

Rapport Technique

Résumé

Ce document se rapporte au Projet Industriel n°4 de l'école des Mines de Saint-Étienne campus Provence de l'année 2020/2021. Nous avons travaillé du 01/09/2020 jusqu'au 02/03/2021 sur la conception détaillée de notre projet, puis du 02/03/2021 jusqu'au 24/06/2021 nous avons travaillé sur la réalisation et la présentation technique de l'application.

Il parcourt les exigences produites du cahier des charges et rend compte de l'état d'avancement de chacune d'entre elles. Puis, il entre en détails sur le développement de notre projet. Les outils/ softwares/ hardware/ langages utilisés seront mentionnés dans les sections du développement ou ils sont utilisés. Ce document sert de rendu pour l'UP **Rapport** du GP **Projet Industriel : Réalisation**. Cependant, il doit surtout permettre au prochain groupe du PI4 de se familiariser avec l'ensemble de notre travail et notre code pour pouvoir prendre en main notre projet et continuer notre lancée. En plus des détails techniques sur la réalisation du projet, nous inclurons également des conseils sur les prochaines étapes qui permettront de mener à bien le projet. De plus, nous avons commenté tous les codes pour que le PI de l'année prochaine puisse se familiariser plus facilement avec notre projet.

Les élèves travaillant sur ce projet sont Hugo Rechatin, Antoine Montagne et Eddy Abou Mansour. La cliente est Mme. Sabine Salmeron et le tuteur est M. Michel Fiocchi.

Sommaire

Remerciements	3
Glossaire	3
Contexte	4
Le cahier des charges	6
I. Affichage des données	6
II. Import des données	8
III. Actualisation de la base de donnée des stages	8
IV. Interface sans contact	9
V. Conformité à la RGPD	9
VI. Simplicité d'utilisation	10
VII. Gestion des statuts des utilisateurs	10
VIII. Données à exporter	11
IX. Méthodes d'export	12
X. Esthétique	12
XI. Gestion de l'énergie	13
Développement technique du projet	14
I. Prérequis informatique	14
II. BDD	16
III. Page Web	18
IV. Caméra	26
V. Nuitrack	28
VI. Projet Unity	28
VII. Lecteur badge	32
Conclusion	34
Bibliographie	35

Remerciements

Nous tenons avant toute chose à remercier M.Fiocchi et Mme.Salmeron pour leurs conseils qu'ils nous ont donnés durant le projet. Nous voulons aussi remercier M.Gracien pour l'ordinateur et l'écran qu'il nous a fourni, Mr Delattre pour ses conseils concernant le lecteur badge, M.Marques pour son organisation des projets industriels en distanciel, M.Uny pour son aide dans la recherche des informations de stage et les divers étudiants ayant répondu à notre Google Forms nous fournissant ainsi leurs liens LinkedIn.

Nous remercions de même Mme. Bec, M. Amer et M.Bataillie pour leur précieux conseils dans la gestion approfondie du projet.

Enfin, nous aimerions remercier M.Féménia pour son aide constructive et apaisante dans le renforcement des liens de notre équipe.

Glossaire

- **IHM** : Interface Homme Machine
- **IG** : Interface Graphique
- **IDE** : Integrated Development Environment
- **SDK** : Software Development Kit
- **RFID** : Radio Frequency Identification
- **FrontEnd** : Code concernant l'aspect graphique et l'interface utilisateur d'une page web. Les langages utilisés pour le développement de tels codes sont typiquement HTML/ JavaScript/ CSS.
- **BackEnd** : Code concernant tous les aspects invisibles à l'utilisateur d'une page web, comme la gestion des données, l'archivage de conversation, etc. Le langage Python est régulièrement utilisé.
- **plugin** : outil composé d'un ensemble de fichiers informatiques, et qui permet d'installer des nouvelles fonctionnalités en marge d'un logiciel auquel il est rattaché.
- **Npm** : Node Package manager
- **BDD** : Base de données

Contexte

L'ISMIN, le client pour ce PI4, souhaite se doter d'un moyen innovant permettant aux élèves, aux personnels et aux visiteurs de l'école d'accéder aux informations concernant les mobilités internationales effectuées par les différentes promotions.

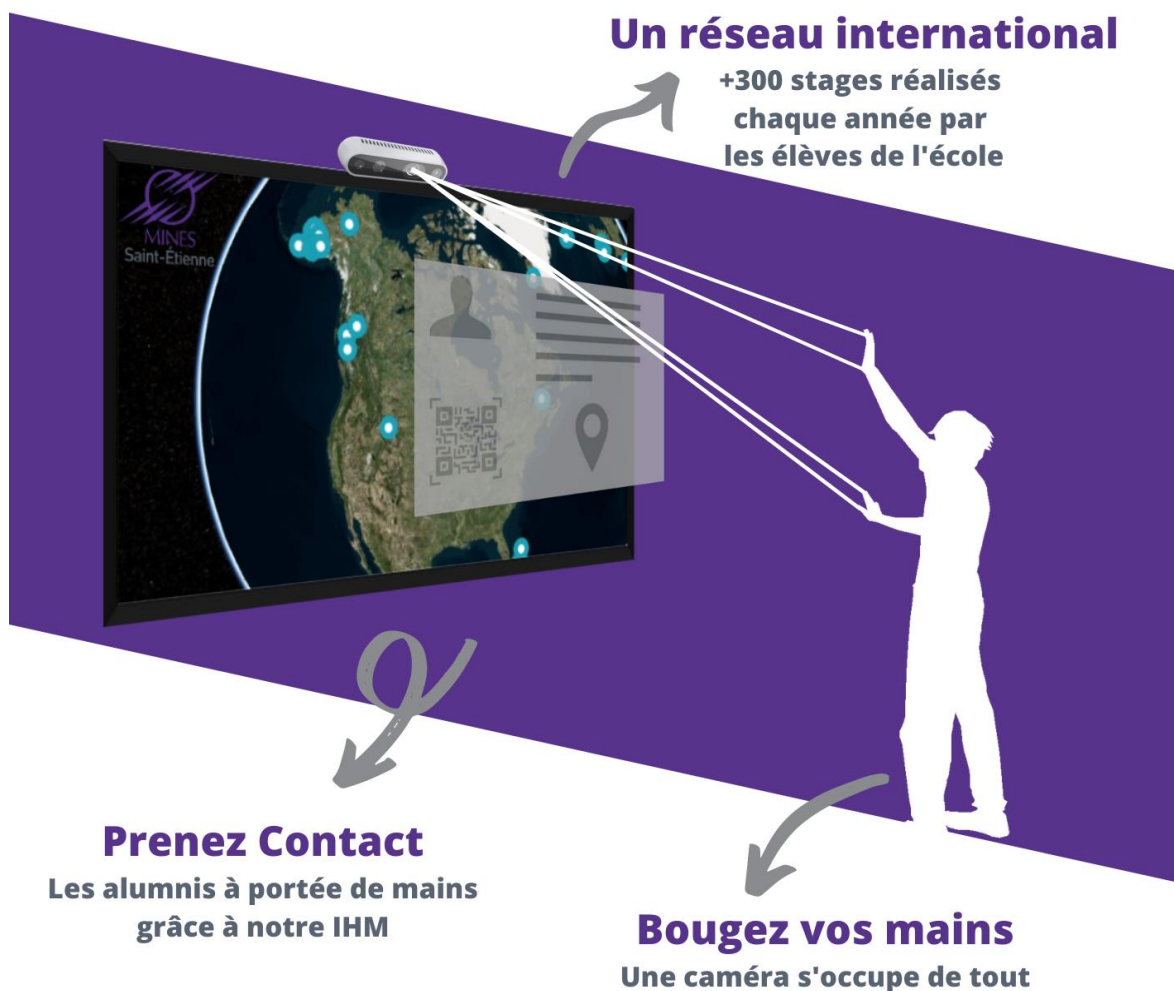
Pour cela, l'école a mis en place un projet industriel qui se résume par le développement d'une IHM gestuelle. Celle-ci permettra de contrôler un globe terrestre sur lequel figure des pins. Une fois sélectionné, un pin procurera des informations sur les étudiants ayant réalisés leur stage à la localisation indiquée.

Ce projet une fois fini sera accessible au sein de l'école afin de faire rayonner celle-ci aux yeux des visiteurs, grâce au savoir-faire de ses élèves et des nombreuses mobilités internationales à leur disposition. De plus, cette interface est un outil idéal pour aider les élèves à développer leurs réseaux afin de faciliter leurs recherches de stage. Pour cette raison, l'interface pourrait être mise à disposition comme une application sur ordinateur une fois le projet terminé.



CONSULTEZ NOS STAGES EN UN COUP DE MAIN

L'OUTIL IDÉAL POUR AGRANDIR VOTRE RÉSEAU
ET FACILITER VOS RECHERCHES DE STAGES
POUR LES ÉTUDIANTS SUR LE CAMPUS DE L'ÉCOLE
DES MINES DE SAINT-ÉTIENNE



PI 4

MEMBRES :
MATHIEU MOREL
ANTOINE MONTAGNE
EDDY ABOU MANSOUR
HUGO RECHATIN
JULIEN MASTRANGELO

TUTRICE INDUSTRIELLE
SABINE SALMERON

TUTEUR ACADÉMIQUE
MICHEL FIOCCHI

Figure 1 : Poster

Le cahier des charges

Le tableau ci-dessous rend compte de l'état de complétion de chaque exigence produit du cahier des charges.

- En vert sont les tâches complétées,
- En rouge les tâches non abordées
- En orange les tâches abordées mais incomplètes

	Urgent	Not Urgent
Important	<ul style="list-style-type: none"> - Affichage des données - Simplicité d'utilisation 	<ul style="list-style-type: none"> - Import des données - Conformité à la RGPD - Interface sans contact - Esthétique
Not Important	<ul style="list-style-type: none"> - Gestion des statuts des utilisateurs (lecteur badge) - Installation complète (PC + écran) 	<ul style="list-style-type: none"> - Actualisation de la BDD - QR Code - Gestion de l'énergie

Table 1 : Etat d'avancements du cdc

I. Affichage des données

Demande Initiale

- Informations à afficher : lieu du stage, l'organisme d'accueil de l'élève, nom de l'étudiant, prénom de l'étudiant, lien vers son profil LinkedIn, retour d'expérience de l'élève sur son stage, titre du stage, promotion, année
- Latitude et longitude pour le géocodage du lieu du stage

Situation actuelle

Nous avons conçu au sein de notre iIG une page d'information contenant toutes les données listées ci-dessus pour chaque stage :

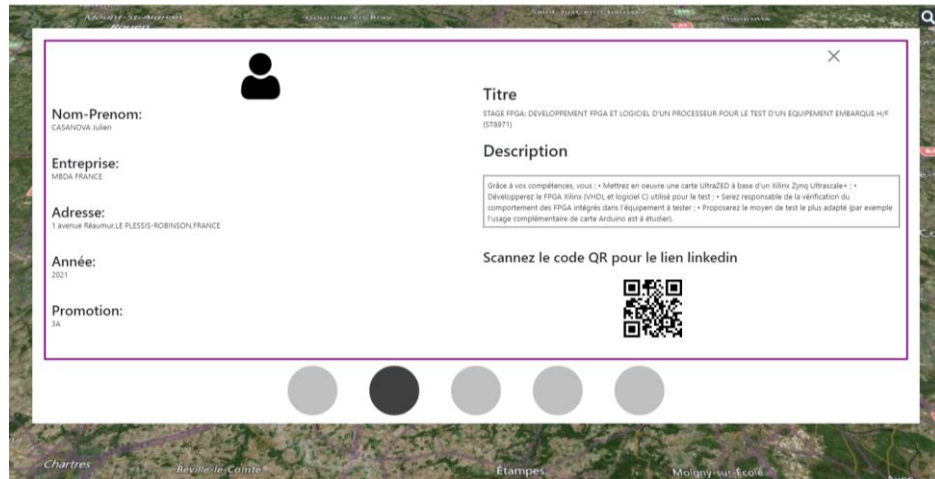


Figure 2 : Fiches d'infos stage de l'interface web

Les pins qui figurent sur le globe sont géolocalisés grâce à la librairie Geopandas en utilisant les adresses des entreprises

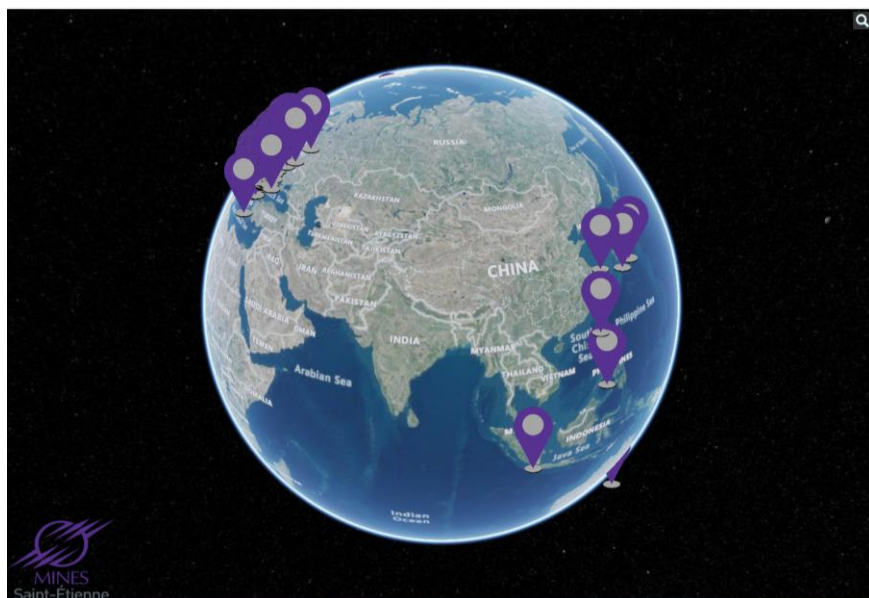


Figure 3 : Globe de l'interface web

Nous sommes tous les trois d'accord pour dire que la fonctionnalité d'affichage des données a été satisfaite et nous ne voyons pas d'ajout nécessaire.

II. Import des données

Demande Initiale

- Lien avec Prométhée
- Obtenir la géolocalisation de l'entreprise, les liens LinkedIn des étudiants et la géolocalisation des organismes

Situation actuelle

Le lien avec Prométhée n'a malheureusement pas pu être effectué. La DSI de Saint-Étienne nous en a refusé l'accès. Nous avons cependant pu récupérer les fichiers excels des stages effectués par les différentes promotions depuis trois ans. Dans ces informations, seul le LinkedIn des étudiants n'y figurait pas. Nous avons donc mis en place des Google Forms pour récupérer les liens LinkedIn des étudiants de l'ISMIN afin de compléter notre BDD.

Cette solution n'est que temporaire, nous espérons que les étudiants pourront par la suite renseigner leur compte LinkedIn pour tout départ en stage.

Il faudra aussi donner un accès de lecture restreint sur Prométhée aux prochains développeurs de notre application pour qu'ils aient accès aux données des stages.

III. Actualisation de la base de données des stages

Demande Initiale

- 3 fois par an : à chaque fin de stage 1A, 2A et 3A
- Délais avant suppression d'une information de stage : entre 3 et 5 ans - Différentiation de la suppression d'un stage en fonction de sa rareté

Situation actuelle

Cette fonctionnalité de notre application est impactée par le fait que nous n'ayons pas pu faire de lien direct entre notre application et la BDD de Prométhée. Les données que nous avons recueillies offrent un panel des stages effectués par nos étudiants sur trois ans. Nous avons codé un script Python permettant de réinitialiser et de remplir notre BDD à partir des différents fichiers Excel comprenant les informations de stage. Ces fichiers sont stockés

dans un dossier de l'ordinateur. Pour actualiser à la BDD, il faut donc remplir le dossier avec les fichiers Excels des nouveaux stages et lancer le script Python.

Il serait intéressant par la suite d'automatiser ce processus.

IV. Interface sans contact

Demande Initiale

- Distance comprise entre 1.5 m et 2 m pour pouvoir interagir avec le dispositif (distance à préciser en fonction de la taille de l'écran)
- Pas d'objet physique en contact avec l'utilisateur
- La solution technique de cette interface et son usage doivent être interchangeables

Situation actuelle

Nous avons eu recours à un SDK (Nuitrack) fourni par 3DIVI qui permet de développer des interfaces gestuelles sans besoin d'objet physique fixé sur l'utilisateur. De plus, le SDK peut fonctionner avec une multitude de différentes caméras précisées sur leur [site](#). Nous parlons notamment dans le développement du projet, des problèmes que notre caméra a causés et des alternatives possibles.

En pratique, nous avons bien respecté la distance de 1.5 - 2 m pour l'interaction. Cela étant dit, nous jugeons qu'il est nécessaire de changer la caméra.

V. Conformité à la RGPD

Demande Initiale

- On peut alors citer quelques articles de la RGPD tels que :
 - o Article 12 : Transparence des informations et des communications et modalités de l'exercice des droits de la personne concernée
 - o Article 17 : Droit à l'effacement, l'oubli
 - o Article 32 : Sécurité du traitement

Situation actuelle

Notre utilisation des données n'est pas considérée comme une utilisation sensible. Il faut néanmoins minimiser les données personnelles collectées, nous avons veillé à seulement utiliser les données nécessaires à la description d'un stage.

Il nous a fallu prévoir une possibilité de suppression des données de notre BDD grâce à un simple script Python qui peut être utilisé à tout moment.

Pour la récupération des liens Linkedin des étudiants, nous avons bien fait attention de préciser le cadre d'utilisation des données dans le Google Form.

VI. Simplicité d'utilisation

Demande Initiale

- Prise en main rapide (<30 secondes)
- Gestes normés à valider avec la cliente
- Limiter à certains gestes

Situation actuelle

Nous avons réussi à créer une interface avec un faible nombre de gestes, 4 pour être exact, tout en comptant le zoom et le dézoom comme deux gestes distincts. Le fonctionnement de chaque geste est représenté dans le [Manuel de gestes.pdf](#) à l'aide d'images.

Nous ne pouvons cependant pas confirmer que la prise en main soit rapide. En effet, notre interface étant toujours instable nous n'avons pas effectué de test utilisateur qui serait indispensable pour regrouper un retour nous permettant d'affiner le design de nos gestes afin d'avoir une IHM plus ergonomique.

VII. Gestion des statuts des utilisateurs

Demande Initiale

- Accès différents selon le statut de l'utilisateur : interne ou externe à l'ISMIN
- Le statut de l'utilisateur doit pouvoir être vérifié
- Lecteur de badge
- Informations de base : lieu et domaine clef de la mobilité

Situation actuelle

À l'aide de la Nucleo-32 et d'un lecteur carte Arduino, nous avons conçu une borne RFID. Celle-ci peut confirmer ou infirmer l'appartenance d'un badge à notre

BDD. Pour les tests et la démo, nous avons créé une base contenant seulement trois cartes.

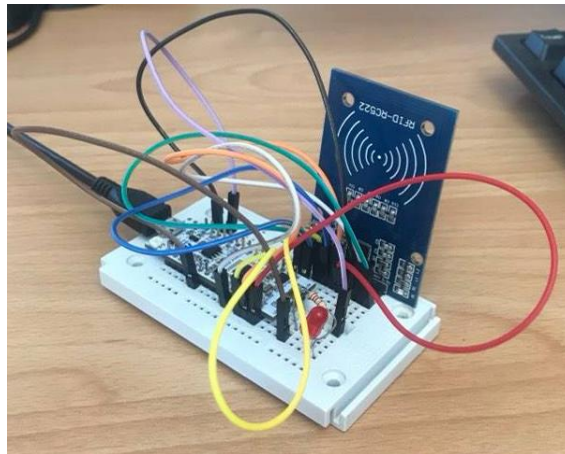


Figure 4 : Lecteur badge Arduino

Le script Python renvoyant un booléen est lié au code de la page web, ainsi en présentant une carte reconnue on peut avoir accès au globe complet, sinon on accède à celui contenant qu'une portion des données.

Les données telles que le lien LinkedIn et le nom de l'étudiant sont réservées à ceux possédant un badge reconnu. Les données montrées à tous sont les noms des entreprises et les descriptions des stages.

Pour le produit final, il faudra récupérer la BDD utilisée pour les portes de l'école pour remplacer notre BDD de test. De plus, il serait question de créer un boîtier pour le lecteur carte assurant une plus longue durabilité.

VIII. Données à exporter

Demande Initiale

- Données devant être extraites : uniquement le LinkedIn de l'élève via son accord pour respecter la RGPD

Situation actuelle

Grâce à une librairie Angular (ngxQRCODE) nous avons pu convertir les liens URL LinkedIn des étudiants en QR code. Nous avons récupéré ces liens grâce à un Google Forms expliquant le PI que nous avons envoyé aux différentes promotions.

Cela étant dit, nous n'avons pas pu récupérer tous les liens. Il faudrait trouver une méthode plus efficace pour récupérer les LinkedIn de tous les étudiants.

IX. Méthodes d'export

Demande Initiale

- 5 secondes pour pouvoir extraire les informations
- Export possible uniquement pour les personnes avec badge
- Format : .pdf, .txt, autres ?
- QR-code ou mail.

Situation actuelle

Nous avons jugé que l'information essentielle à tirer de notre IHM est une liste de contacts pouvant élargir le réseau d'un élève recherchant un stage. Ainsi, la donnée pouvant être récupérée par l'utilisateur est un lien LinkedIn de l'élève ayant réalisé le stage géolocalisé sur notre IHM.

Avec son téléphone, l'utilisateur peut directement accéder à la page LinkedIn de l'étudiant pour ainsi entrer en contact avec lui.

L'IHM pourrait bénéficier d'un second QR code donnant accès à un PDF contenant toutes les informations figurant sur la fiche informative. De plus, s'il vient à exister une version ordinateur de l'IHM, il pourrait être intéressant de remplacer les QR code par des liens direct afin d'accéder aux pages facilement avec une souris.

X. Esthétique

Demande Initiale

- Un globe
- Pointeur représentant les stages
- Modernité

Situation actuelle

Notre IG a été créée avec la simplicité en tête. Nous ne voulons pas noyer l'utilisateur d'informations superficielles, mais garder l'essentiel à l'écran. Les informations des stages se retrouvent sous la forme de pins répartis sur un globe 3D. Pour assurer une simplicité dans le visuel et dans l'interaction avec l'IHM nous avons ajouté des curseurs représentant les mains de l'utilisateur. Bien qu'il nous fût initialement demandé de faire sans

ces derniers, après discussion avec notre cliente, nous avons conclu que sans eux, l'utilisation de l'IHM devenait moins ergonomique.

Comme pour la [Simplicité d'Utilisation](#), nous n'avons pas eu le temps d'organiser un retour d'utilisateur qui serait indispensable avant de considérer l'IG comme complète.

XI. Gestion de l'énergie

Demande Initiale

- Mise en veille au bout de 15 minutes environ si aucune personne n'utilise le système

Situation actuelle

Le système rentre en veille au bout de 35 minutes, seul un mouvement du périphérique de la souris permet de sortir l'écran du mode veille. Nous n'avons pas pris le temps ce semestre d'adresser la question de réveil à l'aide de la caméra. En effet, nous avons jugé qu'il était plus urgent de finir les éléments majeurs du projet comme l'IHM, la BDD et le lecteur carte.

Nous pensons qu'un détecteur de mouvement IR pourrait servir de bouton "Touchless".

Développement technique du projet

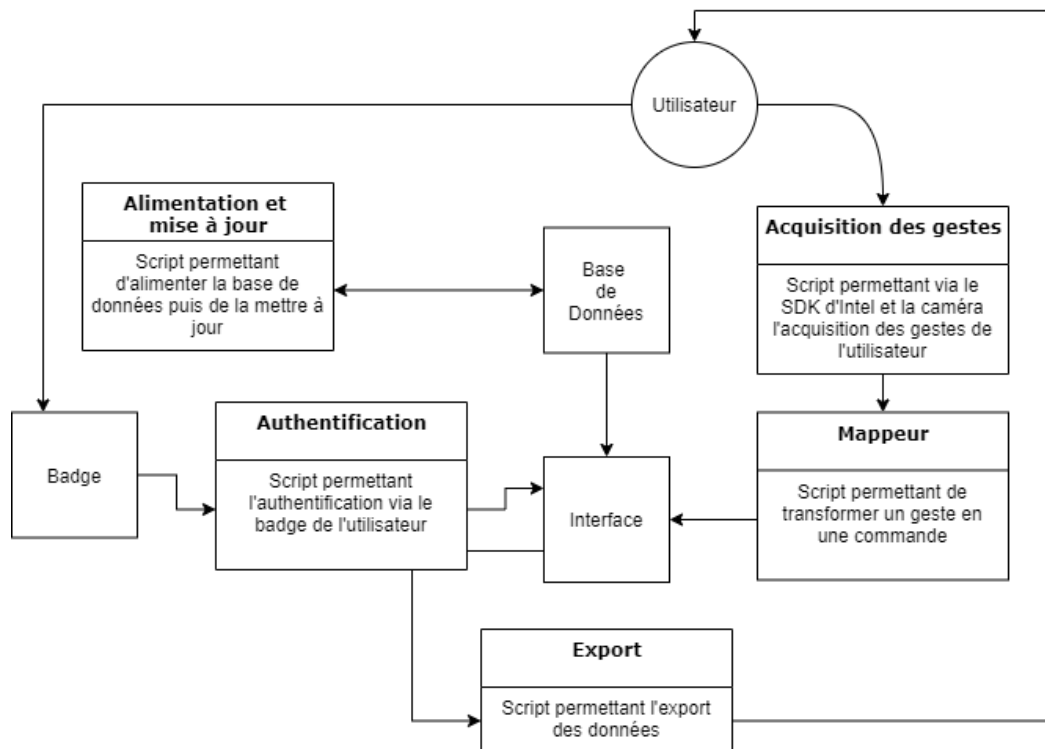


Figure 5 : Diagramme de synthèse du projet

I. Prérequis informatique

Vous avez ci-dessous la liste complète des langages, Librairie et IDE utilisés pour ce projet suivie d'une liste des éléments du projet pour lesquels ils sont utilisés.

Langages

Python : 3.8.3 64-bit ('base' : conda)

- Script du BackEnd liant le code Arduino du lecteur carte au code de la page web. Il permet de confirmer ou non l'appartenance d'un badge à la BDD.
- Script du BackEnd important le contenu de la BDD des stages dans l'IG

CPP

- Script Arduino récupère le code hexadécimal d'un badge et l'envoie à un script python du BackEnd

HTML

- Utilisé pour afficher le contenu FrontEnd de la page web

CSS

- Utilisé pour ajouter le style (format de texte, taille des éléments, couleurs...) au contenu qu'apporte le HTML

TypeScript

- Utilisé pour ajouter du dynamisme à la page web (les événements des boutons, la navigation entre deux pages web et cetera...)

Postgresql

- Utilisé pour le système de gestion des différentes BDD du projet (lecteur carte, informations des stages)

Librairies/ Plugins

Cesium

- Utilisé pour l'affichage des composants comme le globe terrestre.

Psycpg2

- Utilisé pour valider si le code hexadécimal d'une carte appartient à la BDD ou non

SQLalchemy

- Utilisé pour récupérer les informations de la BDD via requête SQL, notamment les informations des stages

Geopandas

- Utilisé pour la géolocalisation des stages à partir du nom des différentes combinaisons entre nom de pays, ville, code postale, adresse de l'entreprise

unity_browser

- Plugin Unity servant à insérer une page web dans un projet unity.

Nuitrack SDK

Utilisé pour la détection des mouvements via la Caméra Intel RealSens D435

IDE

VSCode

- Utilisé pour le FrontEnd et le BackEnd

VStudio

- Utilisé pour modifier les scripts concernant les objets du projet Unity

Unity

- Utilisé pour combiner les différents éléments du projet et produire un exécutable

Arduino

- Utilisé pour télécharger les codes CPP sur notre STM32 et envoyer les données lus par le lecteur carte au BackEnd par un port série

PGAdmin 4

- Interface qui facilite l'utilisation de PostGresql et la visualisation des tables de BDD

II. BDD

Intro

Deux bases de données sont utilisées dans notre projet, elles ont été créées avec le logiciel pgAdmin 4.

La première base regroupe toutes les informations utiles à l'identification du stage d'un étudiant : nom et prénom de l'élève, mail de l'élève, promotion de l'élève, année de l'élève, compte LinkedIn, titre du stage, description du stage, nom de l'entreprise, adresse de l'entreprise, code postal de l'entreprise, ville de l'entreprise et pays de l'entreprise. Ces données nous ont été fournies par la DSI de Saint-Étienne sous forme de fichiers Excel. Nous avons récupéré les LinkedIn des élèves via des Google Forms dont nous avons extrait les informations sous forme de fichiers Csv.

L'autre BDD regroupe les identifiants des cartes des personnes internes à l'ISMIN. Les trois seules données présentes dans cette base sont les identifiants des trois cartes des membres de notre projet industriel.

PgAdmin 4

PgAdmin 4 est une IG permettant de créer une BDD en générant automatiquement le code correspondant. Cette interface utilise le langage PostgreSQL et elle propose des scripts permettant par exemple d'insérer des données dans une table.

Les données contenues dans les bases créées sont du type "text" pour permettre une interaction plus simple avec les informations contenues dans les fichiers Excels fournies par la DSI.

	uid text		nom_prenom [PK] character varying	mail character varying	titre character varying	description character varying
1	136.4.63.96.211		1 ABI RIZK Joseph	joseph.abirizk@etu.em...	Consultant Stagiaire en...	Découverte du métier d...
2	136.4.89.67.150		2 ABOUBAKRI Mehdi	mehdi.aboubakri@etu....	Software intern engineer	- Develop native iOS an...
3	136.4.24.75.223		3 ABOUHACHIM ALAMI Abde...	abdessalam.alami@et...	Ingénieur en électroniq...	Durant ce stage, j'aurai ...
			4 ADLI Karima	karima.adli@etu.emse.fr	Ajout de la fonction réc...	Le stage est découpé e...
			5 AKIN Yavuz	yavuz.akin@etu.emse.fr	Develop an energy pric...	Get up to speed on the ...
			6 ALAUX Camille	camille.alaux@etu.em...	Mise en oeuvre de cart...	Mise en oeuvre de cart...
			7 ALAUX Camille	camille.alaux@etu.em...	Cabin Electronics	Développement de sys...
			8 ALBIGNAC Stéphane	stephane.albignac@et...	Tooling Development f...	Your role will be related...
			9 ALDA Lorys	lorys.alda@etu.emse.fr	Optimization of Defecti...	Support on optimizing ...
			10 ALI Adel	adel.ali@etu.emse.fr	Tool developer- Diagno...	comprendre un outil éc...

Figure 6 : Visualisations des deux tables utilisé pour notre projet (table des stages et table des badges) sur PgAdmin 4

Import de données

Pour remplir la BDD, nous avons décidé de passer par un script Python qui permet une lecture aisée d'un fichier Excel et permet un lien avec une BDD très simple. Deux bibliothèques Python ont été utilisées : la première est "pandas" qui permet de lire un fichier Excel avec la fonction "read_excel" et de stocker les données dans un tableau de deux dimensions nommé "DataFrame", la deuxième bibliothèque est "psycopg2" qui permet de faire le lien avec la BDD.

Bibliothèque "Psycopg2"

La fonction "psycopg2.connect" permet de créer une nouvelle session de connexion à la BDD de notre choix, elle renvoie aussi un objet relatif à la création de cette connexion. Plusieurs paramètres doivent être renseignés pour vous connecter, "host" et "port" possèdent des valeurs par défaut, les autres paramètres dépendent de la façon dont vous avez paramétré pgAdmin.

```
connection = psycopg2.connect(user="postgres", password="PI4PI4", host="127.0.0.1", port="5432", database="postgres")
cursor = connection.cursor()
cursor.execute("DELETE FROM stages_final")
```

Ensuite, la classe "cursor" permet d'exécuter des commandes PostgreSQL, ces curseurs sont créés par la méthode "connexion.cursor". On a ensuite un exemple du type de commandes que l'on peut effectuer en utilisant la méthode "cursor.execute".

Autres fonctions utilisées dans le script :

- "connexion.commit" : après chaque modification de données, cette fonction permet une validation de la requête qui vient d'être effectuée
- "cursor.fetchall" : fonction qui renvoie les résultats d'une requête SQL sous forme de tableau
- "cursor.close" : ferme le curseur
- "connection.close" : ferme la connexion avec la BDD

III. Page Web

Intro

L'IG représentée par une page web est une partie essentielle du projet. Elle a été faite indépendamment de l'interface gestuelle. Ce site peut donc fonctionner avec une souris. Dans le futur, il pourrait être mis en ligne et les élèves y auraient accès depuis chez eux. C'est idéal pour des sessions de recherches plus longues.

Il est néanmoins important de garder en tête que le but du projet est d'implémenter les gestes de mains en tant qu'entrées de notre application remplaçant ainsi la souris. La page web est aussi divisée en deux parties indépendantes : Le FrontEnd et le BackEnd. Dans la suite nous allons expliquer comment faire fonctionner les codes que nous avons déjà écrits.

Le but de cette page web est d'afficher une page d'ouverture d'où on peut accéder au globe qui à son tour charge les informations relatives aux pins du BackEnd et les affiche sur le globe.

FrontEnd

Pour le FrontEnd, nous donnons accès à tout notre code, mais celui-ci doit être utilisé par un autre groupe travaillant sur ce projet. Bien que l'on a installé toutes les dépendances nécessaires sur l'ordinateur fourni par l'école, nos successeurs devront tout de même utiliser leurs propres ordinateurs. Dans ce cas, ils doivent reproduire le même environnement pour le projet, ce qui signifie qu'ils doivent installer les dépendances et les bibliothèques manquantes liées à ce projet.

Nous avons utilisé le framework Angular qui est une plateforme pour construire des applications web en utilisant HTML, TypeScript et CSS. Pour installer Angular, nous devons installer Node.js. L'installation complète se trouve sur le site suivant :

<https://ccbill.com/kb/install-angular-on-windows>

En JavaScript, il existe une bibliothèque pour l'utilisation de globe 3D appelée Cesium. Son rôle est de charger le globe et d'afficher des objets sur ce dernier comme des avions, des pins, des aéroports, etc. Angular est également important pour la facilité d'utilisation concernant la synchronisation des données provenant du BackEnd avec l'interface au moment de l'exécution. Pour cette raison, nous avons trouvé une bibliothèque appelée Angular-Cesium qui combine la puissance impressionnante de Cesium en tant que moteur de carte et Angular en tant que cadre frontal. Le principal avantage d'Angular-Cesium est sa capacité à garder notre carte en parfaite synchronisation avec toutes nos sources de données, qu'il s'agisse de sources de données dynamiques qui envoient de nombreuses données par WebSocket, d'une source de données HTTP unique qui récupère des données une fois, ou d'une action utilisateur qui modifie l'état interne de notre application. Nous avons géré tout cela en utilisant des composants Angular très simples et lisibles, tout en mettant en avant des performances élevées et une utilisation facile.

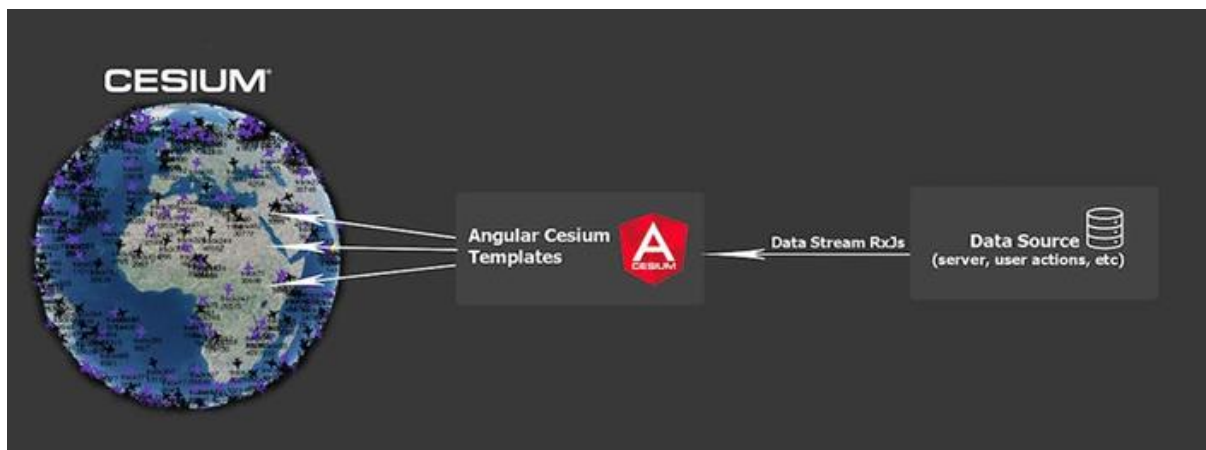


Figure 7 : Flux de données entre Angular-Cesium et la BDD

Après avoir installé NodeJS et Angular, il est temps d'installer Angular-Cesium. Mais d'abord nous devons installer angular CLI dans le dossier FrontEnd sur GitHub.

npm install -g @angular/cli

Puisque le NPM est maintenant une commande dans notre terminal, nous pouvons utiliser :

npm install --save angular-cesium.

Il est très important que le code soit compris et il est impératif d'avoir une expérience de base en Angular pour savoir comment il fonctionne.

Nous allons expliquer le rôle de chaque fichier, de plus, chaque fichier est commenté afin de préciser le rôle de chaque morceau de code. Il faut noter que tous les fichiers ne sont pas originaux. Certains d'entre eux sont créés automatiquement lors de la création d'un projet Angular. Ces fichiers de configuration gèrent les dépendances et nous aident à travailler en mode production ou développement. Dans notre cas, nous travaillons en mode développement puisque le projet n'a pas encore été officiellement mis en production. Il y a des fichiers que nous pouvons modifier et d'autres qui peuvent causer des problèmes si on les modifie de façon incorrecte.

Notre travail se situe principalement dans le dossier FrontEnd/src/app. Nous avons créé trois composants à l'aide de la commande :

ng g c [Nom du composant]

Dans chaque dossier de composant, il y a trois fichiers : Le fichier HTML qui structure la page web, le fichier CSS qui ajoute du design à la page et le fichier TypeScript qui ajoute de la dynamique à la page pendant l'exécution.

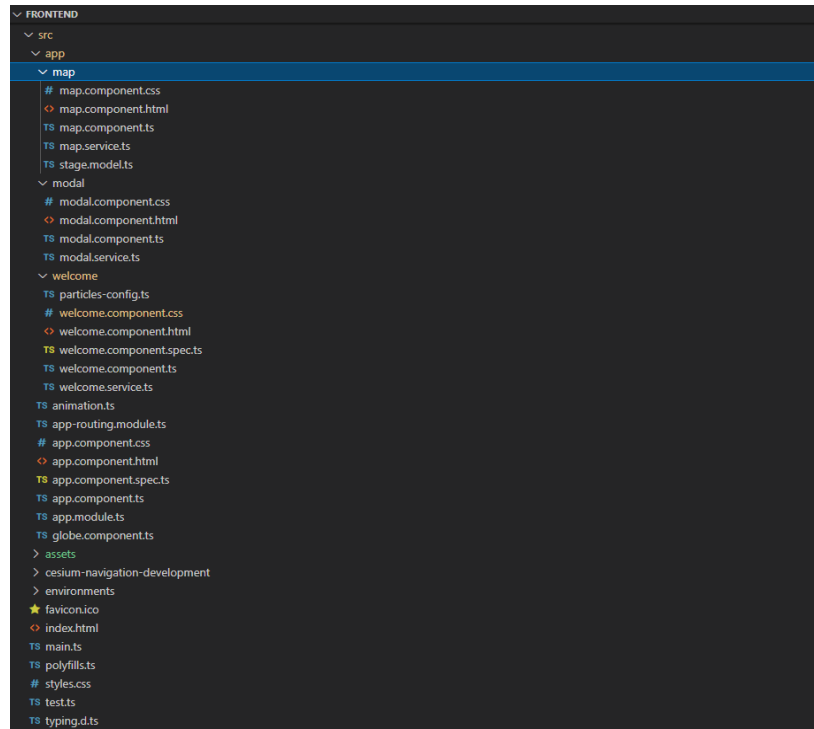


Figure 8 : Hiérarchie des fichiers du FrontEnd

- “MapComponent” est utilisé pour afficher le globe césium et les pins relatifs aux informations sur les stages transmises depuis le BackEnd. La connexion entre ce composant et le BackEnd se fait par le “MapService” que nous avons créé en utilisant cette commande

ng g s [Nom du service]

Le service ouvre une connexion HTTP en utilisant la dépendance HttpClient entre le serveur Angular et le serveur Python qui représente le BackEnd. La rootURL que nous allons utiliser pour nous connecter au serveur BackEnd est `http://localhost:5000`. Nous avons également créé une classe nommée "Stage" qui représente une AcEntity. C'est une entité spécifiée utilisée dans Angular-Cesium, elle est principalement utilisée pour représenter un objet pin. Cette entité a les propriétés suivantes :

nomEleve:string;

description:string;

titre:string;

nomEntreprise:string;

promo:string;

annee:string;

adresse:string;

position:string;

linkedin:string;

id:number;

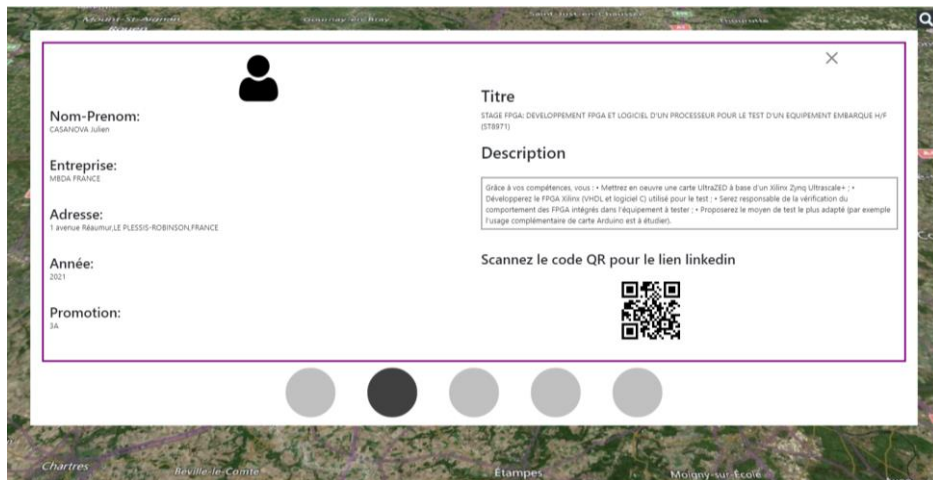


Figure 9 : Fiches d'infos stage de l'interface web

Nous avons utilisé cette dernière pour stocker le flux de données provenant du côté BackEnd.

- “ModalComponent” a été utilisé pour afficher la fiche d'information relative à chaque pin. Nous avons dit précédemment que chaque pin est un objet AcEntity qui contient certaines propriétés utilisées pour définir un stage. Nous avons donc implémenté un événement cliquable qui affiche ce composant modal où nous pouvons voir les informations relatives à chaque stage.
- “WelcomeComponent” qui sert d'écran de veille ou de page Web par défaut du système. L'écran de bienvenue contient deux boutons. Le premier est destiné aux étudiants ou au personnel de l'école qui possèdent une carte d'accès, tandis que l'autre bouton est destiné aux visiteurs qui peuvent accéder au système sans avoir à présenter leur carte. La différence entre les deux types d'utilisateurs est que les utilisateurs vérifiés peuvent accéder aux données personnelles des étudiants ainsi qu'aux données du stage et de l'entreprise, tandis que les visiteurs ne peuvent accéder qu'aux données du stage et de l'entreprise.

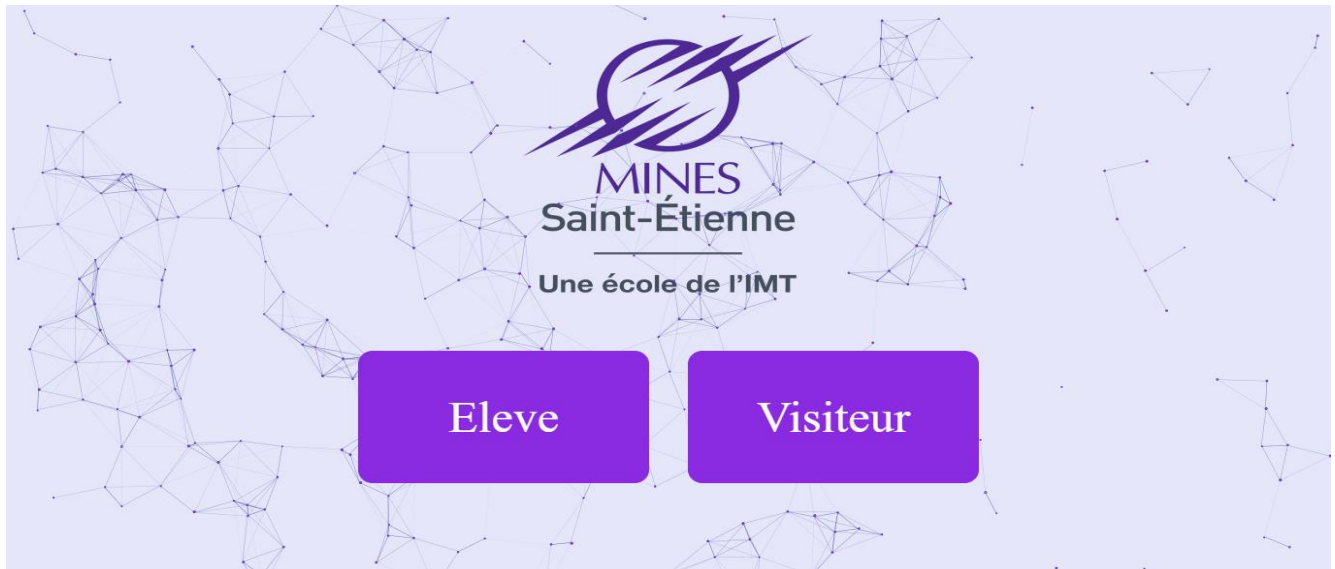


Figure 10 : Welcome screen

Maintenant que nous en avons terminé avec les composants que nous avons créés, nous pouvons parler des composants automatiques créés. Le composant “app” combine tous ces composants et crée une page web structurée. Nous pouvons voir dans le dossier app :

- **app-routing.module.ts** qui définit les routes que nous pouvons atteindre à l'intérieur de la page web.

<http://localhost:8500/map> pour accéder à la carte

<http://localhost:5000/welcome> pour accéder à la page d'accueil

- **app.component.ts, app.component.css, app.Component.html** qui est la structure de base pour créer une page web et qui est utilisée pour connecter les différents composants que nous avons mentionnés plus tôt.
- **animation.ts** est utilisé pour ajouter une animation de particules à la page de bienvenue.
- **app.module.ts** est l'endroit où nous ajoutons tous nos modules, imports, déclinaisons, providers utilisés dans l'application comme NgxQRCodeModule par exemple utilisé pour implémenter la fonctionnalité QR code au système ou le AngularCesiumModule utilisé pour intégrer le globe à l'interface.
- **globe.component.ts** est un composant TypeScript créé par nous pour ajouter la couche de carte Angular-Cesium et changer certaines configurations comme ne pas montrer les altitudes sur le globe ou ne pas montrer le bouton d'accueil et le bouton de recherche. Ces configurations sont appelées **viewerOptions**.

Après le dossier de **app**, nous pouvons voir qu'il y a un dossier **assets**. C'est là que nous conservons toutes nos images, icônes et logos. Le fichier **typing.d.ts** est utilisé pour stocker les variables globales. **Index.html** est le fichier où nous combinons tous nos scripts et composants y compris le dossier des composants de l'application. **Main.ts** ajoute du dynamisme à **index.html** et **styles.css** sert de fichier CSS global pour tous nos composants.

Sachant que chaque fichier et sa fonctionnalité sont présentés. Il ne reste plus qu'à exécuter le programme. La commande est très simple, mais peut être modifiée en fonction de vos ports TCP disponibles.

ng serve --port [spécifiez le numéro du port, dans notre cas, c'est 8500 et la valeur par défaut est 80].

Et voilà, le serveur est opérationnel. Cela étant dit, en naviguant sur le globe, nous pouvons remarquer qu'il n'y a pas de pins. La raison en est que nous n'avons pas encore activé le serveur BackEnd. Nous allons suivre les mêmes étapes pour activer le serveur Python.

BackEnd

Nous avons utilisé Python pour sa simplicité et pour ses nombreuses bibliothèques. La version de python que nous avons utilisée est "Python 3.8.3 64-bit ('base' : conda)". Si une nouvelle version doit être installée, il y aura peut-être des problèmes avec certaines bibliothèques utilisées.

Voici le site officiel de téléchargement de Python : <https://www.python.org/downloads/>

Maintenant, pour installer les bibliothèques, nous avons besoin de la commande qui diffère du packaging Python installé.

Si notre packaging Python est conda, nous devons utiliser : **conda install [Nom de la librairie]**.

Si le packaging Python installé est pip, alors c'est : **pip install [Nom de la librairie]**.

Les librairies Python les plus importantes à installer sont :

- **flask_sqlalchemy** : Flask-SQLAlchemy est une extension pour Flask qui ajoute le support de SQLAlchemy à notre application. Elle vise à simplifier l'utilisation de SQLAlchemy avec Flask en fournissant des valeurs par défaut utiles et des aides supplémentaires qui facilitent l'accomplissement des tâches courantes.
- **flask_cors** : Une extension Flask pour gérer le partage des ressources d'origine croisée (CORS), rendant possible AJAX d'origine croisée. Nous avons utilisé cette

librairie pour permettre la cross-communication entre les deux serveurs angular et flask.

- **flask_migrate** : Flask-Migrate est une extension qui gère les migrations de bases de données SQLAlchemy pour les applications Flask utilisant Alembic. Les opérations de BDD sont disponibles via l'interface de ligne de commande de Flask.

Dans notre code, nous avons utilisé flask-migrate pour migrer notre classe "Stage" vers notre BDD et créer une copie de cette classe sous la forme d'une table. Nous avons juste besoin d'exécuter trois commandes :

1- Avec la classe "Stage", vous pouvez créer un référentiel de migration avec la commande suivante :

flask db init

2- Vous pouvez ensuite générer une migration initiale :

flask db migrate -m "Migration initiale"

3- Vous pouvez ensuite appliquer la migration à la BDD :

flask db upgrade

- **GeoPandas** : GeoPandas est un projet open source visant à faciliter le travail avec des données géospatiales en Python. GeoPandas étend les types de données utilisés par pandas pour permettre des opérations spatiales sur des types géométriques. Les opérations géométriques sont effectuées par shapely. GeoPandas dépend en outre de fiona pour l'accès aux fichiers et de matplotlib pour le traçage.

Dans le code, nous avons utilisé la fonction pour retourner la longitude et la latitude du fournisseur "nominatim" :

```
adresse=geocode("Adresse"), provider='nominatim',  
user_agent='autogis_xx', timeout=4)
```

- **flask_caching** : Flask-Caching est une extension de Flask qui ajoute le support de la mise en cache pour divers BackEnds à toute application Flask. En plus de fournir un support pour tous les BackEnds de mise en cache originaux de werkzeug à travers une API uniforme, il est également possible de développer notre propre BackEnd de mise en cache en sous-classant la classe

- **flask_caching.backends.base.BaseCache**.

Nous avons utilisé le caching dans notre code pour éviter de charger à chaque fois les adresses du serveur Python vers le serveur Angular, car cela prend environ 10

minutes à chaque fois pour charger. Nous avons donc utilisé le caching pour sauvegarder une version dans la mémoire cache pendant 24 heures. Sans avoir besoin de convertir les adresses avec la bibliothèque GeoPandas:

```
@cache.cached(timeout=86400)
```

Ce sont les bibliothèques les plus importantes que nous avons utilisées. Le reste du code est plus détaillé avec les commentaires dans les fichiers source.

Maintenant, pour exécuter le code, il suffit d'exécuter la commande :

flask run

Et le serveur tourne sur le port 5000 par défaut. Maintenant nous pouvons tester la requête http get en utilisant postman ou simplement en essayant le lien

localhost:5000/api/internships sur le navigateur et nous pouvons voir dans la console que les adresses ont commencé à se charger.

```
PS C:\Users\User\Desktop\ISMIN\PI\Github\PI4\BackEnd> flask run
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
Number of addresses: 300
loading addresses: 2%
```

IV. Caméra

Intro

La caméra achetée en début de PI est une Intel RealSens D435. Celle-ci nous sert à capter les gestes de l'utilisateur (grâce au logiciel NuiTrack avec lequel elle est compatible).

Ses limites

Bien qu'elle soit une excellente caméra dans le domaine de la recreation d'environnement 3D, la détection d'objet et de profondeur, cette caméra n'était pas idéale pour le tracking des mains.

En pratique, notre application peine à suivre les mains de l'utilisateur. Bien que fonctionnelle, elle reste frustrante à utiliser. Les curseurs de l'IG peuvent ne pas répondre aux gestes de l'utilisateur, manquer de précision, trembler ou se téléporter sur l'IG.

Nous avons longuement travaillé sur les angles d'approche logiciel qui pourraient augmenter notre stabilisation. Nous avons notamment essayé les choses suivantes :

- Tester différentes valeurs de Min Velocity Interactive Point (un slider associé aux objet Lhand et Rhand dans le projet unity). Nous en avons tiré que deux était la meilleure valeur.
- Dans le scripte C# WebBrowser2D de l'objet Browser2D (dans le projet Unity) nous y trouvons sous le commentaire "//tracker" le code concernant les deux curseurs trackés aux mains de l'utilisateur. Nous avons veillé à ce qu'il n'y ait pas d'interférence entre les deux curseurs lors des différentes actions comme le "drag". En effet on comprend vite qu'en draguant avec la main gauche, l'interface considérait le curseur droit comme le gauche et ne cessait de faire sauter le globe entre les deux positions. Pour éviter les problèmes de la sorte, nous avons assuré que lorsque la main gauche effectue un drag, la main droite soit ignorée. Bien que cette solution ait fluidifié notre IHM, elle n'a réglé qu'un problème logiciel qu'on aurait eu avec n'importe quelle caméra.

Malgré ces différentes tentatives, les curseurs de notre IHM restent insuffisamment stable pour un utilisateur non familiarisé avec elle.

Comme précisé précédemment, le problème révèle plus du hardware que du software. La D435 n'est pas adaptée à la tâche qui lui est demandée.

Cependant, le changement de caméra ne devrait pas affecter le fonctionnement de notre code, il faut cependant que celle-ci soit compatible au SDK de Nuitrack utilisé.

Les bonnes alternatives

La liste des caméras compatibles se trouve sur la page <https://nuitrack.com/>.

Voici celles compatibles qui seraient mieux adaptées à notre application :

- La kinect v2 (kinect de la XBOX ONE)
- Les caméras de la série Astra d'ORBEC

Ces deux caméras sont spécialisées dans le tracking d'objet, notamment des mains.

V. Nuitrack

Intro

Bien qu'on aurait pu se pencher sur la création d'un logiciel capable de capter nos gestes avec notre caméra, ça nous aurait pris beaucoup de temps. Le cahier des charges étant déjà bien rempli, nous devons nous servir de tous les outils à notre disposition.

Nuitrack est un SDK distribué par 3DIVI qui nous a fourni tous les outils nécessaires pour mettre en place une IHM gestuelle.

Licences à acheter

Nous avons travaillé sur notre projet avec la licence gratuite de Nuitrack qui nous procure seulement trois minutes d'activité avec les scripts procurés par le SDK. Afin de développer l'IHM finale, il sera important de prendre la licence professionnelle qui procure un temps d'utilisation illimité. Le [Manuel d'utilisation.pdf](#) explique comment activer la licence.

Le SDK et les aides en lignes

Le SDK auquel on a accès est très riche en contenu. Il est doté de multiples tutoriels et scripts permettant de bien le prendre en main. Il se trouve sur le site une documentation exhaustive. De plus, l'équipe de 3DIVI est très active sur leur GitHub. Nous y avons trouvé un tutoriel qui a servi de fondation pour notre projet Unity. En règle générale, les membres de 3DIVI répondaient à nos problèmes sur GitHub en moins d'une demi-journée ce qui nous a permis de travailler sans jamais rencontrer de gros blocages. Ce fut notamment eux qui nous firent comprendre que notre caméra n'était pas adaptée à notre application.

VI. Projet Unity

Intro

Unity est un outil de jeu 3D/2D et un puissant IDE multiplateforme pour les développeurs. Le logiciel Unity dispose d'un éditeur visuel qui permet aux créateurs de faire simplement glisser et déposer des éléments dans des scènes, puis de manipuler leurs propriétés. Le logiciel Unity offre également une multitude d'autres fonctions et outils utiles :

par exemple, la possibilité de naviguer dans les dossiers du projet ou de créer des animations via un outil de ligne de temps. Lorsqu'il s'agit de coder, Unity bascule vers un éditeur alternatif de votre choix. L'option la plus courante est Visual Studio de Microsoft qui s'intègre en grande partie de manière transparente. Il utilise C#, l'un des langages de programmation les plus conviviaux pour les débutants, pour gérer le code et la logique, avec tout un tas de classes.

Tutoriel 3DIVI (fondation de notre projet)

Dans le package unity SDK que nous avons installé précédemment, il y a un ensemble de tutoriels qui nous ont permis de nous familiariser avec NuiTrack. L'un des tutoriels les plus importants et celui qui nous a servi de base est le [projet de galerie multi-touch](#). Dans ce tutoriel, nous avons appris à créer une galerie virtuelle qui peut être contrôlée par des gestes.

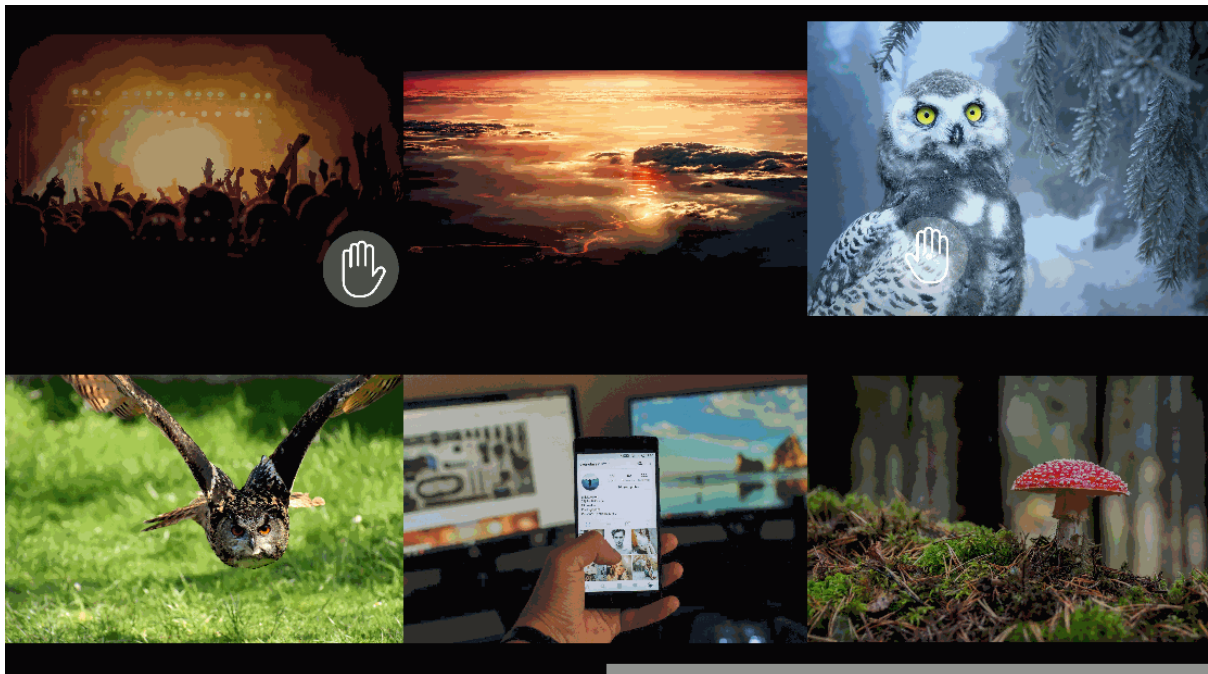


Figure 11 : IG final du tutoriel 3DIVI

La galerie a deux modes :

- Aperçu, plusieurs images affichées sur une page
- Vue, lorsque vous cliquez sur une image, elle s'ouvre en plein écran

La galerie peut être contrôlée avec une ou deux mains (multi-touch) avec les gestes "cliquer", "glisser vers le haut", "glisser vers la gauche", "glisser vers la droite". Nous avons vu que ce tutoriel était idéal comme base de notre projet, nous avons ainsi décidé de développer notre projet sur Unity.

Pour créer ce projet, nous avons eu besoin que de quelques éléments :

- Nuitrack Runtime et Nuitrack SDK
- Tout capteur pris en charge (intel depth camera D435)
- Unity 2017.4 ou supérieur

Notre projet final

Ce que nous avons fait, a été de récupérer ce tutoriel en remplaçant la galerie d'images par le site web que nous avons créé. Nous avons donc intégré avec le plugin unity_browser notre page web dans le projet. Grâce à ce-drenier, nous pouvons utiliser navigateur web à l'intérieur du projet Unity.

Le projet contient maintenant l'objet NuitrackPerfab qui ajoute le SDK Nuitrack dans le code. L'objet caméra principal qui nous permet de changer la vue de la caméra ou l'angle de vue. Pour nous, la caméra fait directement face au navigateur web afin que nous puissions le voir en 2D. L'EventSystem contenant le script HandsInputModule qui reçoit et gère les gestes en utilisant les fonctions du SDK Nuitrack. L'objet Canvas de notre projet Unity contient trois objets :

- RHand et LHand : les objets main droite et main gauche ont chacun deux états. Le premier état est "Hand" représentant une main ouverte et l'autre état est "HandDown" représentant une main fermée.
- Panel : contient l'objet Browser2D responsable de l'affichage de la page webBrowser. Nous avons remplacé l'URL initiale par <http://localhost:8500/welcome>. Ainsi, lorsque nous lançons le projet unity, nous obtenons la page web de bienvenue.

L'objet Browser2D contient un script appelé "Web Browser 2D" qui reçoit les gestes de l'EventSystem et les traduit en simulations de souris permettant à l'utilisateur de contrôler le WebBrowser avec ses mains comme s'il utilisait une souris. Mais bien sûr, c'est différent ici puisque nous avons deux mains qui agissent comme deux curseurs de souris, c'est pourquoi nous avons utilisé la combinaison des mains ouvertes et fermées pour créer ces

simulations de souris. Nous avons donc modifié le script à l'intérieur de l'objet Browser2D pour qu'il effectue les simulations suivantes lorsqu'il reçoit un des gestes suivant :

Geste	Simulation
Main gauche : Ouvert Main droite : Ouvert	Aucun
Main gauche : Ouvert Main droite : Fermé	Click
Main gauche : Fermé Main droite : Ouvert Déplacer la main dans toutes les directions	Drag
Main gauche : Fermé Main droite : Fermé Rapprocher les mains	Zoom in
Main gauche : Fermé Main droite : Fermé Éloigner les mains	Zoom out

Table 2 : Les différents gestes et leur effets

Le résultat de ce projet Unity est un fichier exécutable. Juste en exécutant ce fichier, les serveurs Angular et Python sont en cours d'exécution. Nous pouvons voir notre application démarrer à partir de la page de bienvenue puis nous pouvons utiliser nos mains

comme des simulations de souris pour naviguer sur le site web. Les scripts C# de Unity sont commentés afin que chaque ligne ou chaque groupe de code soit clairement compris.

VII. Lecteur badge

Intro

Notre application possède une fonctionnalité d'identification, elle doit être capable d'identifier le personnel et l'élève de l'école des divers visiteurs qui seront également invités à essayer l'IHM. Pour ce faire, nous utiliserons les badges qui sont fournis aux élèves comme au personnel de l'école. L'IG est constituée d'une page de connexion, elle vous donne le choix entre une accession à notre application en tant que membre de l'ISMIN ou externe. Le choix de membre de l'ISMIN requiert la validation de votre badge pour pouvoir accéder à la suite de l'interface.

Nous avons confectionné un lecteur badge connecté à l'ordinateur qui peut grâce à Arduino et Python récupérer les valeurs hexadécimales des cartes et vérifier leur appartenance à une BDD. Le but étant de bloquer l'accès d'une partie des informations des stages aux personnes extérieures à l'ISMIN.

Deux objets ont été utilisés pour réaliser cette fonctionnalité. Nous nous sommes servi d'une carte ST NUCLEO-32 et d'un module RFID d'Arduino, RFID-RC522.

Lecteur Arduino

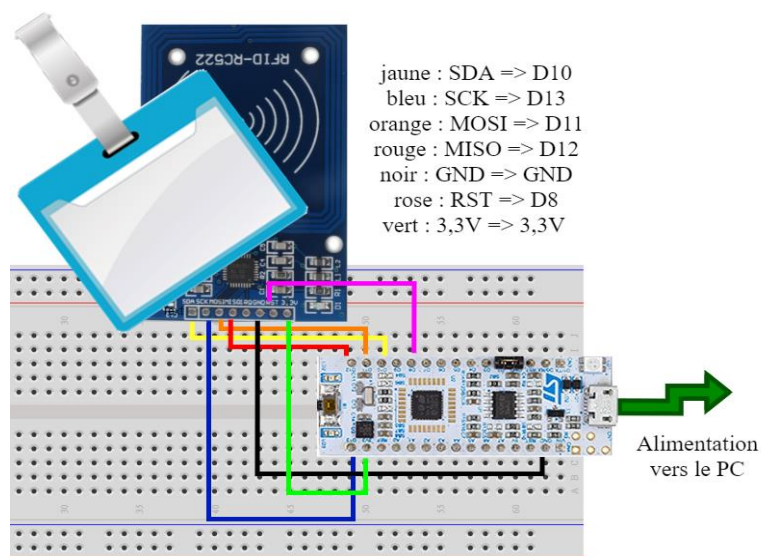


Figure 12 : Schéma de câblage du lecteur badge

Le module RFID-RC522 est utilisé pour écrire ou lire sur des cartes ou des badges utilisant la technologie Mifare. Il utilise une interface de communication Serial Peripheral Interface (SPI), c'est un bus de donnée série synchrone pour communiquer avec l'extérieur, ici, la carte NUCLEO. Le module est alimenté en 3,3 V par la carte.

Nous avons utilisé l'interface Arduino IDE pour commander le module et la carte. Les librairies utilisées sont SPI et RFID, la première pour assurer la communication et la deuxième pour assurer la lecture de la carte RFID. De plus, pour utiliser la carte ST à partir d'Arduino, il nous faut télécharger des gestionnaires JSON afin qu'Arduino connaisse la configuration de la carte NUCLEO-32.

```
if (RFID.isCard()) {  
    if (RFID.readCardSerial()) {  
        digitalWrite(LED, HIGH);  
        for(int i=0;i<=4;i++)  
        {  
            Serial.print(RFID.serNum[i],DEC);  
        }  
    }  
}
```

La première condition "RFID.isCard" vérifie si une carte est à proximité du lecteur de carte et la deuxième condition "RFID.readCard Serial" vérifie si la carte à proximité possède bien un numéro de série. Ensuite, on lit octet par octet les cinq octets de l'identifiant du badge que l'on convertit en décimal puis on affiche le résultat sur le port série.

Script Python

Le but était d'ensuite de vérifier que l'identifiant de la carte appartenait bien aux identifiants répertoriés dans la BDD des cartes de l'école. On va donc utiliser Python pour faire le lien entre le code et la BDD. Pour transmettre les informations de Arduino vers Python, on va utiliser le port série qui nous servait à afficher les résultats sur Arduino. En utilisant la librairie "serial" de Python, on peut se connecter sur le port série et transmettre des informations comme l'identifiant d'une carte. Cet identifiant sera vérifié en envoyant une requête à notre BDD via la librairie "psycopg2". Le paramètre de port de la fonction "serial.Serial" doit correspondre au port sur lequel le périphérique de la carte ST est connecté.

```
arduino = serial.Serial('COM5', 9600, timeout=1)
```

Conclusion

Nous sommes tous les trois fiers du travail réalisé et de notre organisation durant le projet industriel. En effet, nous avons recommencé le projet du départ, la quantité de travail était considérable, néanmoins nous avons pu satisfaire une grande partie des attentes du cahier des charges et de la cliente. Notre projet final est fonctionnel malgré le fait que la caméra ne soit pas adaptée, il peut être utilisé facilement via un ordinateur et sans caméra.

En réalisant ce projet, nous nous sommes rendu compte de la quantité de travail que représente la gestion de projet et la coordination des efforts. Ainsi, ce fut surprenant de se retrouver avec autant de travail technique que logistique.

L'utilisation de l'outil Trello nous a servi considérablement, notamment pour garder une vue d'ensemble sur l'avancée du projet, mais également suivre en détail l'avancé de chaque tâche grâce aux commentaires de chacun. Ainsi après chaque tâche complétée, on pouvait consulter d'abord les objectifs "Pending" afin de voir si un camarade avait besoin d'aide, sinon on se référait à la liste "To Do".

Cet outil n'aurait pas eu la même utilité si nous avions eu des problèmes de communication. Cependant, ce ne fut pas le cas. Profitant de la petite échelle du projet nous ne nous avons pas imposé un rythme de réunion de groupe strict. A la place, vu que nous travaillions la majeure partie du temps en présentiel, nous pouvions rester au courant de ce que chacun faisait avec des échanges brefs et efficaces. De plus, dès que l'un d'entre nous se lançait sur une nouvelle tâche, il l'annonçait aux autres.

En suivant ce mode de travail, nous avons complété environ 75% des exigences du cahier des charges. Compte tenu du contenu de ce dernier, nous sommes très satisfaits par l'état actuel du projet. Le travail restant pour compléter le cahier des charges est d'obtenir un lien direct avec Prométhée, gérer la phase de veille et concevoir un boîtier pour le lecteur de badge.

Nous espérons que ce document en plus des manuels d'utilisation complémentaire sera suffisant pour permettre à tout successeur de prendre en main et finir le projet.

Bibliographie

- Ivan-3DIVI, (2020, 16/10), *Creating an Interactive Multi-Touch Gallery*, GitHub
https://github.com/3DiVi/nuitrack-sdk/blob/master/doc/Unity_Gallery.md
- (2016, 27/04), *RFID-RC522 datasheet* <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>
- Vladimir Kapalrevic, (2020, 19/10), *How to install Angular on Windows*, ccbill
<https://ccbill.com/kb/install-angular-on-windows>