

2020

# Compte rendu projet Projet PSI

**Projet: Application Web permettant de gérer  
des groupes d'individus et de les mettre en  
relation avec des salles de cours.**

git: <https://github.com/rsarujan/PSIProjet.git>



Mai 2019

Enseignant de module: Jean-François PRADAT-PEYRE

**Adja BA & Sarujan RAJARATNAM**

Étudiants en L3 MAGE APP - Promotion 2020

Université Paris Ouest Nanterre La Défense - UFR SEGMI,  
200 Avenue de la République, 92000 Nanterre



## Table des matières

Déroulement du projet .....	4
Fichiers du projet: .....	6
Fonctionnalités prévues mais non implémentées:.....	7
Vue d'ensemble.....	7
Notice d'utilisation:.....	8
Synthèse de ce projet:.....	11
Annexes.....	13

## Déroulement du projet

Ce document a pour but de décrire le déroulement de notre projet de PSI. C'est le résultat du travail qui nous a permis de réaliser une application Web fonctionnelle.

Tout d'abord, on s'est mis à deux comme on travaille tous deux efficacement depuis le début de la Licence MAGE. Afin de travailler rapidement et de manière efficace en dehors des cours malgré la distance et le problème du confinement, nous avons fixé quelques règles pour le bon déroulement du projet.

L'étape suivante était comment réaliser une application Web permettant de gérer des groupes d'individus et de les mettre en relation avec des salles de cours. Au début, nous hésitions longuement par la façon de réaliser ce projet, c'est-à-dire avec quels outils le réaliser, si on utilise ou non un framework permettant de simplifier les tâches, etc... Mais notre choix s'est dirigé vers la création d'une application Web en utilisant un framework PHP MVC qui est un modèle de conception d'application. Ce modèle sépare les données d'application et la logique métier (Model) de la présentation (View). MVC signifie Model, View & Controller. Le framework PHP MVC simplifient le travail en masquant tous les détails d'implémentation, d'utiliser des méthodes standard et développer l'application en veillant à bien utiliser les normes de codage professionnelles. Une fois que nous nous sommes approprié les méthodes pour développer sous le modèle MVC, nous nous sommes donc mis au travail. Nous avons étudié le sujet du projet attentivement et pris des notes sur tout ce que le projet requiert afin de commencer le projet dans les bonnes conditions.

Nous avons établi des méthodes afin de créer un objectif SMART qui nous permettra de s'organiser pour faire le projet et de répartir les tâches. Pour cela, on a établi un cahier des charges en brouillon, où ce cahier des charges reprend le projet pas à pas et définit les objectifs de ce projet et aussi des contraintes qui nous serviront à respecter pas à pas dans le projet.

Le projet consiste à produire une application Web en utilisant les langages de programmation Web tels que le HTML, CSS et le PHP. Cette application Web, permettra aux enseignants et/ou aux administrateurs de suivre un étudiant en fonction d'un groupe et aussi permettra d'attribuer plus tard un groupe d'étudiants à une salle avec un ou plusieurs enseignants.

Nous avons ensuite réfléchi et effectué l'ensemble des diagrammes de cas d'utilisation et de la façon à développer correctement les fonctions demandées. Pour cela nous avons pris des notes sur papier, pour avoir une impression globale de ce projet.

De plus, nous avons écrit les différents problèmes qu'on peut rencontrer tout au long de ce projet. Après avoir bien compris le sujet et bien remarquer les points essentiels de ce projet, nous avons donc commencé à établir le cahier des charges de ce projet. (Cf *Annexe*)

Une fois qu'on a bien défini le but précis de notre projet, nous avons donc commencé à implémenter les fonctionnalités que l'enseignant nous a demandé.

Voici la liste des fonctions à implémenter:

- Fonction permettant de **créer**, de **supprimer** ou de **modifier** un groupe.
- Fonction permettant d'**associer** ou de **supprimer** un individu d'un groupe.
- Fonction permettant d'**importer** des individus à partir d'un fichier csv ou Excel.
- Fonction permettant d'**ajouter** ou de **supprimer** un individu à l'aide d'un formulaire.
- Fonction permettant de **modifier** les caractéristiques d'un individu.
- Fonction permettant d'**exporter** les individus d'un groupe au format csv ou Excel.
- Fonction permettant de **fournir une API** une liste de groupes au format csv ou Excel.

Une fois qu'on a su comment implémenter ces fonctions, on a divisé les tâches en deux pour que chacun d'entre nous puisse avoir une partie à coder.

Pour cela, Adja a codé ces fonctions:

- Fonction permettant de **créer**, de **supprimer** ou de **modifier** un groupe.
- Fonction permettant de **modifier** les caractéristiques d'un individu.
- Fonction permettant de **fournir une API** une liste de groupes au format csv ou Excel.

Sarujan a codé ces fonctions:

- Fonction permettant d'**associer** ou de **supprimer** un individu d'un groupe.
- Fonction permettant d'**importer** des individus à partir d'un fichier csv ou Excel.
- Fonction permettant d'**ajouter** ou de **supprimer** un individu à l'aide d'un formulaire.
- Fonction permettant d'**exporter** les individus d'un groupe au format csv ou Excel.

Il est arrivé que dans des étapes de développement du code, on était bloqué car on ne savait pas comment procéder pour réaliser le développement. Donc à chaque fois, on s'entraider afin de corriger les bugs dus à l'affichage ainsi que les problèmes de développements.

## Fichiers du projet:

Nous avons mis en place le modèle MVC de PHP. Ce modèle MVC contient plusieurs répertoires tels que le Controller, Model, View et bien d'autres.

Le répertoire View contient toutes les pages de présentation des données à l'utilisateur, c'est-à-dire, il contient toutes les pages HTML sous l'extension de PHP. Chaque fichier présent dans le répertoire View est généralement lié à un controller.

Un Controller est la partie qui traite les demandes de ressources grâce à l'utilisation du serveur. Le controller charge le modèle de produits pour récupérer les données, puis affiche les résultats dans la liste. C'est pour cette raison que le Controller a un répertoire propre à lui permettant de relier la View à un Model. Ce répertoire contient 4 fichiers dont une est obligatoire pour qu'elle puisse fonctionner. Le controller Controller\_home va permettre d'afficher la page d'accueil en cas de problème. Le controller Controller\_list va permettre d'afficher une liste de données grâce au Model et à la View en question. Le controller Controller\_set va permettre d'insérer une liste de données dans la base de données grâce au Model et à la View en question.

Un Model est une partie qui concerne la logique du métier et les données d'application. Il peut être utilisé pour valider les données, traiter des données et les stocker dans une base de données. Ce répertoire va permettre d'effectuer des échanges avec la base de données où on met uniquement des requêtes SQL.

Le répertoire Content contient toute la mise en page des pages de présentation telle que les fichiers css, les fonts et les images.

Le répertoire Utils va permettre de convertir un caractère en entité HTML.

On peut aussi voir qu'il y a une page index.php dans le répertoire MVC. Ce fichier est le « chef d'orchestre » de tout le projet qui permet d'indiquer le chemin pour aller vers le Controller, Utils ou Model.

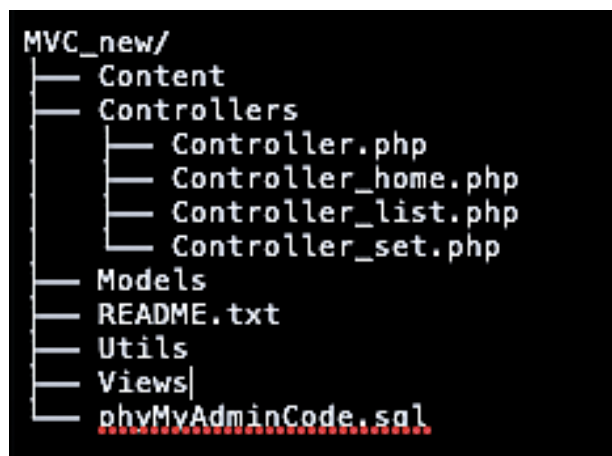


Figure n°1: Arborescence de notre dossier de projet

## Fonctionnalités prévues mais non implémentées:

Si on avait un peu plus de temps et plus d'organisations, on aurait pu mettre une barre de recherche permettant de trier les individus avec un ou plusieurs critères donnés en paramètre et bien d'autres fonctions...

## Vue d'ensemble

La répartition des tâches n'était pas si compliquée que cela en a l'air car chacun savait ce qu'il devait faire, ce qui nous a permis d'éviter une certaine ambiguïté. Cela nous a permis d'éviter à choisir un chef de projet pour ce projet car chacun d'entre nous était le maître de ses fonctions mais aussi du temps qu'il voulait utiliser pour la réalisation de ce projet.

Cependant, on a eu pas mal de problèmes durant le début du développement de ce projet. La principale raison, c'est qu'on était en manque d'organisation notamment à cause de la distance, des différents projets que nous avons à effectuer. ainsi que des différents examens. Nous faisons par ailleurs très fréquemment, de points téléphoniques ce qui nous permettaient de suivre l'avancement du projet, de discuter des différents problèmes que nous rencontrons de notre côté et d'essayer de trouver ensemble des solutions.

Nous communiquons par téléphone la plupart du temps, utilisons Discord pour les échanges et GitHub pour la mise à jour du code.

## Notice d'utilisation:

Lien git: <https://github.com/rsarujan/PSIProjet.git>

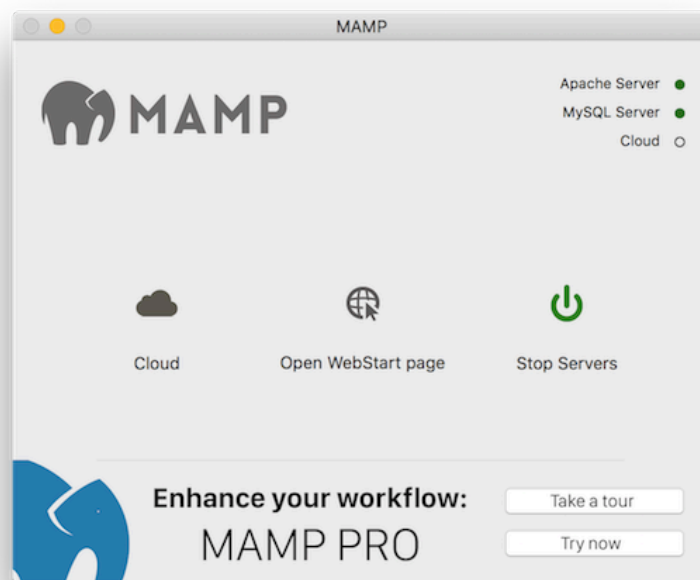
### 1ère étape : Installation des modules PHP ainsi que de MySQL sur la machine.

#### MacOS:

Tout d'abord, il faudrait installer un logiciel permettant de lancer une application Web sur une machine qui est un serveur local. Pour cela, un logiciel qui est très simple à installer sur MacOS est le logiciel **MAMP**. Pour installer le serveur local *MAMP*, il faut aller sur ce lien de téléchargement: <https://www.mamp.info/en/mamp/mac/> .

Après avoir installé correctement, vous pouvez démarrer MAMP et cliquez sur le bouton Démarrer les serveurs. Dans l'affichage d'état dans le coin supérieur droit, l'état de lancement des serveurs est affiché. Si nécessaire, il vous sera demandé votre mot de passe administrateur qui sera dans la plupart du temps *root* ou sinon le mot de passe de votre session exemple: *PN2019miage*.

Le serveur Web (Apache) démarre par défaut sur le port 8888, le serveur de base de données (MySQL) sur le port 8889. Lorsque vous appelez votre page Web dans un navigateur Web (n'importe laquelle), vous devez entrer le port Apache à la fin de l'URL, par exemple: <http://localhost:8888/>



- **Ouvrir la page WebStart**

Permet d'ouvrir la page de démarrage de votre serveur Web local.

- **Démarrer les serveurs / Arrêter les serveurs**

Permet de démarrez ou arrêtez les services Apache / Nginx et MySQL de MAMP.



Si le port 8888 est déjà utilisé par un autre serveur, il suffit d'aller dans *MAMP, Préférences...*, *Ports* et de modifier le numéro de port comme il vous le semble.

Dans l'onglet, PHP, vérifier si la version de PHP choisi est bien le 7.2.8.

Une fois les changements effectués, vous pouvez les sauvegarder en cliquant le bouton **OK**.

### Windows:

Tout d'abord, il faudrait installer un logiciel permettant de lancer une application Web sur une machine qui est un serveur local. Pour cela, un logiciel qui est très simple à installer sur Windows est le logiciel **EasyPHP**. Pour installer le serveur local *EasyPHP*, il faut aller sur ce lien de téléchargement: [https://fr.osdn.net/projects/sfnet\\_quickeasyphp/downloads/OldFiles/EasyPHP-DevServer-OBSOLETE/EasyPHP-DevServer-14.1VC11-install.exe/](https://fr.osdn.net/projects/sfnet_quickeasyphp/downloads/OldFiles/EasyPHP-DevServer-OBSOLETE/EasyPHP-DevServer-14.1VC11-install.exe/).

Après avoir installé correctement, vous pouvez démarrer EasyPHP en allant dans C:\Program Files(x86)\EasyPHP-DevServer-14.1VC11\ et lancer *EasyPHP-DevServer-14.1VC11.exe*. Une fois lancé le logiciel, il suffit d'aller sur la barre de tâche en bas à droite de l'écran et cliquez sur le bouton flèche et faire un clic droit sur le logo de EasyPHP serveur et choisir *Web local*, cela permettrait d'aller sur la page de Web en local qui est <http://127.0.0.1/>.

## 2ème étape : Télécharger le dossier en question et le mettre dans notre serveur local.

### MacOS:

Récupérer le dossier déposer sur la plateforme GitHub et le mettre dans /Applications/MAMP/htdocs/. Une fois déposé, il suffit d'aller sur <http://localhost:8888/>, vous verrez le dossier s'apparaître. Une fois que le dossier s'est apparu, il faut tout d'abord configurer la base de données dans phpMyAdmin et de configurer les accès dans le fichier *Model.php* qui se trouve dans le répertoire du projet *AdjaBA\_SarujanRAJARATNAM/MVC\_new/Model*.

Ouvrir phpMyAdmin sur ce lien: <http://localhost:8888/phpMyAdmin/?lang=en> et copier le script .sql permettant de créer les tables qui se trouve dans le répertoire *MVC\_new* sous le nom de *phyMyAdminCode.sql*

Une fois importé les tables, vérifié si les tables ont bien été importé. Puis ouvrir le fichier *Model.php* qui se trouve dans le répertoire *AdjaBA\_SarujanRAJARATNAM/MVC\_new/Model*.

À la ligne 14 du fichier *Model.php*, vous trouverez ceci:

```
$this->db = new PDO('mysql:host=localhost; dbname=PSI', 'root', 'root');
```

À cet endroit, il faut modifier le '**root**' et '**root**'. En effet, il faudra changer ces deux éléments en fonctions de vos identifiants présent sur phpMyAdmin '**root**' est l'utilisateur de phpMyAdmin et '**root**' est le password de phpMyAdmin.

Une fois effectué toutes ces étapes de configuration, vous pouvez aller sur ce lien: [http://localhost:8888/AdjaBA\\_SarujanRAJARATNAM/MVC\\_new](http://localhost:8888/AdjaBA_SarujanRAJARATNAM/MVC_new) et parcourir le projet comme une application Web.

### Windows:

Récupérer le dossier déposer sur la plateforme GitHub et le mettre dans C:\Program Files (x86)\EasyPHP-DevServer-14.1VC11\data\localweb\my portable files\ . Une fois déposé, il suffit d'aller sur <http://127.0.0.1/my%20portable%20files/>, vous verrez le dossier s'apparaître. Une fois que le dossier s'est apparu, il faut tout d'abord configurer la base de données dans phpMyAdmin et de configurer les accès dans le fichier Model.php qui se trouve dans le répertoire du projet *AdjaBA\_SarujanRAJARATNAM/MVC\_new/Model*.

Ouvrir phpMyAdmin sur ce lien: <http://127.0.0.1/home/index.php> et cliquer sur phpMyAdmin. Une fois arrivé sur phpMyAdmin, copier le script .sql permettant de créer les tables qui se trouve dans le répertoire *MVC\_new* sous le nom de *phyMyAdminCode.sql*. Une fois importé les tables, vérifié si les tables ont bien été importé. Puis ouvrir le fichier *Model.php* qui se trouve dans le répertoire *AdjaBA\_SarujanRAJARATNAM/MVC\_new/Model*.

À la ligne 14 du fichier Model.php, vous trouverez ceci:

```
$this->db = new PDO('mysql:host=localhost; dbname=PSI', 'root', 'root');
```

À cet endroit, il faut modifier le **localhost**, '**root**' et '**root**'. En effet, il faudra changer ces trois éléments en fonctions de vos identifiants présent sur phpMyAdmin **localhost** faut le remplacer par **127.0.0.1**, '**root**' est l'utilisateur de phpMyAdmin et '**root**' est le password qu'on le mettra vide comme on a pas besoin de password pour se connecter à phpMyAdmin.

Une fois effectué toutes ces étapes de configuration, vous pouvez aller sur ce lien: [http://127.0.0.1/my%20portable%20files/AdjaBA\\_SarujanRAJARATNAM/MVC\\_new/](http://127.0.0.1/my%20portable%20files/AdjaBA_SarujanRAJARATNAM/MVC_new/) et parcourir le projet comme une application Web.

## Synthèse de ce projet:

Tout d'abord, nous souhaitons remercier M. PRADAT-PEYRE qui nous a accompagné tout au long de ce projet. Étant donné qu'aucun d'entre nous n'avait pas de base en MVC cela fut une bonne méthode de pratique.

### **Sarujan:**

Le plus difficile dans ce projet, c'était la gestion du temps à consacrer pour chaque tâche. Il y a eu des tâches qui me paraissaient simples lors de la phase de conception mais qui ont pris beaucoup plus de temps lors du développement.

Ce projet m'a aussi appris de nouvelles techniques pour programmer en PHP.

Contrairement aux TP encadrés en cours, qui sont souvent plus courts et moins difficiles, ce projet est nettement plus élaboré et de ce fait, j'ai eu beaucoup de mal à débiter comme on n'a jamais eu un projet assez vague. Ce projet a été une nouvelle expérience. Il y a beaucoup de choses lorsqu'on réfléchit, il peut paraître simple mais lors de la réalisation pouvant être complexes.

Ce que je changerai, si je refais ce projet c'est d'assurer d'avoir un maximum d'informations sur le projet et de bien s'informer par où commencer tout en assurant une gestion de projets assez organisée.

### **Adja:**

Ce projet a été bénéfique pour moi car j'ai pu approfondir mes connaissances en PHP mais surtout de consolider mes lacunes sur certaines notions.

Nous avons su travailler en équipe malgré la distance afin de pouvoir mener à bien le projet.

Au début du projet, la plupart des problèmes rencontrés étaient dus au fait que nous n'avions jamais auparavant travaillé avec un modèle MVC. Il nous a donc fallu du temps pour pouvoir comprendre comment ça marche avant de pouvoir commencer le projet comme il le faut.

Au niveau de la gestion du projet en équipe, nous avons réussi à bien nous répartir les tâches afin de réaliser nos objectifs dans les temps, ce qui a permis de rattraper le retard pris auparavant.

Pour conclure, ce projet m'a permis de travailler à la fois le côté technique en PHP mais également la partie sur la gestion d'un projet, ce qui est tout à fait adéquat avec les matières enseignées en MAGE.



## Annexes

Annexe 1: Cahier des charges .....	14
Annexe 2: Capture d'écran de la page d'accueil.....	16
Annexe 3: Capture d'écran de la page Individus.....	17
Annexe 4: Capture d'écran de la page ajouter un individu.....	18
Annexe 5: Capture d'écran de la page Groupes .....	18
Annexe 6: Capture d'écran de la page mise à jour du nom de groupe.....	19
Annexe 7: Capture d'écran de la page d'attribution groupes .....	19
Annexe 8: Capture d'écran de d'attribution d'un individu à un groupe.....	20
Annexe 9: Extrait de l'API en JSON.....	20

## Annexe 1: Cahier des charges

### CAHIER DES CHARGES - PROJET Développement d'une Application Web permettant de gérer des groupes d'individus et de les mettre en relation avec des salles de cours

#### • Environnement

*Il s'agit d'expliquer ici quel est l'environnement du projet : entreprise, association ... en détaillant l'organisation, le « client ».*

Dans le cadre de Licence 3 MIAAGE, nous devons réaliser un projet informatique afin de valider le module Projet PSI qui est un module compris dans le module Système d'Information dans le langage de programmation qui est le PHP.

#### • Interlocuteur(s)

*Nomment les personnes chez le « client » (fonctions, rôles) qui pourront apporter des précisions sur le cahier des charges.*

*« Client » - Demandeur = rédacteur du CdC, mais il peut se faire assister, notamment fréquemment par le chef de projet.*

Enseignant du module

#### • Contexte

*Il s'agit d'expliquer ici les raisons de ce projet, pour qui et pourquoi ce projet doit être développé.*

*Présenter de façon générale et courte le contexte du projet*

*Expliquer, positionner, si possible, le niveau de complexité, difficulté.*

Nous devons réaliser une application Web, permettant de gérer des groupes d'individus et de les mettre en relation avec des salles de cours. Pour réaliser cette application Web, nous utiliserons les données que notre enseignant nous fournira afin de peupler la base de données.

Le projet a pour but de valider le module Projet PSI mais aussi pour mettre en évidence nos compétences informatiques dans le domaine du Web notamment dans le langage de programmation qui est le PHP.

En revanche, certaines difficultés sont supposées par rapport aux données fournies avec le sujet.

Les fonctions à développer sont:

- Créer / supprimer / modifier un groupe
- Associer des individus à un groupe / supprimer un individu d'un groupe
- Importer des individus à partir de fichiers csv ou Excel
- Ajouter / supprimer à l'aide d'un formulaire un individu
- Modifier les caractéristiques d'un individu
- Exporter sous format csv ou Excel les individus d'un groupe
- Fournir une API permettant de fournir sous format JSON ou XML la liste des individus d'un groupe (lien avec d'autres applications comme la gestion des émargements)

La difficulté apparente : importer un individu dans la base de données tout en respectant le format du fichier et de fournir une API sous JSON ou XML.

#### • Exigences fonctionnelles

*Préciser les fonctions attendues (création, consultation, calculs ...) à partir des besoins exprimés par des phrases courtes, claires et non ambiguës. L'objectif est que les étudiants comprennent ce qu'il y a à développer et se fassent une idée de l'ampleur du projet.*

*Préciser si une connaissance particulière du domaine fonctionnelle est nécessaire.*

- **Exigences NON fonctionnelles**

- **Technique**

*Préciser l'environnement technique : O.S., langages de développement, navigateur(s), SGBD, CMS, progiciel ou composant à (ré)utiliser ...*

Os: Toutes plateformes utilisant un navigateur web

Langage de programmation: PHP, HTML, CSS

- **Performance**

*Préciser, si nécessaire, les contraintes de performances telles que : taux occupation UC, temps de réponse*

Temps de réponse rapide avec chargement de données.

- **Ergonomie**

*Préciser, si nécessaire, les contraintes d'ergonomie en fonction des utilisateurs, de l'interactivité attendue (incorporation de vidéos, images ...)*

Une interface qui soit simple et agréable à l'utilisateur.

- **Volumétrie**

*Préciser si nécessaire, la volumétrie en terme de données (Go, nb. Données, nb. Tables ....), en terme de nombre d'utilisateurs (au total et en simultané) ...*

Tout dépendra des données fournies par l'enseignant.

- **Sécurité**

*Préciser, si nécessaire, les contraintes de sécurité – exemple : paiement en ligne - en fonction de l'utilisation et du niveau de confidentialité des données (données soumises à déclaration CNIIL ...)*

- **Développement durable**

*Préciser, si nécessaire, les contraintes de respect de l'environnement, tant sur le plan éthique que écologique et économique. En informatique, le « green IT » est appelé à se développer considérablement*

Green IT (le code sera annoté avec des commentaires).

- **Autres**

*Préciser, si nécessaire, les autres contraintes telles que : organisation, lieu géographique d'implantation, d'utilisation, commercialisation envisagée (pb. Droit propriété ...)*

- **Risque(s) à ne pas faire le projet**

*Si l'on ne faisait pas ce projet que se passerait-il ? Tous les projets doivent se justifier par un « apport » économique ou non pour l'entreprise*

Il n'y aura pas de données, pas de base de données, pas d'utilisation de l'application Web.

- **Principaux jalons**

*S'ils sont connus, indiquer ici les principales dates clés, souvent date de fin ! – Exemple : date de mise en place d'une réglementation*

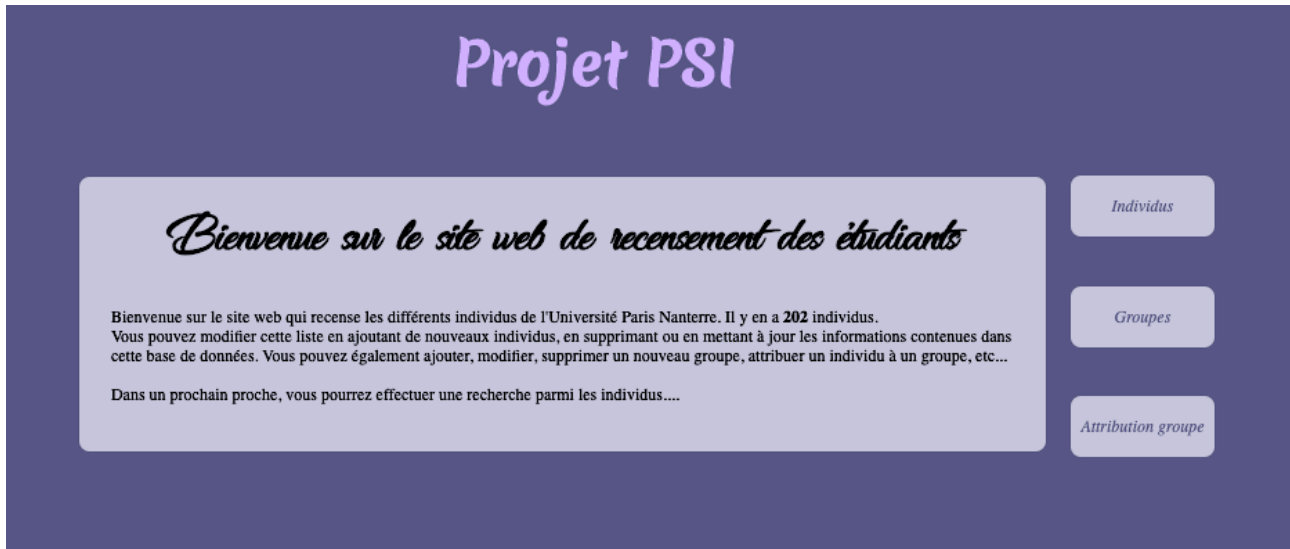
mai 2020 : rendu final

- **Limites**

*Préciser, les limites du projet, ce qui n'est PAS compris dans le projet pour éviter toute ambiguïté*

Le temps est très limité. Mais aussi il y aura des difficultés liées à la programmation comme nous deux nous ne sommes pas opérationnels à 100% dans le langage PHP où chacun apprend petit à petit.

## Annexe 2: Capture d'écran de la page d'accueil





## Annexe 3: Capture d'écran de la page Individus

# Projet PSI

## Liste des individus

[Ajouter un individu](#)

[Importer la liste des individus](#)

[Exporter la liste des individus](#)

Nom	Prenom	Email	Numero		
Ahmed	Noufeine	Noufeine.Ahmed@parisnante.fr	340001	✗	☑
Ait akli	Litissia	Litissia.Ait akli@parisnante.fr	340002	✗	☑
Ba	Adja	Adja.Ba@parisnante.fr	340003	✗	☑
Binous	Wassim	Wassim.Binous@parisnante.fr	340004	✗	☑
Bocoum	Idy	Idy.Bocoum@parisnante.fr	340005	✗	☑
Brochado	Alexandre	Alexandre.Brochado@parisnante.fr	340006	✗	☑
Clebien	Nanginta	Nanginta.Clebien@parisnante.fr	340007	✗	☑
Das	Rahul	Rahul.Das@parisnante.fr	340008	✗	☑
Elarj	Aniss	Aniss.Elarj@parisnante.fr	340009	✗	☑
Fall	Seynabou	Seynabou.Fall@parisnante.fr	340010	✗	☑
Jestin	Gabriel	Gabriel.Jestin@parisnante.fr	340011	✗	☑
Keloute ndamukong	Ubal	Ubal.Keloute ndamukong@parisnante.fr	340012	✗	☑
Khalif	Sofian	Sofian.Khalif@parisnante.fr	340013	✗	☑
Le men	Yann	Yann.Le men@parisnante.fr	340014	✗	☑
Mboup	Gaye	Gaye.Mboup@parisnante.fr	340015	✗	☑
Mouzouri	Ilhame	Ilhame.Mouzouri@parisnante.fr	340016	✗	☑
Ndiaye	Moussa	Moussa.Ndiaye@parisnante.fr	340017	✗	☑
Quelleg	Nathan	Nathan.Quelleg@parisnante.fr	340018	✗	☑
Rajaratnam	Sarujan	Sarujan.Rajaratnam@parisnante.fr	340019	✗	☑
Raypoulet	Hemanath	Hemanath.Raypoulet@parisnante.fr	340020	✗	☑
Sakho	Assane	Assane.Sakho@parisnante.fr	340021	✗	☑
Schauffler	Ophelie	Ophelie.Schauffler@parisnante.fr	340022	✗	☑
Si-mohammed	Samy	Samy.Si-mohammed@parisnante.fr	340023	✗	☑
Sidate	Alexis	Alexis.Sidate@parisnante.fr	340024	✗	☑
Zemali	Lynda	Lynda.Zemali@parisnante.fr	340025	✗	☑

« 1 2 3 4 5 »

Individus

Groupes

Attribution groupe

## Annexe 4: Capture d'écran de la page ajouter un individu

**Projet PSI**

*Ajouter un individu*

Nom:

Prenom:

Email:

Numéro national:

Type d'annuaire:

Type de statut:

## Annexe 5: Capture d'écran de la page Groupes

**Projet PSI**

*Liste des groupes*

<a href="#">Ajouter un groupe</a>	Libelle		
	L3 MIAGE APP	✗	<input checked="" type="checkbox"/>
	M1 MIAGE APP	✗	<input checked="" type="checkbox"/>
	M2 MIAGE APP	✗	<input checked="" type="checkbox"/>
	L2 MIASHS	✗	<input checked="" type="checkbox"/>
	Encadrant	✗	<input checked="" type="checkbox"/>

## Annexe 6: Capture d'écran de la page mise à jour du nom de groupe

**Projet PSI**

Nom du groupe:  
L3 MIAGE APP

Mise à jour Groupe

Individus

Groupes

Attribution groupe

## Annexe 7: Capture d'écran de la page d'attribution groupes

**Projet PSI**

*Liste des groupes attribués à chaque individu*

Exporter:	Attribuer un individu à un groupe	Afficher l'API
--Choisir une option-- Exporter		
Annee	Groupe	Individu
2020	L3 MIAGE APP	<a href="#">Noufeine</a>
2020	L3 MIAGE APP	<a href="#">Litissia</a>
2020	L3 MIAGE APP	<a href="#">Adja</a>
2020	L3 MIAGE APP	<a href="#">Wassim</a>
2020	L3 MIAGE APP	<a href="#">Idy</a>
2020	L3 MIAGE APP	<a href="#">Alexandre</a>
2020	L3 MIAGE APP	<a href="#">Nangninta</a>
2020	L3 MIAGE APP	<a href="#">Rahul</a>
2020	L3 MIAGE APP	<a href="#">Aniss</a>
2020	L3 MIAGE APP	<a href="#">Seynabou</a>
2020	L3 MIAGE APP	<a href="#">Gabriel</a>
2020	L3 MIAGE APP	<a href="#">Ubald</a>

Individus

Groupes

Attribution groupe

## Annexe 8: Capture d'écran de d'attribution d'un individu à un groupe

**Projet PSI**

*Attribution d'un individu à un groupe*

Annee:

Groupe:

Individu:

Individus

Groupes

Attribution groupe

## Annexe 9: Extrait de l'API en JSON

```
[{"libelle": "ETU", "nom": "Ahmed", "prenom": "Noufeine", "email": "Noufeine.Ahmed@parisnante.fr", "num": "340001"}, {"libelle": "ETU", "nom": "Ait akli", "prenom": "Litissia", "email": "Litissia.Ait akli@parisnante.fr", "num": "340002"}, {"libelle": "ETU", "nom": "Ba", "prenom": "Adja", "email": "Adja.Ba@parisnante.fr", "num": "340003"}, {"libelle": "ETU", "nom": "Binous", "prenom": "Wassim", "email": "Wassim.Binous@parisnante.fr", "num": "340004"}, {"libelle": "ETU", "nom": "Bocoum", "prenom": "Idy", "email": "Idy.Bocoum@parisnante.fr", "num": "340005"}, {"libelle": "ETU", "nom": "Brochado", "prenom": "Alexandre", "email": "Alexandre.Brochado@parisnante.fr", "num": "340006"}, {"libelle": "ETU", "nom": "Clebien", "prenom": "Nangninta", "email": "Nangninta.Clebien@parisnante.fr", "num": "340007"}, {"libelle": "ETU", "nom": "Das", "prenom": "Rahul", "email": "Rahul.Das@parisnante.fr", "num": "340008"}, {"libelle": "ETU", "nom": "Elarj", "prenom": "Aniss", "email": "Aniss.Elarj@parisnante.fr", "num": "340009"}, {"libelle": "ETU", "nom": "Fall", "prenom": "Seynabou", "email": "Seynabou.Fall@parisnante.fr", "num": "340010"}, {"libelle": "ETU", "nom": "Jestin", "prenom": "Gabriel", "email": "Gabriel.Jestin@parisnante.fr", "num": "340011"}, {"libelle": "ETU", "nom": "Keloute ndamukong", "prenom": "Ubald", "email": "Ubald.Keloute ndamukong@parisnante.fr", "num": "340012"}, {"libelle": "ETU", "nom": "Khalfi", "prenom": "Sofian", "email": "Sofian.Khalfi@parisnante.fr", "num": "340013"}]
```



