

## Contexte SanayaBio

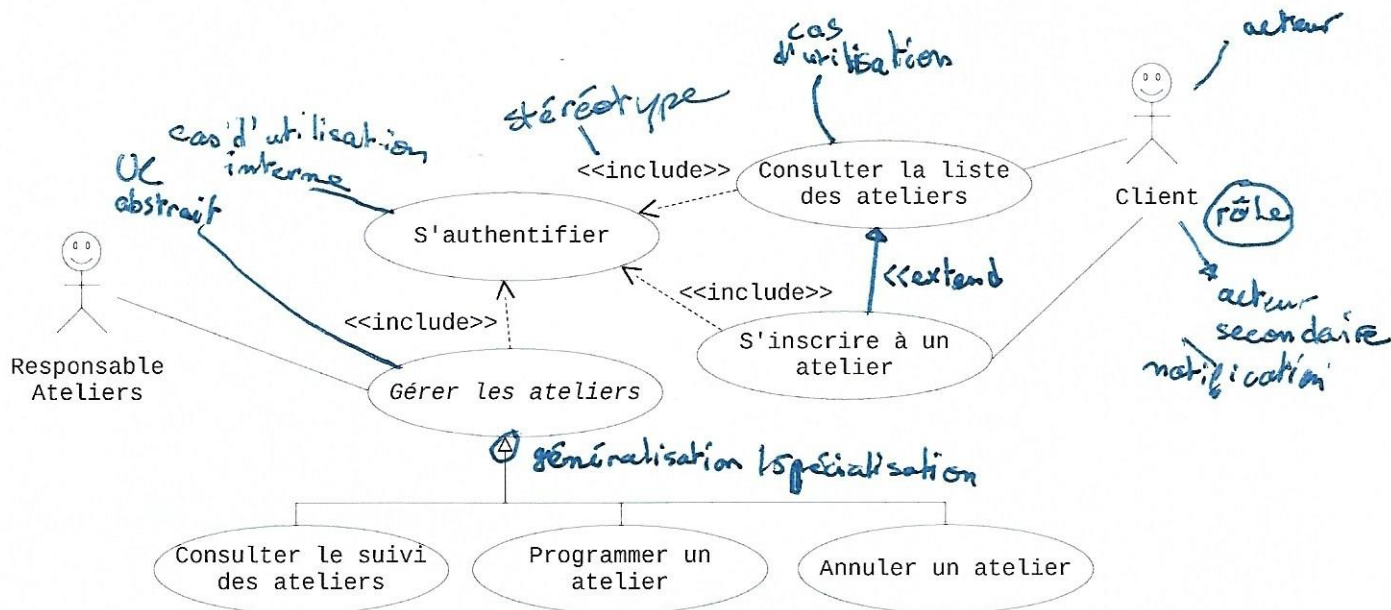
La société *Sanaya Bio* propose des ateliers créatifs (préparation de produits cosmétiques, de produits ménagers...) à ses clients. Ces ateliers se déroulent dans des espaces d'apprentissage et de découverte aménagés dans les points de vente de l'enseigne. Ils sont encadrés par des animateurs spécialisés qui peuvent être des employés ou des intervenants extérieurs.

Une première application web nommée **suivateliers** permet aux responsables d'ateliers de programmer et d'annuler des ateliers ainsi que d'en assurer le suivi.

Une seconde application web nommée **sbateliers**, dans sa version minimale, permet aux clients enregistrés de consulter la liste des ateliers programmés et s'ils le souhaitent, de procéder à leur inscription pour y participer.

### 1- Besoins fonctionnels

Extrait du diagramme des cas d'utilisation



#### Authentifications

- L'authentification d'un client se fait au moyen de son adresse électronique et de son mot de passe.
- L'authentification d'un responsable d'ateliers se fait au moyen de son nom de connexion interne à la société et de son mot de passe.

#### Travail à faire

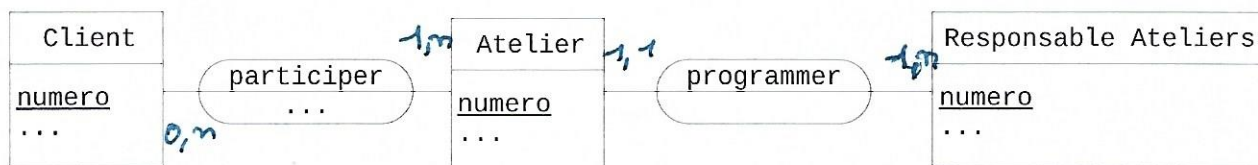
- 1 Compléter le diagramme des cas d'utilisation dans le but de faire apparaître les fonctionnalités énoncées ci-dessous :

- Un client peut consulter son profil ;
- Un client peut annuler sa participation à un atelier créatif programmé ;
- Un visiteur peut s'inscrire pour être recensé en tant que client de l'enseigne.

## 2- Base de données

La base de données nommée *sbateliers* est hébergée par un serveur Linux sur lequel est installé le SGBDR *MariaDB*. Seul le compte *MariaDB* *sanayabio/Sb2022* peut exploiter cette base de données.

Schéma de données (Modèle conceptuel - Modèle Entité-Association) épuré



Données mémorisées

<b>Client</b>	numéro civilité nom prénom date de naissance adresse électronique mot de passe adresse postale code postal ville numéro de mobile
<b>Responsable d'ateliers</b>	numéro nom de connexion nom prénom
<b>Atelier</b>	numéro date d'enregistrement date et heure programmées durée nombre de places thème
<b>Participation</b>	date d'inscription

### Travail à faire

- Compléter le schéma de données.
- Créer le compte *MariaDB* *sanayabio*.
- Créer et peupler la base de données *sbateliers*.
- Écrire et tester les requêtes SQL énoncées ci-dessous : <https://github.com/dev-HR/sbateliers>
  - Vous inscrire en tant que client ; *Insert into client values (2, "Mehdadi", ...);*
  - Afficher tous les clients ; *Select \* from client;*
  - Afficher tous les ateliers passés ; *Select \* from atelier where date\_heure < NOW();*
  - Afficher tous les ateliers du jour ; *Select \* from atelier where select DATE (date\_heure) =*
  - Afficher tous les ateliers programmés ; *Select \* from atelier;*
  - Afficher le nombre de participations à des ateliers de chaque client ; *Select client, count(\*) as*
  - Enregistrer votre participation à deux ateliers programmés ; *participer group by client, from*
  - Afficher les ateliers programmés auxquels vous êtes inscrit-e ;
  - Annuler l'une de vos participations à l'un des ateliers programmés.

## 3- Architecture des applications *sbateliers* et *suivateliers*

L'application web *sbateliers* et *suivateliers* sont écrites en PHP et sont hébergées par un serveur Linux qui dispose du service HTTP Apache 2.

URL de l'application *sbateliers* : <http://domaine/sbateliers/>

URL de l'application *suivateliers* : <http://domaine/suivateliers/>



Ces applications s'appuient sur le **modèle de conception MVC 2 (Modèle-Vue-Contrôleur)**. Le contrôleur frontal (fichier `index.php`) a pour rôle de réorienter toute requête vers un contrôleur secondaire en charge du traitement de celle-ci (**routing**). Le contrôleur secondaire exécute alors le traitement puis renvoie la vue associée au résultat du traitement.

**Exemple :** Application web *sbateliers*

**Organisation des fichiers (dossier `/var/www/html/sbateliers`) :**

```

├── index.php
├── controleurs
│   ├── ctrl-connecter.php
│   ├── ctrl-consulter-ateliers.php
│   ├── ...
│   └── ctrl-deconnecter.php
├── vues
│   ├── vue-connexion.php
│   ├── vue-liste-ateliers.php
│   └── ...

```

### 3.1- Routes

La route `/` correspond à la page d'accueil (`vue-accueil.php`) qui propose un menu constitué de deux items qui permettent au visiteur de s'inscrire ou de s'authentifier s'il possède déjà un compte client.

La route `/clients/enregistrement` correspond à la page d'inscription d'un nouveau client qui se présente sous la forme d'un formulaire. L'enregistrement du nouveau client est réalisée par le contrôleur `ctrl-enregistrer-client.php` (route `/clients/enregistrer`). Une fois l'enregistrement réalisé, le visiteur est redirigé vers la page d'accueil.

La route `/connexion` correspond à la page d'authentification du client qui se présente sous la forme d'un formulaire (`vue-connexion.php`). Cette authentification est réalisée par le contrôleur `ctrl-connecter.php` qui crée une nouvelle **session**, puis qui redirige le client vers son espace personnel. Cet espace personnel propose un menu constitué de deux items qui permettent au client de consulter son profil et la liste des ateliers créatifs programmés (route `/ateliers/programmes`). Si l'authentification échoue, celui-ci est redirigé sur la page d'accueil.

La page dédiée à l'affichage des ateliers créatifs programmés offre la possibilité au client authentifié de s'inscrire à un atelier créatif (route `/ateliers/inscription/numeroAtelier`). L'inscription effectuée, le client est automatiquement redirigé sur la liste des ateliers créatifs programmés.

Les ateliers auxquels est inscrit le client authentifié sont mis en exergue. Celui-ci a la possibilité d'annuler sa participation à un atelier (route `/ateliers/annulation/numeroAtelier`).

La route `/deconnexion` permet au client de mettre fin à session. Il est alors automatiquement redirigé vers la page d'accueil.

### 3.2- Routage

L'objectif est de rediriger toutes les requêtes vers le contrôleur frontal (fichier `index.html`). La vidéo de Boris (PrimFX) intitulée "*Tuto PHP – Créer un routeur*" qui traite de ce sujet présente une solution viable qu'il est possible d'implémenter rapidement.

Cependant, la configuration du service Apache n'y est pas abordée. Il faut commencer par autoriser la réécriture des URL (fichier `/etc/apache2/apache2.conf`) puis activer le module (module `rewrite`) qui permet de le faire :

Autoriser - Fichier <code>/etc/apache2/apache2.conf</code>	Activer
<pre> &lt;Directory /var/www&gt;     ...     AllowOverride None All     ... &lt;/Directory&gt; </pre>	<pre> # a2enmod rewrite # systemctl restart apache2 </pre>

**Travail à faire**

- |   |  |
|---|--|
| 6 | Appliquer, à la configuration du service Apache, les modifications qui sont nécessaires à la réécriture des URL.   |
| 7 | Créer l'arborescence de l'application <i>web sbateliers</i> (dans le dossier /var/www/html).   |
| 8 | Implémenter le routage (fichier .htaccess et contrôleur frontal index.php).  |
| 9 | Implémenter les cas d'utilisation énoncés ci-dessous : <ul style="list-style-type: none"> <li>• Créer un compte client</li> <li>• S'authentifier</li> <li>• Consulter son profil</li> <li>• Consulter la liste des ateliers programmés</li> <li>• S'inscrire à un ateliers</li> <li>• Annuler sa participation à un atelier</li> <li>• Se déconnecter</li> </ul> |

**Annexe – Modèle de conception Modèle-Vue-Contrôleur (MVC)***Design Pattern – DP composite*