UNIVERSITÉ PERPIGNAN VIA DOMITIA

Rapport Projet Trottinette

Réalisation d'un site de location de trottinette

Bessai, Sofiane
Sody, Younes

Sommaire

Modifications

Chapitre concernant les modifications liées au rapport du projet trotinette

Sections	Description	Auteur	Relecteur	Date
Toutes	Mise en page du document	Tous	Tous	Début avril
Présentation	Remplissage de la section	Tous	Tous	Début avril
Toutes	Rapport complété	Tous	Tous	Début mai

Présentation du projet

Introduction

Suite à la création d'une nouvelle entreprise de location dans la ville de Perpignan, l'entreprise a besoin d'une base de donnée qui interagit avec un site internet qui permettra au client de de louer une trottinette, pour cela le site internet doit permettre:
Au client de louer une trottinette dans une station donnée .
Rendre la trottinette .
Calculer le coût de la location .
S'enregistrer pour pouvoir louer une trottinette .
Lister les problèmes sur le réseaux, si la station est pleine, si la une trottinette tombe en panne et permettre à l'employé de choisir un problème à régler dans la liste et d'enregistrer le problème traité dans la base de donnée .
Afficher pour l'employé l'activité du jour (nombre de trottinettes louées, nombre de trottinettes en panne, nombre de problèmes réglés et non-réglés sur le réseau pour une journée donnée) .

Pour réaliser ce projet on a procédé de la façon suivante :

Création d'un modèle relationnel et requête dans l'algèbre relationnel:

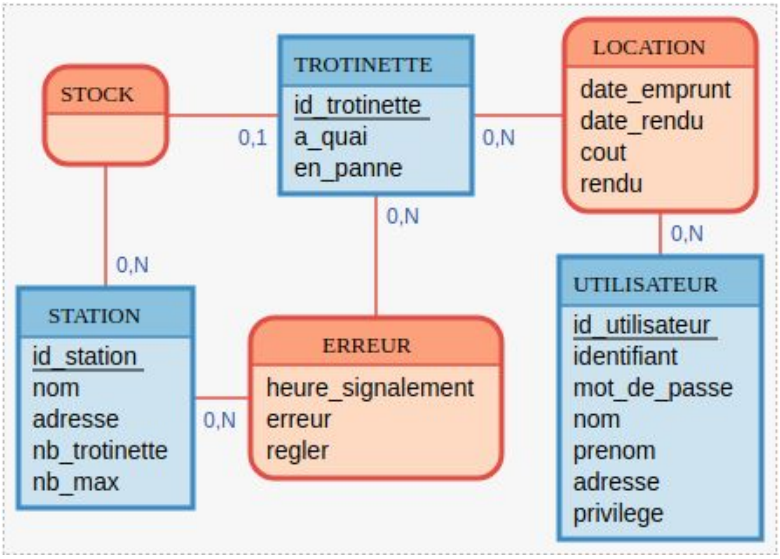
- Création d'un modèle Entité /Association pour le problème de location de trottinettes.
- Création d'un modèle relationnel correspondant .
- Saisie des requête requises dans l'algèbre relationnelle .

Création d'une base de donnée sql :

- Saisie des requêtes SQL pour la création des tables de la base de données .
- Saisie des requêtes SQL nécessaires au formulaire php .
- création d'un script pour insérer des données et pour faire sauvegarder de la base de données.

Création des différents formulaires en php et HTML qui permettent d'effectuer les opérations demandées dans l'énoncé

Modèle Entité/Association



TROTINETTE (id_trotinette, a_quai, en_panne, *id_station*)
LOCATION (*id_utilisateur*, *id_trotinette*, date_emprunt, date_rendu, cout, rendu)
STATION (id_station, nom, adresse, nb_trotinette, nb_max)
ERREUR (*id_station*, *id_trotinette*, heure_signalement, erreur, regler)
UTILISATEUR (id_utilisateur, identifiant, mot_de_passe, nom, prenom, adresse, privilege)

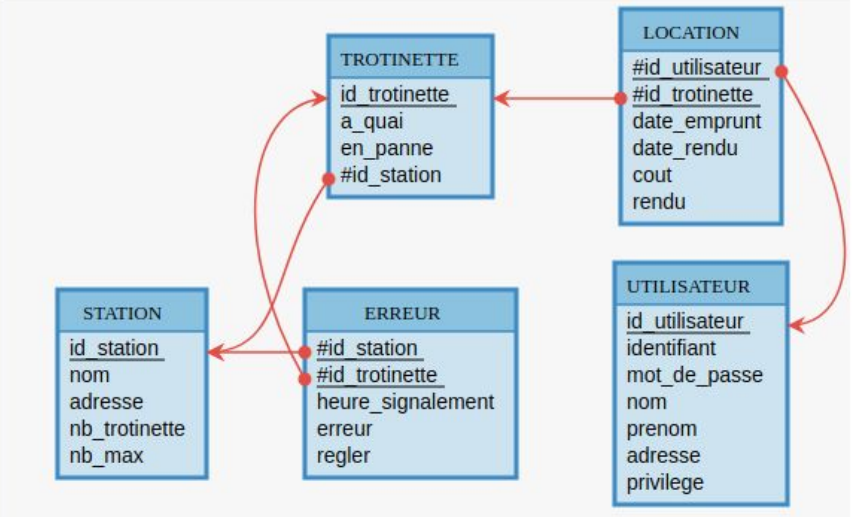
Après une étude du cahier des charges on a constaté qu'on avait besoin de trois entités principales qui sont :

- Station.
- Trottinette.
- Utilisateur.

Ensuite on a créé des relations entres ces trois entitées qui sont :

- Location qui relie l'entité utilisateur à trottinette .
- Erreur qui relie Trottinette à station .
- Stock qui relie station à trottinette

Modèle relationnel



TROTINETTE (id_trotinette, a_quai, en_panne, #id_station)

- Le champ *id_trotinette* constitue la clef primaire de la table. C'était déjà un identifiant de l'entité *TROTINETTE*.
- Les champs *a_quai*, *en_panne* et *#id_station* étaient déjà de simples attributs de l'entité *TROTINETTE*.

LOCATION (#id_utilisateur, #id_trotinette, date_emprunt, date_rendu, cout, rendu)

- Les champs *#id_utilisateur* et *#id_trotinette* constituent la clef primaire de la table. C'était déjà des identifiants de l'entité *LOCATION*.
- Les champs *date_emprunt*, *date_rendu*, *cout* et *rendu* étaient déjà de simples attributs de l'entité *LOCATION*.

STATION (id_station, nom, adresse, nb_trotinette, nb_max)

- Le champ *id_station* constitue la clef primaire de la table. C'était déjà un identifiant de l'entité *STATION*.
- Les champs *nom*, *adresse*, *nb_trotinette* et *nb_max* étaient déjà de simples attributs de l'entité *STATION*.

ERREUR (#id_station, #id_trotinette, heure_signalement, erreur, regler)

- Les champs *#id_station* et *#id_trotinette* constituent la clef primaire de la table. C'était déjà des identifiants de l'entité *ERREUR*.
- Les champs *heure_signalement*, *erreur* et *regler* étaient déjà de simples attributs de l'entité *ERREUR*.

UTILISATEUR (id_utilisateur, identifiant, mot_de_passe, nom, prenom, adresse, privilege)

- Le champ *id_utilisateur* constitue la clef primaire de la table. C'était déjà un identifiant de l'entité *UTILISATEUR*.
- Les champs *identifiant*, *mot_de_passe*, *nom*, *prenom*, *adresse* et *privilege* étaient déjà de simples attributs de l'entité *UTILISATEUR*.

Requêtes

Requêtes Base de Données

Création des table

Requete table utilisateur

```
CREATE TABLE `utilisateur` (
  `id_utilisateur` int(11) NOT NULL,
  `identifiant` varchar(25) NOT NULL,
  `mot_de_passe` varchar(50) NOT NULL,
  `nom` varchar(25) NOT NULL,
  `prenom` varchar(25) NOT NULL,
  `adresse` varchar(50) NOT NULL,
  `privilege` tinyint(1) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Requete table trottinette

```
CREATE TABLE `trottinette` (
  `id_trottinette` int(11) NOT NULL,
  `id_station` int(11) NOT NULL,
  `a_quai` tinyint(1) NOT NULL,
  `en_panne` tinyint(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Requete table station

```
CREATE TABLE `station` (
  `id_station` int(11) NOT NULL,
  `nom` varchar(50) NOT NULL,
  `adresse` varchar(50) NOT NULL,
  `nb_trottinette` int(11) NOT NULL,
  `nb_max` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Requete table location

```
CREATE TABLE `location` (
  `id_location` int(11) NOT NULL,
  `id_utilisateur` int(11) NOT NULL,
  `id_trottinette` int(11) NOT NULL,
  `date_emprunt` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `date_rendu` datetime DEFAULT NULL,
  `cout` float DEFAULT NULL,
  `rendu` int(11) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Requete table erreur

```
CREATE TABLE `erreur` (
  `id_erreur` int(11) NOT NULL,
  `id_station` int(11) NOT NULL,
  `id_trottinette` int(11) NOT NULL,
  `heure_signalement` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `erreur` text NOT NULL,
  `regler` tinyint(1) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Requêtes php

SELECT identifiant, mot_de_passe, privilege FROM utilisateur WHERE id_utilisateur = intval($_SESSION['id_utilisateur'])

$\pi_{identifiant,mots\ de\ passe,\ privilege}\ (\sigma_{id\ utilisateur\ =\ id\ utilisateur\ connecter}\ (Utilisateur)_)$

- la requête permet de sélectionner l'identifiant, le mots de passe et le privilège de la table utilisateur quand id_utilisateur de cette dernière est égal à l'id de l'utilisateur connecté

Utilisé dans la fonction actualiser_session() dans le fichier fonction.php

SELECT COUNT(*) AS exist FROM utilisateur WHERE identifiant = $identifiant

$\pi_{count(*)}\ (\sigma_{identifiant\ =\ "\ .\ \$identifiant\ .\ "}\ (Utilisateur)_)$

- la requête compte le nombre d'occurrence des entités de la table utilisateur et les liste si l'identifiant égal à la variable $identifiant

Utilisé dans la fonction verification_identifiant() dans le fichier fonction.php

SELECT id_trottinette FROM trottinette WHERE id_station = $_POST['id_station'] AND a_quai = '1' AND en_panne = '0' ";

$\pi_{id-trotinette}\ (\sigma_{id-station\ =\ \$post['id-station']\ AND\ a-quai\ =\ 01\ AND\ en-panne\ =\ 0}\ (trotinette)\ )$

- La requête sélectionne les id_trottinette de la table trottinette si la variable retenue du formulaire id_station est égal id_station de la table et à quai vaut 1 et en panne vaut 0 .

Utilisé dans le fichier louer_rendre.php pour trouver une trottinette louable

SELECT COUNT(id_location) AS nbr FROM location WHERE id_utilisateur = $_SESSION['id_utilisateur'] AND rendu = '0'"
• La requête compte le nombre d'occurrence de id_location et les liste si id_utilisateur est égal à l'id de la session.
Utilisé dans la fonction a_une_location() dans le fichier fonction.php

SELECT COUNT(id_station) AS cnt , id_station, nom, adresse, nb_max, nb_trottinette FROM station WHERE nb_trottinette != '0' GROUP BY id_station
• La requête compte le nombre d'occurrence de l'id_station et liste tout les attribut que contient la table station si le nombre de trottinette est différent de et les classes par numéro de station
Utilisé dans le fichier afficher_station.php pour afficher les stations non vide

SELECT COUNT(id_station) AS cnt , id_station, nom, adresse, nb_max, nb_trottinette FROM station WHERE nb_trottinette != nb_max GROUP BY id_station
• La requête compte le nombre d'occurrence de l'id station et liste tous les attributs de la table station par ordre de station si le nombre de trottinette est différent du nombre max
Utilisé dans le fichier afficher_station.php pour afficher les stations non pleine

SELECT id_location, date_emprunt, id_trottinette FROM location WHERE id_utilisateur = $_SESSION['id_utilisateur'] AND rendu = '0'
$\pi_{id-location,date-emprunt,id-trottinette}\left(\sigma_{id-utilisateur=\$_{-session}[id-utilisateur]}(Location)\right)$ • La requête liste l'id de location, date d'emprunt et id de trottinette de la table location si l'id de l'utilisateur est égal à l'id de la session .
Utilisé dans le fichier louer_rendre.php pour récuperer les informations de la location

INSERT INTO location (id_utilisateur, id_trottinette) VALUES ($_SESSION['id_utilisateur'] , $resultat['id_trottinette']
• La requête permet l'insertion de id_utilisateur et l'id de la trottinette récupérer par le formulaire .
Utilisé dans le fichier louer_rendre.php pour valider la location de la trottinette par l'utilisateur

UPDATE location SET cout = $cout , date_rendu = $date_rendu , rendu = '1' WHERE id_location = $resultat['id_location']
• La requête permet le rafraîchissement de la table location en modifiant le coût et la date du rendu et la statue du rendu se met à un ce qui veut dire que notre trottinette a été rendu par l'utilisateur si l'id de location correspond à la variable id_location .
Utilisé dans le fichier louer_rendre.php pour valider le rendu de la trottinette

UPDATE station SET nb_trottinette = nb_trottinette - 1 WHERE id_station = '" . $_POST['id_station'] . "'";
• La requête permet le rafraîchissement de la table station en décrémentant le nombre de trottinette si l'id de la station correspond à l'id de la station récupérer avec la méthode POST .
Utilisé dans le fichier louer_rendre.php pour décrémenter le nombre de trottinettes de la station

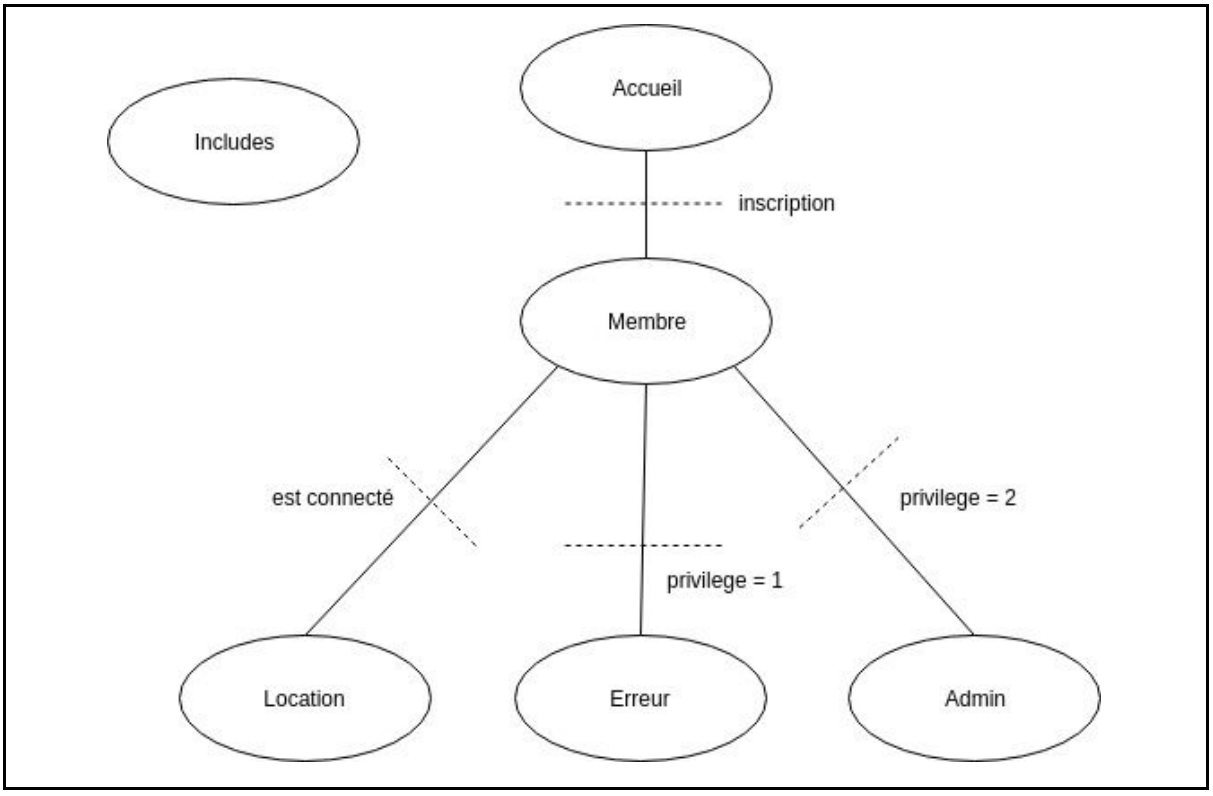
UPDATE station SET nb_trottinette = nb_trottinette + 1 WHERE id_station = $_POST['id_station']
• La requête permet le rafraîchissement de la table station en incrémentant le nombre de trottinette si l'id de la station correspond à l'id de la station récupérer avec la méthode POST .
Utilisé dans le fichier louer_rendre.php pour incrémenter le nombre de trottinettes de la station

SELECT COUNT(id_utilisateur) AS nbr, id_utilisateur, identifiant, mot_de_passe, privilege FROM utilisateur WHERE identifiant = $_POST['identifiant'] GROUP BY id_utilisateur
• La requête compte le nombre d'occurrence de l'id de l'utilisateur et liste l'id de l'utilisateur,identifiant,mots de passe et privilège de la base utilisateur par ordre d'id utilisateur si l'identifiant correspond à l'identifiant envoyer par la méthode POST .
Utilisé dans le fichier connexion.php afin de recuperer les informations et les stocker dans la variable super_globale $_SESSION

INSERT INTO utilisateur (identifiant, mot_de_passe, nom, prenom, adresse) VALUES ('$identifiant', '$sha_mdp', '$nom', '$prenom', '$adresse')
• La requête permet l'insertion des données concernant l'utilisateur dans la table utilisateur .
Utilisé dans le fichier trait-inscription.php pour inscrire le visiteur dans la table des utilisateurs

Le site

Architecture



Accueil

Notre site se compose de plusieurs partie, dans un premier temps de l'accueil étant symbolisé par index.php (l'entrée du site). En tant que visiteur vous pouvez vous connectez si vous êtes inscrit, ou bien dans le cas contraire vous inscrire.
L'inscription se fait en 2 parties, la première le formulaire puis la vérification des informations. Tant que vous avez une erreur lors de l'inscription, l'inscription recommence en gardant les infos valide.

Membre

Une fois connecté vous pourrez en fonction de votre niveau de privilège avoir accès a différentes fonctionnalitées. Comme la location de trottinette ils vous suffit alors de cliquer dans la barre en dessous de la bannière sur la fonctionnalitée de votre choix.

Location

Partie réservée aux membres connecté, Elle permet d'afficher les stations ayant des trottinette disponible et d'en louer une, ou en rendre une si l'on à déjà une location.
(Une seule location possible à la fois.

Erreur

Partie réservée aux employés elle permet de lister les pannes sur les trottinettes et afficher les stations pleines et regler les probleme présents.

Admin

Partie réservée à l'administrateur du site (identifiant : projetBDD et mot de passe : Trottinette2019) ont y trouve un lien menant vers la page de modification des privilège utilisateur, ont peut y augmenter ou descendre les privilège d'un membre.

Connexion à la base de données

Dans notre site la connexion à la base de données se fait à l'aide de la fonction :

```
function connexionbdd()
{
   //Définition des variables de connexion à la base de données
   $bd_nom_serveur='localhost';
   $bd_login='projetBDD';
   $bd_mot_de_passe='trotinette2019';
   $bd_nom_bd='Trotinette';

   //Connexion à la base de données
   $id = mysqli_connect($bd_nom_serveur, $bd_login, $bd_mot_de_passe, $bd_nom_bd);
   mysqli_query($id, "set names 'utf8'");

   return $id;
}
```

Cette fonction se trouve dans le fichier fonction.php

Gestion des erreurs

La major partie des erreurs se produisant sur notre site sont géré à l'aide de la page information.php ont y envoi un tableau composé de l'erreur et des différents retours.

Difficultées

- Contrainte temps , on a pas eu assez de temps pour pouvoir pratiquer du html,css3 et surtout du php .

- Le changement du modèle relationnel de base, ajout et suppression des entitées dans différents tables pour les adaptés à nos besoin .