

ATELIER PRO

Système de Gestion de Tickets

Documentation Technique & Fonctionnelle Complète

PHP • PDO • MySQL

BTS SIO — V2.0

I. PRÉSENTATION DU PROJET.....	3
I.I Architecture générale.....	3
I.II Structure des dossiers.....	3
I.III Technologies utilisées.....	3
II. BASE DE DONNÉES.....	4
II.I Schéma des tables.....	4
II.II Relations entre tables.....	4
II.III Données de démo (mot de passe : role123).....	4
III. GUIDE D'UTILISATION.....	6
III.I Connexion (Login).....	6
III.II Dashboard — Accueil par rôle.....	7
III.III Gestion des utilisateurs (Admin uniquement).....	7
Créer un utilisateur.....	7
Modifier / Supprimer un utilisateur.....	9
III.IV Gestion des tickets (Secrétaire).....	10
Créer un ticket.....	10
Lister et modifier les tickets agents municipaux.....	11
III.V Assignment des tickets.....	12
Mes tickets — Technicien.....	13
Mes tickets — Agent municipal.....	13
Gérer les tickets techniciens (User).....	14
IV. ANALYSE DÉTAILLÉE DU CODE.....	16
IV.I Fichiers de configuration.....	16
config/config.php.....	16
config/function.php.....	16
IV.II Routeur principal (public/index.php).....	16
IV.III Authentification.....	16
IV.IV Contrôleurs utilisateurs.....	17
create_user_ctrl.php — Rôle requis : admin.....	17
IV.V Contrôleurs tickets — Modèle unifié Tickets.php.....	17
Tickets.php remplace : Admins, Agents, Secretaires, Techniciens, Users.....	17
Validation dans le contrôleur — exemple list_tickets_ctrl.php.....	17
IV.VI Contrôleurs mes tickets.....	18
my_tickets_ctrl.php — Technicien.....	18
IV.VII Vues.....	18
IV.VIII Feuille de style — public/assets/style.css.....	18
V. SÉCURITÉ.....	19
VI. INSTALLATION ET CONFIGURATION.....	20
VI.I Prérequis.....	20
VI.II Étapes d'installation.....	20
VII. MCD — MODÈLE CONCEPTUEL DE DONNÉES.....	21
VIII. MPD — MODÈLE PHYSIQUE DE DONNÉES.....	22
IX. REQUÊTES SQL COMPLEXES DÉTAILLÉES.....	23
IX.I Assignment d'un ticket — Tickets::assigner().....	23
IX.II Tickets assignés — Tickets::assignesA().....	23
IX.III Tickets par catégorie — Tickets::parCategorie().....	23
IX.IV Changement de statut.....	23
IX.V Lecture d'un ticket — read_ticket_ctrl.php.....	23

I. PRÉSENTATION DU PROJET

Ce projet est basé sur de la gestion de tickets d'intervention municipale. Elle permet à plusieurs types d'utilisateurs (secrétaire, technicien, agent municipal, utilisateur) d'interagir avec des tickets selon leurs droits respectifs.

I.I Architecture générale

- **Modèle** : classes PHP dans Model/ — requêtes SQL via PDO (Tickets.php, Utilisateurs.php)
- **Vue** : fichiers PHP dans views/ — contiennent uniquement le HTML
- **Contrôleur** : fichiers PHP dans controllers/ — logique métier et validation
- **Routeur** : public/index.php — point d'entrée unique, dispatche les actions

I.II Structure des dossiers

```
Atelier_Pro-2/
├── config/
│   ├── config.php          <- Connexion PDO à la base de données
│   └── function.php        <- Fonctions utilitaires (h, requireRole...)
├── controllers/
│   ├── users/
│   │   ├── create_user_ctrl.php
│   │   ├── list_users_ctrl.php
│   │   ├── edit_user_ctrl.php
│   │   └── delete_user_ctrl.php
│   ├── tickets/
│   │   ├── create_ticket_ctrl.php
│   │   ├── list_tickets_ctrl.php
│   │   ├── edit_ticket_ctrl.php
│   │   ├── read_ticket_ctrl.php
│   │   └── assign_ticket_ctrl.php
│   ├── my_tickets_ctrl.php
│   ├── my_tickets_agent_ctrl.php
│   └── manage_tech_tickets_ctrl.php
├── views/ ...
├── public/
│   ├── index.php          <- Routeur principal (Front Controller)
│   └── assets/style.css
├── Model/
│   ├── Tickets.php        <- Toutes les requêtes SQL tickets (CRUD + filtres)
│   └── Utilisateurs.php    <- Toutes les requêtes SQL utilisateurs (CRUD)
└── database/
    └── tickets.sql
```

I.III Technologies utilisées

Technologie	Usage
PHP 8+	Langage serveur principal
PDO	Couche d'abstraction BDD (requêtes préparées)
MySQL / MariaDB	Système de gestion de base de données relationnelle
HTML5 / CSS3	Interface utilisateur responsive
Sessions PHP	Authentification et gestion des rôles
password_hash / password_verify	Hachage sécurisé des mots de passe (bcrypt)

II. BASE DE DONNÉES

II.I Schéma des tables

Table : utilisateurs

Colonne	Type	Contrainte	Description
id	INT	PK AUTO_INCREMENT	Identifiant unique
nom	VARCHAR(255)	NOT NULL	Nom complet
email	VARCHAR(255)	NOT NULL UNIQUE	Email (identifiant de connexion)
mot_de_passe	VARCHAR(255)	NOT NULL	Hash bcrypt
role	ENUM	NOT NULL DEFAULT technicien	admin/secretaire/technicien/agent municipal/user
created_at	DATETIME	DEFAULT NOW()	Date de création

Table : categories

Colonne	Type	Contrainte	Description
id	INT	PK AUTO_INCREMENT	Identifiant unique
nom	VARCHAR(100)	NOT NULL	Nom de la catégorie

Table : tickets

Colonne	Type	Contrainte	Description
id	INT	PK AUTO_INCREMENT	Identifiant unique
titre	VARCHAR(255)	NOT NULL	Résumé court du problème
corps	TEXT	NOT NULL	Description détaillée
adresse_lieu	VARCHAR(255)	NOT NULL	Localisation de l'intervention
priorite	ENUM(low,medium,hard)	NOT NULL DEFAULT medium	Niveau de priorité
statut	ENUM(...)	NOT NULL DEFAULT en cours	État du ticket
assign_to	INT	NULL — FK utilisateurs(id)	ID de l'utilisateur assigné
categorie_id	INT	NULL — FK categories(id)	ID de la catégorie
date_creation	DATETIME	DEFAULT NOW()	Date de création

II.II Relations entre tables

- tickets.assign_to → utilisateurs.id : ON DELETE SET NULL
- tickets.categorie_id → categories.id : ON DELETE SET NULL

II.III Données de démo (mot de passe : role123)

Nom	Email	Rôle
Administrateur	Mairie@g.fr	admin
Chantal	Chantal@gm.fr	secretaire
Dupont	Monasterol@gm.fr	technicien
Tartampion	Caraf@eau.fr	agent municipal
zaae	ervr@defr.fr	user

III. GUIDE D'UTILISATION

III.I Connexion (Login)

Point d'entrée : `http://localhost/Atelier_Pro-2/public/index.php`

`http://localhost/Atelier_Pro-2/public/index.php`



Connexion

Accédez à votre espace de gestion

Email

Mot de passe

[Se connecter →](#)

Comptes fictifs pour le TP

EMAIL	RÔLE	MOT DE PASSE
Mairie@g.fr	admin	role123
Chantal@gm.fr	secretaire	role123
Monasterol@gm.fr	technicien	role123
Caraf@eau.fr	agent municipal	role123
ervr@defr.fr	user	role123

- **password_verify()** : Compare en temps constant — résistant aux attaques par timing
- **session_regenerate_id(true)** : Protection contre la fixation de session
- **Message générique** : Ne divulgue pas si l'email existe ou non

III.II Dashboard — Accueil par rôle

Rôle	Actions disponibles dans le dashboard
admin	Créer un compte — Voir tous les comptes
secretaire	Créer un ticket — Liste des tickets agents municipaux
technicien	S'assigner un ticket — Mes tickets assignés
agent municipal	S'assigner un ticket — Mes tickets assignés
user	Gérer les tickets techniciens

III.III Gestion des utilisateurs (Admin uniquement)

Créer un utilisateur

- Champs requis : Nom complet, Email, Rôle
- Mot de passe : Automatiquement fixé à « role123 » (haché bcrypt avant insertion)
- Validation : email format, rôle dans liste autorisée, unicité email (erreur PDO 23000)

Modifier / Supprimer un utilisateur

 **Créer un utilisateur**

 Tous les comptes sont créés avec le mot de passe par défaut : **role123**

Nom complet *

Email *

Rôle *

— Choisir un rôle —

Créer l'utilisateur →

Annuler

Lister les utilisateurs

- **Accès** : Dashboard → Tous les comptes | URL : `index.php?action=list_users`
- **Affiche** : Nom, Email, Rôle (badge coloré), Date de création, Boutons Modifier/Supprimer
- **Tri** : Les utilisateurs sont triés par rôle puis par nom

Utilisateurs				
Misshg admin + Nouvel utilisateur Dashboard				
NOM	EMAIL	RÔLE	CRÉÉ LE	ACTIONS
Misshg	Mairie@q.fr	admin	2026-04-26 15:51:10	ModifierSupprimer
Chantal	Chantal@gm.fr	secrétaire	2026-04-26 15:54:41	ModifierSupprimer
Dupont	Monasterol@gm.fr	technicien	2026-04-26 15:55:00	ModifierSupprimer
Tartampion	Caral@eau.fr	agent municipal	2026-04-26 15:55:29	ModifierSupprimer
zazee	ervn@defr.fr	user	2026-04-26 16:36:30	ModifierSupprimer

- Champs modifiables : Nom, Email, Rôle (mot de passe non modifiable ici)
- Protection : Un admin ne peut pas se supprimer lui-même

III.IV Gestion des tickets (Secrétaire)

Créer un ticket

- Priorité : Basse (low), Moyenne (medium), Haute (hard)
- Statut initial : automatiquement « en attente » à la création

- Assignment : assign_to est NULL à la création

Modifier l'utilisateur

Nom complet *

Email *

Rôle *

Utilisateur

Enregistrer Annuler

Supprimer un utilisateur

- **Accès** : Clic sur le bouton Supprimer dans la liste | URL : index.php?action=delete_user&id={id}
- **Confirmation** : JavaScript avant suppression (confirm())
- **Protection** : Un admin ne peut pas se supprimer lui-même



Lister et modifier les tickets agents municipaux

Créer un ticket

Titre *

Description *

Adresse / lieu *

Priorité

● Moyenne

Catégorie *

— Choisir —

[Créer le ticket](#) [Annuler](#)

Lister les tickets agents municipaux

- **Accès** : Dashboard → Liste des tickets agents municipaux | URL : index.php?action=list_tickets
- **Filtre** : Affiche uniquement les tickets de catégorie « Agent municipal »
- **Actions disponibles** : Modifier, Supprimer par ligne

Tickets agents municipaux								← Dashboard
TITRE	DESCRIPTION	ADRESSE	PRIORITÉ	STATUT	ASSIGNÉ À	DATE	ACTIONS	
TP	Tp	TP	● Moyenne	En attente		2026-04-28 14:31:43	Modifier Supprimer	

III.V Assignment des tickets

Modifier le ticket 28

Titre *

TP

Description *

Tp

Adresse / lieu *

TP

Priorité

● Moyenne

Statut

En cours

Catégorie

Agent municipal

Enregistrer

Annuler

III.V Assignment des tickets

L'assignation permet à un technicien ou un agent municipal de prendre en charge un ticket disponible.

S'assigner un ticket

- **Accès** : Dashboard → S'assigner un ticket | URL : `index.php?action=assign_ticket`
- **Filtre** : Affiche uniquement les tickets NON assignés (`assign_to IS NULL`)
- **Filtrage par rôle** : Technicien → catégorie « Technicien » | Agent municipal → catégorie « Agent municipal »
- **Assignation** : `UPDATE tickets SET assign_to = {userId}, statut = « en cours » WHERE id = ? AND assign_to IS NULL`
- **Protection concurrence** : `AND assign_to IS NULL` empêche de « voler » un ticket déjà pris

Tickets disponibles						Tartampion agent municipal < Dashboard
#	TITRE	DESCRIPTION	ADRESSE	PRIORITÉ	DATE	ACTION
28	TP	Tp	TP	● Moyenne	2026-04-28 14:31:43	M'assigner

Mes tickets — Technicien

Mes tickets — Technicien

- **URL** : `index.php?action=my_tickets`
- **Affiche** : Tickets assignés au technicien connecté (statuts « en cours » et « terminé »)
- **Action** : Bouton « Clôt » pour passer un ticket « en cours » à « terminé »

Mes tickets
← Dashboard

ID	TITRE	DESCRIPTION	ADRESSE	PRIORITÉ	STATUT	CATÉGORIE	DATE	ACTIONS
27	dab sf	sfbsfb	sbsb	 Moyenne	En cours	Technicien	2026-04-26 17:01:34	<button>Clôt</button> Voir

Mes tickets
← Dashboard

ID	TITRE	DESCRIPTION	ADRESSE	PRIORITÉ	STATUT	CATÉGORIE	DATE	ACTIONS
27	dab sf	sfbsfb	sbsb	 Moyenne	Terminé	Technicien	2026-04-26 17:01:34	Voir

Ticket 27

TITRE

dab sf

DESCRIPTION

sfbsfb

ADRESSE

sbsb

PRIORITÉ

 Moyenne

STATUT

Terminé

CATÉGORIE

Technicien

DATE DE CRÉATION

2026-04-26 17:01:34

Mes tickets — Agent municipal

- **URL** : `index.php?action=my_tickets_agent`
- **Différence** : Colonne « Technicien assigné » supplémentaire

Mes tickets
← Dashboard

ID	TITRE	DESCRIPTION	ADRESSE	PRIORITÉ	STATUT	CATÉGORIE	AGENT ASSIGNÉ	DATE	ACTIONS
28	TP	Tp	TP	 Moyenne	En cours	Agent municipal	Tartampon	2026-04-28 14:31:43	<button>Clôt</button> Voir

Mes tickets — Agent municipal

- Même fonctionnement que technicien, avec colonne « Agent assigné » supplémentaire

Gérer les tickets techniciens (User)

Atelier Pro — Système de Gestion de Tickets

Mes tickets

[← Dashboard](#)

ID	TITRE	DESCRIPTION	ADRESSE	PRIORITÉ	STATUT	CATÉGORIE	AGENT ASSIGNÉ	DATE	ACTIONS
28	TP	Tp	TP	Moyenne	Terminé	Agent municipal	Tartampion	2026-04-28 14:31:43	Voir

Ticket 28

TITRE	TP
DESCRIPTION	Tp
ADRESSE	TP
PRIORITÉ	Moyenne
STATUT	Terminé
CATÉGORIE	Agent municipal
DATE DE CRÉATION	2026-04-28 14:31:43

Gérer les tickets techniciens (User)

- **URL** : `index.php?action=manage_tech_tickets`
- **Affiche** : Tous les tickets de catégorie « Technicien »
- **Actions** : Modifier (formulaire inline dans le tableau), Supprimer

Tickets techniciens

[← Dashboard](#)

TITRE	DESCRIPTION	ADRESSE	PRIORITÉ	STATUT	TECHNICIEN	DATE	ACTIONS
dsb sf	sfbsfb	sbab	Moyenne	En cours	Dupont	2026-04-26 17:01:34	Modifier Supprimer

Modifier le ticket 27

Titre *

dsb sf

Description *

sfbsfb

Adresse / lieu *

sbsb

Priorité

 Moyenne

Statut

En cours

Catégorie

Technicien

Enregistrer

Annuler

IV. ANALYSE DÉTAILLÉE DU CODE

IV.I Fichiers de configuration

config/config.php

```
<?php
define('DB_HOST',      '127.0.0.1');
define('DB_PORT',      8889);
define('DB_NAME',      'Projet');
define('DB_USER',      'root');
define('DB_PASS',      'root');
define('DB_CHARSET',   'utf8mb4');

$dsn = "mysql:host=".DB_HOST.";port=".DB_PORT.";dbname=".DB_NAME.";charset=".DB_CHARSET;
$options = [
    PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES  => false,
];
try {
    $pdo = new PDO($dsn, DB_USER, DB_PASS, $options);
} catch (PDOException $e) {
    die("Connexion impossible : " . $e->getMessage());
}
```

- **ERRMODE_EXCEPTION** : Toute erreur SQL lance une exception PHP
- **EMULATE_PREPARES = false** : Requêtes préparées réellement côté MySQL — protection maximale contre l'injection SQL
- **FETCH_ASSOC** : Résultats en tableaux associatifs (\$row['nom'] au lieu de \$row[0])

En production, ne jamais mettre les credentials en clair — utiliser des variables d'environnement (.env)

config/function.php

```
function h(?string $v): string {
    return htmlspecialchars((string)$v, ENT_QUOTES | ENT_SUBSTITUTE, 'UTF-8');
}

function requireRole(string|array $roles): void {
    requireLogin();
    if (!in_array($_SESSION['user_role'] ?? '', (array)$roles, true)) {
        http_response_code(403);
        exit('Accès refuse.');
```

- **h()** : Échappe le HTML pour prévenir les attaques XSS — appelée sur toutes les sorties
- **requireRole()** : Accepte string ou array de rôles — HTTP 403 si non autorisé
- **in_array(..., true)** : Comparaison stricte type + valeur

IV.II Routeur principal (public/index.php)

- **\$pages** : Liste blanche des routes autorisées — pas d'include dynamique depuis \$_GET (pas de path traversal)
- **file_exists()** : Vérifie l'existence du fichier avant l'include
- **Déconnexion** : Gérée en priorité dans le routeur (session_destroy())

IV.III Authentification

- **Requête préparée** : L'email vient de \$_POST — paramètre lié, aucun risque d'injection
- **password_verify()** : Comparaison bcrypt en temps constant — résistant au timing
- **session_regenerate_id(true)** : Nouvel ID de session après login — protection fixation de session
- **3 variables uniquement** : user_id, user_name, user_role — le mot de passe n'est jamais stocké en session

IV.IV Contrôleurs utilisateurs

create_user_ctrl.php — Rôle requis : admin

```
requireRole('admin');
$rolesValides = ['admin', 'secretaire', 'technicien', 'agent municipal', 'user'];

if (empty($nom) || empty($email) || empty($role)) {
    $erreur = 'Tous les champs sont requis.'; return;
}
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $erreur = "L'adresse email n'est pas valide."; return;
}
if (!in_array($role, $rolesValides, true)) {
    $erreur = "Le role n'est pas valide."; return;
}

$gestion = new Utilisateurs($pdo);
try {
    $gestion->creer($nom, $email, $motDePasseHache, $role);
} catch (PDOException $e) {
    $erreur = $e->getCode() === '23000'
        ? 'Cet email est deja utilise.'
        : 'Une erreur est survenue.';
}
```

- **Utilisateurs.php** : Uniquement des requêtes SQL — CRUD sans aucune validation
- **PDOException 23000** : Violation UNIQUE — message d'erreur clair à l'utilisateur

IV.V Contrôleurs tickets — Modèle unifié Tickets.php

Tickets.php remplace : Admins, Agents, Secretaires, Techniciens, Users

```
class Tickets {
    public function __construct(private PDO $pdo) {}

    // Lecture
    public function trouver(int $id): array|false { ... }
    public function trouverAssigne(int $ticketId, int $userId): array|false { ... }
    public function parCategorie(string $categorieNom): array { ... }
    public function assignesA(int $userId): array { ... }
    public function nonAssignesParCategorie(string $categorieNom): array { ... }
    public function categories(): array { ... }

    // Ecriture
    public function creer(...): void { ... }
    public function modifier(...): void { ... }
    public function changerStatut(int $ticketId, string $statut): void { ... }
    public function changerStatutAssigne(int $ticketId, string $statut, int $userId): void { ... }
    public function assigner(int $ticketId, int $userId, string $statut): void { ... }
    public function supprimer(int $id): void { ... }
}
```

- **Principe de responsabilité unique** : Le modèle exécute les requêtes, le contrôleur valide les données
- **parCategorie()** : Remplace tousLesTechniciens() et tousLesAgents() — une seule méthode paramétrée
- **changerStatutAssigne()** : WHERE assign_to = ? — un technicien/agent ne peut modifier que ses propres tickets

Validation dans le contrôleur — exemple list_tickets_ctrl.php

```
$statutsValides = ['en attente', 'en cours', 'cloture'];

if (in_array($_POST['statut'], $statutsValides, true)) {
    $gestion->changerStatut((int) $_POST['ticket_id'], $_POST['statut']);
}

$tickets = $gestion->parCategorie('Agent municipal');
```

IV.VI Contrôleurs mes tickets

my_tickets_ctrl.php — Technicien

```
requireRole(['technicien']);
$userId = (int) $_SESSION['user_id'];
$gestion = new Tickets($pdo);

if ($SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['ticket_id'], $_POST['statut'])) {
    if (in_array($_POST['statut'], ['en cours', 'termine'], true)) {
        $gestion->changerStatutAssigne((int)$_POST['ticket_id'], $_POST['statut'], $userId);
        header('Location: index.php?action=my_tickets'); exit;
    }
}
$tickets = $gestion->assignesA($userId);
```

- **Agent municipal** : Même logique mais statuts autorisés = ['termine'] uniquement
- **Double sécurité** : Whitelist PHP dans le contrôleur + AND assign_to = ? en BDD

IV.VII Vues

- **dashboard.php** : Blocs if/elseif par rôle — chaque utilisateur voit uniquement ses actions
- **manage_tech_tickets.php** : Formulaire d'édition inline — déclenché si \$ticket_edit['id'] === \$ticket['id']
- **XSS** : h(\$nom), h(\$role) — toutes les sorties sont échappées via htmlspecialchars()

IV.VIII Feuille de style — public/assets/style.css

Section CSS	Composants couverts
Base	Reset box-sizing, typographie body, h1/h2, liens
Nav / Page-header	Barre de navigation, flex layout
Badge	Étiquette colorée pour les rôles et statuts
Login	Conteneur centré, boîte de connexion
Formulaires	Champs input/select/textarea, focus bleu, form-card
Boutons	Primaire (bleu), danger (rouge), secondaire (gris)
Tableau	table-wrapper scrollable, alternance de couleurs au hover
Card	card-row, card-label, card-value — détail ticket
Messages	error (rouge), success (vert), empty (gris italique)

V. SÉCURITÉ

Menace	Protection mise en place
SQL Injection	Requêtes préparées PDO avec paramètres liés sur toutes les requêtes
XSS	h() = htmlspecialchars() appelée sur toutes les sorties
Fixation de session	session_regenerate_id(true) après chaque connexion réussie
CSRF	Formulaires POST pour les actions sensibles, confirm() côté JS
Accès non autorisé	requireRole() en début de chaque contrôleur + liste blanche des routes
Double assignation	Clause AND assign_to IS NULL dans le UPDATE d'assignation (atomique)
Mots de passe	password_hash() bcrypt + password_verify() résistant au timing
Path traversal	Tableau \$pages liste blanche — pas d'include depuis \$_GET
Auto-suppression admin	Comparaison \$id === (int)\$_SESSION['user_id'] avant DELETE
Validation des statuts	Whitelist in_array() dans chaque contrôleur avant tout appel au modèle

VI. INSTALLATION ET CONFIGURATION

VI.I Prérequis

- PHP 8.0 ou supérieur (promoted properties)
- MySQL 5.7+ ou MariaDB 10.3+
- Serveur web : Apache ou Nginx (MAMP / WAMP / XAMPP en local)

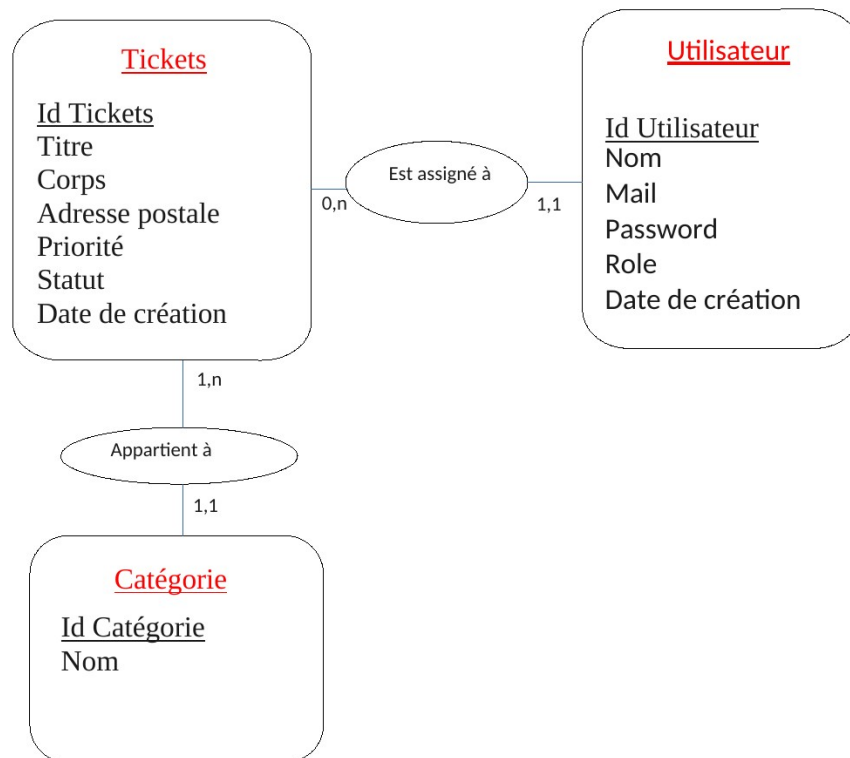
VI.II Étapes d'installation

1. Cloner ou extraire le projet dans le dossier web (ex : htdocs/Atelier_Pro-2/)
2. Importer database/tickets.sql dans MySQL
3. Modifier config/config.php : DB_HOST, DB_PORT, DB_NAME, DB_USER, DB_PASS
4. Accéder à http://localhost/Atelier_Pro-2/public/index.php
5. Se connecter avec un compte de démo (mot de passe : role123)

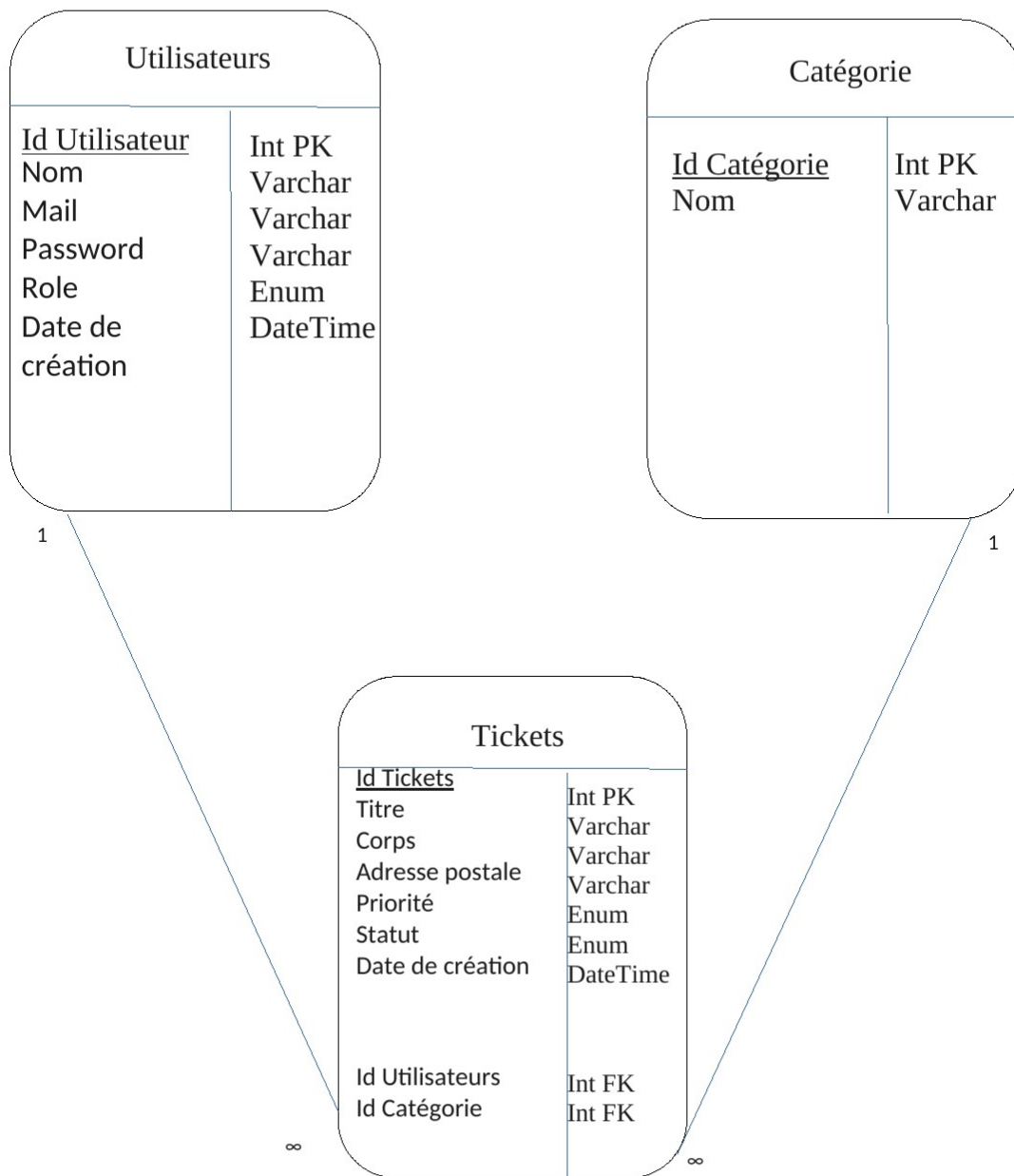
Le port 8889 dans config.php est spécifique à MAMP. Utiliser 3306 pour XAMPP/WAMP ou MySQL standard.

VII. MCD — MODÈLE CONCEPTUEL DE DONNÉES

Un ticket appartient à une seule catégorie. Un ticket peut être assigné à un utilisateur. Un utilisateur peut être assigné à zéro ou plusieurs tickets.



VIII. MPD — MODÈLE PHYSIQUE DE DONNÉES



IX. REQUÊTES SQL COMPLEXES DÉTAILLÉES

IX.I Assignment d'un ticket — Tickets::assigner()

```
UPDATE tickets SET assign_to = ?, statut = ?
WHERE id = ? AND assign_to IS NULL
```

- **AND assign_to IS NULL** : Clause atomique — empêche la double assignment même en accès concurrent
- **MySQL atomique** : Même si deux utilisateurs cliquent simultanément, un seul UPDATE réussira

IX.II Tickets assignés — Tickets::assignesA()

```
SELECT t.id, t.titre, t.corps, t.adresse_lieu,
       t.priorite, t.statut, t.date_creation,
       c.nom AS categorie_nom, u.nom AS assign_nom
FROM   tickets t
LEFT JOIN categories c ON t.categorie_id = c.id
LEFT JOIN utilisateurs u ON t.assign_to = u.id
WHERE  t.assign_to = ?
       AND t.statut IN ('en cours', 'termine')
ORDER BY t.date_creation DESC
```

- **WHERE assign_to = ?** : Isolation — l'utilisateur ne voit que ses propres tickets
- **LEFT JOIN** : Conserve les tickets même si catégorie ou technicien est NULL

IX.III Tickets par catégorie — Tickets::parCategorie()

```
SELECT t.id, t.titre, t.corps, t.adresse_lieu,
       t.priorite, t.statut, t.date_creation,
       u.nom AS assign_nom, c.nom AS categorie_nom
FROM   tickets t
JOIN   categories c ON t.categorie_id = c.id
LEFT JOIN utilisateurs u ON t.assign_to = u.id
WHERE  c.nom = ?
ORDER BY t.date_creation DESC
```

- **Paramètre c.nom = ?** : 'Technicien' ou 'Agent municipal' — remplace tousLesTechniciens() et tousLesAgents()
- **INNER JOIN categories** : Exclut les tickets sans catégorie

IX.IV Changement de statut

```
-- changerStatut() : secrétaire (validation dans le contrôleur)
UPDATE tickets SET statut = ? WHERE id = ?
```

```
-- changerStatutAssigne() : technicien / agent
UPDATE tickets SET statut = ? WHERE id = ? AND assign_to = ?
```

- **AND assign_to = ?** : Sécurité par conception — un utilisateur ne peut modifier que ses propres tickets
- **Whitelist contrôleur** : in_array() valide les statuts autorisés avant tout appel SQL

IX.V Lecture d'un ticket — read_ticket_ctrl.php

```
// Admin : accès total
$ticket = $gestion->trouver($ticketId);
// WHERE t.id = ?
```

```
// Technicien / Agent : accès restreint à ses propres tickets
$ticket = $gestion->trouverAssigne($ticketId, $userId);
// WHERE t.id = ? AND t.assign_to = ?
```

- **Redirection silencieuse** : Si accès refusé → my_tickets.php, sans révéler l'existence du ticket