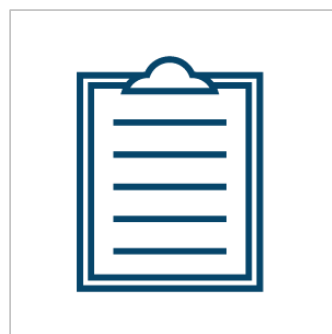


4 images 1 mot



TPI

Kadri Krasniqi / Cin4B

ETML

110 heures

Chef de projet : Gilbert Gruaz

Expert 1 : Didier Viret

Expert 2 : Frédérique Andolfatto

Table des matières

1	SPÉCIFICATIONS	4
1.1	TITRE	4
1.2	DESCRIPTION	4
1.3	SI LE TEMPS LE PERMET	4
1.4	MATÉRIEL ET LOGICIELS À DISPOSITION	4
1.5	PRÉREQUIS	4
1.6	CAHIER DES CHARGES	4
2	PLANIFICATION INITIALE	5
3	ANALYSE	8
3.1	OPPORTUNITÉS	8
3.1.1	<i>Difficultés potentielles :</i>	8
3.2	DOCUMENT D'ANALYSE ET CONCEPTION	8
3.2.1	<i>Maquette interface graphique</i>	8
3.2.2	<i>Analyse base de données</i>	11
3.2.2.1	<i>MCD</i>	11
3.2.2.2	<i>MLD</i>	12
3.2.2.3	<i>MPD</i>	13
3.2.3	<i>Analyse programme</i>	14
3.3	CONCEPTION DES TESTS	15
3.4	PLANIFICATION DÉTAILLÉE	16
4	RÉALISATION	19
4.1	DOSSIER DE RÉALISATION	19
4.1.1	<i>Base de données</i>	19
4.1.2	<i>Interface graphique</i>	21
4.1.3	<i>Connexion C# --> SQL server</i>	22
4.1.4	<i>Exécution des requêtes SQL</i>	22
4.1.5	<i>Récupération des images et du mot en cours</i>	24
4.1.6	<i>Affichage des lettres aléatoire</i>	26
4.1.7	<i>Insertion d'un nouveau mot</i>	27
4.1.8	<i>Gestion des droits</i>	29
4.2	MODIFICATIONS	30
4.2.1	<i>Changement Microsoft SQL Server Management Studio</i>	30
4.2.2	<i>Ajout d'une table « droit » à la base de données</i>	30
5	TESTS	31
5.1	DOSSIER DES TESTS	31
6	CONCLUSION	32
6.1	BILAN DES FONCTIONNALITÉS DEMANDÉES	32
6.2	BILAN DE LA PLANIFICATION	33
6.3	BILAN PERSONNEL	34
7	DIVERS	35

7.1	JOURNAL DE TRAVAIL	35
7.2	WEBOGRAPHIE.....	41
8	ANNEXES	42
8.1	CAHIER DES CHARGES	42
8.2	MISE EN PLACE DE L'INFRASTRUCTURE POUR UTILISER L'APPLICATION	45

1 SPÉCIFICATIONS

1.1 Titre

Réalisation d'un quiz de type 4 images 1 mot en C#

1.2 Description

Ce projet est réalisé dans le cadre des TPI de la session 2017 de l'ETML. Il consiste à créer une application de type « 4 images 1 mot » en c#, qui permettra aux enseignants de l'ETML de soutenir le processus d'apprentissage des élèves dans la matière qu'ils enseignent. Il s'agit d'une application simple à utiliser. À l'ouverture du programme, l'utilisateur aura le choix entre différents thèmes. Lorsqu'il aura choisi, l'application va prendre au hasard dans la base de données 1 mot en rapport avec le thème suivi de ses 4 images. Les enseignants auront la possibilité d'ajouter leurs propres images avec leurs propres mots afin d'enrichir la base de données.

1.3 Si le temps le permet

Si le temps le permet, j'ajouterais :

- Une fonction pour compter les points
- Une fonction pour compter le temps
- Un nouveau formulaire pour pouvoir ajouter des thèmes

1.4 Matériel et logiciels à disposition

1 ordinateur standard ETML, avec la structure habituelle

Visual Studio 2015, SQL Server 2014, Notepad++

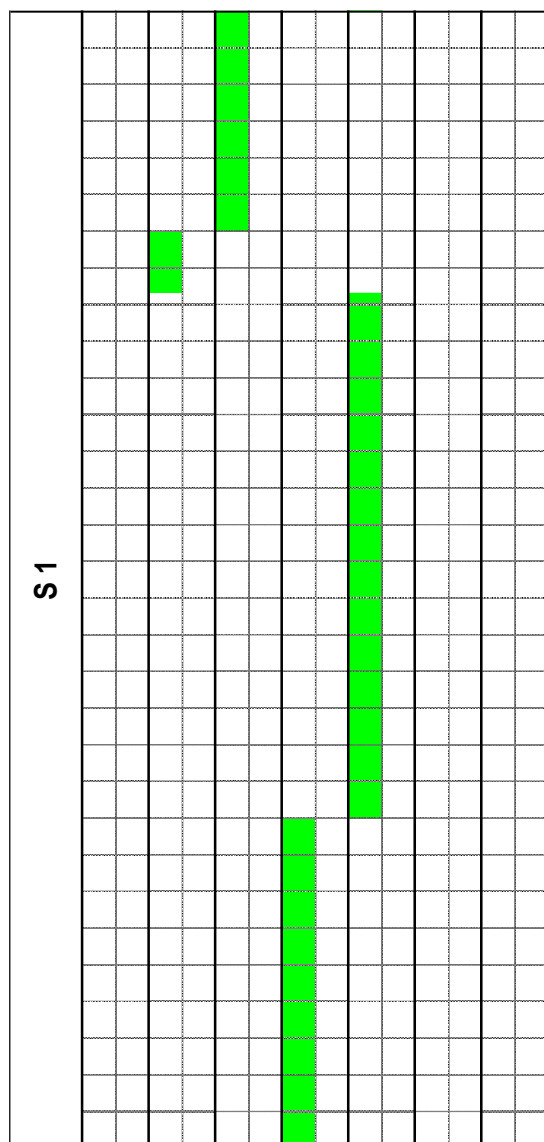
1.5 Prérequis

Avoir suivi les modules ICH à l'ETML, les projets et effectué des stages... Les modules ICH suivants sont les plus impliqués : 100, 104, 105, 303 et 306

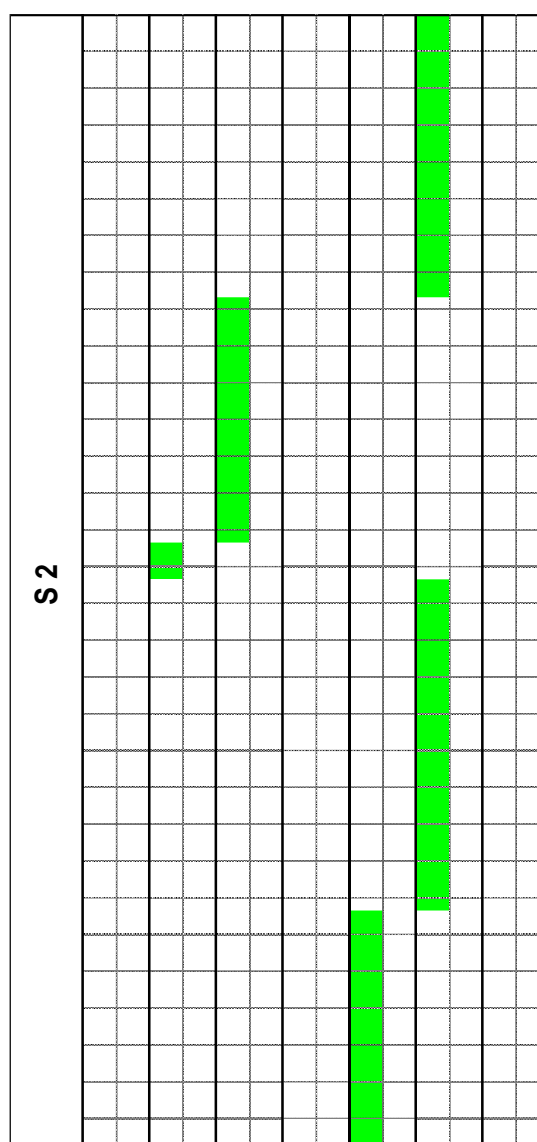
1.6 Cahier des charges

[Lien sur le cahier des charges](#)

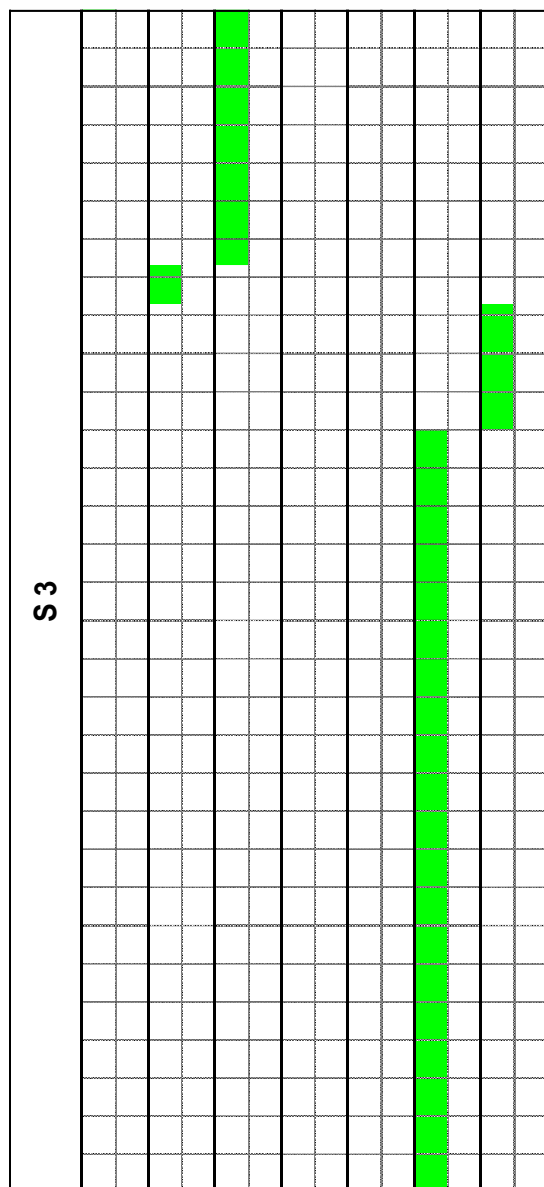
2 PLANIFICATION INITIALE



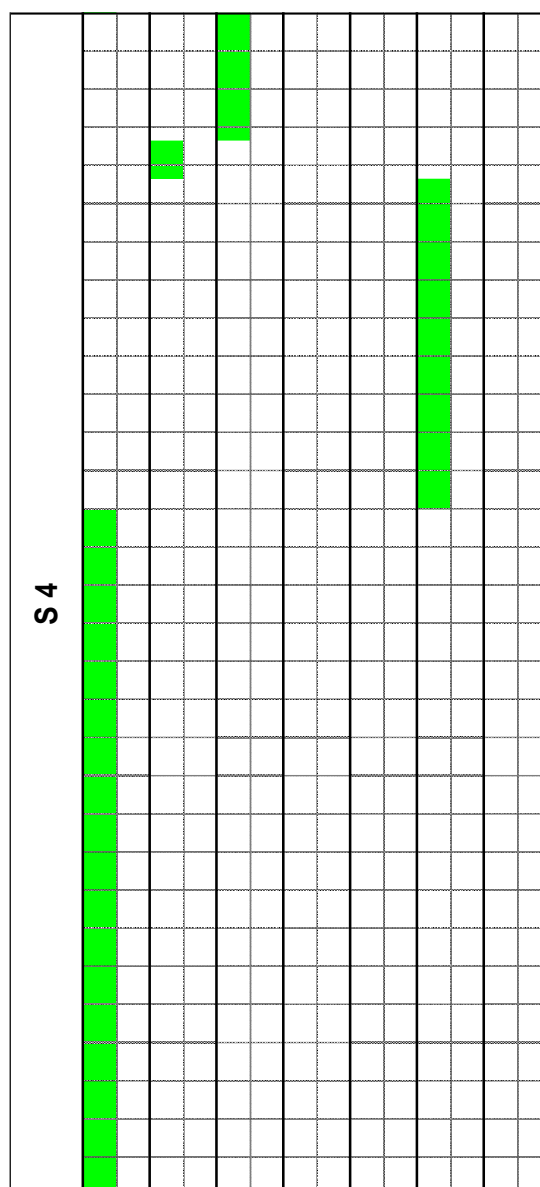
Tâches - objectifs	Nb 1 / 4 heure
Absence - Imprévu	78
Journal de travail	19
Documentation	81
Planification	27
Analyse	63
Implémentation	174
Test	50
Total planifié	492
Total réalisé	0



Tâches - objectifs	Nb 1 / 4 heure
Absence - Imprévu	78
Journal de travail	19
Documentation	81
Planification	27
Analyse	63
Implémentation	174
Test	50
Total planifié	492
Total réalisé	0



Tâches - objectifs	Nb 1 / 4 heure
Absence - Imprévu	78
Journal de travail	19
Documentation	81
Planification	27
Analyse	63
Implémentation	174
Test	50
Total planifié	492
Total réalisé	0



Tâches - objectifs	Nb 1 / 4 heure
Absence - Imprévu	78
Journal de travail	19
Documentation	81
Planification	27
Analyse	63
Implémentation	174
Test	50
Total planifié	492
Total réalisé	0

S 5		Nb 1 / 4 heure	
Tâches - objectifs			
Absence - Imprévus		78	
		0	
Journal de travail		19	
		0	
Documentation		81	
		0	
Planification		27	
		0	
Analyse		63	
		0	
Implémentation		174	
		0	
Test		50	
		0	
Total planifié		492	
Total réalisé		0	

S 6		Nb 1 / 4 heure	
Tâches - objectifs			
Absence - Imprévus		78	
		0	
Journal de travail		19	
		0	
Documentation		81	
		0	
Planification		27	
		0	
Analyse		63	
		0	
Implémentation		174	
		0	
Test		50	
		0	
Total planifié		492	
Total réalisé		0	

3 ANALYSE

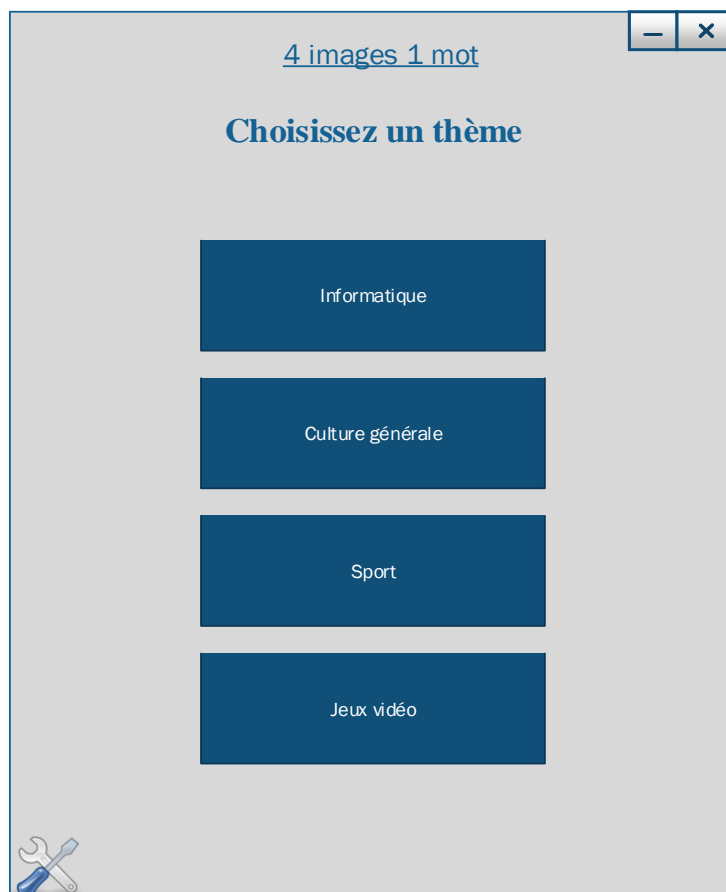
3.1 Opportunités

3.1.1 Difficultés potentielles :

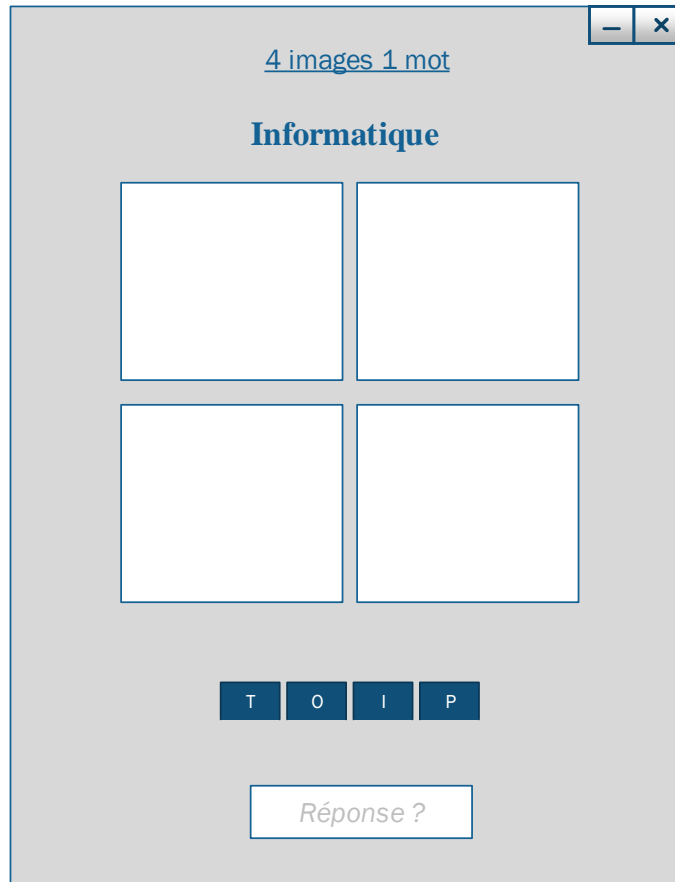
1. Connexion à la base de données depuis C#
2. Récupérer les images stockées dans un répertoire dont le chemin se trouve dans la base de données
3. Organisation des requêtes SQL dans le code C#
4. Distinguer les fonctionnalités en tant qu'enseignant de celle en tant qu'élève

3.2 Document d'analyse et conception

3.2.1 Maquette interface graphique



La page d'accueil ressemblera à cela. Elle propose certains thèmes que l'utilisateur pourra choisir. En bas à gauche, nous avons une icône de « paramétrage » qui va en fait permettre aux enseignants d'ajouter un nouveau mot avec des nouvelles images.

Interface de jeu :

The screenshot shows a game window titled "4 images 1 mot" with a close button (X) in the top right corner. The theme "Informatique" is displayed in blue text. Below the theme, there are four empty square boxes arranged in a 2x2 grid. Underneath the grid, the letters T, O, I, and P are each inside a dark blue square button. At the bottom, there is a text input field with the placeholder text "Réponse ?".

Nous arrivons à la principale fonctionnalité de l'application. Après avoir choisi un thème, l'utilisateur arrivera sur cette page. Nous avons en haut le thème choisi puis en dessous les 4 images en rapport avec le mot à trouver. Après ces images, le programme propose des lettres disposées aléatoirement. Cette liste de lettres contient les caractères pour former le mot à trouver. Une fois que l'utilisateur a trouvé le mot, il peut l'écrire dans la case « Réponse ? ».

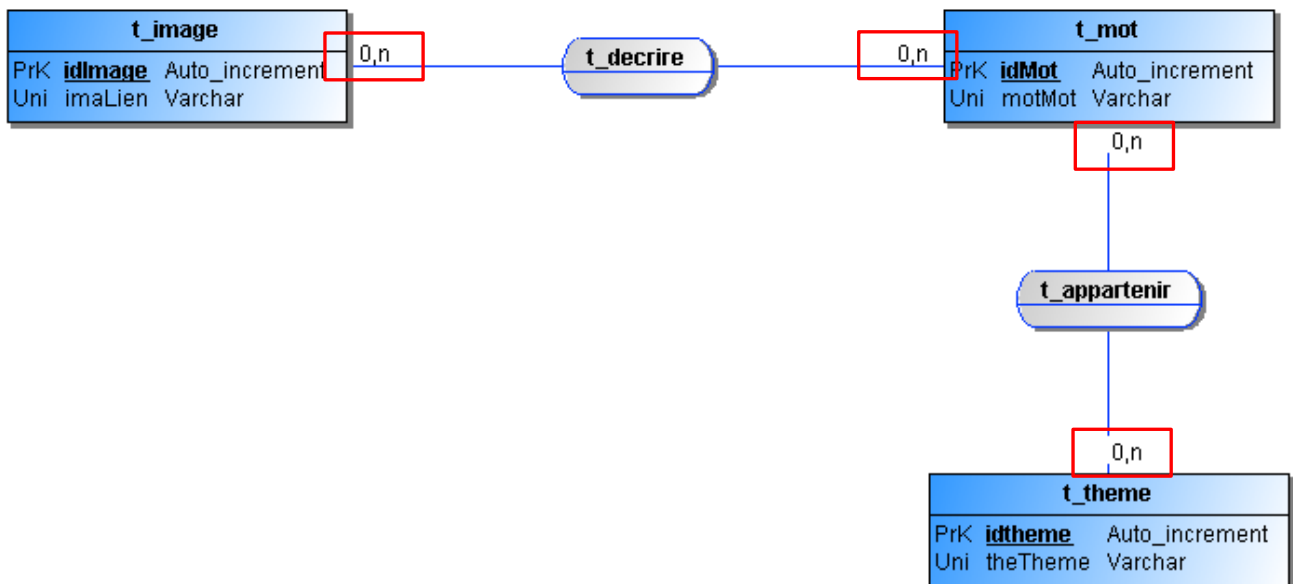
Interface nouveau mot :

La dernière interface graphique est la page qui permet à l'enseignant d'ajouter un nouveau mot. Il s'agit d'une interface très simple et n'importe quel utilisateur peut l'utiliser. Nous avons en premier temps une case qui va permettre à l'utilisateur d'insérer le nouveau mot. Ensuite, une liste proposera les thèmes inscrit dans la base de données et l'utilisateur devra choisir à quel thème appartient son mot. La dernière étape est de sélectionner les images. L'utilisateur est obligé d'ajouter 4 images. Une fois que l'utilisateur a bien choisi ses images dans ses propres répertoires, le programme va lui alors faire une copie dans un répertoire spécifique qui contiendra toutes les images de l'application. Lors du clic sur le bouton « Ajouter », les vérifications vont être effectuées. Tout d'abord de s'assurer que le mot ne contient que des lettres puis de vérifier que l'utilisateur a bien sélectionné 4 images accessible au programme.

3.2.2 Analyse base de données

Pour mon projet TPI, il m'a été imposé de faire une base de données avec SQL server 2014. Celle-ci a été installée sur une machine virtuelle. Après une réflexion et discussion avec le chef de projet, j'ai conçu un modèle conceptuel, un modèle logique puis un modèle physique des données.

3.2.2.1 MCD

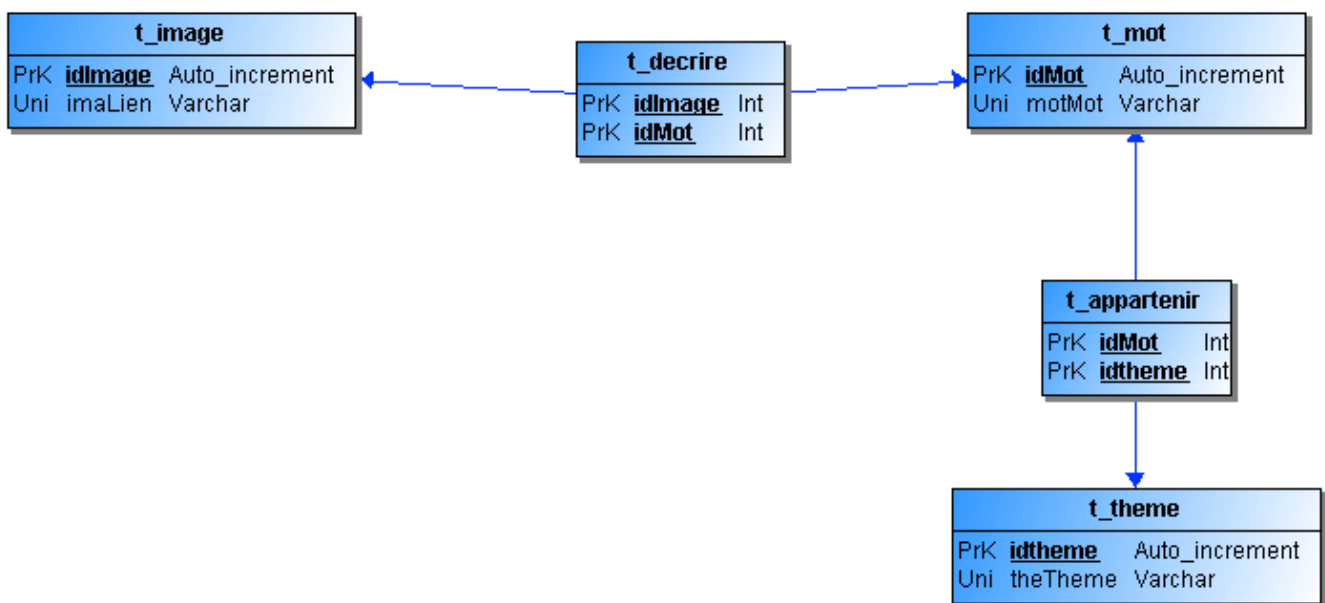


J'ai nommé les entités comme des tables car j'étais embêté par la suite à cause du programme « JMerise ». Le programme permet de mettre les noms seulement dans le MCD, ensuite il ne nous permet plus de les modifier pour le MLD. C'est pour cela que les noms sont déjà des noms de tables.

Il s'agit d'une base de données assez simple. Nous avons 3 entités différentes qui se relient. La première est pour les images. Elle va contenir une propriété « liens » des images utilisées dans l'application. Dans cette entité nous stockeront les liens des images (lien dans le répertoire spécifique d'image). Nous voyons la relation avec l'entité des mots. Le « 0, n » signifie qu'une image peut au minimum contenir aucun mot et pas de restriction pour le maximum. En résumé, une image peut être insérer sans aucun mot

dans la base de données mais ne sera pas utilisée. Elle pourra aussi décrire plusieurs mots. La deuxième entité est celle des mots. Elle contient une propriété simple de type chaîne de caractère qui contiendra le mot à trouver. Cette entité a un lien avec celle que nous avons vu précédemment dont la cardinalité est « 0, n », c'est-à-dire qu'un mot peut avoir au minimum 0 image et pas de maximum. Dans notre cas le mot aura 4 images mais il s'agit d'une logique nous permettant de faire le MLD que nous verrons après. Cette entité possède un autre lien avec la suivante qui est celle pour les thèmes. Nous retrouvons une cardinalité de « 0,n » Elle signifie qu'un mot peut n'appartenir à aucun thème ou à plusieurs thèmes. Pour finir, l'entité des thèmes qui est la dernière possède une propriété de chaîne de caractère ou seront noter tous les thèmes possible. Cette dernière a un lien avec l'entité des mots et la cardinalité est de « 0, n ». Cela veut dire que un thème peut être crée sans aucun mot ou appartenir à un ou plusieurs mots.

3.2.2.2 MLD



Nous avons maintenant le modèle logique des données. Suite aux règles de passage du MCD au MLD, nous avons maintenant des tables et non des entités et nous remplaçons les propriétés par des attributs. Les cardinalités ont été supprimées et nous avons maintenant deux nouvelles tables qui sont des tables qui ont été établies grâce aux règles de cardinalités. Vu que chaque table contenait une relation de « x, n », j'ai alors dû créer des tables pour faire le lien. Ces nouvelles tables ont pour attributs les clés primaires des autres tables.

3.2.2.3 MPD

Nous retrouvons les 5 tables vu précédemment mais sous formes de tableaux d'intégrités. Leurs attributs sont plus détaillés dans le modèle physique des données ci-dessous. Pour chaque table nous avons son type et sa taille, la valeur par défaut, les contraintes, si l'attribut est indexé ou non, si on autorise le nul et autre.

T_Image						
Attribut	Type + Taille	Valeur par défaut	contrainte	Indexe	Nul interdit	Divers
id_image	Int - 100	-	Clé primaire	Non	Oui	NumAuto
imaLien	Varchar - 100	-	-	Non	Oui	Unique

T_Decrire						
Attribut	Type + Taille	Valeur par défaut	contrainte	Indexe	Nul interdit	Divers
idimag	Int - 100	-	Clé étrangère	Non	Oui	NumAuto
idmot	Int - 25	-	Clé étrangère	Non	Oui	NumAuto

T_Mot						
Attribut	Type + Taille	Valeur par défaut	contrainte	Indexe	Nul interdit	Divers
id_mot	Int - 25	-	Clé primaire	Non	Oui	NumAuto
motMot	Varchar - 25	-	-	Non	Oui	Unique

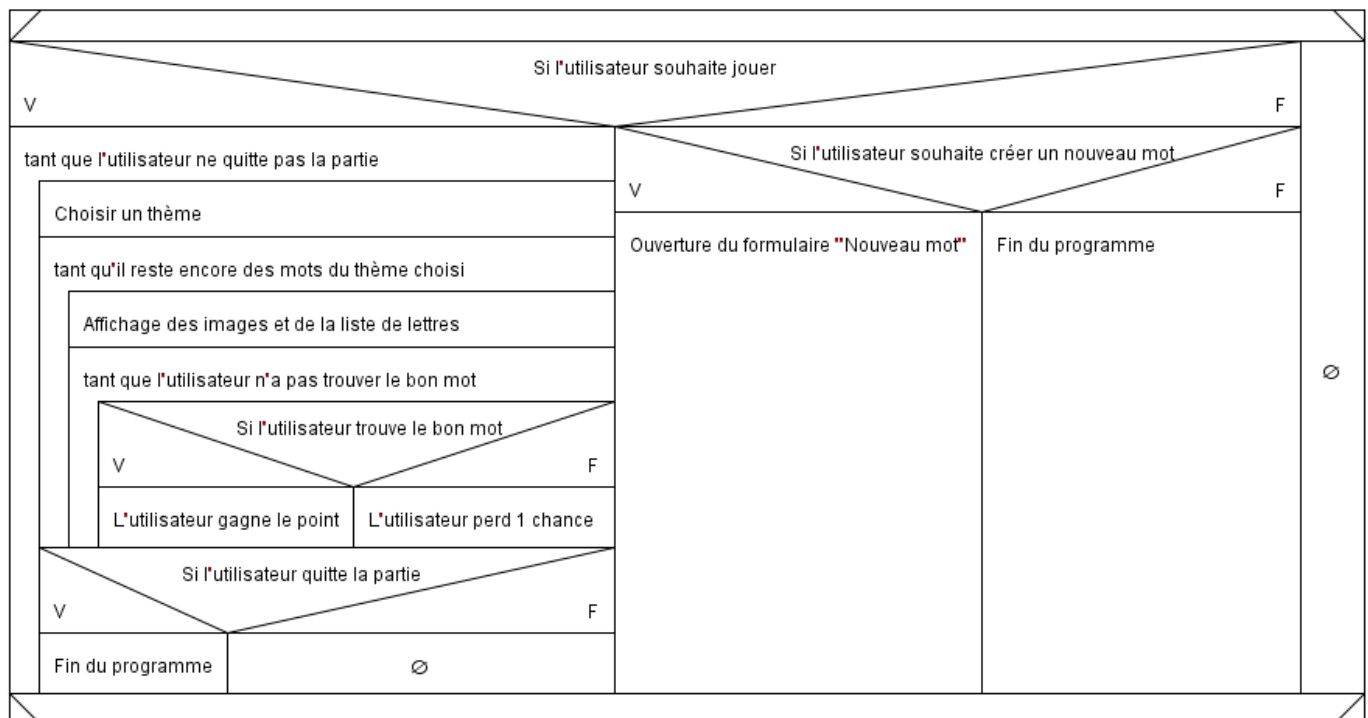
T_appartenir						
Attribut	Type + Taille	Valeur par défaut	contrainte	Indexe	Nul interdit	Divers
idmot	Int - 25	-	Clé étrangère	Non	Oui	NumAuto
idtheme	Int - 25	-	Clé étrangère	Non	Oui	NumAuto

T_Theme						
Attribut	Type + Taille	Valeur par défaut	contrainte	Indexe	Nul interdit	Divers
id_theme	Int - 25	-	Clé primaire	Non	Oui	NumAuto
theTheme	Varchar - 25	-	-	Non	Oui	Unique

3.2.3 Analyse programme

Après avoir lu le cahier des charges et avoir fait une petite analyse sur papier, j'ai créé un structogramme pour expliquer le fonctionnement du programme plus facilement. Il explique étape par étape comment le programme va réagir.

4 images 1 mot



Nous apercevons qu'il y a deux principales fonctionnalités dans le programme. La première est de choisir un thème puis de découvrir le mot qui se cache. La seconde est réservée aux enseignants et sera bloqué par un code d'accès. Elle consiste à pouvoir ajouter un mot avec ses 4 images dans la base de données.

3.3 Conception des tests

Les tests de l'application se feront pendant le projet et après chaque nouvelle fonctionnalité ajoutée. Il s'agira de petits tests pour voir si le programme fonctionne bien. Vers la fin du projet, j'effectuerais des tests un peu plus spécifiques qui auront pour but de tester le programme et de chercher s'il y a des erreurs à corriger.

Les tests généraux du programme à la fin du projet seront ceux-ci :

JEU

CASE « RÉPONSE »	Mettre des mauvaises réponses Mettre des chiffres, caractères spéciaux Ne rien mettre
-------------------------	---------------------------------------------------------------------------------------------

NOUVEAU MOT

CASE « MOT »	Mettre des chiffres, caractères spéciaux Ne rien mettre Mettre un long mot de plus de 20 caractères Mettre un seul caractère
AJOUTER IMAGES	Ajouter un document « word, pdf... » Ne pas mettre d'images Mettre seulement 3 images

Tâches - objectifs	Nb 1/4 heures	S3	S4
Absence - Imprévus	78 0		
Analyse base de données	15 0		
Analyse connexion cf# et sql server	16 0		
Création de l'interface graphique	27 0		
Journal de travail	22 0		
Documentation	85 0		
Création de la base de données	15 0		
Développement : récupérer les images avec le mot et afficher dans l'application	55 0		
Développement : afficher les lettres du mot à trouver aléatoirement	18 0		
Développement : ajouter une fonctionnalité pour ajouter des nouveaux mots avec des images	30 0		
Créer les tests à effectuer	5 0		
Test	40 0		
Planification des tâches	32 0		
Discussion - Réunion	5 0		
Analyse du programme	10 0		
Développement : amélioration	25 0		
Développement : calcul des points	15 0		

Tâches - objectifs	Nb 1/4 heure	S 5										S 6									
Absence - Imprévus	78 0																				
Analyse base de données	15 0																				
Analyse connexion c# et sql server	16 0																				
Création de l'interface graphique	27 0																				
Journal de travail	22 0																				
Documentation	85 0																				
Création de la base de données	15 0																				
Développement : récupérer les images avec le mot et afficher dans l'application	55 0																				
Développement : afficher les lettres du mot à trouver aléatoirement	18 0																				
Développement : ajouter une fonctionnalité pour ajouter des nouveaux mots avec des images	30 0																				
Créer les tests à effectuer	5 0																				
Test	40 0																				
Planification des tâches	32 0																				
Discussion - Réunion	5 0																				
Analyse du programme	10 0																				
Développement : amélioration	25 0																				
Développement : calcul des points	15 0																				

4 RÉALISATION

4.1 Dossier de Réalisation

4.1.1 Base de données

Après avoir créé le MCD, MLD avec le programme JMerise, celui-ci nous permet de générer le script pour pouvoir créer la base de données très facilement. Il nous propose de choisir la plateforme utilisée et produit le code en fonction du langage. Pour ma part, j'ai demandé à cette application de me générer le script pour un SQL Server et voilà le résultat :

```

/*-----
-- Table: t_image
-----*/
CREATE TABLE t_image(
    idImage INT IDENTITY (1,1) NOT NULL ,
    imalien VARCHAR (100) NOT NULL UNIQUE,
    CONSTRAINT prk_constraint_t_image PRIMARY KEY NONCLUSTERED (idImage)
);

/*-----
-- Table: t_mot
-----*/
CREATE TABLE t_mot(
    idMot INT IDENTITY (1,1) NOT NULL ,
    motMot VARCHAR (25) NOT NULL UNIQUE,
    CONSTRAINT prk_constraint_t_mot PRIMARY KEY NONCLUSTERED (idMot)
);

/*-----
-- Table: t_theme
-----*/
CREATE TABLE t_theme(
    idtheme INT IDENTITY (1,1) NOT NULL ,
    theTheme VARCHAR (25) NOT NULL UNIQUE,
    CONSTRAINT prk_constraint_t_theme PRIMARY KEY NONCLUSTERED (idtheme)
);

/*-----
-- Table: t_decrire
-----*/
CREATE TABLE t_decrire(
    idImage INT NOT NULL ,
    idMot INT NOT NULL ,
    CONSTRAINT prk_constraint_t_decrire PRIMARY KEY NONCLUSTERED (idImage,idMot)
);

/*-----
-- Table: t_appartenir
-----*/
CREATE TABLE t_appartenir(
    idMot INT NOT NULL ,
    idtheme INT NOT NULL ,
    CONSTRAINT prk_constraint_t_appartenir PRIMARY KEY NONCLUSTERED (idMot,idtheme)
);

ALTER TABLE t_decrire ADD CONSTRAINT FK_t_decrire_idImage FOREIGN KEY (idImage) REFERENCES t_image(idImage);
ALTER TABLE t_decrire ADD CONSTRAINT FK_t_decrire_idMot FOREIGN KEY (idMot) REFERENCES t_mot(idMot);
ALTER TABLE t_appartenir ADD CONSTRAINT FK_t_appartenir_idMot FOREIGN KEY (idMot) REFERENCES t_mot(idMot);
ALTER TABLE t_appartenir ADD CONSTRAINT FK_t_appartenir_idtheme FOREIGN KEY (idtheme) REFERENCES t_theme(idtheme);

```

Il s'agit d'un script très simple et assez clair. Tout de même j'ai dû faire quelques petites recherches sur internet avant d'exécuter ce code pour bien comprendre le fonctionnement et la création de la base de données.

En effet, je ne comprenais pas la propriété « IDENTITY (1,1) » qui se trouve après chaque clé primaire. Il s'agit en fait tout simplement de définir que l'attribut s'incrémente tout seul. L'équivalent d'un « AUTO_INCREMENT » pour MYSQL. Par contre elle a une fonction en plus qui permet de définir à quel chiffre commence l'incrémentation et de combien va elle augmenter.

Exemple :

```
CREATE TABLE t_exemple(  
    idExemple INT IDENTITY (5,10) NOT NULL  
);
```

Dans ce cas, le premier exemple qui sera créé commencera par l'identifiant numéro 5 et le deuxième sera le numéro 15. Car l'incrémentation se fait de 10 en 10.

4.1.2 Interface graphique

Pour l'interface graphique, j'ai décidé de faire seulement deux formulaires séparés. Le premier est celui du menu d'accueil et du jeu, le deuxième est celui de l'ajout d'un nouveau mot. L'interface de l'accueil a été faite directement grâce à l'espace de travail « Design » de Visual studio. Cet espace de travail permet de créer visuellement l'application, d'ajouter nos boutons ou d'autres contrôles en faisant un « drag and drop » depuis la « toolbox ». C'est un outil très utile qui nous permet de faire des interfaces simples assez rapidement. Les boutons et autres contrôles de la page d'accueil ont tous été mis dans un « panel ». Il s'agit d'un conteneur qui nous permettra par la suite de l'afficher ou le faire disparaître. L'interface de jeu a par contre elle était faite par code et aussi dans un panel. J'ai trouvé mieux de faire comme ça car on garde tout sur une fenêtre et nous n'avons pas besoin d'avoir plusieurs fenêtres inutiles. Pour mieux expliquer, lorsque l'application se lance nous arrivons directement sur l'accueil et donc au choix du thème. Après avoir choisi le thème, je rends le panel invisible et fait apparaître le deuxième qui lui a été créé par code. Il s'agit donc de la même fenêtre c'est juste le contenu qui va changer.

Pour la partie qui permet d'ajouter un nouveau mot j'ai préféré faire un nouveau formulaire car je trouve que c'est une autre fonctionnalité qui doit être indépendante et dont tout le monde n'aura pas l'accès. Cette interface a donc aussi été faite avec l'espace de travail « Design ». Lorsque l'enseignant qui aura les accès clique sur le bouton pour ajouter un nouveau mot. Le nouveau formulaire s'ouvrira et il pourra alors ajouter son mot.

```
private void CreateGame()  
{  
    //Interface graphique  
    pnl_ChoixTheme.Visible = false;  
    this.Size = new Size(500, 600);  
  
    pnl_jeu.Visible = true;  
}
```

Dans ce petit bout de code on peut voir le début de la fonction « CreateGame » qui va créer l'interface de jeu. La première opération qui se fait c'est celle que nous avons vu plus haut. La fonction va cacher le premier panel qui est « pnl_ChoixTheme ».

Ensuite, le programme agrandi le formulaire pour pouvoir afficher l'interface de jeu qui prend plus de place. Nous pouvons voir que l'interface de jeu sera de 500 pixels de largeur et 600 de hauteur.

Après avoir fait tout ça, le programme va rendre le deuxième panel visible pour que le contenu du formulaire soit le jeu.

4.1.3 Connexion C# --> SQL server

J'ai décidé de faire la connexion entre l'application C# et le serveur SQL avec l'espace de noms « System.Data ». Cet espace de noms permet de travailler avec différentes classes qui vont nous permettre de gérer efficacement les données provenant de notre serveur SQL. La classe qui va justement nous permettre de faire la connexion entre ces derniers est la classe « sqlconnection ». Pour cela, il faut d'abord initialiser une nouvelle connexion puis lui passer en paramètre la « connectionString ». Celle-ci sert à indiquer au programme le chemin et la syntaxe pour aller se connecter à la base de données. Exemple :

```
SqlConnection sqlConnection1 = new SqlConnection("Data  
Source=(localdb)\\MSSQLLocalDB;Initial Catalog=4images1mot;Connect  
Timeout=30;ApplicationIntent=ReadWrite");
```

Le nom de la nouvelle connexion est « sqlConnection1 » et la connectionString est en rouge. Elle donne la source de la base de données qui est en local. Le nom de la base de donnée qui est « 4images1mot » et d'autres informations.

4.1.4 Exécution des requêtes SQL

J'ai créé une classe spécifique pour les requêtes SQL qui s'appelle « dbConnexion ». J'ai trouvé que c'était mieux de séparer le code en C# qui va s'occuper de l'application elle-même et le code qui va interagir avec la base de données. Le but était de simplifier le code (le rendre plus compréhensible). Cela me permettra plus tard d'appeler mes requêtes SQL depuis mon code en utilisant seulement les noms des fonctions et sans avoir besoins de mettre du SQL. Exemple :

```
public SqlDataReader getmot()
{
    SqlCommand cmd = new SqlCommand();

    cmd.CommandText = "SELECT * from t_mot";
    cmd.CommandType = CommandType.Text;
    cmd.Connection = dc_connection;

    dc_connection.Open();

    var reader = cmd.ExecuteReader();

    while (reader.Read())
    {
        Console.WriteLine("Orders for " + reader.GetString(1));

        Console.WriteLine(String.Format("{0}, {1}",
            reader["idmot"], reader["motMot"]));
    }

    return reader;
}
```

Nous pouvons voir qu'il s'agit d'une fonction « getmot ». Elle permet de récupérer tous de la table « t_mot ». C'est une fonction test pour voir si la connexion à la base de données à bien fonctionner et voir si l'application arrive à récupérer des données. Nous voyons la classe « SqlCommand » qui va nous permettre de créer une nouvelle commande SQL. Après avoir créé cette variable il faut lui passer les paramètres. Le premier est de lui donner la requête SQL. Ceci se fait directement entre guillemet. Dans ce cas de test, il s'agit d'une requête simple qui est « SELECT * FROM t_mot ». La commande s'exécute avec la fonction « ExecuteReader() » et je récupère le tout dans une variable que j'appelle « reader ». Après avoir récupérer ces données dans une variable, il est encore impossible de les lire. C'est pour cela qu'il faut utiliser la fonction « read » qui va lire les données et les traduire.

L'appelle de cette fonction que nous avons vue est toute simple. Depuis la classe où se trouve le code de l'application il suffit d'initialiser la classe que nous voyons de voir et d'appeler la fonction de connexion à la base de données puis la requête SQL. Exemple :

```
dbConnexion fm_connexion = new dbConnexion();

fm_connexion.getmot();
```

L'initialisation de la base de données s'appelle « fm_connexion » et nous appelons justement la fonction « getmot ».

4.1.5 Récupération des images et du mot en cours

La récupération des images avec le bon mot était, je pense, la plus grande étape de l'application. Tout d'abord j'ai commencé à écrire plusieurs requêtes pour récupérer les données dans des listes. Je ne pouvais pas récupérer plusieurs informations comme par exemple « l'ID avec le lien de l'image » mais je devais récupérer l'id dans une variable puis le lien dans une autre. Ce qui me faisait deux fonctions et deux requêtes différentes. J'ai donc créé deux classes qui contiennent chacune une liste d'objet. La première liste d'objet est « getMotTheme ». Celle-ci va permettre de récupérer l'id du mot et le mot lui-même. Tous les mots du thème choisi seront dans cette liste d'objet.

```
class getMotTheme
{
    3 references
    public int idmot { get; set; }
    2 references
    public string mot { get; set; }
}
```

Nous pouvons voir qu'il s'agit d'une classe très simple. L'objet contient les propriétés suivantes : idmot qui est de type « int » et mot de type « string ».

Après avoir récupéré tous les mots du thème qui nous intéresse, nous allons nous occuper de récupérer les images pour chaque mot. C'est ici que nous allons faire appel à la deuxième classe qui est « getImageMot ».

```
class getImageMot
{
    1 reference
    public int idImage { get; set; }
    5 references
    public string lienImage { get; set; }
}
```

Encore une fois, nous apercevons que la classe est simple. Nous avons en premier la propriété idImage qui est de type « int » et le lien de l'image qui est de type « string ».

Après avoir choisi le thème, l'application va donc chercher tous les mots qui sont dans le thème puis aller chercher les images du premier mot et les afficher. Une fois que le jeu est en place, le programme attend que l'utilisateur clique sur le bouton « valider » pour donner sa réponse. Le programme va vérifier si le mot est correct ou pas et s'il est juste, il va appeler la fonction pour récupérer les images en lui donnant comme paramètre le deuxième mot.


```
private void btn_valider_Click(object sender, EventArgs e)
{
    //Si l'utilisateur a trouvé la bonne réponse
    if(tb_reponse.Text == lst_motOfTheme[int_motSuivant -1].mot)
    {
        //Si il reste encore des mots du thème
        if((int_motSuivant) < lst_motOfTheme.Count)
        {
            //Changer les images du jeux
            showImages(Convert.ToString(lst_motOfTheme[int_motSuivant].idmot));
        }
    }
}
```

Ci-dessus nous avons la fonction de validation. La première étape est de vérifier si le mot est égal au mot en cours. Le programme regarde ensuite s'il reste encore des mots du thème choisi, si oui il appelle la fonction qui gère les images « showImages » que nous pouvons voir ci-dessous.

```
private void showImages(string mot)
{
    //Initialisation de la classe dbconnexion
    dbConnexion si_connexion = new dbConnexion();

    //Récupérer la liste des images du mot
    List<getImageMot> lst_imageOfMot = si_connexion.getImageOfMot(Convert.ToString(mot));

    //Récupérer le lien des images
    pb_ima1.ImageLocation = lst_imageOfMot[0].lienImage;
    pb_ima2.ImageLocation = lst_imageOfMot[1].lienImage;
    pb_ima3.ImageLocation = lst_imageOfMot[2].lienImage;
    pb_ima4.ImageLocation = lst_imageOfMot[3].lienImage;

    //Passer au mot suivant
    int_motSuivant++;
}
```

La fonction « showImages » va nous permettre de récupérer les images puis de les afficher. Premièrement elle initialise la classe qui contient les requêtes « dbConnexion » en « si_connexion ». Ensuite, elle va appeler la fonction qui se charge d'exécuter la requête pour retourner les images du mot. En paramètre, nous donnons le mot en cours. Après avoir récupéré ces images, nous allons les afficher dans les pictureBox dans l'interface graphique du programme.

4.1.6 Affichage des lettres aléatoire

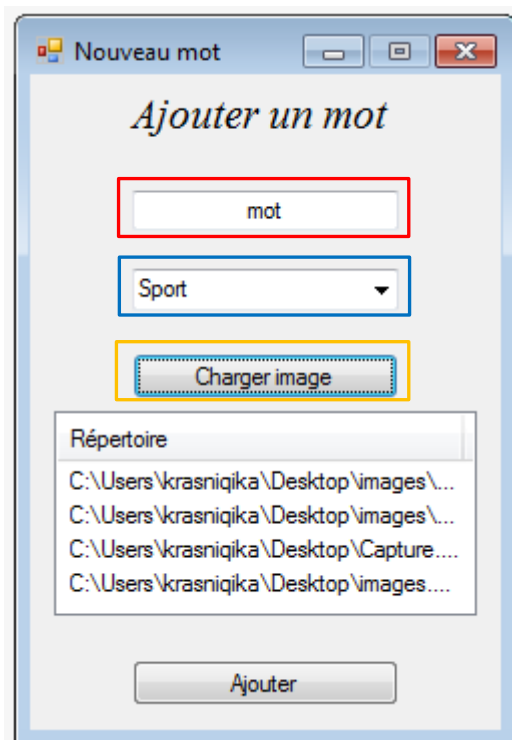
L'affichage des lettres aléatoire, du mot, permet d'aider l'utilisateur à trouver le mot. Cette fonction a été plus simple que prévue et j'ai passé moins de temps que prévu dans la planification pour cette partie. Le mot a été découpé et toutes les lettres du mot ont été placées dans la table (Chars). Ce qui va nous permettre de compter le nombre de lettres puis de les mélanger. Voici la fonction :

```
for (int i = 0; i < int_nbChar; i++)  
{  
    int j = rnd.Next(0, int_nbChar - 1);  
    temp = chars[i];  
    chars[i] = chars[j];  
    chars[j] = temp;  
}
```

Il s'agit en effet d'une boucle for qui va s'exécuter au nombre de lettres que contient le tableau (chars). Le programme va ensuite choisir un numéro aléatoire qui évidemment sera un chiffre entre 0 et le nombre de lettres. Après avoir choisi ce chiffre, la fonction copie la lettre se trouvant dans l'index du chiffre aléatoire pour la mettre dans la première case du tableau, mais avant ça, elle copie la lettre qui sera remplacée pour ne pas la perdre et la met à la place de celle qui la remplacée.

4.1.7 Insertion d'un nouveau mot

La deuxième grande fonctionnalité du programme est de permettre à l'enseignant d'insérer un nouveau mot avec 4 images. Il s'agit d'une nouvelle interface graphique qui a été créée depuis l'espace de travail « Design ». Voici à quoi elle ressemble :



Nous pouvons voir qu'il s'agit d'un formulaire simple pour l'utilisation et pour que n'importe quel utilisateur puisse s'en servir.

La première étape est d'inscrire le nouveau mot. Ensuite, l'utilisateur doit choisir le thème auquel appartient son mot dans la liste en dessous. Après avoir remplis ces informations, il doit choisir les 4 images en lien avec le mot. Le programme affiche ensuite les liens des images tout en bas. Une fois que l'utilisateur a tout bien rempli, il peut cliquer sur le bouton « Ajouter » et le programme se chargera d'ajouter le mot.

En cas d'erreur de sélection d'une image, il est possible de la supprimer de la liste. Il suffit de sélectionner le lien de l'image que l'on souhaite supprimer puis cliquer sur la touche « Delete » de notre clavier.

Lorsque l'utilisateur clique sur le bouton « Ajouter », toutes les vérifications se font. L'application vérifie évidemment que l'utilisateur a bien mis un mot et donc que la case mot n'est pas vide puis contrôle que le mot n'est pas seulement un seul caractère. Les images sont aussi vérifiées et le programme regarde si le mot contient bien 4 images.

Une fois que les vérifications ont été effectuées, l'application va copier les images dans un répertoire spécifique pour qu'elles soient utilisables à n'importe quel endroit. Voici la fonction qui nous permet de faire ça :

```
//Copier les images dans un répertoire spécifique
for(int i = 0; i < lv_lienImage.Items.Count; i++)
{
    //Récupérer le nouveau mot
    string str_mot = tb_mot.Text;

    //Répertoire ou l'image sera copier
    string str_destination = @".\images\" + str_mot + "" + int_noImage + ".png";
    int_noImage++;

    //Copie de l'image dans le répertoire
    if (!File.Exists(str_destination))
    {
        File.Copy(Convert.ToString(lv_lienImage.Items[i].Text), str_destination);
    }
}
```

Il s'agit d'une boucle qui va s'exécuter au nombre d'images (donc 4). Le programme récupère tout d'abord le nouveau mot dans une variable « str_mot ». Ensuite, une variable est créée pour initialiser la destination de l'image (l'endroit où l'image sera copiée). Toutes les images sont copiées dans le même répertoire qui est « .\images\ ». J'ai choisi de nommer les images avec le nom qu'elles décrivent puis d'un numéro qui s'incrémente entre 0 et 3. Le programme vérifie si le mot n'existe pas déjà puis va copier.

```
//Appel de la fonction pour insérer le nouveau mot
string idmot = fn_connexion.insertMot(tb_mot.Text, Convert.ToString(idtheme));

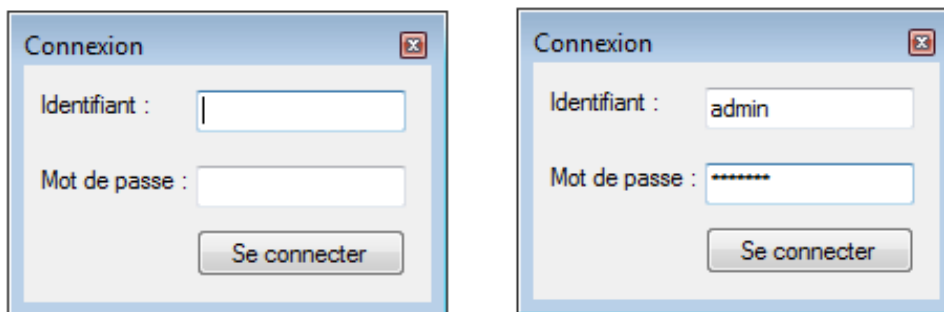
//Boucle pour insérer les images dans la base de données
for(int j = 0; j < lv_lienImage.Items.Count; j++)
{
    fn_connexion.insertImage(lv_lienImage.Items[j].Text, idmot);
}
```

Nous pouvons voir ici que la première requête appelée est celle d'ajouter un mot « InsertMot ». Nous lui donnons en paramètres le nouveau mot ainsi que l'id du thème auquel il appartient. Cette fonction va insérer le nouveau mot. Récupérer son ID et faire le lien entre le mot et le thème dans la table « t_appartenir ». Une fois que le mot est bien inscrit dans la base de données, nous allons nous occuper des images. Il y a encore une boucle for qui s'exécute au nombre d'images (4). Nous travaillons donc image par image. La fonction utilisée est « InsertImage » et nous passons en paramètres le lien de l'image et l'id du mot. La fonction va insérer la nouvelle image avec son lien puis récupérer son id. Elle va ensuite faire le lien entre l'image et le mot dans la table « t_decrire ».

4.1.8 Gestion des droits

Pour la gestion des droits de l'application (l'ajout de nouveaux mots). J'ai tout d'abord penser à utiliser le LDAP pour accéder à l'active directory de l'ETML. C'est quelque chose que j'avais déjà fait durant mon stage et c'était très efficace car nous allons rechercher directement les vraies données. Si une personne change de groupe, ses droits auraient changé directement dans l'application. Grâce à cela j'aurais pu gérer les droits très facilement. Une fois l'utilisateur connecter, il aurait suffi de rechercher si l'utilisateur appartient au groupe des enseignants et si oui, lui autoriser l'accès au formulaire pour créer un nouveau mot. Malheureusement je n'ai pas pu faire cette méthode car le SIE (service informatique de l'ETML), n'autorise pas l'accès à leur AD aux élèves.

J'ai finalement choisi de gérer cette partie avec un simple compte administrateur. Tout d'abord j'ai créé une nouvelle table dans la base de données contenant les informations de connexion au compte. Lorsque l'utilisateur clique sur le bouton pour ajouter un nouveau mot, un formulaire s'affiche et demande de se connecter :



Lorsque l'utilisateur clic sur le bouton « Se connecter », le programme va chercher les informations dans la base de données et les compare avec celle insérer par l'utilisateur. Si les informations sont correctes le formulaire de création d'un nouveau mot s'ouvre, sinon, un message d'erreur s'affiche tant que l'identifiant ou le mot de passe sont faux.

Voici la méthode qui permet de « hasher » le mot de passe insérer par l'utilisateur pour pouvoir le comparer après.

```
var hash = (new SHA1Managed()).ComputeHash(Encoding.UTF8.GetBytes(tb_mdp.Text));  
string mdpSha1 = string.Join("", hash.Select(b => b.ToString("x2")).ToArray());
```

J'ai choisi cette méthode de connexion car il s'agit d'une méthode pas très compliquée et assez sécurisée.

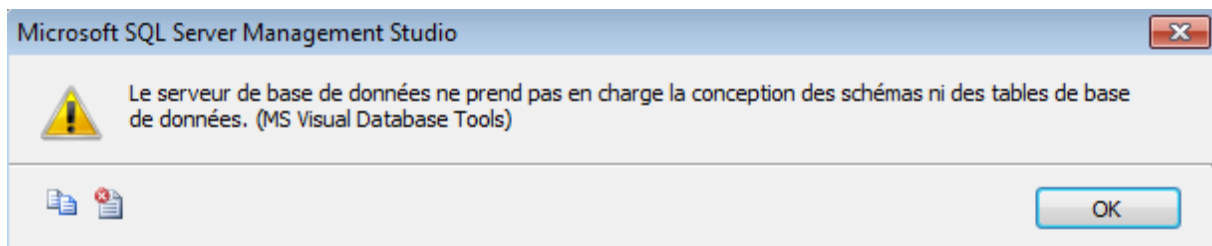
Identifiant : admin

Mot de passe : tpi\$123

4.2 Modifications

4.2.1 Changement Microsoft SQL Server Management Studio

Après avoir installé mon serveur SQL 2014, j'ai créé une nouvelle base de données grâce au script de JMerise. La base de données a très bien fonctionné et j'arrivais bien à récupérer des données depuis l'application C#. Le problème était que je n'arrivais pas à ouvrir mes tables en mode création et à utiliser quelques autres fonctionnalités depuis le SQL Server Management Studio. Voici le message d'erreur :



Suite à quelques recherches, j'ai vu qu'il s'agissait d'un problème de version. J'ai donc essayé d'installer la dernière version du management studio qui est la 2016 et après avoir installé celle-ci, j'ai enfin réussi à manipuler mes tables et base de données comme je le voulais avec toutes les fonctionnalités que propose Microsoft.

4.2.2 Ajout d'une table « droit » à la base de données

J'ai créé une nouvelle table dans la base de données qui contient les informations de connexion à un compte administrateur. Ce compte permettra de se connecter à la fonctionnalité « Ajouter un nouveau mot ». Au début, j'avais prévu de faire d'une autre manière et c'est pour cela que la table n'a pas été créée plus tôt. Il s'agit d'une table ne contenant aucun lien avec d'autres tables.

MCD / MLD :

t_connexion		
PrK	<u>idConnexion</u>	Auto_increment
Uni	conIdentifiant	Varchar
	conMotDePasse	Varchar

5 TESTS

5.1 Dossier des tests

JEU

CASE « RÉPONSE »	Mettre des mauvaises réponses	OK
	Mettre des chiffres, caractères spéciaux	OK
	Ne rien mettre	OK

NOUVEAU MOT

CASE « MOT »	Mettre des chiffres, caractères spéciaux	OK
	Ne rien mettre	OK
	Mettre un long mot de plus de 20 caractères	OK
	Mettre un seul caractère	OK
AJOUTER IMAGES	Ajouter un document « word, pdf... »	OK
	Ne pas mettre d'images	OK
	Mettre seulement 3 images	OK

Il y a eu des tests unitaires qui ont été faits au fur et à mesure de l'avancement du programme. Après chaque nouvelle fonctionnalité, j'effectuais des tests pour voir si tout fonctionne bien. Les tests finaux m'ont permis de trouver des améliorations pour rendre l'application plus facilement utilisable.

Voici les améliorations trouvées après les tests :

1. Permettre à l'utilisateur de supprimer les images sélectionnées et les changer dans "Ajouter un nouveau mot".
2. Dans la fonctionnalité "Ajouter un nouveau mot" il était possible d'ajouter un mot déjà existant dans la base de données et ça faisait planter le programme. J'ai donc ajouté une vérification avant d'insérer le mot pour voir s'il existe déjà.
3. Vérifier si le thème possède des mots ou pas. S'il ne possède pas de mots afficher un message. À la fin des mots du thème afficher un message et le nombre de points.

6 CONCLUSION

6.1 Bilan des fonctionnalités demandées

Dans l'ensemble, le programme est fonctionnel. Les fonctionnalités demandées ont été réalisées. Il s'agit en effet d'une application simple à utiliser par n'importe quel utilisateur, le programme propose différents thèmes, la gestion entre utilisateur et enseignant a été mis en place. Le seul point dont je n'ai pas pu réaliser était de faire une installation simple, comme sur les smartphones. Étant donné que nous avons choisi un SQL Serveur comme base de données, il n'était plus vraiment possible d'installer simplement l'application.

Idées d'améliorations :

1. L'une des plus grandes améliorations serait d'automatiser les thèmes. Il faudrait ajouter la possibilité d'ajouter des nouveaux thèmes aux utilisateurs. Comme pour ajouter un nouveau mot, les utilisateurs devraient pouvoir aussi insérer un thème qui n'est pas dans la liste. Cette fonctionnalité apporterait beaucoup de changement au code. Il faudrait aussi au menu d'accueil « Choix du thème » afficher les thèmes en recherchant directement dans la base de données les thèmes insérés.
2. Pour le moment, l'application affiche les mots du thème à la suite logique. C'est-à-dire que lorsque l'utilisateur choisit un thème, le programme va afficher les mots depuis le premier au dernier et tout le temps dans le même ordre. Pour l'instant ce n'est pas très dérangement car le jeu contient pas énormément de mots mais lorsqu'il y aura beaucoup d'images, ça pourrait être une bonne idée d'afficher les mots aléatoirement et pas toujours dans le même ordre.
3. Pour que l'application soit facile à installer, il faudrait choisir une base de données plus légère qu'un serveur SQL. Par exemple une base de données SQL Lite.
4. Je pense qu'il serait assez intéressant de faire le programme en utilisant le WPF et non pas le Windows Forms. Cela pourrait donner une application plus agréable graphiquement.
5. Pour la gestion des droits, si l'application devrait être utilisée. Il faudrait sans doute utiliser un LDAP pour différencier l'utilisation entre élèves et celles entre professeurs.

6. Une amélioration qui est à faire sans cesse, c'est enrichir la base de données. C'est quelque chose de très important. Plus la base de données est grande et plus les gens pourront utiliser l'application plus longtemps.
7. Il pourrait être intéressant de développer l'application sous téléphone portable, tablette et tout autre appareil.
8. Ajouter une option qui serait pour aider l'utilisateur lorsqu'il n'arrive pas à trouver le mot. Il pourrait cliquer sur l'aide et ça lui afficherait le mot caché. L'utilisateur perdrait des points lorsqu'il utiliserait l'aide. Après lui avoir afficher la solution, le programme passerait au mot suivant.

6.2 Bilan de la planification

La planification a été assez bien suivi dans l'ensemble. J'avais prévu un peu plus de temps pour les fonctionnalités mais après avoir fait la connexion entre l'application et la base de données, les fonctionnalités ont été assez vite effectuées. Plus vite que prévu. Ce qui m'a pris du temps c'était les petites améliorations et petits détails. Il y a plusieurs manières d'effectuer les fonctionnalités (code) et j'ai parfois dû corriger et faire d'une autre manière pour que ça fonctionne avec mon programme.

6.3 Bilan personnel

Si c'était à refaire :

Pour commencer, je ne ferais pas mon application avec Windows Forms car j'ai remarqué qu'il n'était pas vraiment possible de faire des interfaces graphiques très jolies. Je ferais surement une application WPF. Ensuite, je choisirais une autre base de données qu'un SQL Serveur, mais une base de données portable et moins lourde. Une chose pourrait être intéressant, ça serait de développer l'application pour application mobile aussi.

Le projet c'est plutôt bien déroulé et nous étions vraiment dans de bonnes conditions de travail. Ça nous a permis de bien nous concentrer et de bien être dans notre projet. Il s'agissait d'un travail très complet, mettant en œuvre pleins de modules appris durant notre formation, et je pense qu'il s'agit d'une très bonne manière de conclure notre formation.

Je tenais à remercier, mon chef de projet M. Gruaz, qui m'a aussi suivi pendant mes stages en entreprise, et qui m'a accompagné tout au long du projet. Mes deux experts de projet M. Viret et Mme. Andolfatto. Mais aussi M. Marengo et M. Jorge qui ont pris du temps pour tester mon application et trouver des petites failles.

7 DIVERS

7.1 Journal de travail

Lundi			
Discussion - Réunion	3	Mise en route du TPI	
Analyse du programme	3	Analyse du projet	
Planification des tâches	6	Création de la planification initiale	
Mercredi			
Planification des tâches	9	Création de la planification initiale	
Planification des tâches	4	Mise en place du GIT	
Documentation	2	Début du rapport de projet (spécifications)	
Création de l'interface graphique	10	Interface graphique du programme avec les différents menus	GIT "Analyse" fichier : MM-MaquetteInterfaceGraphique.vsd
Analyse base de données	2	Analyse de la base de données (MCD)	
Jeudi			
Analyse base de données	9	Analyse de la base de données (MCD, MLD, MPD)	GIT "Analyse" fichiers : MM-Analyse-BD.vsd , MM-TPI-MPD-Krasniqika.docx
Création de l'interface graphique	2	Correction interface graphique	
Documentation	5	Ajout de la planification initiale, interface graphique avec explications	Rapport de projet Chapitre 2 et 3.2.1
Analyse du programme	9	Analyse du programme + structogramme	GIT "Analyse" fichier : MM-Structo.PNG
Journal de travail	2	Mise à jour du journal de travail	
Vendredi			
Documentation	12	Ajout de l'analyse base de données et du structogramme	Rapport de projet Chapitre 3.2.2 et 3.2.3
Analyse connexion c# et sql server	15	Recherche moyen de faire la connexion c# -> SQL server + test sur visual studio	
Total semaine	93	Max. 93	

Semaine 2		Date: lundi 8 mai 2017	
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Lundi			
Discussion - Réunion	2	Discussion avec le chef de projet	Modification de la base de données
Analyse base de données	6	Correction de l'analyse de la base de données (MCD, MLD, MPD)	GIT "Analyse" fichier : MM-ConceptionBDJmerise-Krasniqika (Ouvrable avec l'application Jmerise) , MM-MCD-Krasniqika.PNG + MM-MLD-Krasniqika.PNG , MM-TPI-MPD-Krasniqika.docx
Planification des	4	Planification détaillée	
Mercredi			
Documentation	3	Correction de l'analyse de la base de données	Rapport de projet chapitre 3.2.2
Planification des tâches	3	Planification détaillée	GIT "Documentation" fichier : T-TPI-PlanificationDetaillee-Krasniqika.xlsm
Journal de travail	1	Modification du journal de travail (initiale -> détaillé)	
Création de l'interface graphique	17	Interface graphique sur visual studio. Page d'accueil avec thèmes : OK, Page de jeu : OK, page "ajouter un nouveau mot" : Presque OK, je cherche encore une petite fonction pour pouvoir récupérer les images grâce à l'explorateur windows avec le composant "SaveFileDialog"	SaveFileDialog : https://msdn.microsoft.com/fr-fr/library/sfez97z(v=vs.110).aspx L'interface graphique est codée sous visual studio et sera bientôt dans le rapport de projet
Analyse base de données	1	Commentaire pour les différentes tables dans Jmerise	GIT "Analyse" fichier : MM-ConceptionBDJmerise-Krasniqika (Ouvrable avec l'application Jmerise)
Journal de travail	2	Mise à jour du journal de travail	
Jeudi			
Création de l'interface graphique	6	Ajout du bouton retour depuis la page de jeu, Améliorer l'affichage des images, Récupérer les fichiers (images) sélectionner par l'utilisateur : OK	
Création de la base de données	6	Script sql server (compréhension du code) + création de la base de données	Information sur le code : https://blog.developpez.com/sqlpro/p6135/ms-sql-server/auto_increment_identity_avec_sql_server + https://msdn.microsoft.com/fr-fr/library/ms190457.aspx ,
Création de la base de données	8	Installation de "SQL Server Management Studio 2016" car avec le 2014 je n'arrivais pas à ouvrir certaines fonctionnalités comme ouvrir les tables en mode "création" pour les modifier par exemple.	<u>Lien ou l'ai trouver la solution au problème :</u> http://stackoverflow.com/questions/25146474/the-backend-version-is-not-supported-to-design-database-diagrams-or-tables
Documentation	6	Mise à jour de la documentation (Réalisation interface graphique et base de données)	Rapport de projet chapitre 4.1.1 et 4.1.2
Journal de travail	1	Mise à jour du journal de travail	
Vendredi			
Absence - Imprévu	2	Sauvegarde sur une clé usb de la machine virtuelle avec laquelle je travail	
Développement : ajouter une fonctionnalité pour ajouter des nouveaux mots avec des images	2	Recherche pour différencier l'utilisation enseignant de l'utilisation élève	LDAP : https://www.mssqltips.com/sqlservertip/2580/querying-active-directory-data-from-sql-server/
Développement : récupérer les images avec le mot et afficher dans l'application	4	Connexion à la base de données	http://csharp.net-informations.com/data-providers/csharp-sql-server-connection.htm , http://stackoverflow.com/questions/15631602/how-to-set-sql-server-connection-string , https://msdn.microsoft.com/fr-fr/library/cc437981(v=vs.71).aspx Rapport de projet : 4.1.3
Développement : récupérer les images avec le mot et afficher dans l'application	6	Récupérer des données de la table qui contient les mots	Rapport de projet 4.1.4
Test	2	Vérifier si l'application arrive bien à récupérer les données et si elles sont juste	
Développement : récupérer les images avec le mot et afficher dans l'application	6	Amélioration du code pour séparer la partie code application c# et la partie interaction avec la base de données SQL.	Rapport de projet 4.1.4
Documentation	3	Mise à jour de la documentation (Connexion à la base de données, Récupération des données d'une table)	Rapport de projet : 4.1.3 et 4.1.4
Journal de travail	2	Mise à jour du journal de travail	
Total semaine	93	Max. 93	

Semaine 3		Date: lundi 15 mai 2017	
Tâche	Durée (1/4 h.)	Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Lundi			
Développement : récupérer les images avec le mot et afficher dans l'application	2	Recherche d'images sur internet pour pouvoir tester l'application	
Créer les tests à effectuer	1	Création des tests à effectuer	
Documentation	2	Transmission des tests dans la documentation	Rapport de projet : 3.3
Développement : récupérer les images avec le mot et afficher dans l'application	6	Crée fonction pour récupérer les images, recherche de conversion du type "Datareader" en type list	
Journal de travail	1	Mise à jour du journal de travail	
Mercredi			
Discussion - Réunion	2	Visite de Mme. Andolfatto (Expert 2)	
Développement : récupérer les images avec le mot et afficher dans l'application	2	Conversion du type "Datareader" en liste	http://stackoverflow.com/questions/320180/how-can-i-convert-sqldatareader-to-list
Développement : récupérer les images avec le mot et afficher dans l'application	9	Création des fonctions (Requêtes) pour récupérer : le ID du thème choisi, les mots appartenants au thème choisi, les images appartenants au mot	Git (Programme C#4Images1Mot) : dbConnexion.cs
Développement : récupérer les images avec le mot et afficher dans l'application	2	Insertion des liens des images dans la base de données (Petit problème d'insertion des liens car les liens était trop long par rapport à la taille maximum. J'ai donc du modifier la taille maximum)	
Développement : récupérer les images avec le mot et afficher dans l'application	5	Amélioration du code (simplification des requêtes SQL), création de classes pour récupérer les données dans des listes d'objet et non juste des listes car je ne pouvais pas récupérer plusieurs données différentes mais seulement un seul. Par exemple je pouvais seulement récupérer les ID sans d'autre champs	Git (Programme C#4Images1Mot) : getMotTheme.cs , getImageMot.cs + dbConnexion.cs
Test	4	Test du programme : ajout d'images, ajout des liens entre images et mot, ajout thème, ajout liens entre mot et thème, test si le programme récupère bien seulement les mots du thème choisi et pas d'autre, test si le programme récupère bien seulement les images du mot et pas d'autres images en plus	Test OK
Développement : récupérer les images avec le mot et afficher dans l'application	2	Changement de la méthode pour afficher les images car c'était une boucle et je ne pouvais pas l'arrêter pour attendre la réponse de l'utilisateur	Git (programme C#4Images1Mot) : formMenu.cs
Journal de travail	1	Mise à jour du journal de travail	
Jeudi			
Développement : récupérer les images avec le mot et afficher dans l'application	2	Chercher des images et des mots pour enrichir la base de données	
Développement : ajouter une fonctionnalité pour ajouter des nouveaux mots avec des images	3	Copier et coller l'image sélectionner par l'utilisateur dans un répertoire spécifique	https://openclassrooms.com/forum/sujet/copier-coller-fichier-en-c https://www.developpez.net/forums/d1336394/dotnet/langa-gestsharp/programme-copier-image-d-dossier-csharp/
Documentation	4	Mise à jour de la documentation (Récupérer les images avec le mot en cours)	Rapport de projet : 4.1.5
Développement : afficher les lettres du mot à trouver aléatoirement	4	Affichage des lettres du mot aléatoirement	https://www.developpez.net/forums/d348889/dotnet/genera-l-dotnet/melanger-chaine-caractere/
Développement : récupérer les images avec le mot et afficher dans l'application	2	Amélioration du code pour ajuster automatiquement les images de n'importe quelle taille	http://stackoverflow.com/questions/16822138/fit-image-into-picturebox
Documentation	4	Mise à jour de la documentation (Modification MSSMS, Affichage des lettres aléatoire)	Rapport de projet : 4.1.6, 4.2.1
Développement : calcul des points	3	Petit système de calcul des points	
Test	1	Test du programme : test de tous les boutons, test du jeu	
Développement : amélioration	2	Amélioration du programme : Vérifier si le thème possède des mots ou pas si il ne possède pas de mots afficher un message, fin des mots du thème afficher un message et le nombre de points	
Développement : amélioration	1	Mettre des commentaires pour le code	Programme : pages de code
Journal de travail	1	Mise à jour du journal de travail	

Vendredi			
Développement : ajouter une fonctionnalité pour ajouter des nouveaux mots avec des images	2	Vérification du nouveau mot entré par l'utilisateur, des images	Git (Programme C#4Images1Mot) : FormNew.cs
Développement : ajouter une fonctionnalité pour ajouter des nouveaux mots avec des images	2	Changement de la méthode (Copier coller image sélectionnée par l'utilisateur dans un répertoire spécifique). Elle se fait désormais à la fin du programme lorsque l'utilisateur clique sur le bouton "valider".	Git (Programme C#4Images1Mot) : FormNew.cs
Développement : ajouter une fonctionnalité pour ajouter des nouveaux mots avec des images	6	Insérer le nouveau mot dans la base de données puis récupérer son id, lier le mot et le thème	Git (Programme C#4Images1Mot) : FormNew.cs, dbConnexion.cs
Test	2	Test pendant le développement ci-dessus. (création de nouveaux mots, création de liens)	
Développement : ajouter une fonctionnalité pour ajouter des nouveaux mots avec des images	6	Insérer les images dans la base de données, récupérer les id, lier les images avec le mot	Git (Programme C#4Images1Mot) : FormNew.cs, dbConnexion.cs
Test	3	Test d'ajout des nouveaux mots avec images (petits bugs)	
Documentation	4	Mise à jour de la documentation (Insertion d'un nouveau mot)	Rapport de projet : 4.1.7
Journal de travail	2	Mise à jour du journal de travail	
Total semaine	93	Max. 93	

Semaine 4		Date: lundi 22 mai 2017	
Tâche	Durée (h4 h.)	Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Lundi			
Test	2	Test du programme avec un collègue	
Développement : amélioration	1	Ne pas autoriser l'utilisateur à modifier le thème car lorsque l'utilisateur voulait créer un nouveau mot il pouvait modifier la combobox et donc mettre un thème qui n'existe pas	http://stackoverflow.com/questions/598447/how-to-disable-editing-of-elements-in-combobox-for-c
Documentation	1	Mise à jour de la documentation (gestion des droits)	Rapport de projet : 4.1.8
Développement : ajouter une fonctionnalité pour ajouter des nouveaux mots avec des images	2	Gestion des droits : Rechercher un moyen pour faire la gestion des droits. (Idap dans le sql serveur, table avec les droits, compte utilisateur avec mot de passe hasher)	
Discussion - Réunion	1	Discussion avec le chef de projet. Conclusion : Important d'enrichir la base de données du jeu	
Développement : récupérer les images avec le mot et afficher dans l'application	1	Ajout d'images avec des mots pour enrichir la base de données	
Développement : récupérer les images avec le mot et afficher dans l'application	2	Découverte d'un petit problème. Lors de l'insertion des liens des images dans la base de données, le programme reprenait le lien de la source et non pas du répertoire spécifique	
Développement : récupérer les images avec le mot et afficher dans l'application	2	Ajout d'images avec des mots pour enrichir la base de données	
Mercredi			
Absence - Imprévu	2	Sauvegarde sur une clé usb de la machine virtuelle avec laquelle je travail	
Développement : récupérer les images avec le mot et afficher dans l'application	2	Ajout d'images avec des mots pour enrichir la base de données	
Développement : ajouter une fonctionnalité pour ajouter des nouveaux mots avec des images	2	Petit bug découvert. Lorsque l'on cliquait sur le bouton pour charger une image, l'explorateur windows nous permettait de sélectionner seulement les fichiers .jpg. J'ai donc ajouté les autres extensions d'images.	https://stackoverflow.com/questions/4710881/multiple-file-extensions-in-openfiledialog
Test	7	Installation d'une nouvelle machine virtuelle Windows, puis différentes installations pour utiliser l'application c# (Installation .net Framework, sql serveur, création de la base de données, création des tables, exporter et importer les données dans les tables)	
Documentation	6	Installation d'une nouvelle machine virtuelle Windows, puis différentes installations pour utiliser l'application c#. Ajout du cahier des charges	Rapport de projet : 8.1 - 8.2
Création de la base de données	3	Recherche un moyen pour exporter les données simplement	http://www.scribfluent.fr/blog/expertise/SQL-Exporter-le-contenu-d'une-table
Documentation	1	Ajout de commentaire dans le code	
Test	2	Test du programme et des fonctionnalités dans la nouvelle machine virtuelle	Test OK, les fonctionnalités fonctionnent comme sur la machine de base
Journal de travail	2	Mise à jour du journal de travail	

Jeudi			
Absence - Imprévu	27		
Vendredi			
Absence - Imprévu	27		
Total semaine	93	Max. 93	

Semaine 5		Date: lundi 29 mai 2017	
Tâche	Durée (1/4 h.)	Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Lundi			
Test	3	Tests du programme (petite amélioration trouvée)	
Développement : amélioration	2	Amélioration suite aux tests : Dans la fonctionnalité "Ajouter un nouveau mot" il était possible d'ajouter un mot déjà existant dans la base de données et ça faisait planter le programme. J'ai donc ajouté une vérification avant d'insérer le mot pour voir si il existe déjà	
Développement : ajouter une fonctionnalité pour ajouter des nouveaux mots avec des images	3	Gestion des droits	https://stackoverflow.com/questions/17292366/hasing-with-sha1-algorithm-in-c-sharp
Création de l'interface graphique	2	Gestion des droits	
Documentation	1	Mise à jour de la documentation (amélioration)	Rapport de projet : 6.1.1
Journal de travail	1	Mise à jour du journal de travail	
Mercredi			
Création de la base de données	3	Modification de la base de données, Analyse Jmersie, script SQL (ajout d'une table pour les droits)	
Documentation	1	Modification de la base de données	Rapport de projet : 4.2.2
Développement : ajouter une fonctionnalité pour ajouter des nouveaux mots avec des images	5	Ajout d'un formulaire de connexion lorsque l'utilisateur souhaite ajouter un nouveau mot.	
Documentation	2	Mise à jour de la documentation (connexion)	Rapport de projet : 4.1.8
Test	3	Tests du programme (petite amélioration trouvée)	Rapport de projet : 5.1
Journal de travail	1	Mise à jour du journal de travail	
Absence - Imprévu	12	Examen de culture générale	
Jeudi			
Absence - Imprévu	2	Sauvegarde sur une clé usb de la machine virtuelle avec laquelle je travail	
Documentation	3	Amélioration du rapport de projet chapitre : 4.7, corrections, début de la conclusion Chapitre : 6	
Test	3	Tests de l'application petite amélioration découverte (permettre à l'utilisateur de supprimer les images sélectionnées et les changer dans "Ajouter un nouveau mot")	Rapport de projet 4.7
Développement : amélioration	2	amélioration permettre à l'utilisateur de supprimer les images sélectionnées et les changer dans "Ajouter un nouveau mot"	https://stackoverflow.com/questions/16829678/how-to-delete-selected-items-from-a-listview-by-pressing-the-delete-button
Absence - Imprévu	2	Séance de classe	
Documentation	6	Amélioration du rapport, Planification détaillée	Rapport de projet : 3.4 - 6.1 - 5.1, 6.2 - 6.3 - 7.2
Développement : récupérer les images avec le mot et afficher dans l'application	3	Ajout d'images avec des mots pour enrichir la base de données	
Test	3	tests de l'application en ajoutant des nouveaux mots. Petit bug découvert (à l'ajout d'un mot, lorsque l'on ajoute un mot le programme ne donne aucun message)	
Développement : amélioration	2	Ajouter un message lorsque l'utilisateur ajout un nouveau mot pour lui dire que son mot a bien été ajouté puis remettre à zéro le formulaire de nouveau mot.	
Journal de travail	1	Mise à jour du journal de travail	
Vendredi			
Test	3	Test du programme, test du programme sur une autre machine physique	
Développement : afficher les lettres du mot à trouver aléatoirement	2	Afficher d'autres lettres en plus des lettres du mot	https://stackoverflow.com/questions/15243138/pick-random-char
Documentation	3	Mise à jour de la documentation	Rapport de projet : 6.1, 6.3
Test	3	Installation d'une machine virtuelle et installation de l'application dans une machine vierge en suivant la procédure	Rapport de projet : 8.2
Documentation	2	Correction de la procédure d'installation	Rapport de projet : 8.2
Documentation	1	Solution du jeu	
Absence - Imprévu	3	Recherche d'un moyen de mettre à disposition la machine virtuelle pour le chef de projet et les experts. CD/DVD pas assez de place. Solution : cloud ou machine virtuelle	
Test	3	Test de l'application, vérifier que tout fonctionne bien pour le rendu final	
Documentation	3	Amélioration de la documentation, ajout du journal de travail (essai)	
Absence - Imprévu	2	Copie de la machine virtuelle sur un disque dur externe pour le chef de projet	
Journal de travail	2	Mise à jour du journal de travail	
Total semaine	93	Max. 93	

Semaine 6		Date: lundi 5 juin 2017	
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Documentation	5	Corrections de la documentations, ajout du journal de travail	Rapport de projet : 7.1
Journal de travail	2	Mise à jour du journal de travail	
Absence - Imprévu	5	Imprimer et relier le rapport de projet	

7.2 Webographie

[https://msdn.microsoft.com/fr-fr/library/cc437981\(v=vs.71\).aspx](https://msdn.microsoft.com/fr-fr/library/cc437981(v=vs.71).aspx)

<http://www.windows8facile.fr/telecharger-installer-sql-server-2014/>

[https://msdn.microsoft.com/fr-fr/library/9yyz8a6c\(v=vs.110\).aspx](https://msdn.microsoft.com/fr-fr/library/9yyz8a6c(v=vs.110).aspx)

<https://www.developpez.net/forums/d1375619/general-developpement/alm/connexion-base-donnees-sql-server-csharp/>

https://www.tutorialspoint.com/entity_framework/entity_framework_environment_setup.htm

<https://www.codeproject.com/Articles/823854/How-to-connect-SQL-Database-to-your-Csharp-program>

<https://www.developpez.net/forums/d348889/dotnet/general-dotnet/melanger-chaine-caractere/>

<http://www.softfluent.fr/blog/expertise/SQL-Exporter-le-contenu-dune-table>









<https://stackoverflow.com/questions/17292366/hashing-with-sha1-algorithm-in-c-sharp>

<https://stackoverflow.com/questions/6829678/how-to-delete-selected-items-from-a-listview-by-pressing-the-delete-button>

8 ANNEXES

8.1 Cahier des charges

1-INFORMATIONS GENERALES

Candidat :	Nom : KRASNIQI	Prénom : KADRI
	 : mailto:krasniqika@etml.educaenet2.ch	 :
Lieu de travail :	ETML, Sébeillon 12 1004 Lausanne	
Chef de projet :	Nom : Gruaz	Prénom : Gilbert
	 : mailto:gilbert.gruaz@vd.educaenet2.ch	 : 079 338 7808
Expert 1 :	Nom : Viret	Prénom : Didier
	 : mailto:didier.viret@bluewin.ch	 : 078 678 22 41
Expert 2 :	Nom : Andolfatto	Prénom : Frédérique
	 : mailto:frederique.andolfatto@cprv.ch	 : 077 206 66 45
Dates de réalisation :	Du lundi 1 mai au mercredi 7 juin à 11H25	
Horaire de travail : (Basé sur l'horaire officiel)	Lundi → 08h00-11h25 →	Pentecôte, 5 juin 2017
	Mardi	
	Mercredi → 08h00-12H15/13h10-16h35 →	Exo CG 31 mai 2017 après-midi
	Jeudi → 08h00-11H25/12h20-16h35 →	Pont de l'Asc, 25 mai 2017
	Vendredi → 08h00-12H15/13h10-16h35 →	Pont de l'Asc, 26 mai 2017
Présentation :	Entre le mercredi 14 et jeudi 15 juin 2017	
Nombre d'heures :	Environ 110 heures	
Planning (en H ou %)	Analyse : 15%, Implémentation 50%, Tests 15%, Doc. 20%	

2-PROCÉDURE

Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le 1er jour.

Le cahier des charges est approuvé par la i-CQ VD. Il est en outre présenté, commenté et discuté avec le candidat. Par sa signature, le candidat accepte le travail proposé.

Le candidat a connaissance de la feuille d'appréciation avant de débiter le travail.

Le candidat est entièrement responsable de la sécurité de ses données.

En cas de problèmes graves, le candidat avertit au plus vite les deux experts et son chef de projet.

Le candidat a la possibilité d'obtenir de l'aide, mais doit le mentionner dans son dossier de projet.

A la fin du délai imparti pour la réalisation du TPI, le candidat doit transmettre par courrier électronique le dossier de projet aux deux experts et au chef de projet. En parallèle, une copie papier du rapport doit être fournie sans délai en trois exemplaires. Cette dernière doit être en tout point identique à la version électronique.

3-TITRE

Réalisation d'un quiz de type 4 images 1 mot en **c#**

4-SUJET

Il s'agit d'implémenter une application de type "4 images 1 mot" (exemple : <http://4images1mot.info>) , en c#, en vue d'offrir un moyen pédagogique aux enseignants de l'etml pour soutenir le processus d'apprentissage des élèves dans la matière qu'ils enseignent.

5-MATÉRIEL ET LOGICIEL À DISPOSITION

1 ordinateur standard etml, avec la structure habituelle →
(Visual Studio 2015, SQL Server 2014, **Notepad++**)

6-PRÉREQUIS

Avoir suivi les modules ICH à l'ETML, les projets et effectué des stages... Les modules ICH suivants sont les plus impliqués : 100, 104, 105, 303 et 306

7-DESCRIPTIF DU PROJET

Les enseignants ne disposent jamais de trop d'outils pour soutenir le processus d'apprentissage des notions de la matière qu'ils enseignent aux élèves.

L'idée de ce projet est de permettre aux enseignants qui l'utilisent d'enregistrer des mots relatifs à la matière enseignée et de placer des images qui permettront aux élèves de déduire le mot.

Par exemple, s'il s'agit de découvrir le mot "cellule" pour décrire une cellule Excel, il y a 4 images qui représentent une cellule de prison, une cellule biologique, une cellule de nid d'abeilles et une cellule d'intersection de ligne et de colonne.

L'enseignant va pouvoir ajouter des images avec les informations nécessaires pour les utiliser dans les questions, ainsi que les mots de vocabulaire du sujet à traiter.

Les élèves pourront utiliser l'application comme cela est le cas avec les jeux qui se téléchargent pour les applications de smartphones.

Le dispositif aura un système avec des catégories de mots selon des branches, ou des modules (par exemple, réseau, **sabot**, serveurs, français, math, etc.).

Cette application doit être simple à mettre en œuvre et à utiliser, tant d'un point de vue "enseignant" que d'un point de vue "élève". Les utilisateurs ne doivent pas avoir de compétences particulières d'informaticiens. Le candidat devra donc concevoir une architecture (bibliothèques, **sabot**, etc.) qui respecte ce besoin et indiquer comment préparer les machines des utilisateurs de l'application pour que celle-ci puisse fonctionner. De même, c'est lui qui décide et conçoit comment distinguer l'utilisation en tant qu'enseignant de celle en tant qu'élève. Il n'y a pas de besoins de serveurs, de centralisations ou de partages. →

En bref, la mise en œuvre doit ressembler à ce qui se fait sur les smartphones : j'installe – j'utilise.

Lors de l'utilisation, l'application affiche 4 images ayant un dénominateur commun, les positions correspondantes des caractères du mot représentant le dénominateur commun, et un ensemble de lettres parmi lesquelles il faut puiser celles qui représentent le mot :



NTCZAIPOCRE

La solution est 5 lettres **ECRAN**

8-POINTS ÉVALUÉS DURANT LE PROJET

- Le comportement et l'engagement durant toute la période pour le travail
- La bonne tenue à jour, systématique, des documents (JNLTRAV, rapport, etc.)
- Les interactions avec le chef de projet

9-LIVRABLES

Le candidat est responsable de mettre à disposition, pour son chef de projet et les deux experts, **un dépôt GIT**, sur un cloud, respectant les conditions suivantes :

- La planification initiale (délai mercredi 3 mai)
- Le rapport de projet (mis à jour au moins 1 fois par semaine)
- Le journal de travail (mis à jour quotidiennement)
- Le code qu'il produit
- Toutes les informations nécessaires de mise en œuvre sur une machine nouvellement installée avec Windows 7 ou 10

10- POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

En plus de cela, le travail sera évalué sur les trois points spécifiques suivants :

- 1.→Le bon fonctionnement de ce qui sera livré, avec des données réalistes en nombre suffisant pour donner une idée du fonctionnement de l'application.
- 2.→La ressemblance avec les applications traditionnelles pour les smartphones
- 3.→Les idées d'améliorations par rapport à la solution proposée

8.2 Mise en place de l'infrastructure pour utiliser l'application

1. Pour commencer, installez une machine avec un système d'exploitation Windows 64 bits. Une fois que celle-ci est bien mise en place, il vous faut installer le .net Framework 4.6.2. Voici le lien :

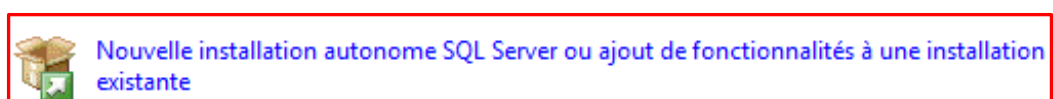
<https://www.microsoft.com/en-us/download/details.aspx?id=53345>

2. Après avoir installé le .net Framework 4.6.2, il faut installer le SQL server 2014. C'est la plus grande installation à faire pour mettre en place l'application. Voici le lien :

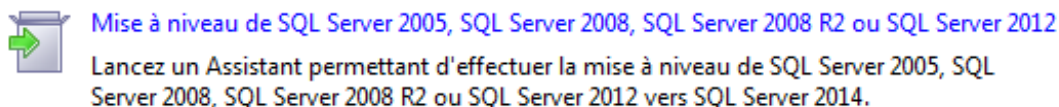
<https://www.microsoft.com/fr-FR/download/details.aspx?id=42299>

3. Aller sur le site puis sélectionner la langue que vous souhaitez et cliquer sur télécharger. Le site vous propose différentes versions à télécharger. Choisissez la version « ExpressAndTools 64\SQLEXPRT_x64_FRA.exe » et cliquez sur Next. Après le téléchargement, cliquer l'exécutable pour lancer l'installation et ensuite choisissez un répertoire où extraire vos fichiers.

Installation :

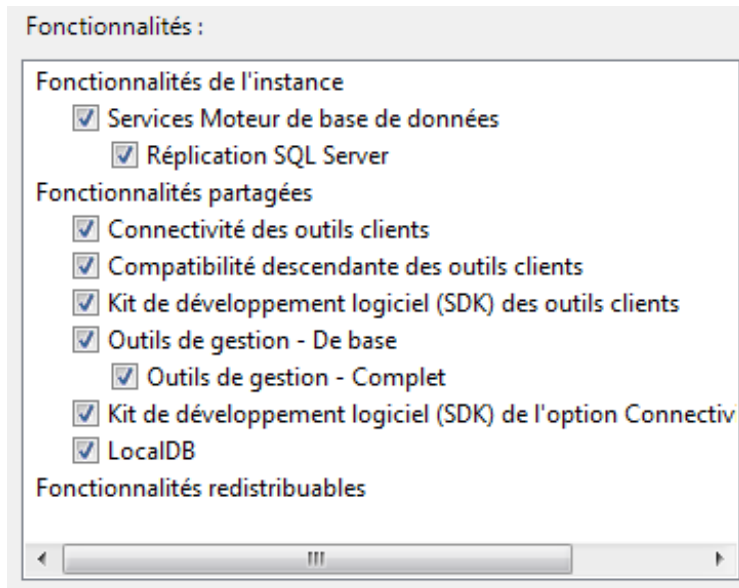


Lancez un Assistant permettant d'installer SQL Server 2014 dans un environnement non cluster ou d'ajouter des fonctionnalités à une instance existante de SQL Server 2014.

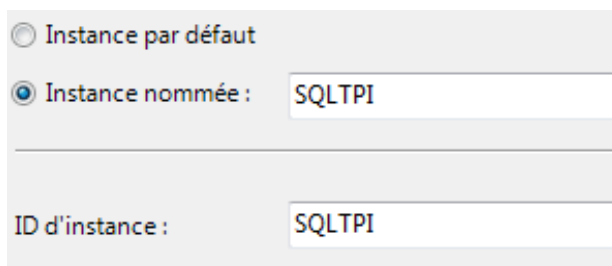


Lancez un Assistant permettant d'effectuer la mise à niveau de SQL Server 2005, SQL Server 2008, SQL Server 2008 R2 ou SQL Server 2012 vers SQL Server 2014.

4. Vous avez maintenant cette interface devant vous. Sélectionnez la première option « Nouvelle installation » puis dans la page suivant, acceptez les termes du contrat de licences. Nous avons la possibilité d'activer les mises à jours mais nous n'allons pas les activer pour éviter les risques de compatibilité.



5. Dans la page suivante, veuillez cocher toutes les fonctionnalités afin de toutes les installer puis cliquer sur suivant.

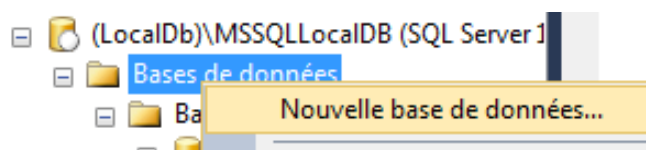


6. Vous avez maintenant la possibilité de choisir un nom à l'instance SQL qui est pas défaut « SQLEXPRESS ». Dans cet exemple, nous pouvons voir qu'il s'agit du nom « SQLTPI ».
7. Pour la partie « Comptes de service », veuillez cliquer sur suivant.
8. Le mode d'identification nous permet de choisir la méthode qui sera utilisé pour nous connecter au serveur. Nous allons rester avec le mode d'identification Windows qui est choisi par défaut et cliquer sur suivant.
9. L'installation va démarrer et il ne vous restera plus qu'à attendre.
10. Une fois l'installation terminée, le programme vous demandera de redémarrer votre machine. Cliquer sur OK. Vous avez maintenant la dernière page de l'installation qui s'affiche vous montrant que toutes les opérations ont été réussies. Vous pouvez fermer cette page et redémarrer la machine.

11. Après avoir redémarrer, vous pouvez lancer le Microsoft SQL Server Management Studio et vous connectez à la base de données local en inscrivant dans la case « Nom du serveur : » le nom suivant : « (LocalDb)\MSSQLLocalDB ». L'authentification reste « Authentification Windows ».

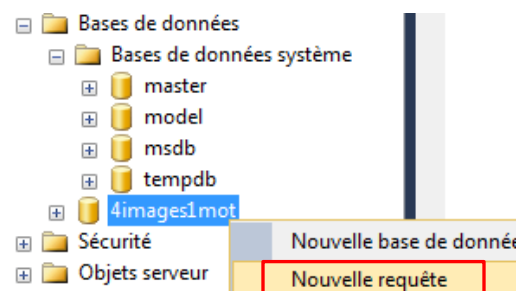


12. Nous allons maintenant créer la base de données. Sous le répertoire « Bases de données », il suffit de faire un clic droit puis « Nouvelle base de données » :



13. Une nouvelle fenêtre va s'ouvrir et il faudra insérer le nom de la base de données qui est : « 4images1mot » puis cliquer sur OK. **ATTENTION : il faut absolument mettre ce nom pour la base de données et pas un autre.**

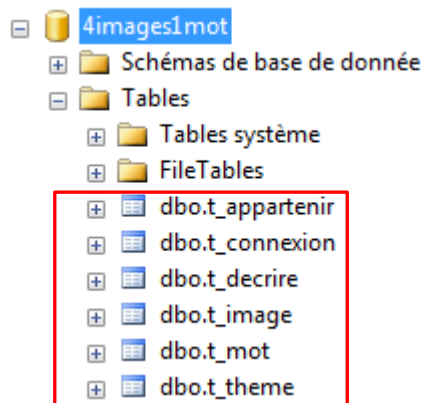
14. Nous allons créer les tables. Dans le répertoire de la base de données, faites un clic droit sur la base de données que nous venons de créer et puis sélectionner la fonctionnalité « Nouvelle requête ». Une nouvelle page vierge s'ouvrira et vous allez pouvoir insérer le script SQL pour créer la base de données.



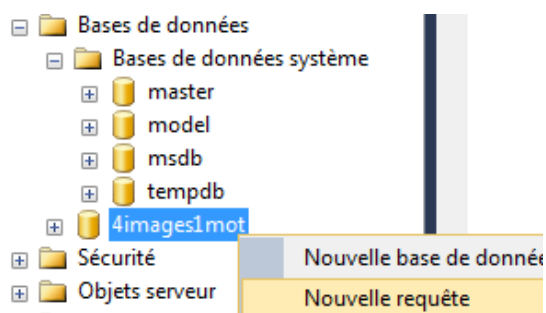
15. Ouvrez le script SQL, puis copier le code et coller le dans la page de création d'une nouvelle requête et pour finir cliquer sur « Exécuter » qui est toujours dans la barre d'outil en haut.



Nous pouvons voir que la base de données contient bien les bonnes tables.



16. Une fois les tables créées, il faut insérer les données. Pour cela il suffit d'exécuter un script SQL. Pour ce faire il faut comme vu précédemment créer une nouvelle requête sur la base de données.



17. Une nouvelle fenêtre va de nouveau s'ouvrir et là il vous faut ouvrir le script de données, copier le contenu et le coller dans cette nouvelle fenêtre vierge. Il vous restera plus qu'à exécuter le script.
18. Pour finir, vous pouvez copier le répertoire du programme et le coller n'importe où dans votre machine virtuelle, puis faire un double-clic sur l'application « 4Images1mot » qui se trouve dans « 4Images1Mot\4Images1Mot\bin\Debug »