aide.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Table des matières

Remerciement.....	1
Préface	3
1. Introduction	4
1.1. Les lampes d'un aéroport	5
2. L'application.....	9
2.1. Lamp.....	11
2.2. Lists	14
2.3. New	15
2.4. Update.....	15
2.5. Site	17
2.5.1. Runway.....	20
2.5.2. Taxiway	20
2.5.3. Exemple d'utilisation.....	21
3. Analyse et Conception	22
3.0.1. Logiciel et version – Back-End :	22
3.0.2. Logiciel et version – Front-End :	22
3.0.3. Versioning	23
3.1. Base de données	24
3.2. Back-End.....	25
3.3. Front-end	26
3.3.1. Exemple d'utilisation d'un composant	28
3.4. Évolution	29
4. Conclusion.....	30
5. Table des illustrations	32
6. Liste des tableaux.....	33
7. Liste des annexes	33
8. Bibliographie	34

Préface

Ce document est tiré de la version « démo » d'un projet beaucoup plus vaste qui englobe un grand nombre de service au sein de la défense belge.

Dans ce document, vous ne trouverez aucune information sur la couche physique. Ce qui inclut gestion de l'adressage IP, nombre de lampes, ...

Page | 3

Ce document est censuré par la défense qui ne souhaite pas que certaines informations sensibles soient rendues publique

1. Introduction

WINDOM Beaconing project est un outil de management du système d'éclairage permettant la circulation d'aéronef dans les aéroports de la défense Belge. En jargon, nous appelons cela le balisage – beaoning en anglais – des aéroports, c'est le terme que nous utiliserons par la suite.

L'ICAO (OACI en français) ou International Civil Aviation Organization (Organisation de l'Aviation Civile Internationale) – agence spécialisée de l'ONU qui élabore les politiques et les normes concernant la standardisation du transport aéronautique – a dédié un chapitre complet aux systèmes de balisage dans l'annexe 14 de sa convention.

Page | 4

J'ai puisé certaines de mes informations de l'édition 7 de juillet 2016, mais aussi dans le « Aerodrome Design Manual » Part 2 toujours édité par l'ICAO



Figure 1 : ICAO

L'atterrissage (landing en anglais) et le roulage (taxiing en anglais) sont des phases délicates pour un aéronef. Le pilote doit gérer un nombre énorme d'informations durant ses deux phases et toutes les aides sont les bienvenues.

Le balisage est une de ces aides, il permet au pilot d'aéronef de se situer facilement lors du taxiing sur l'aérodrome mais aussi de visualiser la piste lors de son approche et surtout se centrer sur son axe.



Figure 2 : approche de Chicago



Figure 3 : feu d'approche

L'atterrissage se fait sur une runway, elle est délimitée de part et d'autre par un nombre suivis dans certains cas des lettres « C » - « R » ou « L ». Il s'agit de l'axe de la piste, si vous ôtez le plus petit nombre du plus grand, vous obtiendrez toujours 18. Dans notre cas $22-4 = 18$. Ce qui implique que notre piste est accessible de venant avec un axe de 220° ou 40° .

Un taxiway est une voie de sortie/entrée de piste permettant au pilote de se rendre de son parking au début de la piste et inversement. Il est reconnaissable au panneau qui se trouve à des endroits clés.



Figure 4 : marquage d'une piste



Figure 5 : marquage d'un taxi

1.1. Les lampes d'un aéroport

Chaque voie de circulation est définie par une série de lampes qui se trouvent à des endroits bien précis. Certaines clignotent, d'autres pas. Elles ont aussi une couleur spécifique :

- Blanc/jaune
- Bleu
- Vert
- Rouge
- Orange

L'espace entre chaque type de lampe est fixé par l'annexe 14 de l'ICAO.

Elles ont deux alimentations indépendantes l'une de l'autre. Leur couleur est fixe, elle ne change pas selon l'angle de descente de l'aéronef.

Pour expliquer les différents rôles de chaque lampe, je vais me référer au schéma (cf. figure 6) que j'ai réalisé pour m'aider dans la réalisation de ce TFE.

La manche à vent vous indique l'axe de la piste en utilisation – 22 dans le cas présent –, la configuration est identique dans l'autre sens.

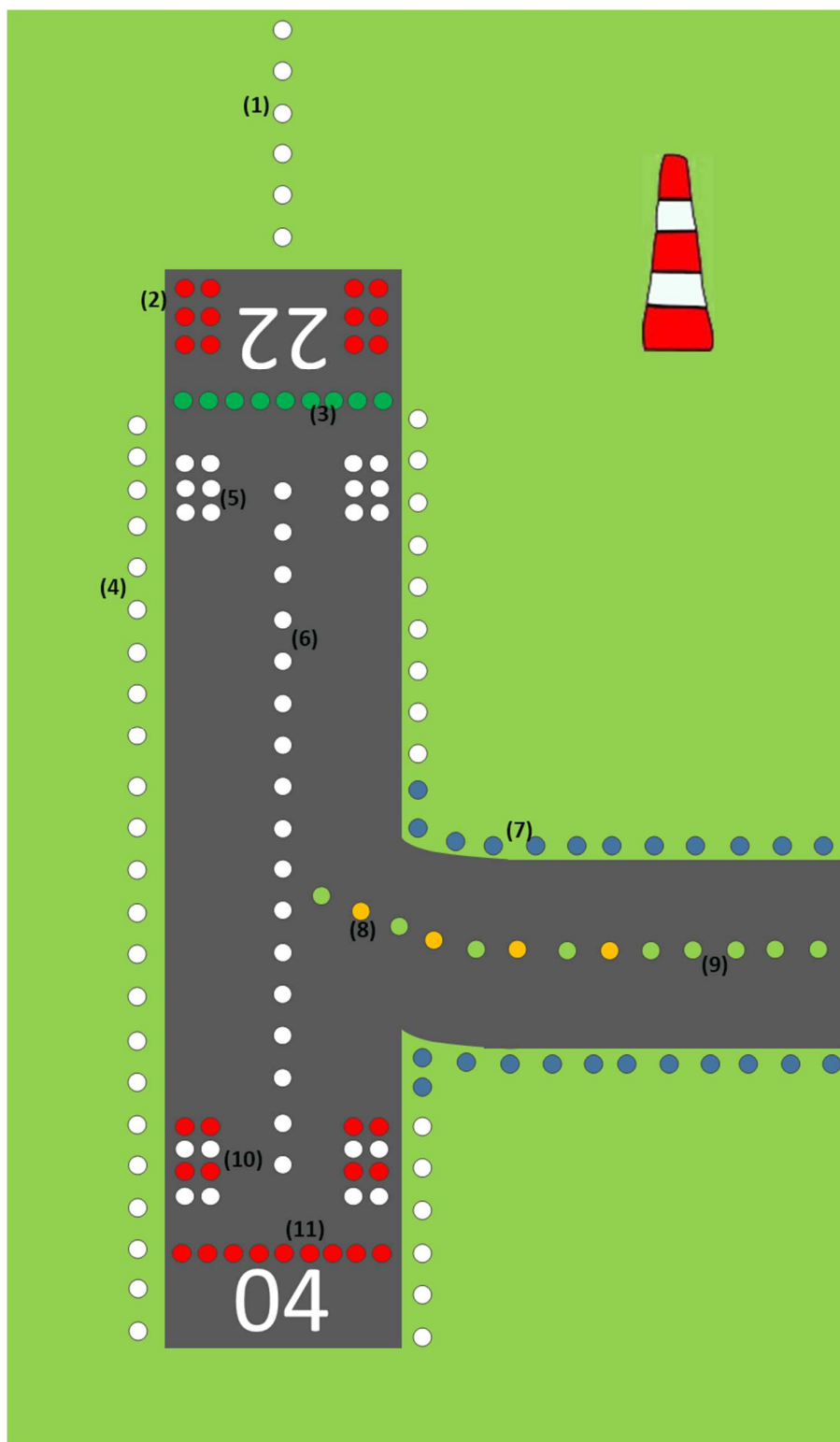


Figure 6 : schéma général de balisage EBBE

Numéro	Description	Données techniques
(1)	Axe de la piste, elle se trouve en amont de la piste dans les champs.	Couleur : blanche Flashing : true Hors sol – sur portique
(2)	Zone de piste – Tarmac, elle indique une zone de la piste ou aucune activité n'est permise.	Couleur : rouge Flashing : false Intégrée
(3)	Début de la piste, elle indique le début de la piste ou l'activité est permise.	Couleur : verte Flashing : false Intégrée
(4)	Bord de piste, elle délimite la piste sur la gauche et la droite sur toute sa longueur.	Couleur : blanche Flashing : false Hors-sol
(5)	Zone de touchdown, c'est la partie de la piste où l'avion va poser ses roues.	Couleur : blanche Flashing : false Intégrée
(6)	Axe de piste	Couleur : blanche Flashing : false Intégrée
(7)	Bord de taxi, sur la runway, annonce la proximité du taxi.	Couleur : bleue Flashing : false Hors sol
(8)	Axe central du taxi. Indique la sortie/entrée sur la runway, elle est toujours en quinconce avec une lampe verte	Couleur : orange Flashing : false Intégrée
(9)	Axe central du taxi. Se trouve sur toute la longueur du taxi. Elle est en quinconce avec une lampe orange quand elle indique l'entrée/sortie de la piste.	Couleur : verte Flashing : false Intégrée
(10)	La fin de piste est proche. Alternance avec les lampes blanches de zone de toucher	Couleur : rouge Flashing : false Intégrée
(11)	Fin de piste.	Couleur : rouge Flashing : false Intégrée

Tableau 1 : les différents types de lampes

Nous avons vu qu'il existait 3 types de lampes :

- Intégrées
- Hors-sol
- Hors-sol – sur portique



Figure 7 : lampe intégrée



Figure 8 : lampe hors-sol



Figure 9 : lampe hors-sol sur portique

Il s'agit d'une balise intégrée, elle comporte deux cotés lumineux.

Elle peut servir de lampe de fin de piste (rouge) ou bien de début de piste (vert).

Cela dépend de l'axe de piste en utilisation.

L'avantage est qu'une seule lampe est nécessaire pour deux utilisations. Cependant, elle réagisse comme deux lampes distinctes l'une de l'autre

Type : 2 – 3 – 5 – 6 – 8 – 9 – 10 – 11

Il s'agit d'une balise hors-sol qui se trouve sur les extrémités des différentes runway et taxiway.

Type : 4 – 7

Balise hors-sol sur portique donnant l'axe de piste.

La hauteur du portique varie en fonction de la distance de la piste.

Type : 1

2. L'application

L'application est accessible via un browser internet.

L'adresse pour s'y connecter dépend de votre serveur de production, cependant, le port sera toujours 8184.

Page | 9

La langue de travail choisie est l'anglais.

Passage obligé par la case login, sans authentification, vous n'avez accès qu'à la page de démarrage.

Pour ce faire, un lien « Login » se trouve à l'extrême droite de la barre de navigation.

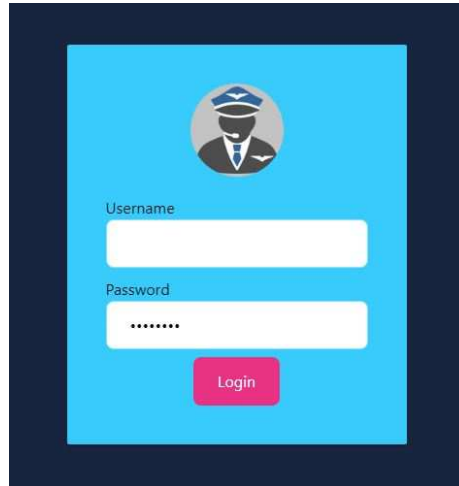


Figure 10 : login

Un rôle est alloué à chaque user. Il y a 3 rôles possibles :

- ADMIN
- CONTROL
- TECH

Le rôle CONTROL est donné à un contrôleur aérien.

Le rôle TECH est donné au technicien qui va gérer les différentes lampes, il s'agit le plus souvent d'électricien.

Le rôle ADMIN a le droit de tout faire, c'est le rôle que nous utiliserons.

La barre de navigation se compose maintenant de deux parties bien distinctes.

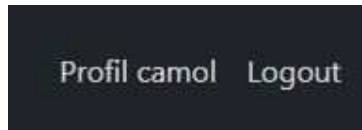


Figure 11 : barre de navigation - partie de droite

Vous pouvez vous déconnecter à l'aide du lien « Logout » et voir votre profil en cliquant sur « Profil <username> »

La partie gauche de la barre de navigation est plus importante.

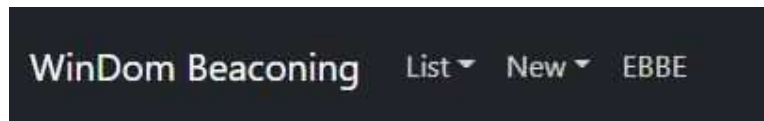


Figure 12 : barre de navigation - partie de gauche

« WinDom Beaconsing » vous ramène à la page de démarrage.

« List » va ouvrir un menu où vous pourrez sélectionner chaque élément qui constitue l'application.

« New » vous permettra de créer un nouvel élément.

« EBBE » ouvrira la page concernant l'aéroport de Beauvechain.

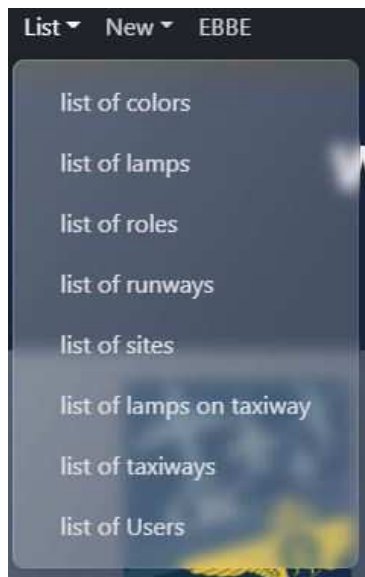


Figure 13 : menu "List"

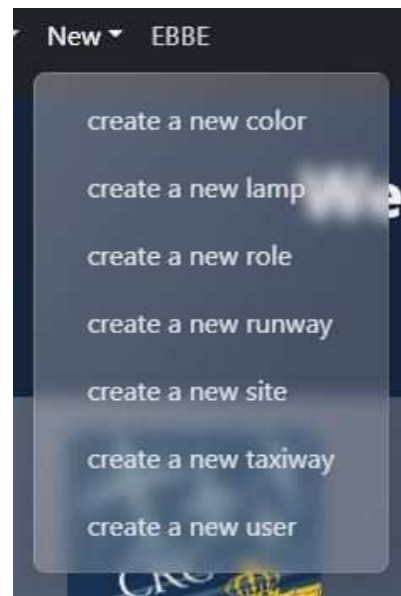


Figure 14 : menu "New"

2.1. Lamp

La pièce angulaire de notre système est la lampe. Il est donc logique que celle-ci se retrouve au cœur de cette application.

Le moyen d'accéder à une lampe spécifique est de passer soit par « list of lamps », « list of lamps on runway » ou « list of lamps on taxiway ».

Page | 11



Figure 15 : affichage d'une lampe

En plus des informations déjà connues, tel que type et couleur, une lampe est caractérisée par un état, une durée de vie, un nombre de cycle, une adresse IP.

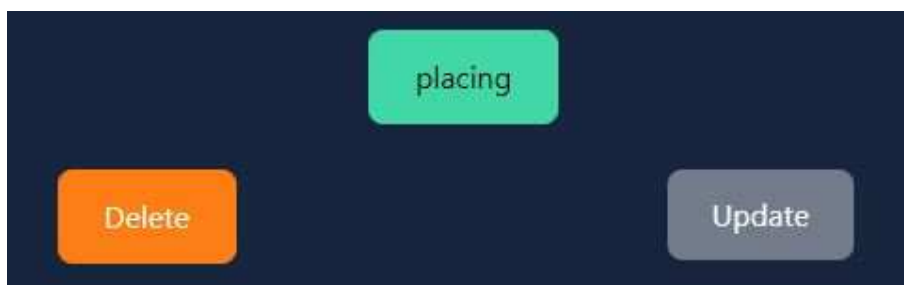


Figure 16 : affichage d'une lampe - placée

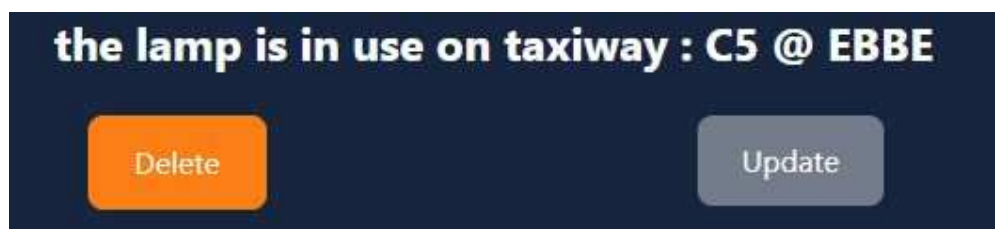


Figure 17 : affichage d'une lampe - à placer

Caractéristique	Choix possible	Description
Type	Fixed	Spécifie s'il s'agit d'une lampe clignotante ou non.
	Flashing	
State lamp	On	Spécifie si la lampe est allumée ou éteinte.
	Off	
Lifetime	<numbers>	Indique la durée en seconde ou la lampe a été dans l'état « On »
Cycles	<numbers>	Indique le nombre de cycle On/Off, particulièrement important pour une lampe clignotante.
Color	Red	Spécifie la couleur de la lampe.
	Blue	
	White	
	Green	
	Orange	
IP Address	<ip_address> <network_mask>	Spécifie l'adresse IP d'une lampe. Monitoring et gestion.

Tableau 2 : caractéristique d'une lampe

Vous avez la possibilité de faire une modification / update de votre lampe en cliquant sur le bouton « update »

update Lamp

type of lamp

☒ Fixed
☐ Flashing

color of lamp

-- please select a color --

IP Address

10.0.0.101

Update

Figure 18 : modification d'une lampe

Attention cependant, vous ne pouvez pas tout modifier.

En effet, la durée de vie et le nombre de cycle ne sont pas des valeurs éditables, ce sont des valeurs reçues de la lampe. La couleur émise ainsi que son type sont des caractéristiques intrinsèques de la lampe, et ne sont donc pas modifiable.

Cependant, ces valeurs ne sont pas reçues de la lampe. Il est donc possible de modifier ces deux paramètres dans le système. Cela ne changera en rien le comportement réel de la lampe. Page | 13

Il est possible de modifier l'adresse IP, mais cela peut fortement influencer le comportement du système, c'est pour cela que le formulaire est en couleur « orangered ».

Bien évidemment, si vous pouvez modifier, vous pouvez aussi supprimer. Cela ne la supprimera pas de la base de données, cela réinitialisera certains de ses paramètres tel que l'état et l'adresse IP.

Ça nous permet de garder une trace des lampes utilisées et de faire des statistiques d'utilisation.

Comme vous pouvez le voir sur les figures 16 et 17 – il s'agit de l'affichage qui se trouve juste après l'adresse IP – l'information affichée est différente.

En effet, nous en arrivons à une autre information importante concernant une lampe, il s'agit de son utilisation.

Oui, une lampe bleue sera toujours une lampe de taxi, mais l'aéroport de Beauvechain compte à lui seul 15 taxis.

Si je prends le cas d'un taxi, il est allumé selon l'axe de la piste choisie, de même que les lampes d'approche – (1), cf. figure 6 – elles ne sont allumées que dans le sens où l'atterrissage est autorisé. Nous avons donc regroupé les lampes selon leurs types d'utilisations et le leurs lieux d'utilisation.

Il faut donc assigner une lampe à un lieu d'utilisation et c'est ici que l'information est créée / reçue.

À la figure 17, nous voyons que la lampe est en utilisation dans le taxi C5 qui se situe à Beauvechain. Et à la figure 16, vu que la lampe n'est pas en utilisation, vous avez la possibilité de placer cette lampe.

À la figure 19, vous pouvez voir le formulaire de placement d'une lampe.

Vous devez sélectionner le site sur lequel placer la lampe et ensuite le taxi. Il s'agit d'une version démo – cf. préface – dans la version en production, vous devez en plus ajouter l'adresse IP de la lampe.

Figure 19 : placement d'une lampe

2.2. Lists

Les listes sont sensiblement toutes identiques, c'est pour cela que je ne les passerai pas toutes en revues.

list of site					
site	longitude	latitude	see	update	delete
EBBE	4° 46' 06" Est	50° 45' 31" Nord			
EBBL	5° 28' 16" Est	51° 10' 09" Nord			
EBFN	2° 39' 15" Est	51° 05' 29" Nord			
EBFS	4° 38' 48" Est	50° 14' 37" Nord			

Remove All

Figure 20 : exemple de liste - list of site

Pour chaque liste, les 1^{ères} colonnes sont différentes car elles représentent des caractéristiques de chaque ligne.

Dans le cas de la figure 20, il s'agit du nom des sites et de leur longitude et latitude.

Les 3 colonnes suivantes sont identiques pour chaque liste.

- See, vous permet de voir une fiche personnalisée sur la ligne sélectionnée.
- Update, sert à la mise à jour
- Delete vous permet de supprimer l'enregistrement. Cette action sera différente sur chaque type d'enregistrement.

Dans la liste des lampes sur une piste ou un taxi, la fonction « Delete » peut ne pas être disponible, il s'agit d'une fonction de statistique demandée par le client.

2.3. New

Tout comme pour les listes, l'ajout d'un ou l'autre enregistrement reste assez semblable.

Page | 15



Figure 21 : exemple de nouvel équipement - nouveau Taxi

Les différents champs qui composent un enregistrement sont ici présents et – comme pour notre exemple – lorsque notre enregistrement est lié à un autre – ici, un Taxi appartient à un Site – il s'agit d'une liste de choix.

2.4. Update

Nous en avons déjà parlé dans la partie lampe, le formulaire de mise à jour / update est presque identique à celui de la création.

Cependant, la mise à jour de l'utilisateur est un peu différente des autres.

En effet, un utilisateur a un nom d'utilisateur, un rôle et un mot de passe. Et c'est ce dernier qui demande une attention particulière.

Comme vous pouvez le voir à la figure 22, vous pouvez mettre à jour votre utilisateur via le bouton « update », vous mettrez à jour le rôle et le nom d'utilisateur.

Cependant pour modifier le mot de passe, vous devez passer par un formulaire spécial.

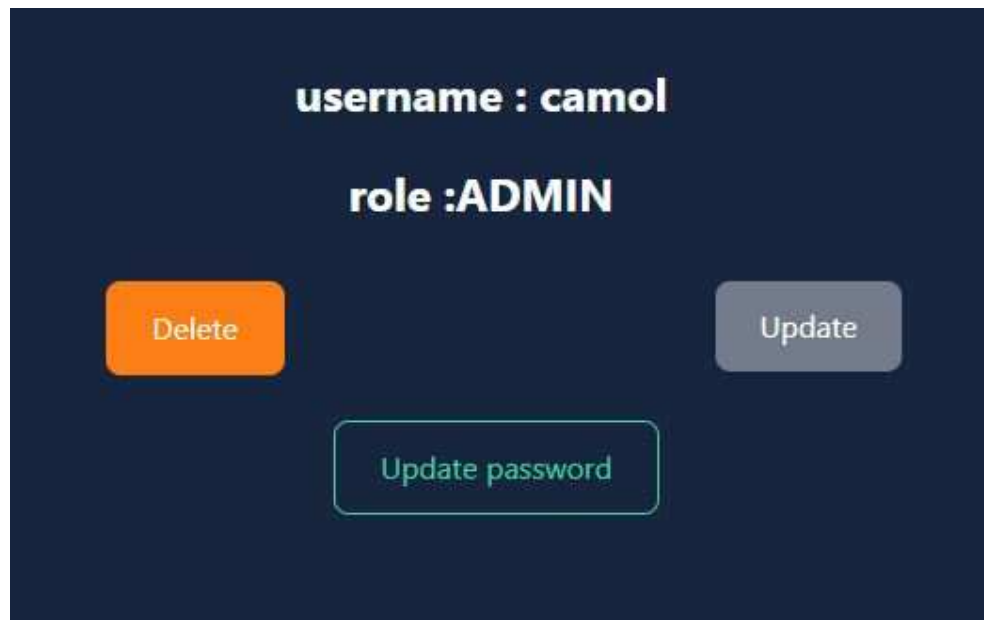


Figure 22 : affichage d'un utilisateur

A dark blue rectangular interface. At the top, there is an orange banner with the text 'ATTENTION PASSWORD MODIFICATION' in white. Below the banner, there are two input fields, each with the placeholder text 'password'. At the bottom center, there is an orange button with the text 'update password' in white.

Figure 23 : modification d'un mot de passe

Vous devez entrer le même mot de passe dans chaque champ, sinon, le système rejettera la requête de mise à jour, de plus, seuls les administrateurs ont le droit de modifier les mots de passe, concernant les règles du mot de passe liée au nombre de caractère, de la présence de caractères spéciaux et de mixité de caractère, il a été expressément demandé de ne pas la mettre en place.

2.5. Site

Un site est un ensemble de piste d'atterrissage et taxi. J'ai utilisé une « chart » pour travailler.

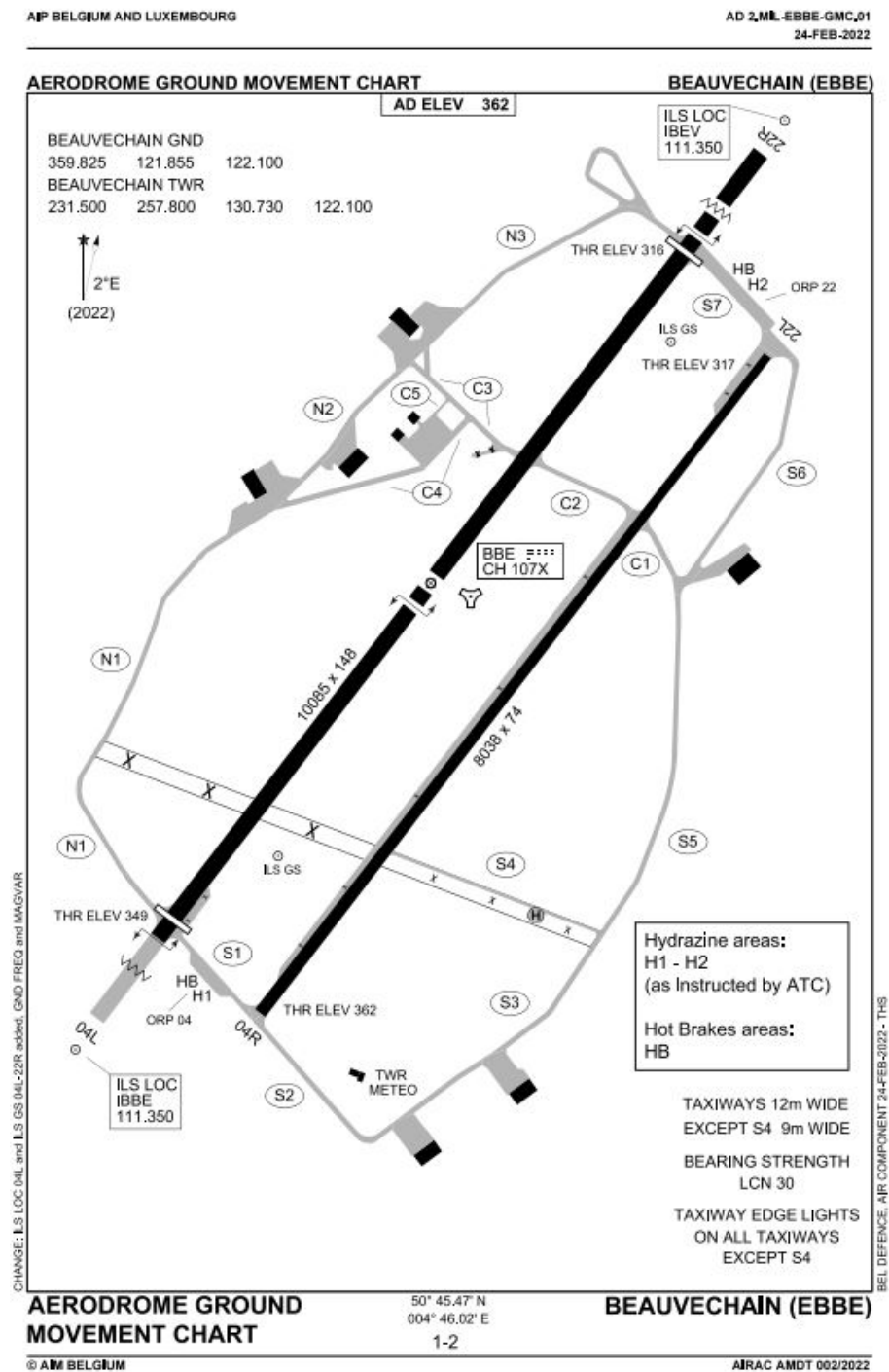


Figure 24 : Aerodrome Ground Movement Chart - EBBE

Il y a beaucoup d'informations importantes sur ce document utilisé principalement par les pilotes lors de la préparation de leur activité de vol.

Qu'est-ce que ce document nous apprend d'important et qui peut nous être utile ?

Runway : 4

22L – 22R – 04L – 04R

Page | 18

Taxiway : 15

S1 – S2 – S3 – S4 – S5 – S6 – S7

N1 – N2 – N3

C1 – C2 – C3 – C4 – C5

C'est l'axe de la piste principale – axe 22R-04L – qui fait la différence entre les S – pour South – et les N – pour North – et les C – pour Central. Les taxis centraux passent au milieu des pistes.

Mais elle nous apprend aussi – en bas à droite – que tous les taxiways sont munis de lampes excepté le taxiway S4.

On peut trouver les mêmes documents pour chaque aéroport belge.

À la figure 25 – voir page suivante – vous pouvez voir l'affichage de l'aéroport dans l'application.

Elle se compose de 3 parties bien distinctes :

- Une carte
- Une carte runways
- Une carte taxiways

Le design est sombre permettant une utilisation aussi bien de jour comme de nuit.

L'interface est pratique et simplifiée au maximum afin de garantir un confort d'utilisation.

La carte est tirée du document à la figure 24, retravaillée sommairement avec Photoshop® afin de supprimer le fond et les informations non-utiles

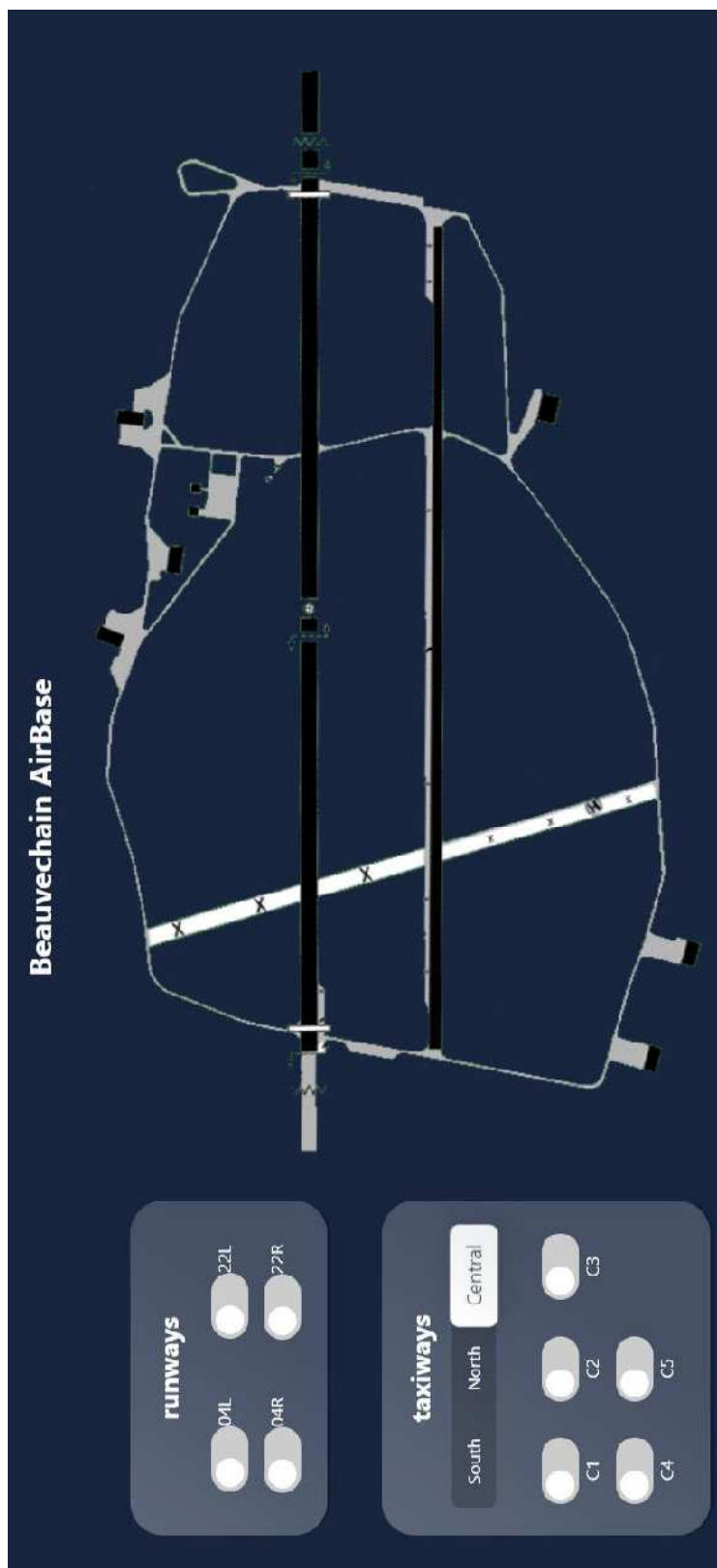


Figure 25 : Beauvechain AirBase

2.5.1. Runway

L'utilisation d'une piste d'atterrissage est liée à la direction du vent – l'aéronef décolle et atterri face au vent – il doit donc être impossible d'atterrir sur l'axe 04L et en même temps sur l'axe 22R.



Figure 26 : choix de la piste – désactivée



Figure 27 : choix de piste - activée

2.5.2. Taxiway

Chaque taxi est sélectionnable indépendamment, il n'y a pas de scénario préétabli. Ils sont regroupés par localisation.



Figure 28 : taxiway - South @ EBBE



Figure 29 : taxiway - North @ EBBE

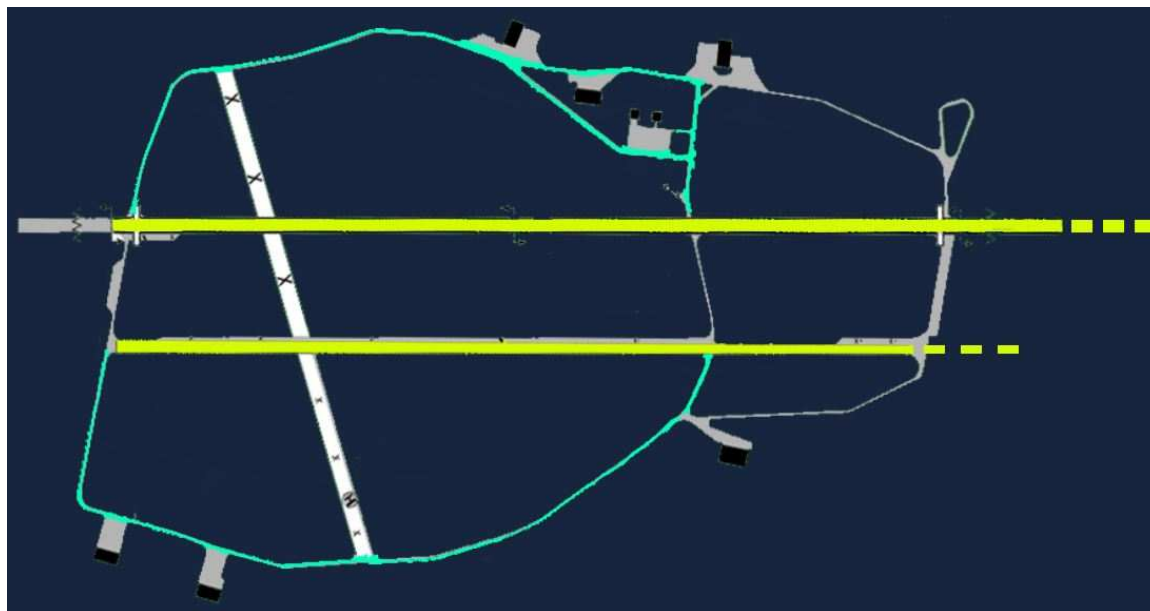


Figure 30 : taxiway - Central @ EBBE

2.5.3. Exemple d'utilisation

Les pistes en 22 sont en utilisations.

Nous activons les taxis S2, S3, S5 et C1 pour l'accès à la partie Sud et les taxi N1, N2, C3, C4 et C5 pour la partie Nord



Page | 21

Figure 31 : exemple d'utilisation - Carte

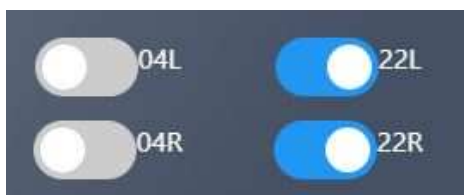


Figure 32 : exemple d'utilisation - runways

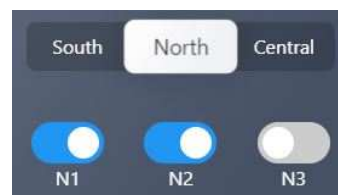


Figure 33 : exemple d'utilisation - Taxiway North



Figure 34 : exemple d'utilisation - Taxiway Central

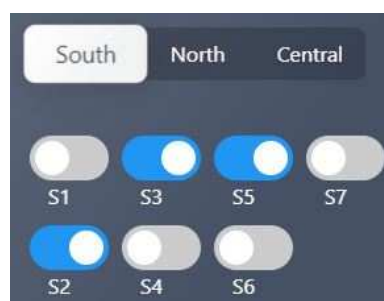


Figure 35 : exemple d'utilisation - Taxiway South

3. Analyse et Conception

J'ai utilisé Node pour le Back-End et React pour le Front-End. Et j'ai une base de données MySQL.

3.0.1. Logiciel et version – Back-End :

Node – version 16.14.2

Page | 22

Dépendances :

Axios : 0.22.0
bcrypt : 5.0.1
Body-Parser : 1.19.0
cors : 2.8.5
express : 4.17.1
jsonwebtoken : 8.5.1
mysql2 : 2.3.0
sequelize : 6.6.5

Dépendance développement :

nodemon : 2.0.15

3.0.2. Logiciel et version – Front-End :

React – Version 16.0.2

Dépendances :

@fortawesome/fontawesome-free : 5.15.4
@testing-library/jest-dom : 5.11.4
@testing-library/react : 11.1.0
@testing-library/user-event : 12.1.10
Axios : 0.23.0
Bootstrap : 5.1.3
Bootswatch : 5.1.3
react-async : 10.0.1
react-bootstrap : 2.3.0
react-client-session : 0.0.8
react-dom : 16.0.2
react-redux : 7.2.5
react-router-dom : 5.3.0
react-scripts : 4.0.3
react-validation : 3.0.7
redux : 4.1.1
redux-thunk : 2.3.0
validator : 13.6.0
web-vitals : 1.0.1

3.0.3. Versioning

Demo version : V.1.0.0

Production version : V.2.1.0

Historique des versions

Page | 23

Production :

V.1.0.0

Back-End full JAVA

abandonné à la suite de la demande du client.

V.2.0.0

Back-End en Node et Front-End en Angular

abandonné à la suite de la demande du client. Il voulait revenir à du JAVA.

V.2.1.0

Back-End en Node et Front-End en React

Version actuelle – abandonnée à la production fin mars 2022, elle devient version Demo.

V.3.0.0

prévue pour juin 2023 – application C++.

3.1. Base de données

J'utilise la suite WAMP version 3.2.4 avec PHP en version 7.4.9 et MySQL 8.0.22

Comme d'habitude, je prépare toujours ma DB en 1^{er} lieu. J'utilise un crayon et du papier pour poser par écrit mes 1^{ères} impressions et je le traduis dans un fichier Excel. (Cf. annexe 1)

Page | 24

users					
nom	type SQL	size/AI	nullable	type IDE	
id	int	AI	no	integer	
roleId	int		no	roles	
username	varchar	40	no	string	
password	varchar	50	no	string	

roles					
nom	type SQL	size/AI	nullable	type IDE	
id	int	AI	no	integer	
name	varchar	20	no	string	

lamps					
nom	type SQL	size/AI	nullable	type IDE	
id	int	AI	no	integer	
status	bool		no	boolean	
genre	bool		no	boolean	
colorId	int		no	color	
lifetime	int		no	integer	
modele	bool		no	integer	
IPADD	varchar	15	yes	string	
integrated	bool		yes	boolean	
cycle	int		yes	integer	

sites					
nom	type SQL	size/AI	nullable	type IDE	
id	int	AI	no	integer	
name	varchar	60	no	string	
longitude	varchar	60	no	string	
latitude	varchar	60	no	string	

runways					
nom	type SQL	size/AI	nullable	type IDE	
id	int	AI	no	AI	
siteId	int		no	0	
axe	varchar	5	no	5	

taxis					
nom	type SQL	size/AI	nullable	type IDE	
id	int	AI	no	integer	
siteId	int		no	sites	
name	varchar	20	no	string	

taxilamps					
nom	type SQL	size/AI	nullable	type IDE	
inUsage	int		no	boolean	
lampId	0		no	lamp	
taxiwayId	0		no	taxiway	

relation					
users.id	roles.id	OneToMany			
roles.id	users.id	ManyToOne			
sites.id	runways.id	OneToMany			
runways.id	sites.id	ManyToOne			
sites.id	taxiways.id	OneToMany			
taxiways.id	sites.id	ManyToOne			
lamps.id	colors.id	OneToMany			
colors.id	lamps.id	OneToMany			
runways.id	lamps.id	ManyToOne			
lamps.id	runways.id	ManyToOne			
taxiways.id	lamps.id	ManyToOne			
lamps.id	taxiways.id	ManyToOne			

colors					
nom	type SQL	size/AI	nullable	type IDE	
id	int	AI	no	integer	
name	varchar	20	no	string	

runlamps					
nom	type SQL	size/AI	nullable	type IDE	
inUsage	bool		no	boolean	
lampId	int		no	lamps	
runwayId	int		no	runways	

Figure 36 : aperçu de la base de données en Excel

La DB se compose de 9 tables et de 8 relations.

Il y a 2 tables intermédiaires qui sont créées par sequelize dû à la relation ManyToMany. Une des particularités de ces 2 tables, est qu'elles n'ont pas de champs ID mais elles ont 2 Primary Key.

Vous trouverez le schéma complet dans l'annexe 2.

J'ai utilisé les booléens le plus souvent possible parce qu'un ami m'a rapidement expliqué comment fonctionnait une IA, il faut réduire le champ des possibilités au maximum. La 1^{ère} version de la base de données ne contenait pas la table « Colors » et la table « Lamps » elle avait des champs au nom des différentes couleurs, j'ai modifié cela quand une personne avisée m'a expliqué que cela demanderait beaucoup d'aménagements si une couleur de lampe devait être ajoutée par la suite.

lamps	
- id : int - AI - PK	
- status : bool	
- genre : bool	
- colorId : int	
- lifetime : int	
- modele : bool	
- IPADD : varchar(15)	
- integrated : bool	
- blue : bool	
- white : bool	
- red : bool	
- green : bool	
- orange : bool	
- cycle : int	

Figure 37 : ancienne table "lamps"

3.2. Back-End

J'ai choisi Node JS parce qu'il est un acteur majeur dans le secteur, qu'il est très bien soutenu et que des formations sont facilement trouvables sur Internet.

J'ai travaillé avec le framework Express.js qui n'est autre que le framework standard pour le développement de serveur en NodeJS.

Page | 25

Durant ces 4 années de cours, j'ai toujours apprécié l'approche contrôleur – à la Symfony –, je suis donc parti sur le même principe, j'ai lu beaucoup sur la documentation Express, sur le site de BezKoder et j'ai donc la hiérarchie suivante :

```
Répertoire racine
├── config
│   ├── auth.config.js
│   └── db.config.js
├── controllers
│   └── <name_class>.controller.js
├── middleware
│   ├── authJwt.js
│   ├── index.js
│   └── verifySignUp.js
├── models
│   ├── <name_class>.model.js
│   └── index.js
├── node_modules
├── Routes
│   └── <name_class>.routes.js
├── package.json
└── server.js
```

Tableau 3 : hiérarchie Back-End

Le dossier « config » contient deux fichiers pour la gestion de la connexion DB et l'authentification.

Le dossier « controllers » contient les fichiers des actions à prendre en DB pour les fonctions CRUD et ce pour chaque table de la DB.

Le dossier « middleware » contient des fonctions concernant la sécurité de la connexion pour le server front-end.

Le dossier « Models » contient quant à lui, la composition des tables.

Le dossier « node_modules » contient les fichiers des différents outils utilisés.

Le dossier « routes » contient pour chaque table, une route pour joindre le contrôleur et ce pour chacune des fonctions du CRUD.

Le fichier « package.json » contient les informations générales du projet, le script de lancement et le nom des dépendances installées aussi bien en développement qu'en production.

Le fichier « server.js » initialise le serveur et les différentes constantes, tel que le port en utilisation, il initialise la connexion avec le serveur de base de données ainsi que le CORS.

3.3. Front-end

J'ai choisi React car j'avais déjà travaillé avec sur un projet en commun pour un autre cours.

J'y ai appliqué les principes appris au cours de Web et j'utilise Bootstrap auquel j'implémente Bootswatch qui me permet de définir un jeu de couleur. J'utilise Redux pour la gestion des états. Axios va gérer la communication avec le Back-end.

```
Répertoire racine
├── node_modules
├── public
├── src
│   ├── action
│   │   ├── <name_class>.action.js
│   │   └── types.js
│   ├── assets
│   ├── components
│   ├── reducers
│   │   ├── <name_class>.reducer.js
│   │   └── index.js
│   ├── services
│   │   ├── <name_class>.service.js
│   │   └── auth-header.js
│   ├── App.css
│   ├── App.js
│   ├── http-common.js
│   └── index.js
├── .env
└── package.json
```

Tableau 4 : hiérarchie Front-End

Le fichier « .env » contient les variables d'environnement, dans mon cas, le port sur lequel va se lancer le serveur.

Le fichier « package.json » contient les informations générales du projet, le script de lancement et le nom des dépendances installées aussi bien en développement qu'en production.

Le dossier « node_modules » contient les fichiers des différents outils utilisés.

Le dossier « public » contient les fichiers de rendu nécessaires aux pages Web.

Le dossier « src » contient tous les documents sources de l'application, c'est dans ce dossier que nous allons le plus souvent travailler.

Le fichier « App.css » contient le CSS général de notre application.

Le fichier « App.js » contient le cœur de notre application, c'est ici que je définis les éléments fixes à l'application.

Le fichier « Index.js » gère le rendu du DOM.

Le fichier « http-common.js » est un fichier axios qui crée et contient les informations pour la connexion vers le Back-End.

Dans le dossier « actions », il y a un fichier pour chaque table de la base de données et chaque fichier contient les actions à prendre pour exécuter le CRUD. Chaque fichier est lié à son penchant dans le dossier « services ».

Le fichier type.js centralise tous les types/actions pour toutes les tables.

Dans le dossier « services », les fichiers sont en liens avec ceux dans le dossier « actions » et contiennent les routes pour chaque actions CRUD pour chaque table.

Page | 27

Le fichier « auth-header.js » gère les droits d'accès à chaque route.

Le dossier « reducers » est une compilation pour redux et le CRUD. Il y a un fichier par table et un fichier qui combine le tout.

Le dossier components est particulier, c'est dans ce dossier que nous allons retrouver tous les composants de l'application.

React travaille avec des composants réutilisables et personnalisables.

J'ai découpé mon application en plusieurs petits composants regroupés par action du CRUD et par tables. Par exemple pour le switch qui gère l'affichage d'une piste/taxi, j'ai deux fichiers :

- toggleSwitch.component.js
- toggleSwitch.css

Le fichier toggleSwitch.component.js contient le code de rendu du toggleSwitch (Cf. figure 38) et le fichier toggleSwitch.css reprend le CSS spécifique au toggleSwitch.

En ayant un fichier spécifique au CSS pour le toggleSwitch, cela me permet de rendre mon compte plus lisible. Tout le CSS ne se retrouve pas dans un seul et gros fichier ou trouver une information peut-être par moment pénible.

3.3.1. Exemple d'utilisation d'un composant

À la figure 38, se trouve le code du toggleSwitch.

J'ai une classe toggleSwitch qui implémente Component. Et qui importe e.a. le fichier css qui lui est propre.

À partir de la ligne 10, jusqu'à la ligne 13, ce sont des « props » que je vais pouvoir passer à mon composant pour qu'il réagisse comme je le veux.

Page | 28

J'ai 4 props :

- id : qui rend unique l'instance.
- disabled : qui permet de désactiver l'instance (ndlr : utile pour S4)
- onclick : qui permet d'assigner une action lors d'un clic sur le switch
- onchange : qui permet d'assigner une action lors d'un changement survenu.

```
1 import React, { Component } from "react";
2 import './toggleSwitch.css';
3
4 export default class toggleSwitch extends Component{
5   render()
6   {
7     return (
8       <label className="switch">
9         <input type="checkbox"
10           id={this.props.id}
11           disabled={this.props.disabled}
12           onChange={this.props.onChange}
13           onClick={this.props.onClick}/>
14         <span className="slider round"> </span>
15       </label>
16     )
17   }
18 }
```

Figure 38 : exemple de composant -initialisation du toggleSwitch

Je l'utilise dans l'affichage des aérodromes.

```
260 <div className="col-3 text-center">
261   <ToggleSwitch id="S3" onChange={() => setSouthThree(!southThree)}/> <span>S3</span>
262   <br/>
263   <ToggleSwitch id="S4" disabled={true}/><span>S4</span>
264 </div>
```

Figure 39 : exemple de composant - utilisation du toggleSwitch

3.4. Évolution

Comme je l'ai expliqué dans la partie versioning, le client a remanié le projet à plusieurs reprises.

Ce projet passera en production en 2023, mais après une refonte totale du concept.

WINDOM Beaconing Project sera écrit – probablement – en C++ et la DB sera construite plus que
probablement avec Oracle.

Page | 29

Un mode jour/nuit sera implémenté et le tactile fera son apparition.

Une liaison avec le système météorologique est à prévoir afin d'afficher la direction du vent.

4. Conclusion

Le maître mot de ce TFE sera : déception.

Ces 4 années d'études débouchent sur un projet colossal, un projet rempli de défis techniques importants qui ne pourront pas être abordés dans ce TFE.

Page | 30

La partie physique est un défi, parvenir à faire communiquer une lampe avec un système d'information complexe, faire réagir en même temps plusieurs lampes et récupérer les données en provenance de ces lampes.

Toute une histoire que je ne peux vous conter, à la suite de trop nombreuses censures professionnelles. La défense n'est pas la seule en cause, la firme qui nous a fourni la lampe ne veut pas non plus que son savoir se retrouve dans un mémoire de fin d'étude.

Revenons-en un peu à la genèse de ce TFE.

Les officiers m'ont donné le choix entre trois projets :

- Un système de ticketing
- Un système de balisage d'aérodrome
- Un VCS – Voice Communication System

J'ai choisi le système de ticketing, car il y avait un réel intérêt pour le service, un premier jet a été réalisé très rapidement, mais aussi vite, le projet est mis en stand-by, à la suite d'une mésentente entre officiers pour la gestion des équipements.

Je ne voulais pas d'un VCS, car, cela dépasse de loin mes compétences.

Le WINDOM Beaconing Project a donc vu le jour.

Sur le point de vue Analyse, cela a été un réel plaisir, j'ai organisé plusieurs réunions avec les contrôleurs afin de comprendre le balisage, les différentes possibilités d'action, j'ai ensuite pris contact avec les électriciens en charge de la partie technique. Ils m'ont appris comment le système fonctionnait et en discutant avec eux des nouvelles technologies, on est parti sur l'idée des lampes contrôlées par l'informatique.

Ensuite, il a fallu convaincre ma hiérarchie directe – ce fut la partie la plus facile – et lancer la procédure d'achat d'un exemplaire de la lampe. Enfin, nous sommes allés à Bruxelles pour défendre le projet auprès des officiers dirigeants. Ils ont accepté sous réserve de donner un budget, je vous avoue que durant cette dernière phase, je n'étais qu'un des experts techniques présents autour de la table et que je n'ai pas eu grand-chose à faire.

À l'issue de cette réunion, j'ai reçu un serveur de virtualisation vierge – donc à installer –, un routeur CISCO 3945, un switch CISCO Catalyst 3850 et un bureau au calme.

J'ai pensé DB, mais aussi plan d'adressage IP, quel OS installer sur le serveur, comment déployer en production, configuration des équipements réseaux.

Puis j'ai proposé JAVA comme langage, quand j'avais bien avancé, la hiérarchie a dit stop à JAVA et on est partis vers javascript puis retour à JAVA et enfin, mi-mars de cette année, on est passé sur du plus lourd, du C++. Mais par manque de temps et un agenda professionnel chargé, je suis revenu sur le javascript qui était la version la plus élaborée du projet et j'avais une petite base en React. J'ai donc développé cette application de démonstration qui servira de base au projet C++.

C'est donc avec un gout d'amertume que je termine ce TFE, celui de ne pas avoir pu vous le partager, vous partager mes acquis durant ce périple.

Enfin, ce projet est toujours sur la bonne voie, en octobre avec un collègue bachelier nous entamerons une formation C++ et nous nous lancerons dans la programmation du WINDOM Beaconing Project 2.0.

Page | 31

Enfin, terminons sur une note positive, cette version démo sera notre fil conducteur, elle permet de montrer au client vers quoi nous nous dirigeons, les retours de la phase démo auprès des contrôleurs sont encourageants, l'interface répond à leur demande tout en n'étant pas trop compliquée.

La hiérarchie voit que malgré la charge de travail toujours plus grandes, nous parvenons à avancer.

Et malgré tout, pour moi, c'est une grande fierté de pouvoir me dire : « oui, j'y suis arrivé »

5. Table des illustrations

Figure 1 : ICAO	4
Figure 2 : approche de Chicago.....	4
Figure 3 : feu d'approche	4
Figure 4 : marquage d'une piste	5
Figure 5 : marquage d'un taxi	5
Figure 6 : schéma général de balisage EBBE	6
Figure 7 : lampe intégrée	8
Figure 8 : lampe hors-sol.....	8
Figure 9 : lampe hors-sol sur portique.....	8
Figure 10 : login.....	9
Figure 11 : barre de navigation - partie de droite.....	10
Figure 12 : barre de navigation - partie de gauche.....	10
Figure 13 : menu "List"	10
Figure 14 : menu "New"	10
Figure 15 : affichage d'une lampe.....	11
Figure 16 : affichage d'une lampe - placée	11
Figure 17 : affichage d'une lampe - à placer	11
Figure 18 : modification d'une lampe	12
Figure 19 : placement d'une lampe	14
Figure 20 : exemple de liste - list of site	14
Figure 21 : exemple de nouvel équipement - nouveau Taxi.....	15
Figure 22 : affichage d'un utilisateur	16
Figure 23 : modification d'un mot de passe	16
Figure 24 : Aerodrome Ground Movement Chart - EBBE	17
Figure 25 : Beauvechain AirBase.....	19
Figure 26 : choix de la piste – désactivée	20
Figure 27 : choix de piste - activée.....	20
Figure 28 : taxiway - South @ EBBE	20
Figure 29 : taxiway - North @ EBBE.....	20
Figure 30 : taxiway - Central @ EBBE.....	20
Figure 31 : exemple d'utilisation - Carte	21
Figure 32 : exemple d'utilisation - runways	21
Figure 33 : exemple d'utilisation - Taxiway North	21
Figure 34 : exemple d'utilisation - Taxiway Central.....	21
Figure 35 : exemple d'utilisation - Taxiway South	21
Figure 36 : aperçu de la base de données en Excel	24
Figure 37 : ancienne table "lamps"	24
Figure 38 : exemple de composant -initialisation du toggleSwitch	28
Figure 39 : exemple de composant - utilisation du toggleSwitch.....	28

6. Liste des tableaux

Tableau 1 : les différents types de lampes	7
Tableau 2 : caractéristique d'une lampe.....	12
Tableau 3 : hiérarchie Back-End.....	25
Tableau 4 : hiérarchie Front-End	26

7. Liste des annexes

Annexe 1 : Analyse de la base de données en Excel

Annexe 2 : Base de données – WINDOM Beaconsing

8. Bibliographie

- Abramov, D. (2022, 5 02). *React Redux*. Retrieved from React Redux: <https://react-redux.js.org/>
- BezKoder. (2022, 05 12). *Node.js Rest APIs example with Express, Sequelize & MySQL*. Retrieved from BezKoder: bezkoder.com/node-js-express-sequelize-mysql/
- BootStrap Team. (2022, 05 17). *React-Bootstrap · React-Bootstrap Documentation*. Retrieved from React-Bootstrap: <https://react-bootstrap.github.io/>
- Debret, J. (2022, 05 17). *Exemples de remerciements d'un mémoire*. Retrieved from scribbr: <https://www.scribbr.fr/memoire/remerciements-exemples/>
- ICAO. (2005). *Aerodrome Design Manual* (4 ed., Vol. Part 2).
- ICAO. (2016). *Annex 14 to the Convention on International Civil Aviation* (7 ed., Vol. 1).
- Mozilla Foundation. (2022, 05 17). *JavaScript | MDN*. Retrieved from Mozilla: <https://developer.mozilla.org/fr/docs/Web/JavaScript>
- Open JS Foundation. (2022, 05 12). Retrieved from Node JS: nodejs.org
- Open JS Foundation. (2022, 05 12). *Open JS Foundation*. Retrieved from openjsf.org
- Refsnes Data. (2022, 05 17). *JavaScript Tutorial*. Retrieved from W3 schools: <https://www.w3schools.com/js/default.asp>
- Sandy, L. (2022, 04 11). *Fullstack React 17, GraphQL, Apollo, Google Sign & Paypal | Udemy*. Retrieved from udemy: <https://www.udemy.com/course/fullstack-react-17-graphql-apollo-google-sign-paypal/>
- Simon, D. (2022, 04 11). *NodeJS: Développer Votre Première API Rest avec MySQL (2022) | Udemy*. Retrieved from udemy: <https://www.udemy.com/course/nodejs-tutorial-francais-express-api-rest-mysql/>
- Skeyes. (2022, mai 09). *Z:\Charting\charts\MIL_Charts\EBBE\AD 2.MIL-EBBE-GMC01_p02_Frame.dgn*. Retrieved from Skeyes: https://ops.skeyes.be/html/belgocontrol_static/eaip/eAIP_Main/graphics/eAIP/EB_AD_2_EBBE_GMC_01_en_v09.pdf
- Stack Exchange Inc. (2022, 05 17). *stackoverflow*. Retrieved from stackoverflow: <https://stackoverflow.com/>
- Thomas, P. (2022, 05 17). *Bootswatch: Free themes for Bootstrap*. Retrieved from Bootswatch: <https://bootswatch.com/quartz/>
- Wikipedia. (2022, 05 12). *Express.js*. Retrieved from Wikipedia: <https://fr.wikipedia.org/wiki/Express.js>
- Wikipedia. (2022, 05 12). *Node.js*. Retrieved from Wikipedia: <https://fr.wikipedia.org/wiki/Node.js>