Rapport de projet

Table des matières

[1. Analyse préliminaire 3](#_Toc8282947)

[1.1 Introduction 3](#_Toc8282948)

[1.2 Organisation 3](#_Toc8282949)

[1.3 Objectifs 3](#_Toc8282950)

[2. Maquettage 3](#_Toc8282951)

[2.1 Login 3](#_Toc8282952)

[2.2 Inscription 4](#_Toc8282953)

[2.3 Accueil 5](#_Toc8282954)

[2.3.1 Réservation 5](#_Toc8282955)

[2.4 Profils d’un utilisateur 6](#_Toc8282956)

[2.5 Vue de tous les utilisateurs 7](#_Toc8282957)

[2.6 Paramètres administrateurs 8](#_Toc8282958)

[3. Analyse organique 9](#_Toc8282959)

[EUser 9](#_Toc8282960)

[Exemple d’utilisation 9](#_Toc8282961)

[Propriété 9](#_Toc8282962)

[Méthode 9](#_Toc8282963)

[EResevation 10](#_Toc8282964)

[Exemple d’utilisation 10](#_Toc8282965)

[Propriété 10](#_Toc8282966)

[Méthode 10](#_Toc8282967)

[ECourt 10](#_Toc8282968)

[Exemple d’utilisation 10](#_Toc8282969)

[Propriété 11](#_Toc8282970)

[Méthode 11](#_Toc8282971)

[ERole 11](#_Toc8282972)

[Exemple d’utilisation 11](#_Toc8282973)

[Propriété 11](#_Toc8282974)

[Méthode 11](#_Toc8282975)

# Analyse préliminaire

## 1.1 Introduction

Le site de réservation de courts de squash permet aux membres de réserver un court de squash. Pour pouvoir s’enregistrer, la personne doit spécifier : une adresse email valide, un mot de passe et un « nickname ». La validation de l’enregistrement se fait à l’aide d’un email d’activation envoyé à l’adresse email utilisée lors de l’enregistrement. L’administrateur du site est également informé par email de l’inscription. Un utilisateur peut compléter/modifier ses informations personnelles : Nickname, Nom, Prénom, Téléphone de contact, Description et Image du profil. L’accès à l’application est régi selon deux niveaux d’habilitation : Membre et Administrateur

Le code respecte un design MVC. Les technologies utilisées sont : PHP, HTML 5, CSS 3, Javascript.

## 1.2 Organisation

## 1.3 Objectifs

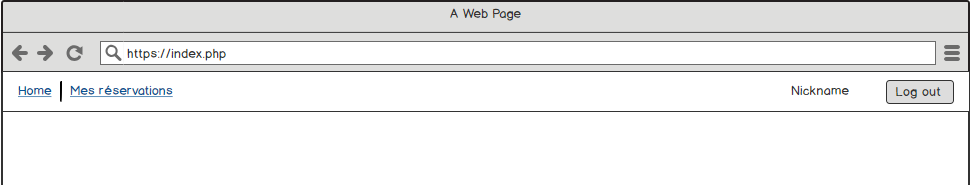
# Maquettage

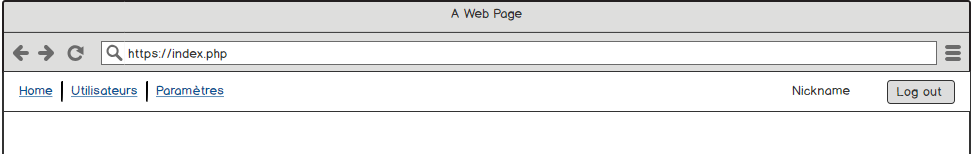
## Introduction

Le site de réservation de cours de squash est accessible uniquement aux utilisateurs enregistrés. Il existe deux types de profils au sein du site : membres et administrateurs. Les membres peuvent visualiser et réserver un court pour une période donnée

## Barre de menu

La barre de menu s’affiche en fonction du profil de l’utilisateurs

Pour les membres : 

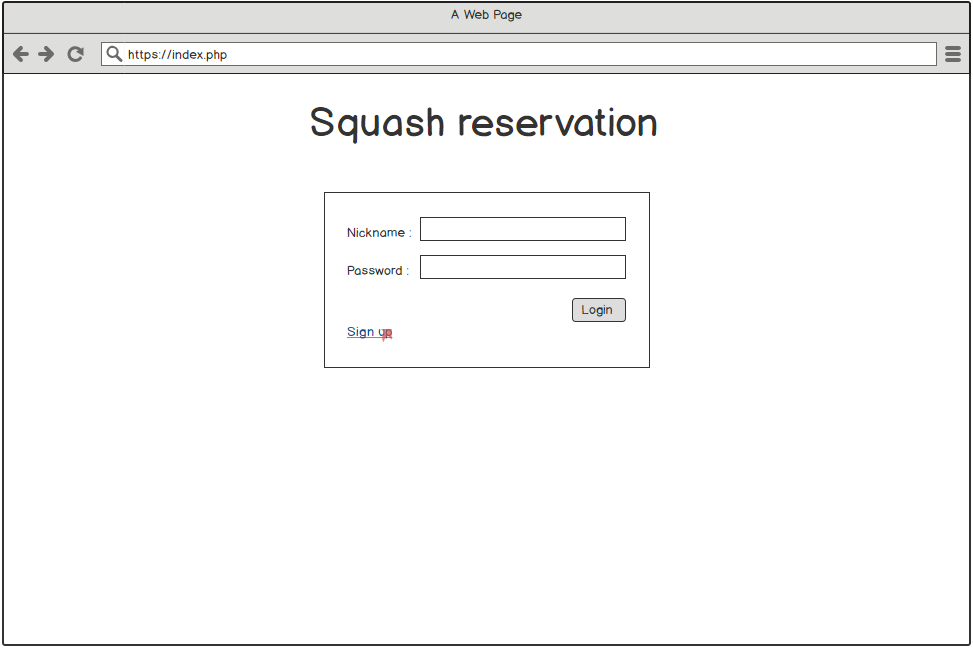
Pour les administrateurs : 

## Login

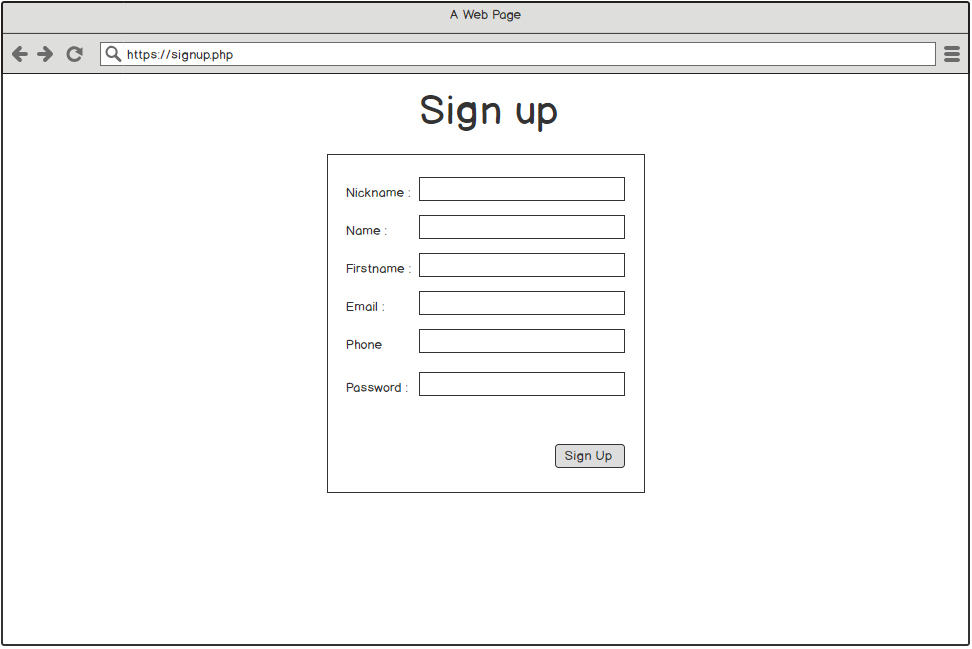
En arrivant dans le site, l’utilisateur étant non connecté, il est redirigé directement sur la page de login, tel ci-dessous

La page principale du site présente un formulaire de connexion du fait que le site est accessible uniquement aux utilisateurs enregistrés et authentifiés.

Si l’utilisateur n’est pas encore enregistré, il peut le faire depuis le lien Sign up.

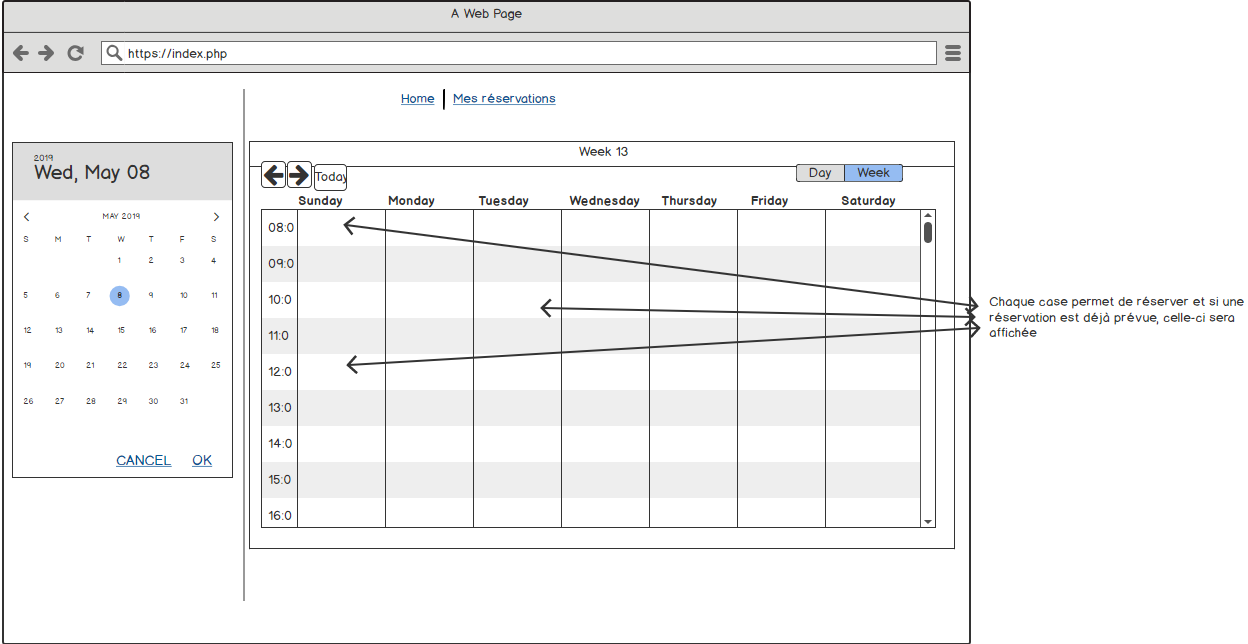


## Inscription

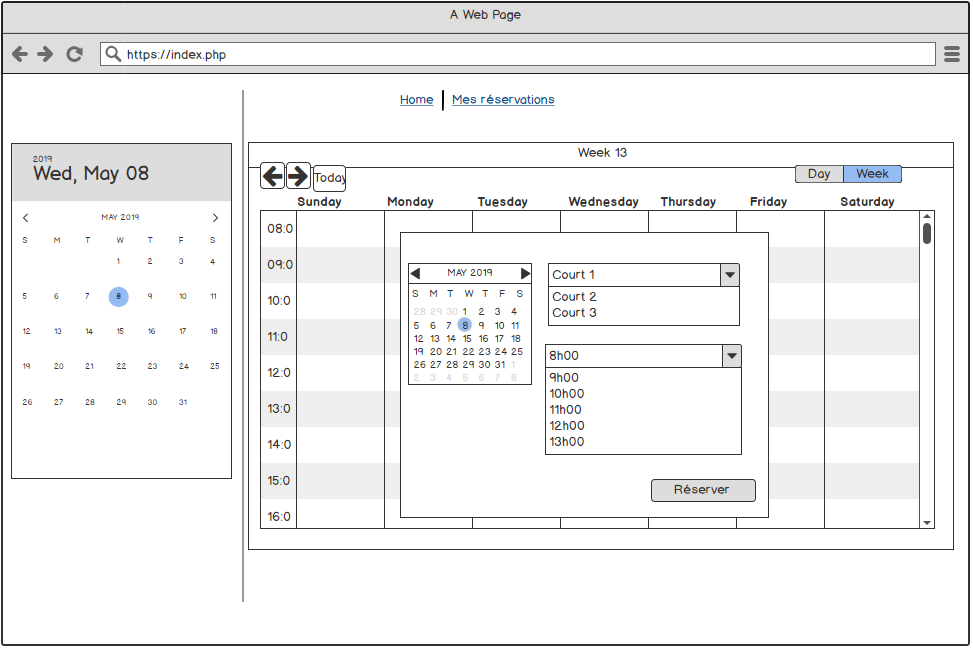
Si l’utilisateur n’est pas encore inscrit, il peut s’inscrire grâce à cette page :

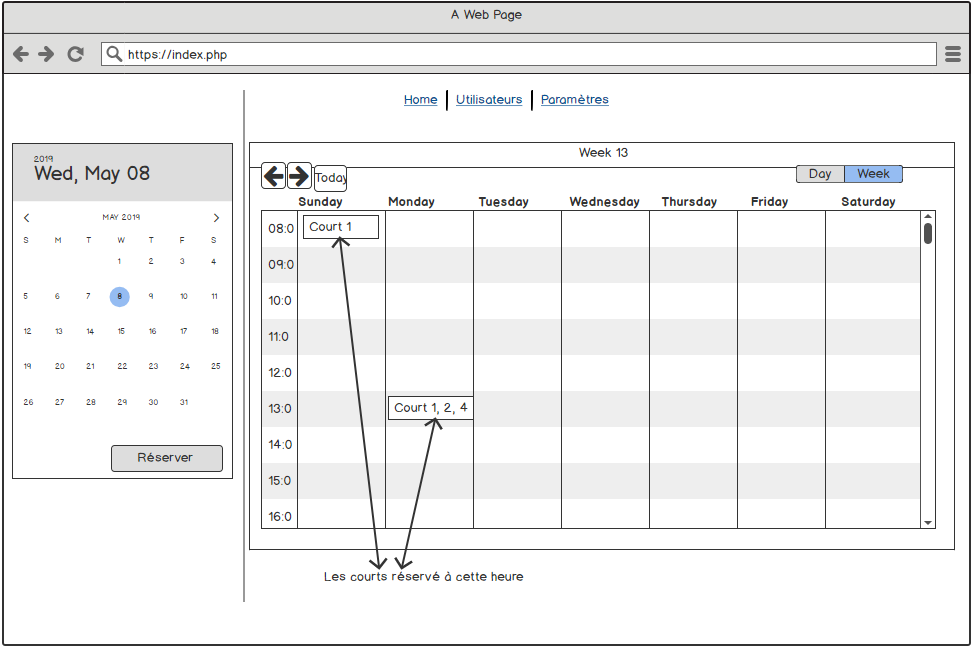
Attention, pour réserver un court, il faut confirmer son compte grâce à l’email qui aura été envoyé à l’adresse indiqué dans le champ « email ». Après avoir cliqué sur le bouton pour s’inscrire, l’utilisateur est redirigé vers la page ci-dessous pour qu’il aille confirmer son compte par email

## Accueil

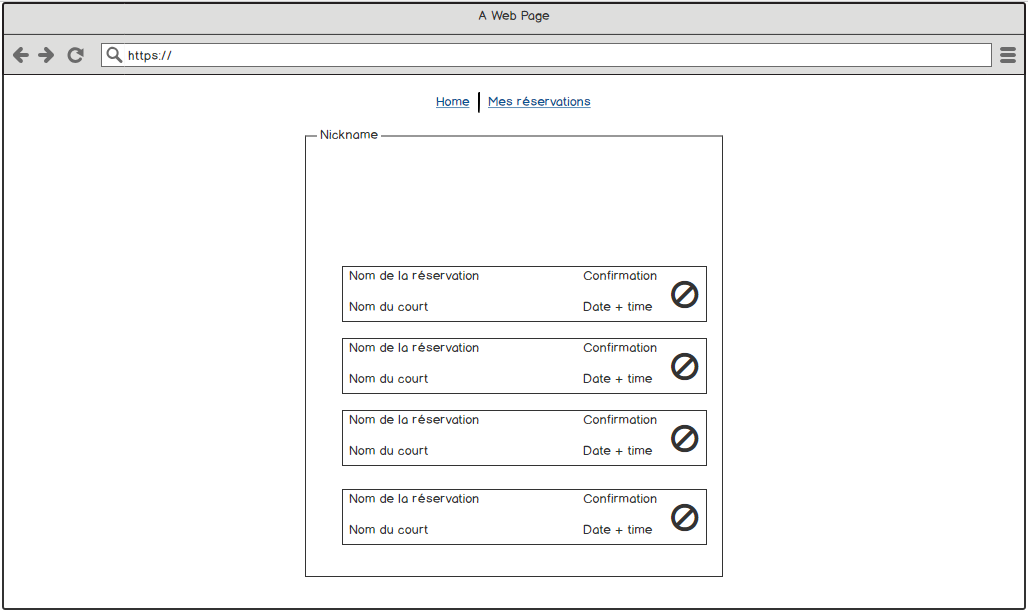
En se connectant, un utilisateur arrivera sur la page suivante : 

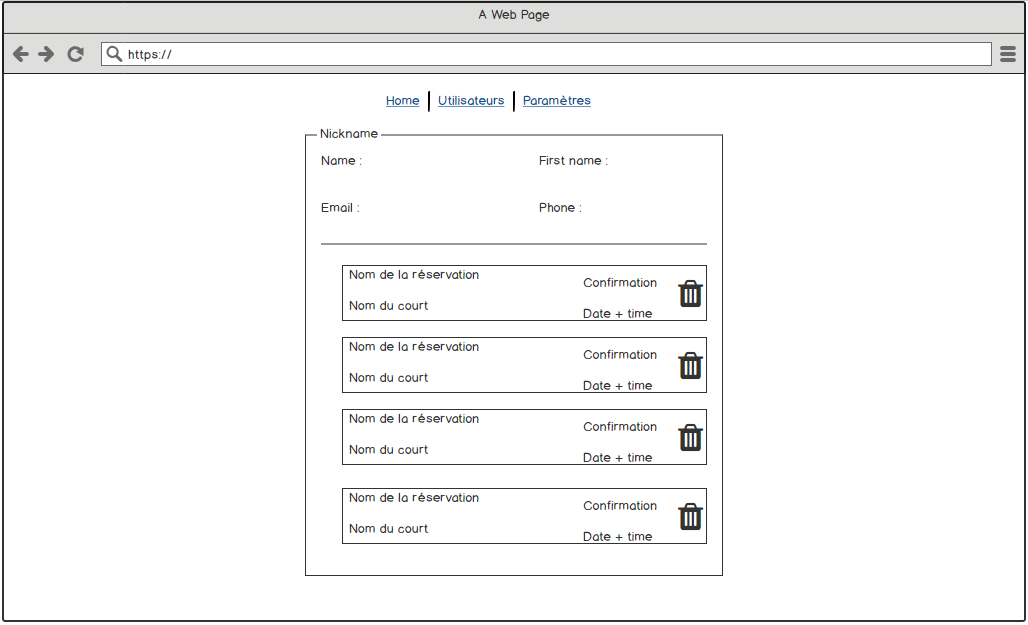
## Réservation

Quand on clique sur une case vide, une pop-up s’ouvre et nous permet de faire une réservation (La pop-up permet de changer la date et l’heure mais par défaut ces informations seront mis selon les données de la case sur laquelle on a cliqué) 

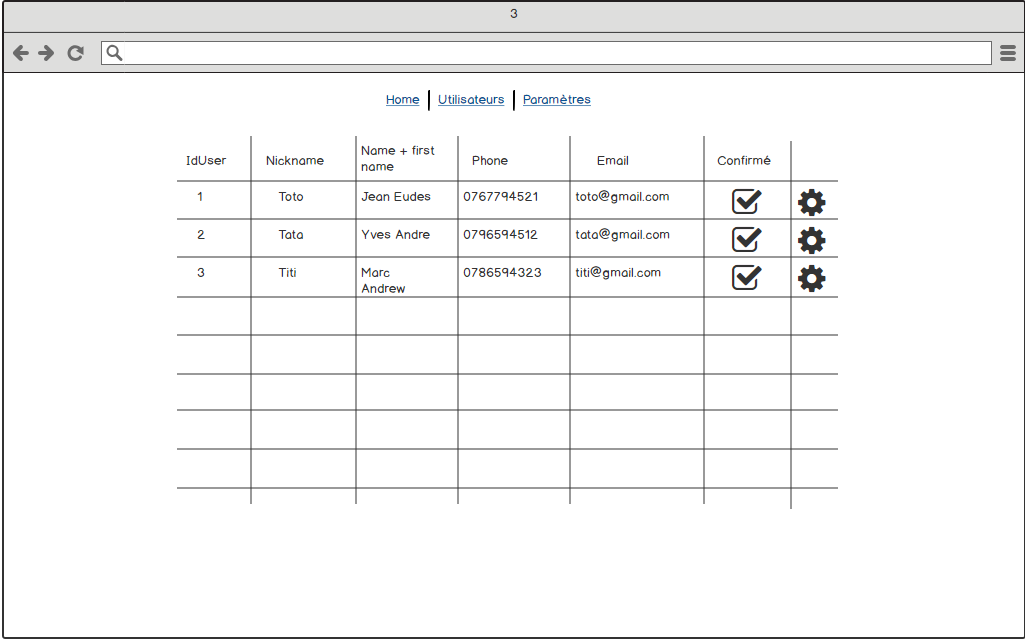
Si l’utilisateur connecté a également les droits d’administrateurs la barre de navigations changera tel que ci-dessous :

## Profils d’un utilisateur

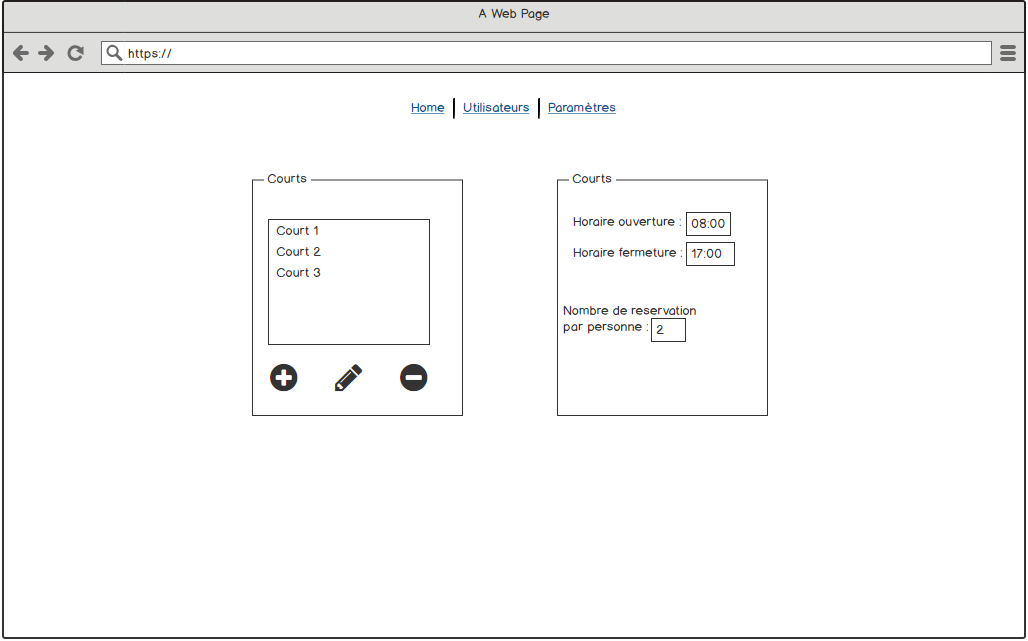
Cette page permet à un utilisateur de voir ses différentes réservations et il est possible d’annuler avant 24h du début de la réservation :

Cette page est aussi utilisée par les administrateurs qui, en allant sur un profil, peuvent supprimer les réservations : 

## Vue de tous les utilisateurs

Disponible seulement en ayant les droits administrateurs, cette page permet de voir tous les utilisateurs. Il permet également de confirmer l’inscription et si l’on veut le « dé confirmé » pour le bloquer. Enfin depuis cette page il peut accéder à n’importe quels profils et ainsi pouvoir supprimer les réservations. 

## Paramètres administrateurs

Cette page est accessible seulement aux administrateurs. Elle permet de gérer les différents cours. Elle permet aussi de gérer les horaires ainsi que le nombre de réservations que chaque utilisateur peut prendre par jour. 

# Analyse organique

## EUser

### Exemple d’utilisation

Cet objet est utilisé comme conteneur

//Exemple d'utilisation 1

$u = new EUser();

$u->Email = "loic@burnand.com";

$u->Nickname = "Test";

$u->Name = "BURNAND";

$u->FirstName = "Loïc";

$u->Phone = "0767756644";

$u->Role = new ERole();

$u->IsConfirmed = false;

//Exemple d'utilisation 2

$u = new EUser("loic@burnand.com", "Test", "BURNAND", "Loïc", "0767756644", new ERole(), false);

### Propriété

#### Email

String, L'email de l'utilisateur

#### Nickname

String, Le pseudo de l'utilisateur

#### Name

String, Le nom de l'utilisateur

#### FirstName

String, Le prénom de l'utilisateur

#### Phone

String, le numéro de téléphone de l’utilisateur

#### IdRole

Int, l'id du rôle de l'utilisateur

#### IsConfirmed

Bool, Confirmation de l’inscription de l’utilisateur

### Méthode

#### \_\_construct(string InEmail, string InNickname, string InName, string InFirstName, string InPhone, ERole InRole, Bool InCofirmation)

Le Constructor appelé au moment de la création de l'objet. new EUser();

@param InEmail L'email de l'utilisateur. (Optionel) Defaut ""

@param InNickname Le nickname de l'utilisateur. (Optionel) Defaut ""

@param InName Le nom de l'utilisateur. (Optionel) Defaut ""

@param InFirstName Le prénom de l'utilisateur. (Optionel) Defaut ""

@param InPhone Le numéro de téléphone de l'utilisateur. (Optionel) Defaut ""

@param InRole Le rôle de l'utilisateur. (Optionel) Defaut null

@param InCofirmation Est-ce que l'utilisateur est confirmé. (Optionel) Defaut false

## EResevation

### Exemple d’utilisation

Cet objet est utilisé comme conteneur

//Exemple d'utilisation 1

$u = new EResevation();

$u->Court = new ECourt();

$u->User = new EUser();

$u->IsConfirmed = false;

$u->Date = "";

//Exemple d'utilisation 2

$u = new EResevation(new ECourt(), new EUser(), false, "");

### Propriété

#### Court

ECourt, le terrain qui a été réservé

#### User

EUser, l’utilisateur qui a réservé

#### Date

DateTime, la date et l’heure du début de la réservation

### Méthode

#### \_\_construct(ECourt InCourt, EUser InUser, bool InConfirmed, DateTime InDate)

Le Constructor appelé au moment de la création de l'objet. new EResevation();

@param InCourt Le terrain réservé. (Optionel) Defaut null

@param InUser L'utilisateur qui a réservé. (Optionel) Defaut null

@param InConfirmed La confirmation de la réservation. (Optionel) Defaut false

@param InDate La date et l'heure de la réservation. (Optionel) Defaut ""

## ECourt

### Exemple d’utilisation

Cet objet est utilisé comme conteneur

//Exemple d'utilisation 1

$u = new ECourt();

$u->Name = "Court Test";

$u->Desc = "Ce terrain est un terrain reserver au entrainement";

$u->Deleted = true;

//Exemple d'utilisation 2

$u = new ECourt("Court Test", "Ce terrain est un terrain reserver au entrainement", true);

### Propriété

#### Name

String, Le nom du court

#### Desc

String, La description du court

#### Deleted

Bool, est-ce que le court est fermé ou pas

### Méthode

#### \_\_construct(string InName, string InDesc, bool InDeleted)

Le Constructor appelé au moment de la création de l'objet. new ECourt();

@param InEmail L'email du Court. (Optionel) Defaut ""

@param InDesc La description du court. (Optionel) Defaut ""

@param InDeleted Est-ce que le court est fermé ou pas. (Optionel) Defaut false

## ERole

### Exemple d’utilisation

Cet objet est utilisé comme conteneur

//Exemple d'utilisation 1

$u = new ERole();

$u->IdRole = 1;

$u->Name = "Admin";

//Exemple d'utilisation 2

$u = new ERole(1, "Admin");

### Propriété

#### IdRole

Int, L’id du rôle

#### Name

String, le nom du rôle

### Méthode

#### \_\_construct(Int InIdRole, String InName)

Le Constructor appelé au moment de la création de l'objet. new ERole();

@param InIdRole L'Id du rôle

@param InName Le nom du rôle. (Optionel) Defaut ""

## EPreferences

### Exemple d’utilisation

Cet objet est utilisé comme conteneur

//Exemple d'utilisation 1

$u = new EPreferences();

$u->Updated = new DateTime('2011-01-01T15:03');

$u->BeginTime = new DateTime('2019-01-01T15:00');

$u->EndTime = new DateTime('2019-01-01T16:00');

$u->NbReservation = 2;

### Propriétés

#### Updated

DateTime, Date et heure de la dernière update

#### BeginTime

DateTime, L'heure d'ouverture des courts

#### EndTime

DateTime l'heure de fermeture des courts

#### NbReservations

Int, Le nombre de réservation que l'utilisateur peut faire par jour

### Méthode

#### \_\_construct()

## ModelUsers

### Exemple d’utilisation

Cette classe est static et permet de gérer la table User

//Exemple d'utilisation 1

$users = ModelUsers::GetAllUsers();

//Exemple d'utilisation 2

$u = new EUser();

$u->Email = "loic@burnand.com";

$u->Nickname = "Test";

$u->Name = "BURNAND";

$u->FirstName = "Loïc";

$u->Phone = "0767756644";

$u->Role = new ERole();

$u->IsConfirmed = false;

ModelUsers::AddUser($u);

### Méthodes

#### GetAllUsers ()

Retourne le tableau des utilisateurs de type EUser

#### GetUserByNickname (String nickname)

Cherche un utilisateur par son Id dans la table. Retourne le premier utilisateur du résultat de la requête

##### Paramètres

Nickname Le nicknamecsous forme de string a recherché dans la base de donnée

#### AddUser (EUser User)

Ajoute un utilisateur dans la base de donnée.

##### Exemple

$u = new EUser();

$u->Email = "loic@burnand.com";

$u->Nickname = "Test";

$u->Name = "BURNAND";

$u->FirstName = "Loïc";

$u->Phone = "0767756644";

$u->Role = new ERole();

$u->IsConfirmed = false;

ModelUsers::AddUser($u);

#### UpdateUser (EUser User)

Permet de metter à jours les donnée d'un utilisateur. Retourne true si l'update est réussie, sinon retourne false

##### Paramètres

User l'utilisateur que l'on souhaite modifier

#### GetUserRole (EUser User)

Retourne le nom du rôle de l'utilisateur. La recherche se fait d’abord s’il y a, sur l’id du rôle stocké dans le paramètre User. S’il n’y est pas, il va chercher l’id en ayant cherché l’utilisateur grâce à la fonction GetUserByNickname pour ensuite chercher le rôle.

##### Exemple

// Exemple 1

$u = new EUser();

$u->IdRole = "1";

$roleName = ModelUsers::GetUserRole($u);

// Exemple 2

$u = new EUser();

$u->nickname = "Test";

$roleName = ModelUsers::GetUserRole($u);

## ModelRoles

### Exemple d’utilisation

// Exemple d'utilisation 1

$roles = ModelRoles::GetAllRoles();

// Exemple d'utilisation 2

$role = ModelRoles::GetRolesById(1);

### Méthodes

#### GetAllRoles ()

Retourne un tableau de ERole contenant tous les rôles stockés en base de donnée

#### GetRolesById ($Id)

Chercher un role par son Id dans la table. Retourne le premier rôle du résultat de la requête.

##### Paramètres

Id L'id a recherché dans la base de donnée

## Modèle de donnée

