

HEH Campus Technique

Projet PHP

Réalisation d'un site basé sur le jeu Hearthstone

Titulaires : MALAISE Antoine & SCOPEL Fabrice

Arnaud Urbain
02/06/2017

HEH Campus Technique

Projet PHP

Réalisation d'un site basé sur le jeu Hearthstone

Titulaires : MALAISE Antoine & SCOPEL Fabrice

Table des matières

| | |
|---------------------------------------|----|
| Présentation générale du projet | 6 |
| Base de données | 7 |
| Premier diagramme UML | 7 |
| Deuxième diagramme UML..... | 8 |
| Site : arborescence du projet | 9 |
| Page d'accueil | 9 |
| Formulaire d'inscription..... | 9 |
| Formulaire de connexion..... | 10 |
| Gestion des utilisateurs..... | 10 |
| Cartes..... | 11 |
| Decks | 11 |
| Header / contact admin..... | 11 |
| Scripts MYSQL..... | 12 |
| Conclusion..... | 14 |

Présentation générale du projet

L'objectif principal de ce projet est de réaliser un site dont le fonctionnement est dynamique, c'est-à-dire que toutes les données proviennent et sont enregistrées dans une base de données.

Le choix du projet étant libre, je me suis orienté vers la création d'un site basé sur le jeu Hearthstone (Heroes of Warcraft).

Sur ce site, les utilisateurs inscrits ont accès à une base de données comprenant 120 cartes, toutes classes confondues, tirées tout droit de l'univers de leur jeu favori.

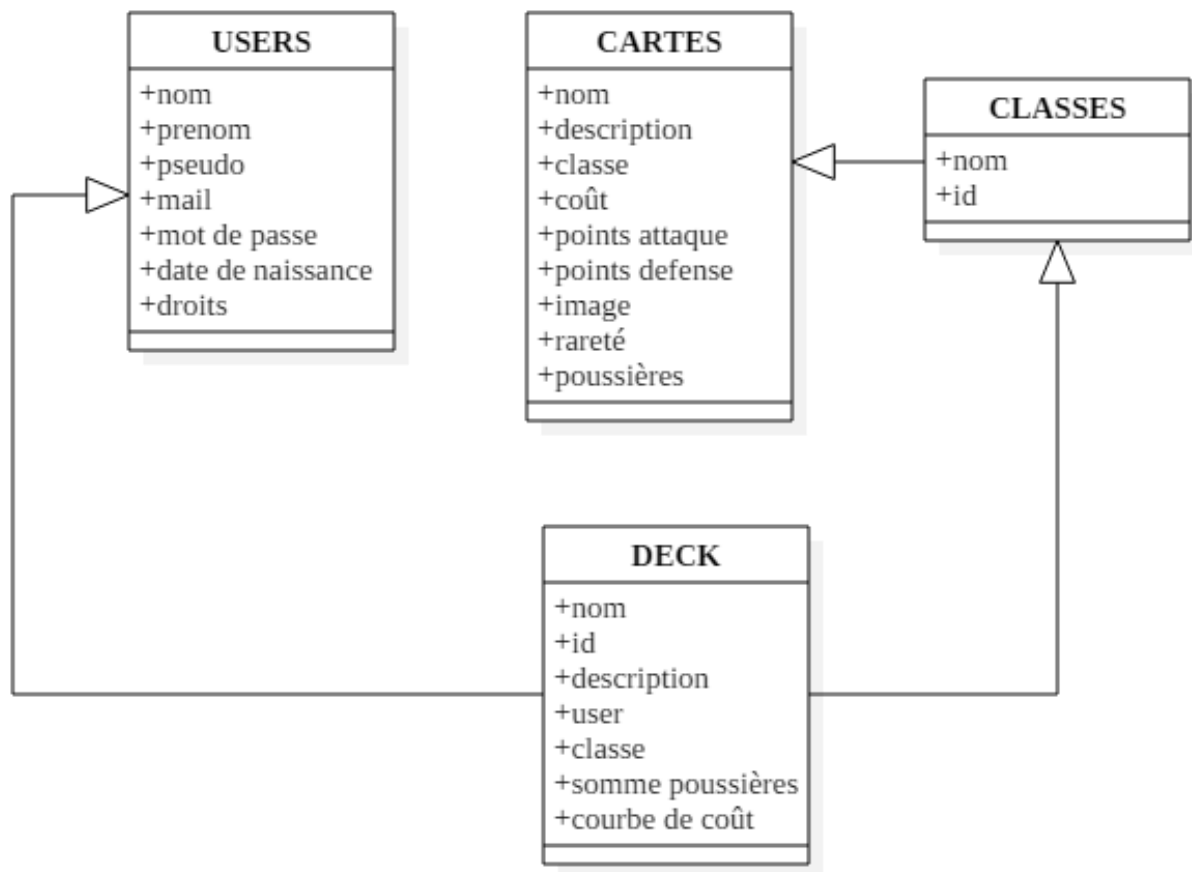
Ces cartes leur permettent de créer leur propre deck (30 cartes) et de le partager avec le reste de la communauté.

Les utilisateurs possèdent chacun leur propre profil qu'ils peuvent éditer à n'importe quel moment !

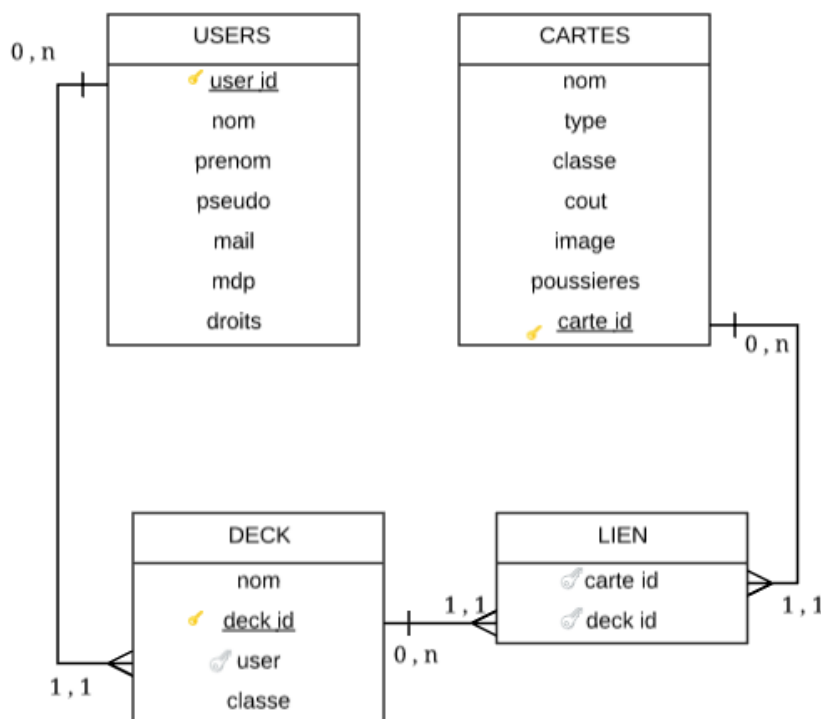
Un onglet contact leur permet de communiquer leurs impressions et des suggestions afin d'améliorer le site.

Base de données

Premier diagramme UML



Deuxième diagramme UML

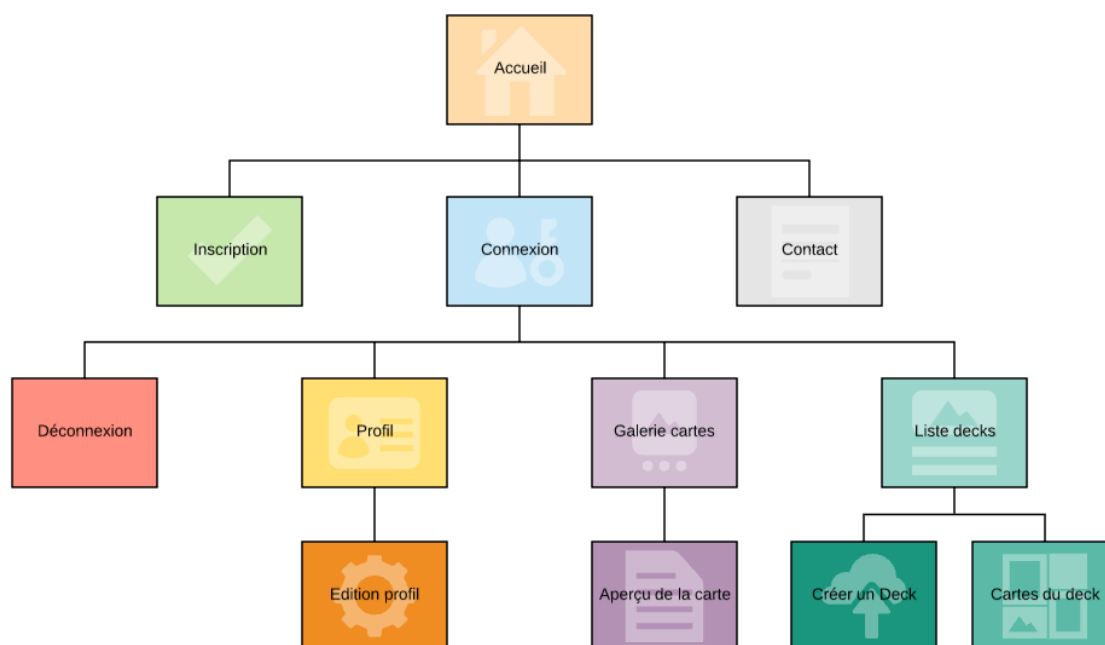


L'idée de départ, pour la base de données, était d'enregistrer les utilisateurs selon tous les caractéristiques reprises dans la table 'USERS' ci-dessus. Par la suite, j'ai supprimé le champ 'date de naissance' car je ne le trouvais pas utile pour ce projet. J'ai cependant ajouté le champ 'user_id' qui est devenu la clé primaire ('remplaçant le mail dans ce rôle') de ma table et qui s'auto-incrémente à chaque création d'un compte utilisateur (l'admin possède l'user_id 1).

Pour la table 'CARTES', j'ai supprimé le champ 'description', 'points d'attaque' et 'points défense' car après avoir commencé l'encodage des cartes dans la BDD, je me suis rendu compte que ces champs n'étaient pas nécessaires au bon fonctionnement du site (les points d'attaque défense n'étant présents que sur les créatures). Un champ 'type' a vu le jour et définit si la carte est soit une créature, soit un sort ou une arme. Un autre champ a été ajouté, le 'carte_id' qui remplace le 'nom' en tant que clé primaire et qui s'auto-incrémente à chaque carte.

La table 'classe' a été supprimée car inutile pour le projet (non-utilisée). Une nouvelle table 'lien' relie les tables DECK et CARTES ensemble. Finalement, les champs 'description' et 'courbe de coût' ont eux aussi été supprimés pour les mêmes raisons citées ci-dessus.

Site : arborescence du projet



Page d'accueil

La page d'accueil est on ne peut plus simple : un titre et un message invitant les utilisateurs à s'inscrire sur le site via l'onglet 'Inscription'. Trois onglets sont disponibles pour un utilisateur non connecté : connexion, inscription et contact.

Formulaire d'inscription

Le formulaire d'inscription demande des informations à l'utilisateur comme son nom, son prénom, un pseudo (pour se connecter), son adresse mail et un mot de passe. Chacun des champs est sécurisé et vérifie que l'utilisateur respecte un regex défini dans le code PHP. Ces regex permettent de s'assurer que l'utilisateur n'utilise pas de chiffres ou de caractères spéciaux dans son nom et prénom, le mot de passe doit faire entre 8 et 16 caractères et peut contenir des lettres (minuscules comme majuscules) ainsi que des chiffres.

Un champ d'avertissement est mis en place pour prévenir l'utilisateur de toutes erreurs commises dans ce formulaire.

Le pseudo étant le champ utilisé lors de la connexion, une vérification lors de l'inscription permet de voir si le pseudo choisi est déjà utilisé par quelqu'un d'autre (idem pour l'e-mail).

Formulaire de connexion

Le formulaire de connexion est composé de deux champs : le pseudo et le mot de passe de l'utilisateur. Si les valeurs des deux champs existent dans la base de données et qu'elles correspondent, la connexion est autorisée et une redirection s'effectue vers le profil de l'utilisateur. Dans le cas contraire, un message d'avertissement apparaît, expliquant que les champs saisis ne sont pas corrects.

C'est vers le formulaire de connexion qu'une redirection se fait si quelqu'un ne possédant pas de droits essaie d'afficher l'une ou l'autre page du site.

Gestion des utilisateurs

A la création du compte, un utilisateur est identifié dans la base de données par un id qui s'incrémente automatiquement. Cet id est la clé primaire de la table 'users' et est utilisée pour définir le contenu appartenant à l'utilisateur connecté. Lors de la connexion d'un utilisateur, l'id de celui-ci va être stocké dans la variable \$_SESSION['id'] qui permet de savoir qui est connecté afin de lui afficher son propre contenu (profil, etc...).

Un utilisateur lambda possède des droits normaux, lui permettant d'accéder au contenu du site comme les autres utilisateurs lambda.

Un admin est créé lors de la mise en place de la base de données, celui-ci possède des droits supérieurs aux utilisateurs lambda et a donc accès à des outils de modération.

L'admin peut modifier les informations des autres utilisateurs, il peut bannir et 'débannir' voir même élire un utilisateur lambda comme modérateur du site.

Lorsqu'un utilisateur souhaite se désinscrire, tout le contenu lui appartenant (decks, etc...) sont supprimés en cascade dans la base de données et la session est détruite ce qui ramène l'utilisateur à la page d'accueil.

Cartes

La table ‘cartes’ de la base de données est remplie de 120 cartes possédant chacune des caractéristiques qui lui sont propres comme un nom, un coût, etc...

L’affichage des cartes se fait à l’aide d’une boucle qui va parcourir et afficher chacun des éléments compris dans la table. Celles-ci sont tout d’abord triées par coût et nom en ordre croissant.

Deux filtres sont proposés aux utilisateurs : un leur permettant de définir la classe et l’autre le type des cartes à afficher. Ces deux filtres peuvent être utilisés séparément ou simultanément.

Si l’utilisateur clique sur une des cartes, celui-ci est redirigé vers une page d’aperçu. Sur cette page, plusieurs informations sont ajoutées à l’image de la carte comme le nom, la rareté, le coût de création en poussières, etc...

Decks

La gestion des decks se fait en plusieurs étapes. Tout d’abord, l’onglet ‘Decks’ affiche tous les decks qui ont été créés par les utilisateurs inscrits sur le site. L’affichage donne comme information : le nom, la classe et le nom de l’auteur du deck. Il est possible de le consulter afin de voir les cartes composant ce deck. Le créateur d’un deck peut le supprimer via cette page.

La création d’un deck se passe en quelques étapes. Tout d’abord l’utilisateur entre un nom pour son deck et choisit la classe qui le définit. Le deck est créé (vide) et l’utilisateur est redirigé vers une page lui montrant toutes les cartes disponibles pour sa sélection de classe. L’utilisateur peut entrer 0 à 30 cartes dans son deck. Un message s’affiche lorsque la limite des 30 cartes est atteinte et l’ajout de carte n’est plus possible.

Header / contact admin

Le fichier header.php est inclus dans la plupart des autres fichiers car il exécute la commande ‘session_start() ;’ qui crée la session. Lorsqu’une personne consulte le site, son \$_SESSION[‘droits’] est défini à 0, ce qui me permet d’empêcher tout visionnage du contenu du site par un utilisateur non connecté (utilisateur connecté défini à 1).

Les onglets de navigation sont affichés depuis ce fichier. Certains onglets ne sont disponibles que pour les utilisateurs connectés.

Ce fichier comporte aussi les méta-balises html définissant le format d’encodage, l’auteur et qui créent les liens css utiles à l’affichage des éléments.

Scripts MYSQL

```

1  CREATE DATABASE IF NOT EXISTS project CHARACTER SET utf8
2  COLLATE utf8_general_ci;
3
4  USE project;
5
6
7  CREATE TABLE users (
8      user_id INT NOT NULL AUTO_INCREMENT,
9      nom VARCHAR(255) NOT NULL,
10     prenom VARCHAR(255) NOT NULL,
11     pseudo VARCHAR(255) NOT NULL,
12     mail VARCHAR(255) NOT NULL,
13     mdp TEXT NOT NULL,
14     droits INT NOT NULL DEFAULT 1,
15     CONSTRAINT pk_users PRIMARY KEY(user_id)
16 )ENGINE=InnoDB;
17
18 CREATE TABLE cartes (
19     nom VARCHAR(255) NOT NULL,
20     type INT NOT NULL,
21     classe VARCHAR(255) NOT NULL,
22     cout VARCHAR(255) NOT NULL,
23     image VARCHAR(255) NOT NULL,
24     poussieres INT NOT NULL,
25     carte_id INT NOT NULL AUTO_INCREMENT,
26     CONSTRAINT pk_cartes PRIMARY KEY(carte_id)
27 )ENGINE=InnoDB;
28
29 CREATE TABLE deck (
30     nom VARCHAR(255) NOT NULL,
31     deck_id INT NOT NULL AUTO_INCREMENT,
32     user INT NOT NULL,
33     classe VARCHAR(255) NOT NULL,
34     CONSTRAINT pk_deck PRIMARY KEY (deck_id),
35     CONSTRAINT FK_deck FOREIGN KEY (user) REFERENCES users(user_id) ON DELETE CASCADE
36 )ENGINE=InnoDB;
37
38 CREATE TABLE lien (
39     carte_id INT NOT NULL,
40     deck_id INT NOT NULL,
41     CONSTRAINT FK_lien FOREIGN KEY (deck_id) REFERENCES deck(deck_id) ON DELETE CASCADE
42 )ENGINE=InnoDB;

```

Je commence en créant la base de données ‘projet’ et en définissant le format d’encodage UTF-8.

Je crée ensuite les 4 tables dont j’ai besoin en y indiquant le nom et le type de chacune des colonnes. Chacune des tables possède sa clé primaire (sauf ‘lien’). Des clés étrangères sont définies dans les tables cartes, deck et lien ce qui me permet d’effectuer une suppression du contenu en cascade lorsqu’un utilisateur se désinscrit du site.

```

1  INSERT INTO users (nom, prenom, pseudo, mail, mdp, droits) VALUES (
2  [      'Urbain', 'Arnaud', 'Wanxed', 'arnaud-3@outlook.com', SHA1('XXXXXXXXXX'), 2);
3
4  TRUNCATE TABLE cartes;
5  INSERT INTO cartes (nom, type, classe, cout, image, poussieres) VALUES
6      ('Eclair de jade', 0, 'Chaman', '04', 'img/1.jpg', 40),
7      ('Volcan', 0, 'Chaman', '05', 'img/2.jpg', 100),
8      ('Esprit farouche', 0, 'Chaman', '03', 'img/3.jpg', 100),
9      ('Griffes de jade', 2, 'Chaman', '02', 'img/4.jpg', 100),
10     ('Eauracle jinyu', 1, 'Chaman', '04', 'img/5.jpg', 100),
11     ('Chef de jade', 1, 'Chaman', '07', 'img/6.jpg', 40),
12     ('Totem Langue de feu', 1, 'Chaman', '02', 'img/7.jpg', 0),
13     ('Chose venue d'en bas', 1, 'Chaman', '06', 'img/8.jpg', 100),
14     ('Regard de nacre', 1, 'Chaman', '05', 'img/9.jpg', 1600),
15     ('Al\Akir, seigneur des vents', 1, 'Chaman', '08', 'img/10.jpg', 1600),
16     ('Tir réflexe', 0, 'Chasseur', '02', 'img/11.jpg', 40),
17     ('Tir explosif', 0, 'Chasseur', '05', 'img/12.jpg', 100),
18     ('Compagnon animal', 0, 'Chasseur', '03', 'img/13.jpg', 0),
19     ('Arc long du gladiateur', 2, 'Chasseur', '07', 'img/14.jpg', 400),
20     ('Elekk du roi', 1, 'Chasseur', '02', 'img/15.jpg', 40),
21     ('Hyène charognarde', 1, 'Chasseur', '02', 'img/16.jpg', 40),
22     ('Gentille grand-mère', 1, 'Chasseur', '02', 'img/17.jpg', 40),
23     ('Grande crinière des savanes', 1, 'Chasseur', '06', 'img/18.jpg', 100),
24     ('Roi du marais Dred', 1, 'Chasseur', '07', 'img/19.jpg', 1600),
25     ('Roi Krush', 1, 'Chasseur', '09', 'img/20.jpg', 1600),
26     ('Potion rougefurie', 0, 'Demoniste', '03', 'img/21.jpg', 100),
27     ('Flammes infernales', 0, 'Demoniste', '04', 'img/22.jpg', 0),
28     ('Plaie funeste', 0, 'Demoniste', '05', 'img/23.jpg', 400),
29     ('Néant distordu', 0, 'Demoniste', '08', 'img/24.jpg', 400),
30     ('Diablotin de Malchezaar', 1, 'Demoniste', '01', 'img/25.jpg', 40),
31     ('Chef du gang des diabolotins', 1, 'Demoniste', '03', 'img/26.jpg', 40),
32     ('Garde funeste', 1, 'Demoniste', '05', 'img/27.jpg', 100),
33     ('Trafiquante de la Kabale', 1, 'Demoniste', '06', 'img/28.jpg', 400),
34     ('Krul le Déchaîné', 1, 'Demoniste', '09', 'img/29.jpg', 1600),
35     ('Seigneur Jaraxxus', 1, 'Demoniste', '09', 'img/30.jpg', 1600),

```

Je commence par insérer le compte administrateur dans la base de données (avec des droits définis par la valeur 2). La ligne 4 vide la table 'cartes' afin de m'assurer qu'aucune entrée enregistrée auparavant ne puisse fausser ma table. J'encode ensuite les 120 cartes qui seront disponibles à l'affichage des cartes et lors de la création de deck.

Le coût des cartes est défini au format 01, 02, 03, ... car lors de l'affichage de l'ensemble des cartes, j'effectue une requête en ORDER BY cout ASC. Cette requête effectue un tri par ordre alphabétique sur la colonne cout. Ce tri voit la colonne comme une chaîne de caractère et regarde la valeur du premier élément de la chaîne avant de passer au suivant. Si je défini le format en 1, 2, 3, ... une carte dont le coût est supérieur à 9, se place entre la dernière carte de coût '1' et la première de coût '2' car le premier élément de la chaîne '10' est '1' (qui est plus petit que 2).

Conclusion

Pour conclure ce projet, j'ai découvert un langage dynamique permettant de réaliser une multitude de choses très intéressantes. Les différents projets déjà réalisés dans le domaine web et développement ne m'ont pas apporté autant de satisfaction que celui-ci. Le simple fait de pouvoir afficher toutes les cartes enregistrées dans la base de données selon certains filtres (conditions) m'ont émerveillé !

Durant mon travail, j'ai vu mon projet évoluer dans différentes directions prouvant que les possibilités de réaliser une action sont multiples.

Pour ce qui est des objectifs fixés, j'en ai réalisé la plupart comme je le pensais, notamment l'affichage de l'ensemble des cartes et la gestion des utilisateurs, tout en me rendant compte que plusieurs conditions devaient être respectées afin de protéger l'intégrité des données (les regex, la fonction htmlspecialchars, isset, ...).

Certaines fonctionnalités du site ne sont pas encore implantées mais l'idée est bien présente et j'ai pris conscience que la charge de travail que je m'étais fixée était bien plus conséquente que prévu.

Voici les fonctionnalités que j'envisagerai d'ajouter pour ce travail :

- La récupération d'un mot de passe,
- L'esthétisme du site,
- La modification d'un deck,
- La suppression de cartes dans un deck,
- Un compteur montrant le nombre de copie présente dans le deck,
- L'ajout de la carte dans un tableau représentant le deck en cours de remplissage,
- Un espace commentaire sous chaque deck qui permettrait à la communauté de discuter sur les choix effectués,
- Une page permettant à l'administrateur d'ajouter des cartes dans la base de données.

