Documentation fonctionnelle et technique

Association sportive version C#

I - Les données

1) Le dictionnaire de données (page 2)

2) MEA (page 2)

3) Schéma relationnel (page 3)

4) Le script (page 4 à 6)

II - Fonctionnalités

1) Backlog : Liste des fonctionnalités (page 8)

2) L’arborescence (page 9 à 10)

3) Les cas d’utilisation (page 11 à 12)

4) Les maquettes du site final (page 13)

III – Dossier Technique

1) Focus sur 2 fonctionnalités (page 14)

2) Pour chaque fonctionnalité : une capture d’écran et un code commenté (page 14 à 15)

IV – Veille ou autoformation

Autoformations de Mehdi et Quentin (page 17)

V – Conduite du projet

1) Qui a fait quoi et quand ? (page 17)

2) 1 compte rendu de réunion (page 17)

**Contexte :**

Une Association Sportive d’un lycée privé nous demande de réaliser une application de gestion de différents types d’utilisateurs telle que : les élèves adhérents, la comptabilité, les évènements sportifs, les inscriptions aux évènements sportifs et les statistiques.

L’application devra être réalisée dans sa totalité, étant donné qu’elle n’existe pas.

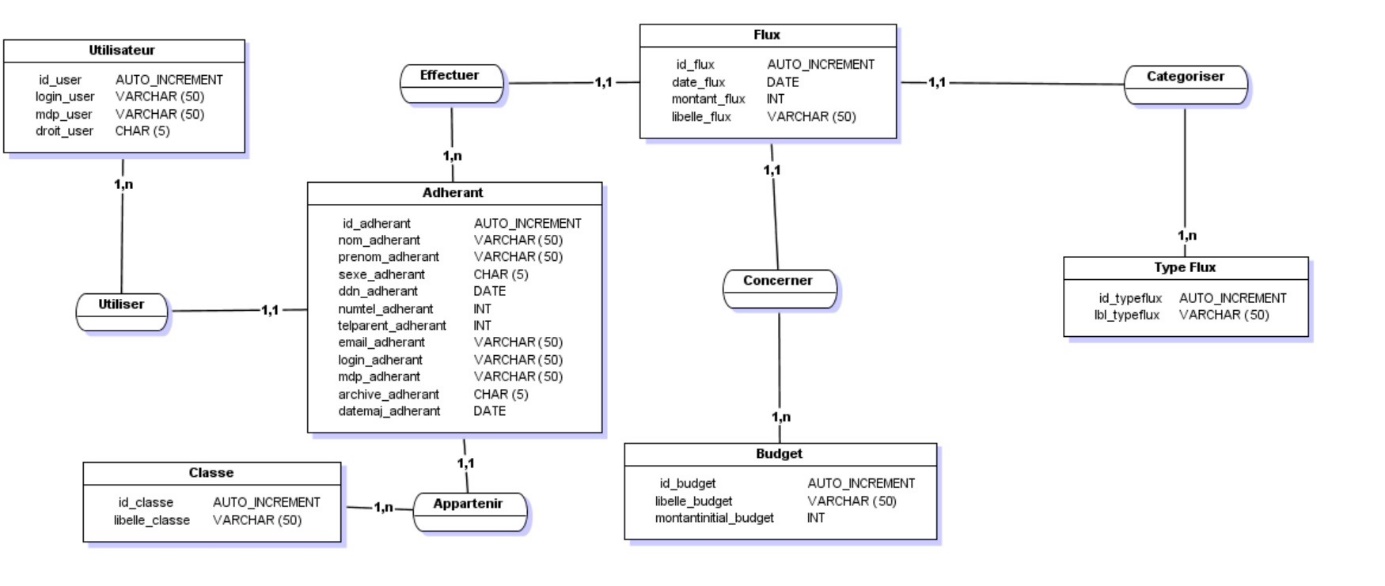
I – Les données

1) Le dictionnaire de données

Une image contenant table

Description générée automatiquement

MEA :



**Schéma relationnel :**

**UTILISATEUR** (Id\_user, Login\_user, mdp\_user, droit\_user)

Clé primaire = Id\_utilisateur

**CLASSE** (Id\_classe, Libelle\_classe)

Clé primaire = Id\_classe

**ADHERENT** (Id\_adherent, Nom\_adherent, Prenom\_adherent, Ddn\_adherent, numtel\_adherent, email\_adherent, numparent\_adherent, autoprelev\_adherent, sexe\_adherent, login\_adherent, mdp\_adherent, datemaj\_adherent, archive\_adherent, #Id\_utilisateur, #Id\_classe)

Clé primaire = Id\_adherent

Clé étrangère = #Id\_user en référence à Id\_user de la table UTILISATEUR

Clé étrangère = #Id\_classe en référence à Id\_classe de la table CLASSE

**BUDGET** (Id\_budget, lbl\_budget, montantInitial\_budget)

Clé primaire = Id\_budget

**TYPE\_FLUX** (Id\_type\_flux, lbl\_typeflux)

Clé primaire = Id\_type\_flux

**FLUX** (Id\_flux, lbl\_flux, Date\_flux, Montant\_flux, #Id\_adherent, #Id\_typeflux)

Clé primaire = Id\_flux

Clé étrangère = #Id\_adherent en référence à Id\_adherent de la table ADHERENT

Clé étrangère = #Id\_type\_flux en référence à Id\_type\_flux de la table TYPE\_FLUX

Clé étrangère =#Id\_budget en référence à Id\_evenement de la table EVENEMENT

**Scripts SQL :**

-- Création de la base de données

CREATE DATABASE ASSOCSPORTIVE;

GO

-- Utilisation de la table ASSOCSPORTIVE

USE ASSOCSPORTIVE;

GO

-- Création des tables

CREATE TABLE CLASSE

(

id\_classe INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,

lbl\_classe VARCHAR(255)NOT NULL

);

GO

CREATE TABLE UTILISATEUR

(

id\_user INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,

login\_user VARCHAR(255),

mdp\_user VARCHAR(255),

droit\_user CHAR

);

GO

CREATE TABLE ADHERENT

(

id\_adherent INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,

nom\_adherent VARCHAR(255),

prenom\_adherent VARCHAR(255),

ddn\_adherent DATE,

sexe\_adherent CHAR,

email\_adherent VARCHAR(255),

numtel\_adherent INT,

date\_maj\_adherent DATETIME,

numparent\_adherent INT,

autoprelev\_eleve CHAR,

login\_adherent VARCHAR(255),

mdp\_adherent VARCHAR(255),

archive\_adherent CHAR,

id\_user INT NULL FOREIGN KEY(id\_user) REFERENCES UTILISATEUR(id\_user),

id\_classe INT NULL FOREIGN KEY(id\_classe) REFERENCES CLASSE(id\_classe)

);

GO

CREATE TABLE BUDGET

(

id\_budget INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,

lbl\_budget VARCHAR (255)NOT NULL,

montantntial\_budget INT

);

GO

CREATE TABLE TYPE\_FLUX

(

id\_type\_flux INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,

lbl\_type\_flux VARCHAR(255),

);

GO

CREATE TABLE FLUX

(

id\_flux INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,

montant\_flux INT,

date\_flux DATE,

id\_adherent INT NULL FOREIGN KEY (id\_adherent) REFERENCES ADHERENT(id\_adherent),

id\_type\_flux INT NULL FOREIGN KEY(id\_type\_flux) REFERENCES TYPE\_FLUX(id\_type\_flux),

id\_budget INT NULL FOREIGN KEY(id\_budget) REFERENCES BUDGET(id\_budget)

);

II - Fonctionnalités

1) Backlog : Liste des fonctionnalités (page 8)

|  |  |
| --- | --- |
| **Page d’accueil** | Le formulaire d’inscription d’un utilisateur dans les champs et vérifier que les données soient valides auquel cas un message d’erreur devra être affiché. |
| **Formulaire de connexion** | Le formulaire de connexion devra permettre de se connecter en renseignant le nom d’utilisateur et le mot de passe. |
| **Gestion des élèves adhérant** | Le formulaire d’ajout d’une question permet de créer une question et de le stocker dans la base de données. |
| **Gestion de comptabilité** | La vue d’une question permet de voir une question, de les liker et de voir et créer des réponses. |
| **Gestion des budgets** | Vous devrez vérifier que les informations saisies soient valides, auquel cas un message d’erreur devra prévenir l’utilisateur que certains champs sont invalides. |
| **Espace d’administration** | Un espace d’administration, accessible uniquement aux profils “administrateurs” pourront accéder à la liste des questions dans une page dédiée et pourront les modérer en les supprimant. |
| **Ecran de statistique** | Un espace où on peut voir des statistiques en utilisant les données venant de la base. |

2) L’arborescence

Non Connecté :

Page accueil

Inaccessible

Ecran statistique

Gestion de comptabilité

Formulaire de connexion

Gestion des budgets

Gestion des élèves adhérents

Administrateur :

Page accueil

Ecran statistique

Gestion de comptabilité

Formulaire de connexion

Gestion des budgets

Gestion des élèves adhérents

Comptable :

Page accueil

Inaccessible

Ecran statistique

Gestion de comptabilité

Formulaire de connexion

Gestion des budgets

Gestion des élèves adhérents

**Cas d’utilisation d’un admin :**

**Cas d’utilisation d’un comptable :**

Maquette :

**Une image contenant texte

Description générée automatiquement**

**Une image contenant table

Description générée automatiquement**

**Une image contenant table

Description générée automatiquement**

III – Dossier technique

1) Focus sur 2 fonctionnalités

La page d’ajout d’adhérent : La page d’ajout adhérant permet à l’utilisateur de saisir un nouvel élève qui sera inséré dans la base de données.

Des messages d’erreurs sont affichés si un ou plusieurs champs sont vides et n’envoie pas les données tant que l’utilisateur n’a pas saisi les données dans les champs demandés.

La page de gestion des adhérents: La page de gestion des élèves permet l’affichage des élèves enregistrés dans la base de données. Cette permet de sélectionner un élèves où l’utilisateur peut modifier ou supprimer.

2) Pour chaque fonctionnalité : une capture d’écran et un code commenté

Inséré les images de code des fonctionnalités.

IV – Veille ou autoformation

Mehdi :

Quentin :

V – Conduite du projet

A) Qui a fait quoi et quand ?

Mehdi :

Quentin :

B) Compte rendu de réunion