

Rapport de stage

Réécriture et optimisation de progiciels de python dans le
langage Golang

Quentin Agneray

16 août 2018

Table des matières

I Remerciements	3
II Introduction	4
III Présentation	5
III.1 Alloa Voyages	5
III.1.a L'agence de voyage	5
III.1.b L'agence et le domaine de l'informatique	5
III.2 Technologies utilisées	6
III.2.a Golang et GopherJS	6
III.2.b gRPC et Protocol Buffers	7
III.2.c PostgreSQL	8
III.2.d Bootstrap	8
IV Missions	9
IV.1 Présentation et Organisation	9
IV.2 Hyooper	10
IV.3 Site Commercial	12
IV.3.a Client	12
IV.3.b Serveur	14
V Bilan	18
VI Conclusion	20
VII Abstact	21

Table des figures

IV.1 logiciel hyooper module Nos Zones (Destinations)	11
IV.2 Page web responsive utilisation de la fonction inspecter et de Toggle device toolbar	13
IV.3 Page contact avec affichage d'une carte avec la localisation de l'agence Alloa	14
IV.4 Affichage des hébergements correspondant à la destination Amérique du Nord	17

I | Remerciements

Je tiens dans un premier temps à remercier tous les membres de l'entreprise que j'ai côtoyé pendant ce stage, en particulier monsieur Michel Dewintre gérant de l'agence qui a eu l'amabilité de m'accueillir pendant trois mois.

Je remercie également les membres du jury qui ont accepté de présider lors de ma soutenance.

Je remercie mon responsable de formation ainsi que mon tuteur Monsieur Eric Timmerman et l'ensemble des professeurs et intervenants qui m'ont suivi pendant mon parcours d'étudiant en DUT Informatique.

II | Introduction

Le rapport que je vais vous exposer a été fait dans le cadre de ma deuxième année d'études en tant qu'étudiant en Diplômes Universitaires de Technologie (DUT) ayant pour spécialité l'informatique.

L'objectif de ce stage est de mettre à profit les compétences que j'ai acquises dans le cadre de mes études, d'apprendre de nouvelles choses dans un environnement professionnel.

Le stage que j'ai effectué a eu une durée d'environ trois mois du 14 mai 2018 au 3 août 2018. Ce stage a été effectué au sein de l'agence de voyage Alloa située au 72 Rue Léon Gambetta à Lille. J'ai choisi cette entreprise car je voulais apprendre un nouveau langage, le Golang(Go) ainsi que d'autres technologies comme GopherJS, gRPC, Protocol Buffers, PostgreSQL, Bootstraps.

Ma mission lors de ce stage a été de réécrire et d'optimiser les progiciels de l'agence écrits en python dans le langage Golang.

Tout d'abord je vais vous faire une présentation de l'entreprise Alloa, de ses acteurs ainsi que l'environnement de travail (logiciels utilisés, répartition des tâches,...). Ensuite je vous parlerai de l'avancement de mes différentes missions, de la partie développement ainsi que de l'analyse de code. Pour finir, je vous exposerai les différentes compétences que j'ai acquises qui pourront être utiles pour mon futur emploi.

III | Présentation de l'entreprise

III.1 Alloa Voyages

III.1.a L'agence de voyage

L'entreprise Alloa Voyages est située à Lille à l'adresse 72, rue Léon Gambetta. L'agence a été créée en 1981 et a comme nom Apolinea Club, c'est une association. En 1983 l'agence adopte le statut juridique de la SARL et se renomme en Amphitrite. La devise de l'agence est ne me demandé pas la lune je ne fait que le monde.

L'entreprise n'a pas de clientèle principale et travail avec les entreprises, les particuliers ainsi que les familles. L'agence ne possède pas de concurrent direct dans son domaine d'activité et propose des voyages sur mesure à ces clients.

III.1.b L'agence et le domaine de l'informatique

L'agence Alloa est familiarisé depuis longtemps avec l'informatique. L'entreprise a créé le premier programme d'agence de voyage pour la comptabilité d'abord sur minitel puis sur internet. La licence choisie fut propriétaire puis open source.

Elle possède ces propre serveurs et utilise comme plateforme d'orchestration de containers Kubernetes. Les serveurs sont regroupés et forme une grappe de serveur ou cluster¹ Le système d'exploitation utilisé sur les serveurs est Debian car le système d'exploitation est fiable et accessible.

1. un cluster consiste à associer divers noeuds afin de les faire travailler ensemble en partageant des ressources communes

III.2 Technologies utilisées

III.2.a Golang et GopherJS

Golang est un langage de programmation open source compilé proche du C ou du Pascal développé en interne dans les laboratoires de Google à partir de 2007. Ce langage à été développé par Robert Griesemer qui a contribué au compilateur JavaScript V8 par Rob Pike qui a travaillé sur l'UTF-8 et Kenneth Lane Thompson qui a travaillé sur l'écriture du système Unix.

Quelques différences entre le Golang et le langage Java que j'ai appris lors de mon cursus. Le Golang permet de retourner plusieurs valeurs contrairement au Java. Autre différence en Golang nous utilisons des structures et non des classes. Pour finir en golang pour rendre une variable visible à l'extérieur du package nous mettons la première lettre de la variable en capital et n'utilisons pas des mots clefs comme public, private et protected.

GopherJS est un transpileur² développé par Richard Musiol à partir d'aout 2013 qui permet de convertir du code Golang en code JavaScript exécutable sur tous les navigateurs.

2. traduit un code source dans un langage de programmation et le compile dans un autre langage

III.2. TECHNOLOGIES UTILISÉES CHAPITRE III. PRÉSENTATION

Voici un exemple de code écrit en GopherJS et son équivalent écrit en javascript inclut dans une page html à l'aide de balise `<script>...</script>`

```
1 package main
2
3 import "fmt"
4 import "github.com/gopherjs/gopherjs/js"
5
6 func main() {
7     js.Global.Call("alert", "hello world")
8 }
```

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <script>
5 function maFonction() {
6     alert("hello world");
7 }
8 </script>
9 <body>
10 </html>
```

III.2.b gRPC et Protocol Buffers

gRPC est un framework RPC (Remote Procedure Call) développé par Google. RPC est un protocole IP qui permet de développer des applications client-serveur. RPC se situe dans la couche 5 dite Session du modèle OSI (Open Systems Interconnection).

Protocol Buffers est un système de sérialisation de données³. Il est utilisé avec le langage Java et le Golang qui ne supporte pas très bien le JSON qui est un autre système de sérialisation de données.

3. processus de conversion d'un objet en un flux d'octets pour stocker l'objet ou le transmettre

Voici un exemple de fichier protobuf :

```
1 package web;
2
3
4 import "github.com/johanbrandhorst/protobuf/proto/gopherjs.
   proto"
5
6 message Destination {
7     string lib = 1;
8     string uuid = 2;
9     repeated Pays pays = 3;
10 }
```

Ce fichier est un fichier protobuf avec l'extension `.proto`. Ce fichier contient la définition d'un message ici `Destination`. Chaque message commence avec la directive `message`. Ce message contient une liste de champs avec leur type comme `string`, `int32`, `bool` ... et un identifiant. Le mot clé `repeated` correspond à une liste de type. Ici il s'agit d'un tableau de `pays`. A chaque message nous associons un numéro unique qui sert à identifier le champ.

III.2.c PostgreSQL

PostgreSQL est un système de bases de données relationnelles objet développé à l'université de Berkeley en Californie en 1986 sous le nom de `POSTGRES` puis `PostgreSQL` à partir de 1996. Pour administrer notre base de donnée nous avons utilisé `pgAdmin` qui est un outil pour PostgreSQL. Ce logiciel m'a permis de tester mes requêtes sur la base de donnée et de voir le contenu des tables. Pour finir j'ai travaillé sur `pgModeler` pour visualiser le MCD (Modèle Conceptuel de Données), ce qui m'a permis de voir les tables et les différents liens, relations entre celles-ci.

III.2.d Bootstrap

Bootstrap est un framework web côté client *frontend*. Bootstrap inclut de l'HTML du CSS ainsi que du JavaScript. L'avantage de Bootstrap est qu'il est facile à utiliser et est compatible avec de nombreux navigateurs. L'autre avantage majeur est qu'il s'adapte au téléphone, tablettes ainsi que pour les moniteurs d'ordinateur de bureau. Ceci s'appelle le RWD (Responsive Web Design) et ce concept vient du web designer Ethan Marcotte.

IV | Missions

IV.1 Présentation et Organisation

Mes missions lors de ce stage ont été de réécrire et d'optimiser les progiciels de l'agence de Python en Golang. Le choix du Golang pour la réécriture des progiciels n'a pas été la seule option envisagée. En premier lieu des tests ont été faits avec Angular et Vue.js.

Angular est un framework côté client qui permet de construire des applications mobiles et web. Angular utilise Typescript, qui est un langage qui permet le développement d'applications écrites en JavaScript. Vue.js quand à lui est un framework JavaScript pour créer des interfaces utilisateurs.

Nous n'avons pas retenue cette solution car trop lourde. Nous avons préféré utiliser le Golang qui répond plus à notre besoin. En effet le Golang prend en charge des machines multi-cœur et permet d'être déployer facilement avec Kubernetes que nous utilisons. Golang peut être utilisé pour construire un CMS(Content Management System) qui permet de gérer des sites web.

Pour mener à bien les différents projets nous avons utilisé Git qui est un logiciel de gestion de version et Gitlab qui nous a permis de pouvoir gérer nos dépôts Git. Sur Gitlab nous avons utilisé un outil qui s'appelle une issue qui permet de suivre l'évolution du projet avec la liste des tâches à effectuer et leur avancement avec un système de To Do (A Faire) , Doing (Faire) et Done (Terminé). Système similaire à celui que j'ai utilisé lors des projets Agile fait lors de mon cursus.

Comme logiciel de communication nous nous sommes servi de pidgin qui est un logiciel de messagerie instantanée. Ce logiciel nous a permis de nous envoyer de la documentation technique et du code.

Au niveau de l'organisation, nous nous sommes organisé de la manière suivante. J'ai tous d'abord travaillé sur la réécriture du logiciel Hyooper côté Back-End (Serveur) avec l'aide d'un autre stagiaire Valerian. Celui-ci m'a expliqué le projet et son organisation hiérarchique (packages, comment compiler le projet, les branches sur Git, ...). Par la suite je me suis occupé du côté Front-End (HTML, CSS, JavaScript, GopherJS) pour prendre la place d'un stagiaire Quentin qui a commencé le site commercial de l'agence. Pour finir j'ai travaillé sur le site commercial de l'agence côté Back-End avec Quentin et Front-End avec Maxime, stagiaire que j'ai du former.

Pour ce qui est du suivi du projet nous discutons entre nous et avec monsieur Dewintre aussi souvent que possible pour voir les différentes difficultés rencontrées et des améliorations éventuelle à apporter.

Au niveau de mon organisation personnel pour mon travail, j'ai beaucoup travaillé à l'aide d'un papier et d'un crayon. Cela m'a permis de prendre des notes et de coucher par écrit mes idées sur comment résoudre certaine difficultés et consigner les bugs que j'ai rencontré pour pouvoir les remonter.

IV.2 Hyooper

Hyooper est un progiciel de gestion globale de l'agence de voyage. Ce logiciel a pour but de faciliter l'ajout, la suppression et la modification d'informations sur des composantes d'un voyage par exemple comme une destination, un hébergement, un circuit touristique ... Mais pas seulement il permet également de pouvoir gérer tous ce qui concerne le marketing, la direction , les sous traitant ...

Le travail que j'ai effectué a été de créer des "modules" sur la composante d'un voyage comme les villes , les régions, les pays ... Mon but a été de créer des fonctions pour récupérer des informations sur une ville par exemple en faisant une requête de type SELECT sur la base de donnée. J'ai également créer des structures pour définir par exemple qu'est ce qu'une destination et qu'une destination ce compose d'un Uuid(Universally Unique IDentifier) , d'une référence, d'un libellé français et anglais ... Pour obtenir les informations concernant le contenu des modules (tables), j'ai utilisé PgModeler pour voir les données à récupérer ainsi que leur types(chaine de caractères , entier, booléen , date ,...).

Nos Zones

Imprimer Ajouter Rechercher

Search:

Référence	Libellé FR	Libellé EN	Thematique	Publique	Statut	Date de Modification
02	Afrique -	Africa	false	true	Validée	2010-11-04 11:53:00
03	Amérique dNord	Amérique du Nord	false	true	Validée	2010-11-04 11:53:00
30	Polynésie - Hawaï - Îles marquises	Polynesia - Hawaii - Marquises Islands	false	false	Validée	2010-11-04 11:53:00
06	Proche et Moyen Orient	Proche et Moyen Orient	false	true	Validée	2010-11-04 11:53:00
01	Europe	Europa	false	true	Validée	2010-11-04 11:53:00
29	Iles Mascareignes	Madagascar & Mascareigne Islands	true	true	Validée	2010-11-04 11:53:00
a05	Sports d'hiver	Winter sports	true	true	Validée	2010-11-04 11:53:00
ref	test 2 modif	test 2	false	false	Validée	2018-07-03 14:58:28
13	France, Bénélux, Irlande et Royaume uni	dqsqsd	false	false	Validée	2010-11-04 11:53:00
11	Europe du Nord de l'est et Scandinavie	Poles	false	true	Validée	2010-11-04 11:53:00
220	Afrique de l'Ouest et Centrale	Mascarene & Indian Ocean	false	true	Validée	2010-11-04 11:53:00

Showing 1 to 30 of 30 entries

Supprimer Modifier

Référence: 01

Fr Lib: Europe

Ref à la une:

Gb Lib: Europa

Ref banniere: gif_europe.gif

A revoir le: 1970-01-01

☒ Publique ☐ Thematique

☒ Continent

Descriptions Descriptions Pays/Régions Notes Descriptions Pays/Régions

FIGURE IV.1 – logiciel hyooper module Nos Zones (Destinations)

Pour ce qui est du côté client, je me suis assuré de la mise en page ainsi que de la récupération des données et de leur ajout dans le fichier HTML en utilisant GopherJS. Pour la mise en page du site nous avons utiliser Bootstrap et en particulier SB Admin 2 qui est un template¹. La création des zones de texte pour l’affichage de la description d’un module c’est fait grâce à l’API(Application Programming Interface) Quill.

1. un template est un modèle pour l’architecture du design d’une page

IV.3 Site Commercial

IV.3.a Client

La création du site commercial c'est basé sur l'ancien site se trouvant à l'adresse suivante : <https://www.alloa.com> . Pour ce qui est du template Bootstrap nous avons utilisé Bushido.

Je me suis occupé à créer les différentes pages html qui vont composer le site comme la page d'accueil , la page de contact , ... en m'assurant de respecter la mise en page voulu par monsieur Dewintre en me basant sur des croquis. Dans un second temps je me suis attelé à ce que ces pages soit responsive donc adapté aux smartphones et tablettes.

Responsive Web Design

Pour visualiser le rendu que le site pouvait donner en version mobile j'ai utiliser la fonction inspecter que l'on obtient en faisant un clic droit sur une page web puis j'ai utiliser l'option Toggle device toolbar.

Pour rendre le site responsive j'ai du apprendre le système de grille de Bootstrap avec les colonnes(col) et les lignes(row). Pour ce faire, le contenu du site web doit être contenu dans une balise dans lequel on applique la class container. Ensuite on crée une ligne avec la class row et on positionne chaque élément dans une colonne à l'aide de la class col. Dans cette configuration les éléments sont placés les uns à côté des autres sur une même ligne sur un écran d'ordinateur et se positionnent les uns en dessous des autres en version mobile.

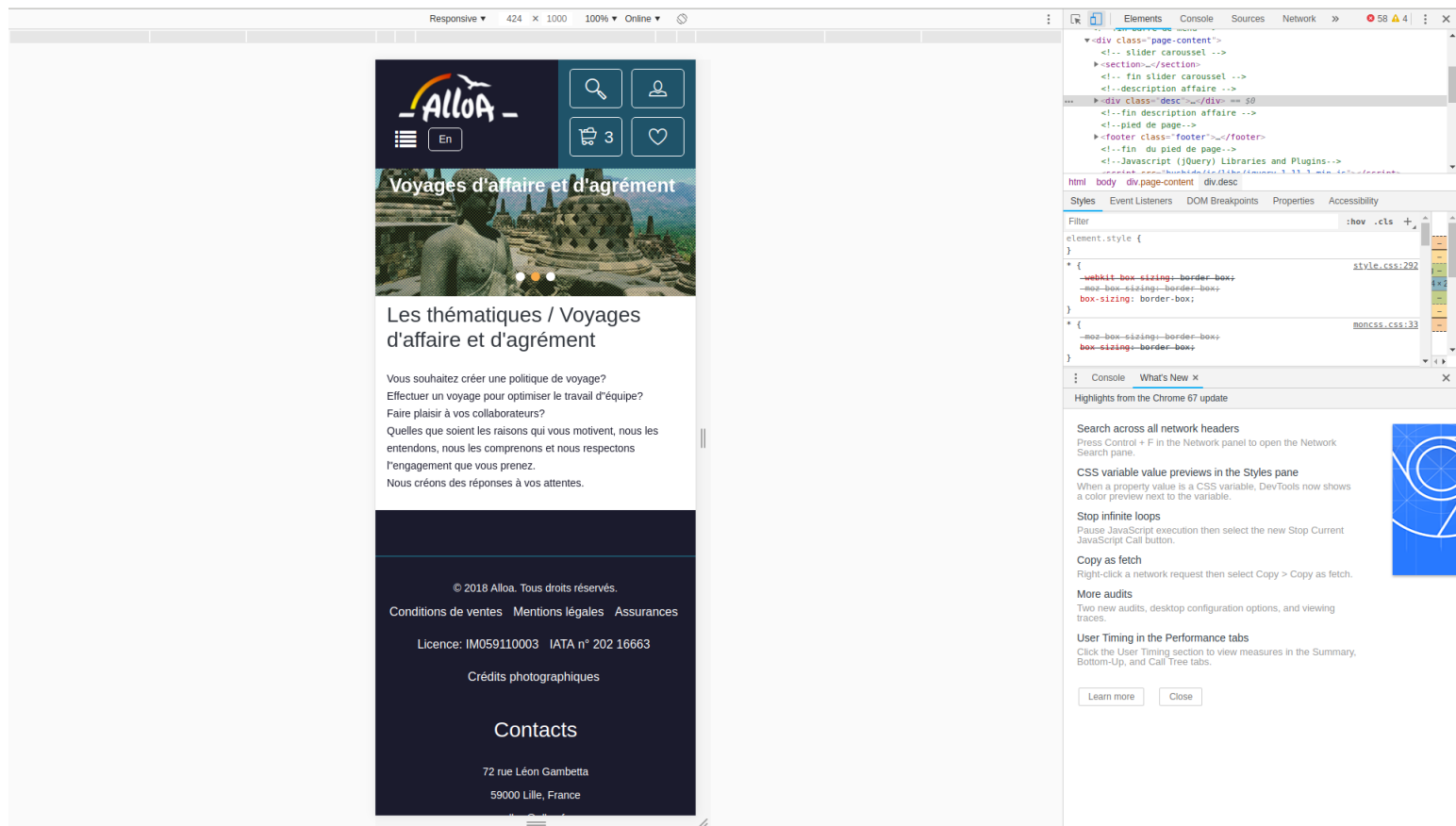


FIGURE IV.2 – Page web responsive utilisation de la fonction inspecter et de Toggle device toolbar

Leaflet

Sur le site nous avons eu besoin d'afficher des cartes et pour se faire nous avons utiliser Leaflet² qui à été développé par Vladimir Agafonkin. Leaflet est une bibliothèque JavaScript open-source pour l'affichage de carte interactive. A l'aide de cette outil nous avons pu afficher des localisations à l'aide de coordonnées GPS (latitude et longitude), positionner un marqueur à un endroit voulu, lui assigner une pop up ... Cela nous à été utile sur la page contact pour la localisation de l'agence de voyage dans Lille et pour la localisation des hébergements et des choses à voir à proximité de ceux-ci ainsi que pour les circuits ou nous avons reliée les différentes étapes du tour.

2. <https://leafletjs.com>

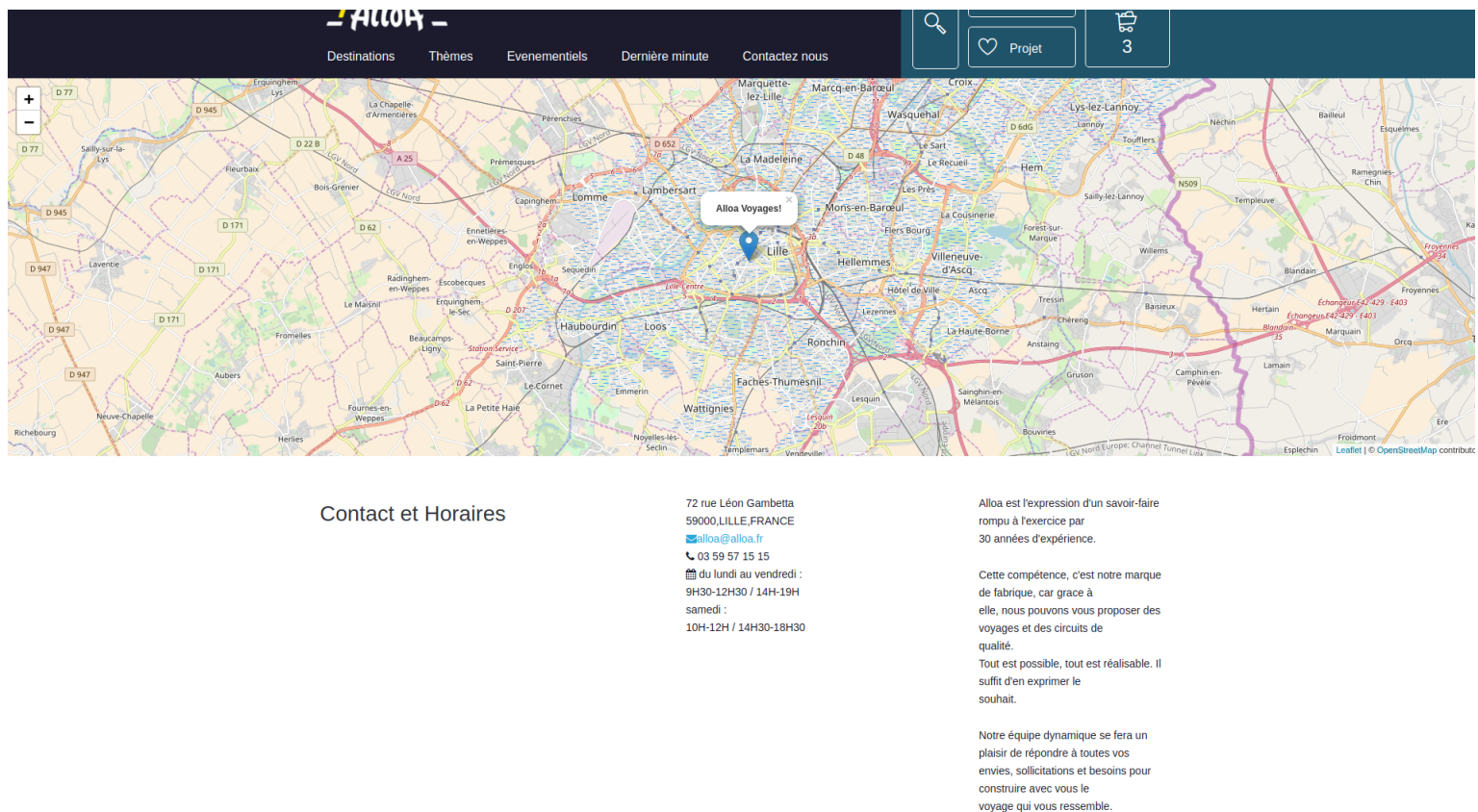


FIGURE IV.3 – Page contact avec affichage d’une carte avec la localisation de l’agence Alloa

IV.3.b Serveur

Je me suis occupé des requêtes en base de donnée pour la récupération des listes de destinations/pays/régions/villes ainsi que de l’affichage des produits (tours touristiques et hébergements) proposé par l’agence.

La gestion des menus deroulants

J’ai du créer les requêtes pour par exemple récupérer à partir de l’uuid d’une destination les pays correspondant à cette destination. Pour stocker les listes de destinations j’ai utilisé en html la balise select qui permet de créer un menu déroulant. Dans ce select j’ai créer autant d’option qu’il y avait de destination et pour chaque destination dans la value correspondant à l’option je lui est associé son uuid.

Je créer ensuite une fonction en GopherJS ou j'ajoute un écouteur à mon menu déroulant. J'écoute l'événement click sur ce menu pour savoir la destination que l'utilisateur a choisi en récupérant la value (ici l'uuid) lié à la destination. Ensuite je passe l'uuid en paramètre de la requête pour obtenir la liste des pays en lien avec cette destination et ainsi de suite pour les régions et les villes.

La récupération des gateways

Une autre tâche qui m'a été confié à été à partir de coordonnées GPS d'un point de trouver les gateways se trouvant à une certaine distance de ce point. Un gateway est une passerelle et correspond à tous ce qui est aéroport, gare , port, métro ...

Tous d'abord j'ai écrit une fonction de conversion de degré en radian. Ensuite j'ai créer une fonction qui calcul la distance entre deux points à partir de leur latitude et leur longitude et du rayon de la Terre que j'ai fixé à 6378000 mètres .

Le calcul étant :

```
distance = RayonTerre * (math.Pi/2 - math.Asin( math.Sin(latitudeCoord2)
* math.Sin(latitudeCoord1) + math.Cos(longitudeCoord2 - longitudeCoord1)
* math.Cos(latitudeCoord2) * math.Cos(latitudeCoord1)))
```

Ensuite il suffit de comparer si la distance calculé entre notre points (ex : Rome) et un gateway quelconque (ex : Aéroport de Milan Malpensa) est inférieur ou supérieur à la distance passé en paramètre. Si elle est inférieur alors le gateway ce trouve dans le périmètre d'étude sinon non.

La récupération et l'affichage des produits

La récupération et l'affichage des produits c'est fait de la manière suivante. J'ai créé des messages dans un fichier .proto pour définir ce à quoi correspond un hébergement et un tour touristique. J'ai ensuite créé des rpc pour définir ce que je souhaite passé en paramètre et ce que je souhaite retourner. Ici je passe en paramètre une DestinationLang qui correspond à un uuid et une lang et je retourne soit un TourList soit un HebergementList.

En second temps je crée une fonction pour appeler ma requête qui prendra en paramètre un pointeur sur un server.DestinationLang et me retournera un pointeur sur un server.TourList ou une erreur si la requête échoue.

Voici un exemple de requête :

```
SELECT DISTINCT ON(tour.uuid_tour) tour.frlib, tour.uuid_tour, tour.uuidville
image.frlib , image.ref_image, image.extension
FROM catalog.tour, catalog.z_liens_regions_villes ,
map.zliens_destinations_regions
WHERE tour.uuid_tour = z_liens_image.uuidcomposant::uuid
AND z_liens_image.uuidimage::uuid = image.uuid_image
AND zliens_regions_villes.uuid_nosregions_nosregions
= zliens_destinations_regions.uuid_nosregions_nosregions
AND tour.uuidville = zliens_regions_villes.uuid_villes_villes
AND zliens_destinations_regions.uuid_noszones = noszones.uuid_noszones
```

J'ai utilisé DISTINCT pour éviter d'avoir des doublons au niveau des résultats. L'utilisation des :: permet de convertir un type dans un autre.

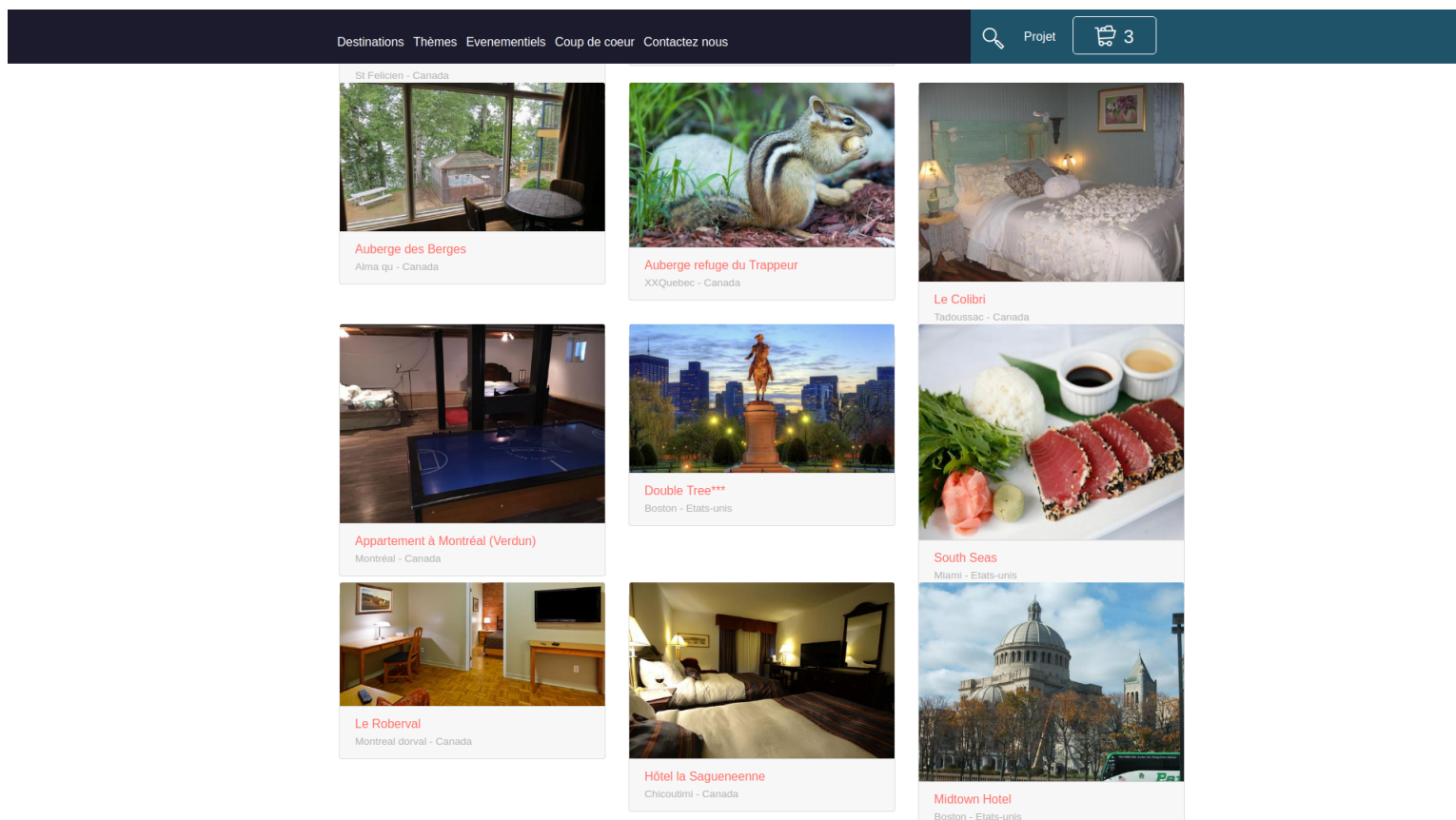


FIGURE IV.4 – Affichage des hébergements correspondant à la destination Amérique du Nord

V | Bilan

Le début du stage à été un peu difficile pour moi pour plusieurs raisons.

Tout d'abord je ne connaissais pas le langage Golang ni le GopherJS donc il m'a fallut un peu de temps pour apprendre les bases du langage. Pour le Golang cela à été plutôt facile car il y a pas mal de documentation sur internet comme <https://godoc.org> qui est un équivalent à la javadoc en java ou encore <https://gobyexample.com> . Pour le GopherJS cela à été plus compliquer car il y a très peu de documentation sur internet.

Pour la partie client(HTML,CSS,JavaScript) j'ai suivi les exemple du site <https://www.w3schools.com> ainsi que ceux sur le site <https://getbootstrap.com>

Pour ce qui est du SQL je me suis servie du site <https://sql.sh> . J'ai également discuté avec monsieur Dewintre sur comment diminué le temps des requêtes en base de donnée en utilisant des index et en créant des liens entre les tables qui été manquant.

Si je ne trouver pas les informations que je recherché j'ai parcouru les forum comme <https://stackoverflow.com> ou je discuter avec les autres stagiaires pour voir si il n'avait pas déjà rencontré les mêmes difficultés.

L'autre difficulté à été de m'intégrer à un projet qui été déjà avancé et de trouver ma place, ce que je pouvais apporter au projet et de fixer moi même mes propres tâches à effectuer chaque jours.

Ce stage m'a appris à être totalement autonome à combattre ma timidité et à m'ouvrir pour demander de l'aide quand j'en avais vraiment besoin. Le stage m'a permis également de croire en moi, et de voir que malgré les difficultés rencontré j'ai réussie à produire des choses et que j'ai réussie à m'intégrer à une équipe.

J'ai également aimé le fait de transmettre les connaissances que j'avais appris à un autre stagiaire pour qu'il puisse reprendre mon travail.

Le projet qui m'a le plus intéressé à été le site web car il y a un travail de création de mise en forme du site qui m'a plus et j'ai également aimé le travail sur les bases de données.

VI | Conclusion

Ce projet d'une durée de 12 semaines à vraiment été intéressant pour plusieurs aspect car j'ai appris de nouvelles technologies comme le Golang. J'ai apprécié aussi le fait de pouvoir travailler sur les deux aspects d'un site, le côté frontend et le côté backend. J'ai pu également mettre en pratique mes connaissances apprises en base de données ainsi qu'en programmation web.

J'ai réussi à faire avancer les différents projets en collaboration avec les autres stagiaires. J'ai appris à leur côté et j'ai pu transmettre mes connaissances en retour ce qui est vraiment enrichissant.

Le seul regret que je peux avoir et de n'avoir pas réussi à finir le site web suite à beaucoup de modifications au jour le jour et le fait que j'ai dû former un stagiaire pour poursuivre mon travail. J'ai dû lui expliquer le projet ainsi que l'utilisation de Git ce qui a demandé un peu de temps.

VII | Abstract