23/12/2016

Christophe Sadowski

CFPT Informatique

Project M306

C#/Sqlite

Table des matières

[Introduction 2](#_Toc470707022)

[Organisation 2](#_Toc470707023)

[Objectif 2](#_Toc470707024)

[Analyse 2](#_Toc470707025)

[Analyse concurrentielle 2](#_Toc470707026)

[Cahier des charges détaillé 2](#_Toc470707027)

[Définition de l’audience 2](#_Toc470707028)

[Définition du contenu et de fonctionnalités 2](#_Toc470707029)

[Détermination des points faibles 3](#_Toc470707030)

[Stratégie de tests 3](#_Toc470707031)

[Planification détaillée 3](#_Toc470707032)

[Conception 4](#_Toc470707033)

[Analyse de l’environnement 4](#_Toc470707034)

[Définition de la charte graphique 4](#_Toc470707035)

[Maquette graphique 4](#_Toc470707036)

[Conception de la base de données 6](#_Toc470707037)

[Interrogation 7](#_Toc470707038)

[Insertion 7](#_Toc470707039)

[Modification 7](#_Toc470707040)

[Suppression 7](#_Toc470707041)

[Réalisations 8](#_Toc470707042)

[Dossier de réalisation 8](#_Toc470707043)

[Description des tests 8](#_Toc470707044)

[Erreurs restantes 8](#_Toc470707045)

[La vie du projet 9](#_Toc470707046)

[Conclusion 9](#_Toc470707047)

# Introduction

Pour ce projet, j’ai décidé de m’inspirer d’un projet que j’ai réalisé dans le cours de Mme Weber, qui consistait à faire un site web. Je me suis donc servit du même sujet, hearthstone.

Mon application permet donc de voir une liste de cartes, et de voir les informations spécifiques à la carte choisit. En plus de ceci, une fois connecté, l’administrateur peut créer, modifier ou bien de supprimer une carte.

# Organisation

Élève : Sadowski Christophe, [christophe.sdwsk@eduge.ch](mailto:christophe.sdwsk@eduge.ch)

# Objectif

* Afficher les cartes contenues dans la base de données.
* Se connecter pour avoir accès au fonctions suivante :
  + Ajouter des cartes dans la base de données grâce à l’interface C#.
  + Modifier la carte sélectionnée grâce à l’interface C#.
  + Supprimer la carte sélectionnée grâce à l’interface C#.

# Analyse

## Analyse concurrentielle

Sources :

* http://www.hearthstone-decks.com/carte

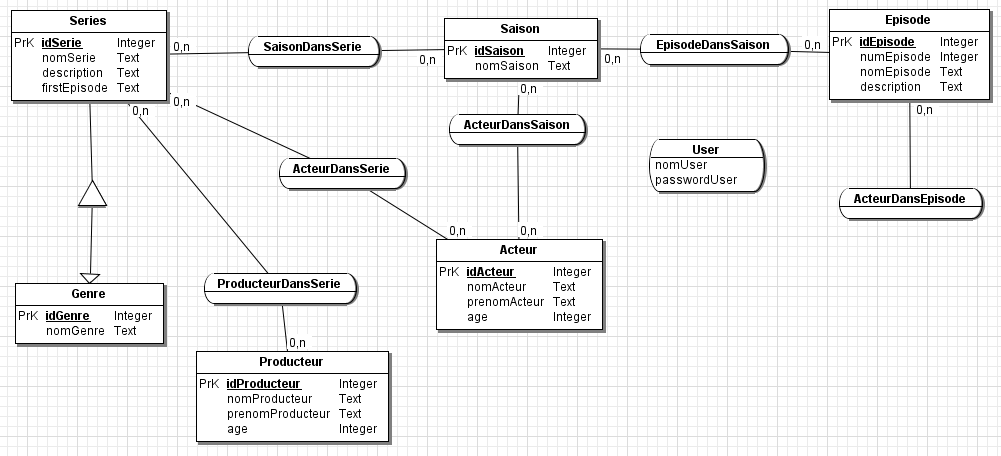
# Cahier des charges détaillé

## Définition de l’audience

Les cibles des personnes sont les personnes qui recherchent des informations sur des cartes hearsthtone.

## Définition du contenu et de fonctionnalités

* Fonctions :
  + Fonction 1 : Connexion entre base de données et l’application C#.
  + Fonction 2 : Afficher les cartes dans la forme frmCarte.
  + Fonction 3 : Ajouter des cartes dans la forme frmAjouter.
  + Fonction 4 : Modifier des cartes dans forme frmModifier.
  + Fonction 5 : Supprimer des cartes dans la forme frmCarte.
  + Fonction 6 : Se connecter dans la forme frmConnexion.
* MCD :



## Détermination des points faibles

Ajouter une base de données SQLite dans un programme C#.

## Stratégie de tests

* Tests :
  + Test 1 : Affichage des divers objets (cartes et informations)
    - Démarrer l’application
    - Choisir une carte dans la liste et cliquer dessus pour voir les informations de celle-ci
  + Test 2 : Ajouter une carte :
    - Démarrer l’application et se connecter
    - Cliquer sur « Ajouter »
    - Cliquez sur « Valider » après avoir remplis et sélectionner les champs
  + Test 3 : Modifier une carte :
    - Démarrer l’application et se connecter
    - Cliquez sur « Modifier » après avoir choisi une carte
    - Cliquez sur « Valider » après avoir modifier les champs
  + Test 4 : Supprimer une carte :
    - Démarrer l’application et se connecter
    - Cliquez sur « Supprimer » après avoir choisi une carte.
* Pas de navigateur
* Données de test à prévoir ? Oui
* Intervention de testeurs externes éventuels ? Oui, 1 testeur viendra tester le programme.

## Planification détaillée

Voir le fichier « SADOWSKI\_Planning.xlsx »

# Conception

## Analyse de l’environnement

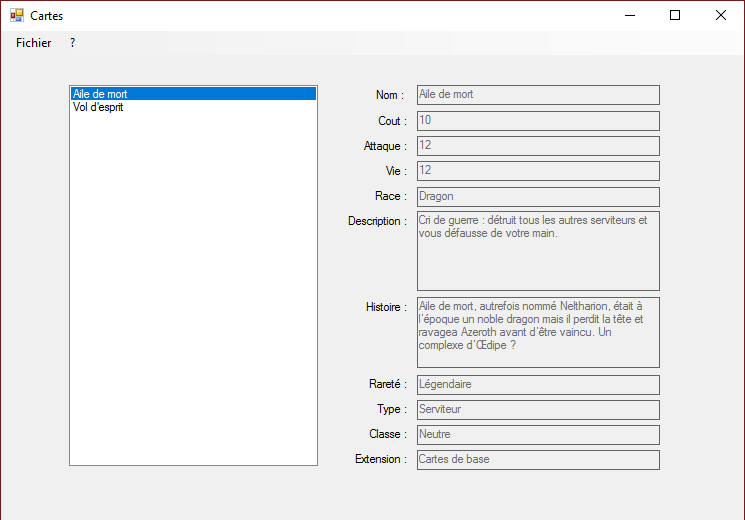
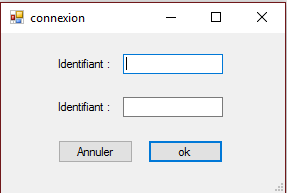
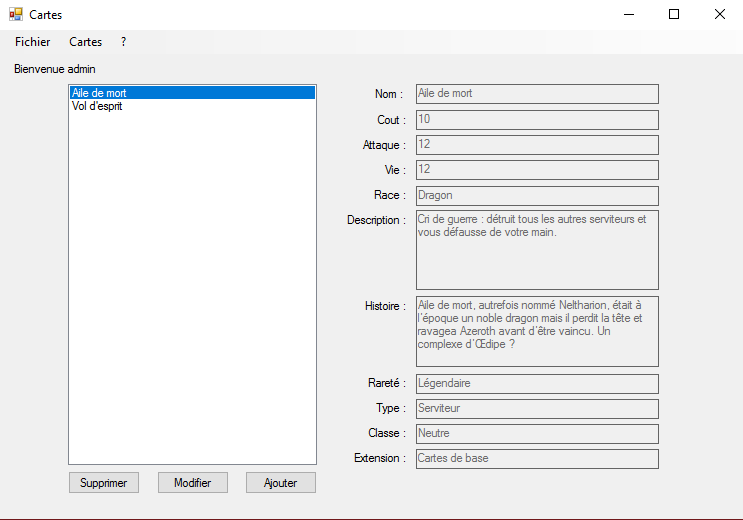
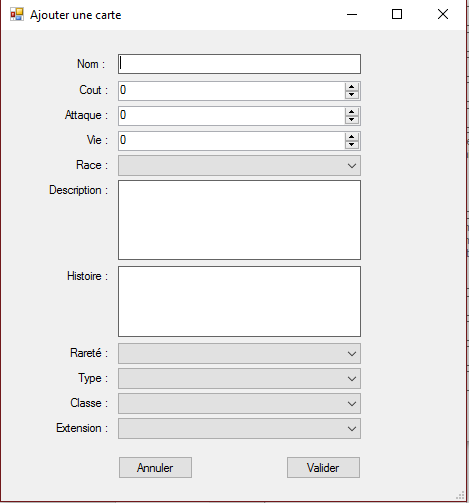
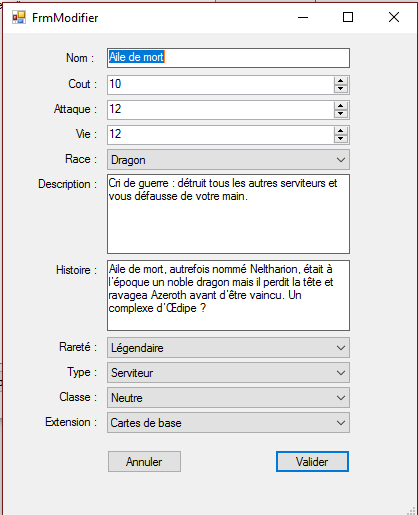
Logiciels utilisés :

* JMerise
* Visual Studio
* DB Browser

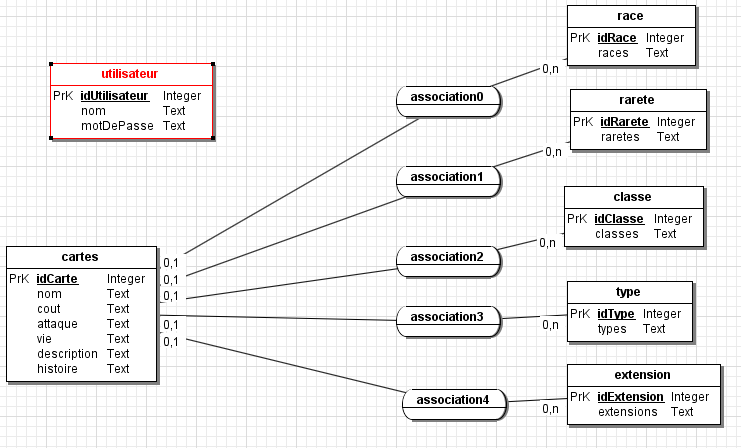
## Définition de la charte graphique

Design de base.

## Maquette graphique

## Conception de la base de données



La base de données comprendra 7 tables et 5 tables de liaisons.  
La table 1, cartes, comprendra toutes les informations relatives aux cartes.  
La table 2, race, comprendra toutes les races.  
La table 3, rarete, comprendra toutes les raretés.  
La table 4, classe, comprendra toutes les classes.  
La table 5, type, comprendra tous les types.  
La table 6, extension, comprendra toutes les extensions.   
La table 7, utilisateur, comprendra toutes les informations relatives aux utilisateurs.

Toutes les associations permettent de lier les tables 2,3,4,5,6 à la table cartes.

## Interrogation

**Requête :** SELECT \* FROM cartes WHERE nom = " Aile de mort" ;  
**Explication :** Sélectionne toutes les données relatives à la carte selon son nom.

**Requête :** SELECT \* FROM cartes ;  
**Explication :** Sélectionne toutes les données relatives à la table cartes.

**Requête :** SELECT classes FROM classe WHERE idClasse = "1" ;  
**Explication :** Sélectionne toutes les données relatives à la classe selon son id.

**Requête :** SELECT extensions FROM extension WHERE IdExtension = "1" ;  
**Explication :** Sélectionne toutes les données relatives à l’extension selon son id.

## Insertion

**Requête :** INSERT INTO cartes VALUES (null,"Aile de mort",1,1,1,"Description de la carte", "Histoire de la carte",1,1,1,1,1) ;  
**Explication :** Ajoute une carte en fonction des informations entrées.

## Modification

**Requête :** UPDATE cartes SET nom = "Aile de mort", cout = 1, attaque = 1, vie = 1, description = "Description de la carte", histoire = "Histoire de la carte", idRace = 1, idRarete = 1, idClasse = 1, idType = 1 WHERE idCarte = 1 ;  
**Explication :** Modifie une carte en fonction des informations entrées.

## Suppression

**Requête :** DELETE FROM cartes WHERE nom = " Aile de mort" ;  
**Explication :** Supprime toutes les données relatives à la carte selon son nom.

# Réalisations

## Dossier de réalisation

Le projet est dans un fichier Visual Studio

* **WF\_Cartes** – Fichier contenant tous les fichiers
  + **Bin** – Fichier contenant plusieurs fichiers
    - **Debug** – Fichier qui contient les exe et la base de données
  + **Properties** – Fichier qui contient les propriétés de l’application
  + **Ressources** – Fichier contenant les fichiers SQLite

## Description des tests

Voir les fichiers « SADOWSKI\_planDeTest » et « SADOWSKI\_rapportDeTest ».

## Erreurs restantes

Aucune

# La vie du projet

Voir le fichier « FERNANDES\_PlanningFinal.xlsx »

# Conclusion

Objectifs atteints :

* Afficher une liste de séries avec leurs acteurs, leurs genres, leurs réalisateurs et leur description
* Afficher une liste de saison avec leurs épisodes
* Afficher une liste d’épisodes avec leur description et leur nombre d’épisode.

Objectif non-atteints :

* Ajouter une série avec ses acteurs, ses genres, ses réalisateurs et sa description
* Ajouter une saison avec ses épisodes
* Ajouter un épisode avec sa description et son nombre d’épisode.
* Modifier une série avec ses acteurs, ses genres, ses réalisateurs et sa description
* Modifier une saison avec ses épisodes
* Modifier un épisode avec sa description et son nombre d’épisode.
* Supprimer une série
* Supprimer une saison
* Supprimer un épisode

Points positifs :

* Ergonomique
* Facile d’utilisation

Points négatifs :

* Incomplet

Difficultés particulières :

* Connexion du SQLite avec le C#
* Communication entre le programme et la base de données

Suites possibles :

* Rajouter un ajout de classes
* Ajouter plus de propriétés dans la barre de recherches
* Ajouter les extensions comme information pour les cartes