

INFO PLAGE – CHANG TOMA

BTS SIO SLAM

30 / 05 / 2022

VUE D'ENSEMBLE

1. Informations et description du projet

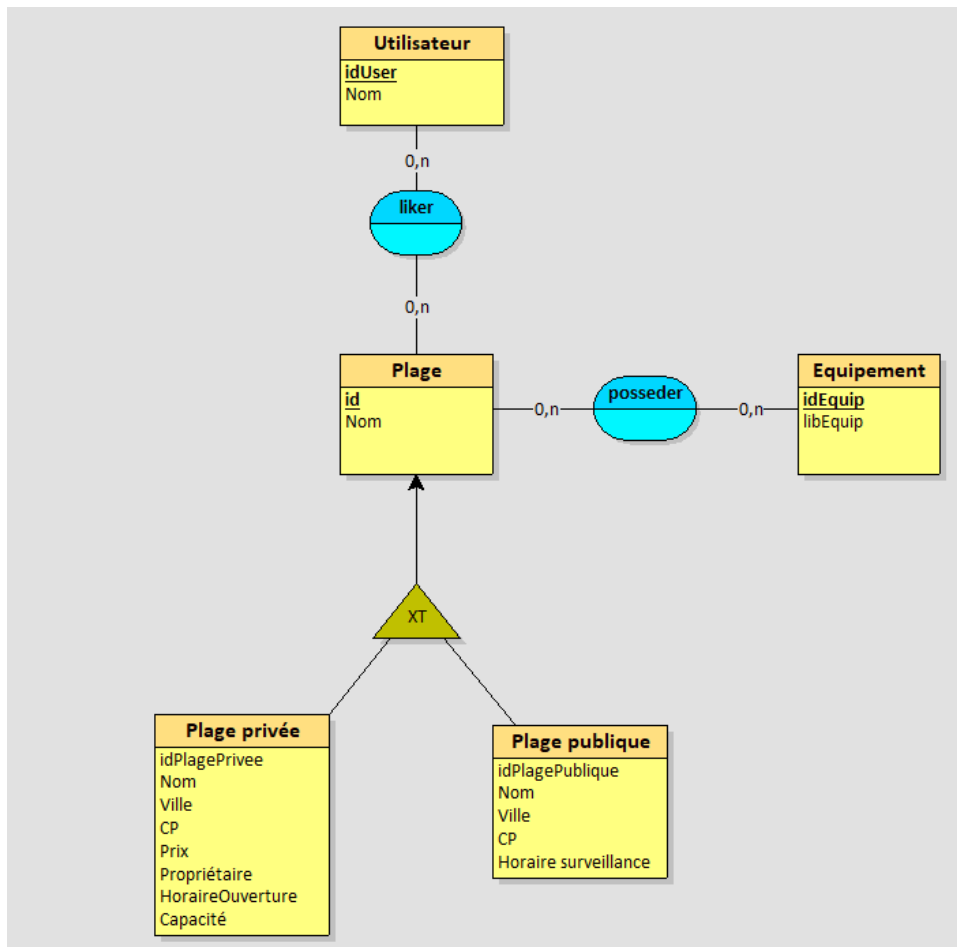
i Cette application Web a pour **but** de **donner** des **informations** concernant des **types** de **plages** (**Plage publique** ou encore **Plage privée**). Plus que la diffusion de **plage** et de ses **équipements**, elle permet de **gérer** un système **d'interaction** avec **l'utilisateur** qui aura la **capacité d'aimer** une ou **plusieurs plages**. L'application Web a une **interface original** et **ergonomique**, permettant de gérer **efficacement** et **simplement** la diffusion de plage tout en proposant de **nombreuses fonctionnalités**.

2. Contraintes technique du projet

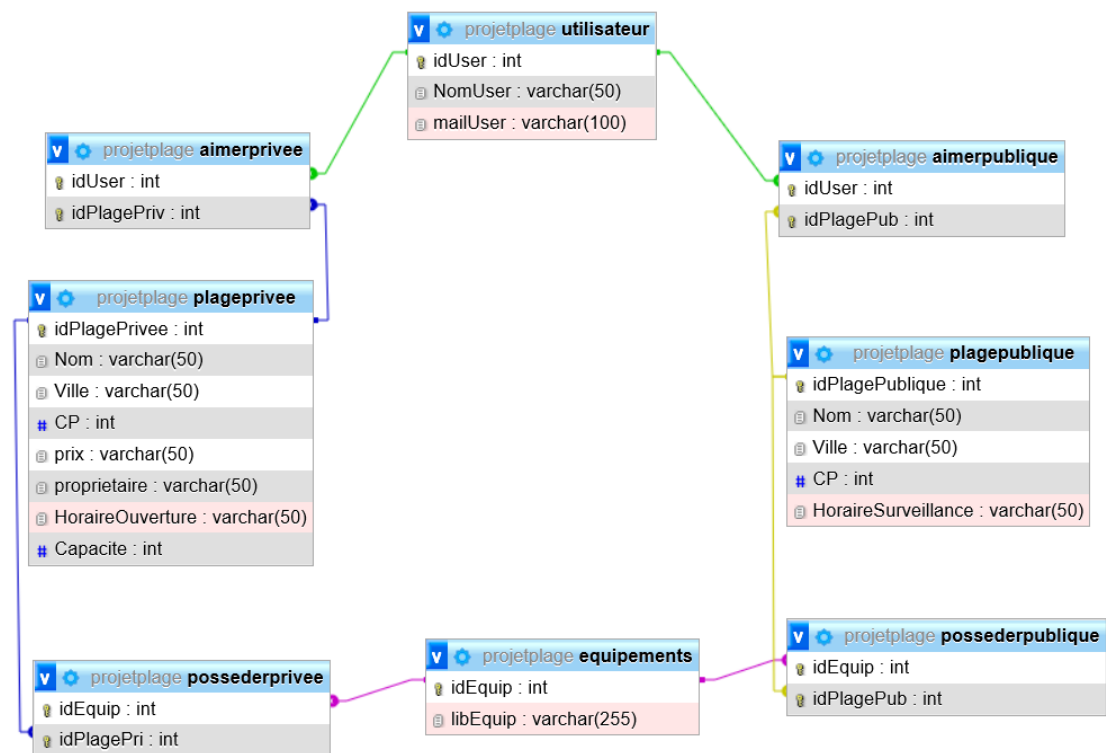
i Mon projet se décompose en différentes contraintes:

- Une **structure** principale d'application selon le **modèle MVC**
- Un projet codé en **PHP**
- Un **développement** orienté **objet**, qui comporte différentes **classes** permettant de gérer les **objets**
- Un système de **gestion** de **base** de **données MySQL**

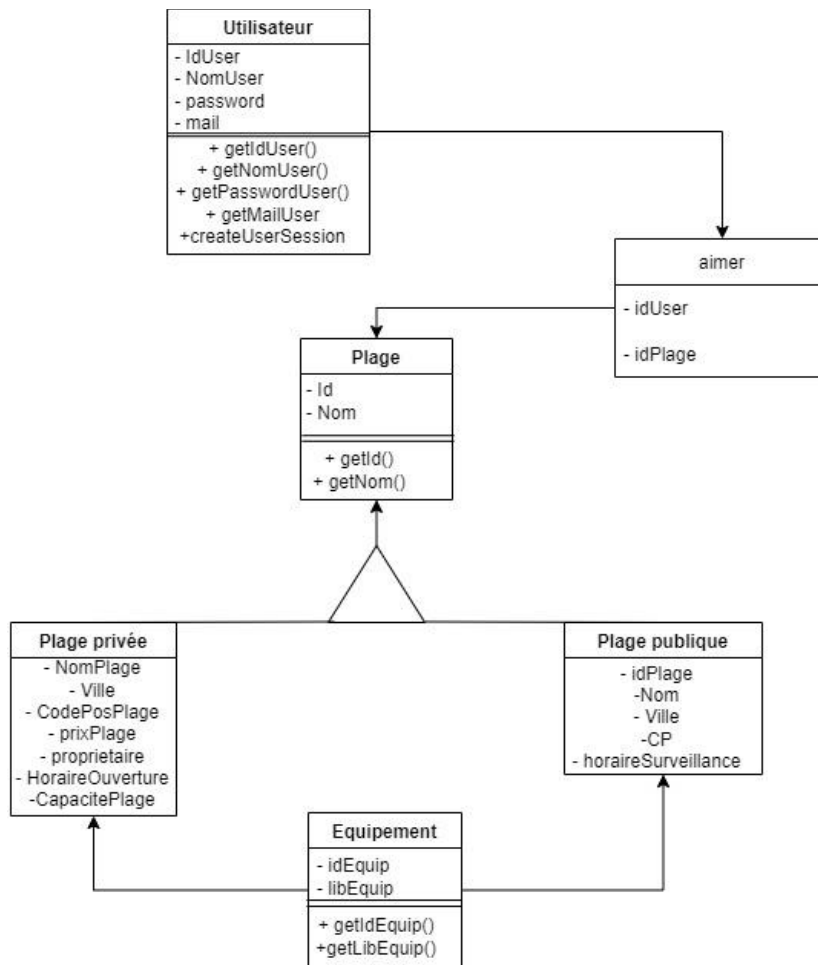
3. Schéma conceptuel



4. Schéma relationnel



5. Diagramme de classes



6. Controleur User – ControlUser.php



Ce controleur se compose de toutes les fonctionnalités principales concernant l'utilisateur :

- **Register** : Il prend en compte le nom / le mail / le password de l'utilisateur
 - Si un ou plusieurs champs est manquant, le site signalera l'utilisateur, en cas contraire :
 - **Premièrement**, on vérifie si le nom et le mail ne sont pas utilisés
 - **Deuxièmement**, on vérifie si le password et le confirmPassword sont identiques
 - Si les conditions sont validées, le compte de l'utilisateur sera enregistré dans la base de données, via l'appel d'une fonction `registerDAO($name, $mail, $password)` provenant du modèle **DAOUser.php** et enfin l'appel à la vue nécessaire, cependant dans le cas où ces conditions ne sont pas respectées, aucune données ne sera prise en compte et l'utilisateur devra entrer de nouveau ses données.
- **Login** : récupère les données saisie par le formulaire de Login, et qui ensuite les attribues à des variables spécifiques : `$name` et `$password`
 - On regarde si l'utilisateur a bien un compte, qu'on vérifie via la fonction `loginDAO($name, $password)` du **DAOUser.php**, dans le cas où l'utilisateur est vérifié, nous pouvons donc démarrer une session et ainsi appeler les vues nécessaires, si non l'utilisateur devra saisir à nouveau ses données.
- **Logout** : Lors de l'appel de cette fonctionnalités, il permettra de détruire la session.
- **New Password** : Pour permettre de changer le mot de passe, il faut nous assurer qu'il s'agit du bon utilisateur, pour cela, nous faisons une demande à l'utilisateur de son **Nom**, son **mot de passe actuel** et enfin de son **nouveau mot de passe**, une vérification qui se fait via la fonction `loginDAO` du modèle, si la vérification est positive, le changement de mot de passe se fera à travers une

fonction **UpdatePassword(\$OldPassword, \$NewPassword)**, et ainsi l'appel à la **vue** appropriée.

Modèle : *Utilisateur.php*

Vues : *VueRegisterUser.php* / *VueLoginUser.php* / *VueNewPassword.php*

7. Controleur Plage Publique – ControlPlagePu.php

i Ce controleur nous permet de récupérer les **informations** des **plages publiques** via l'appel à des **fonctions** tel que **getPlages()** ainsi que **getLibEquip()** pour les **équipements** de la **plages**. Des fonctions qui sont **instanciées** à des **variables spécifiques** qui nous permettra de les **utilisées** dans la **vue**.

Modèle : *DAOPlagePubLique.php*

Vue : *VuePlagePublique.php*

8. Controleur Plage Privée - ControlPlagePriv.php

i Ce controleur **reflète** le **controleur** de la **plage publique**, le seul **changement** est les **informations diffusées**.

Modèle : *Modele/DAOPlagePriv.php*

Vue : *VuePlagePriv.php*

9. Controleur Like – ControlAimer.php

i Dans ce controleur nous pouvons retrouver les **fonctionnalités** qui permettra à **l'utilisateur d'aimer** ou de **supprimer** son **"like"**, nous récupérons le **nom** de **l'utilisateur** via la **méthode \$_SESSION** et ainsi **l'id** de la **plage** via la **vue**.

Une interaction qui se fait via la fonction **addAimerPlagePub(\$idUser, \$idPlage)** ou **addAimerPlagePriv(\$idUser, \$idPlage)**, si cette **fonction** est **validée**, l'appel de la **vue nécessaire** sera **exécutée**.

L'utilisateur pourra aussi **supprimer** son **"like"** via la fonction **DeleteAimerPub(\$idUser, \$idPlage)** ou **DeleteAimerPriv(\$idUser, \$idPlage)**, ce qui dependra du choix de **l'utilisateur**.

Modèle : *DAOAimerPlage.php* / *Utilisateur.php*

Vue : *VueAccueil.php*

10. Base de données

i **Utilisateur :**

Table utilisateur(IdUser, NomUser)

Clé primaire : IdUser

Plage publique :

Table aimerpublique(IdUser, IdPlage)

Clé primaire : IdUser, IdPlage

Clé étrangère : IdUser references IdUser de la table User

Clé étrangère : IdPlage references IdPlage de la table Plage

Table Plagepublique(IdPlage, Nom, Ville, CP, HoraireSurveillance)

Clé primaire: IdPlage

Plage privée :

Table aimerprivee(IdUser, IdPlage)

Clé primaire : IdUser, IdPlage

Clé étrangère : IdUser references IdUser de la table User

Clé étrangère : IdPlage references IdPlage de la table Plage

Table Plageprivee(IdPlage, Nom, Ville, CP, Prix, Propriétaire, HoraireOuverture, capacite)

Clé primaire : idPlage

Equipements :

Table Equipements(idEquip, LibEquip)

Clé primaire : idEquip

Table PossederPrivee(idEquip, idPlagePri)

Clé primaire : idPlage, idEquip

Clé étrangère : idPlage reference idPlage de la table PlagePrivée

Clé étrangère : idEquip reference idEquip de la table Equipement

Table PossederPublique(idEquip, idPlagePub)

Clé primaire : idPlage, idEquip

Clé étrangère : idPlage reference idPlage de la table PlagePublique

Clé étrangère : idEquip reference idEquip de la table Equipement

| | | | | | | | | | | | | | | |
|--------------------------|------------------|-------|-----------|-----------|------------|---------|-------|-----------|---|--------|--------------------|--------------------|------|---------|
| <input type="checkbox"/> | aimerprivee | ★ | Parcourir | Structure | Rechercher | Insérer | Vider | Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 32,0 | kio | - |
| <input type="checkbox"/> | aimerpublique | ★ | Parcourir | Structure | Rechercher | Insérer | Vider | Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 1,0 | kio | - |
| <input type="checkbox"/> | equipements | ★ | Parcourir | Structure | Rechercher | Insérer | Vider | Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 1,0 | kio | - |
| <input type="checkbox"/> | plageprivee | ★ | Parcourir | Structure | Rechercher | Insérer | Vider | Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 16,0 | kio | - |
| <input type="checkbox"/> | plagepublique | ★ | Parcourir | Structure | Rechercher | Insérer | Vider | Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 1,0 | kio | - |
| <input type="checkbox"/> | possederprivee | ★ | Parcourir | Structure | Rechercher | Insérer | Vider | Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 32,0 | kio | - |
| <input type="checkbox"/> | possederpublique | ★ | Parcourir | Structure | Rechercher | Insérer | Vider | Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 1,0 | kio | - |
| <input type="checkbox"/> | utilisateur | ★ | Parcourir | Structure | Rechercher | Insérer | Vider | Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 1,0 | kio | - |
| 8 tables | | Somme | | | | | | | | 0 | MyISAM | utf8mb4_0900_ai_ci | 85,0 | kio 0 o |