4 images 1 mot

  
  
  
  
  
  
Kadri Krasniqi / Cin4B

ETML

110 heures

Chef de projet : Gilbert Gruaz

Expert 1 : Didier Viret

Expert 2 : Frédérique Andolfatto

Table des matières

[1 Spécifications 3](#_Toc481763003)

[1.1 Titre 3](#_Toc481763004)

[1.2 Description 3](#_Toc481763005)

[1.3 Matériel et logiciels à disposition 3](#_Toc481763006)

[1.4 Prérequis 3](#_Toc481763007)

[1.5 Cahier des charges 3](#_Toc481763008)

[2 Planification Initiale 4](#_Toc481763009)

[3 Analyse 7](#_Toc481763010)

[3.1 Opportunités 7](#_Toc481763011)

[3.2 Document d’analyse et conception 8](#_Toc481763012)

[3.2.1 Maquette interface graphique 8](#_Toc481763013)

[3.2.2 Analyse base de données 11](#_Toc481763014)

[3.2.2.1 MCD 11](#_Toc481763015)

[3.2.2.2 MLD 13](#_Toc481763016)

[3.2.2.3 MPD 14](#_Toc481763017)

[3.2.3 Analyse programme 15](#_Toc481763018)

[3.3 Conception des tests 16](#_Toc481763019)

[3.4 Planification détaillée 16](#_Toc481763020)

[4 Réalisation 16](#_Toc481763021)

[4.1 Dossier de Réalisation 16](#_Toc481763022)

[4.2 Modifications 16](#_Toc481763023)

[5 Tests 16](#_Toc481763024)

[5.1 Dossier des tests 16](#_Toc481763025)

[6 Conclusion 16](#_Toc481763026)

[6.1 Bilan des fonctionnalités demandées 16](#_Toc481763027)

[6.2 Bilan de la planification 16](#_Toc481763028)

[6.3 Bilan personnel 17](#_Toc481763029)

[7 Divers 17](#_Toc481763030)

[7.1 Journal de travail 17](#_Toc481763031)

[7.2 Bibliographie 17](#_Toc481763032)

[7.3 Webographie 17](#_Toc481763033)

[8 Annexes 17](#_Toc481763034)

# Spécifications

## Titre

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

## Description

Le projet 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.

## Matériel et logiciels à disposition

1 ordinateur standard ETML, avec la structure habituelle

Visual studio 2015, SQL Server 2014, NotePad++

## 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

## Cahier des charges

Le document fourni par le chef de projet fait foi. Il doit être mis en annexe 🡺 Lien sur CDC

# Planification Initiale







# Analyse

## Opportunités

Difficultés potentielles :

Ce paragraphe énumère la liste des difficultés potentielles de tout ordre :

Liste des compétences à acquérir ou approfondir

Liste du matériel à exploiter

Recherche d’informations particulières

Gestion du travail en équipe & collaboration

Ainsi que les solutions possibles

Si les spécifications de départ ne laissent pas de doutes sur la manière de réaliser un projet, ce chapitre ne fera que renvoyer le lecteur aux spécifications.

## Document d’analyse et conception

### 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 enfaite permettre aux enseignants d’ajouter un nouveau mot avec des nouvelles images.



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 à trouver le mot, il peut l’écrire dans la case « Réponse ? ».



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 clique 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 à bien sélectionné 4 images accessible au programme.

### 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 installé 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.

### 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 chaine 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 chaine 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.

### 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 créé des tables pour faire le lien. Ces nouvelles tables ont pour attributs les clés primaires des autres tables.

### 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 |
| fkmot | Int - 25 | - | Clé étrangère | Non | Oui | - |

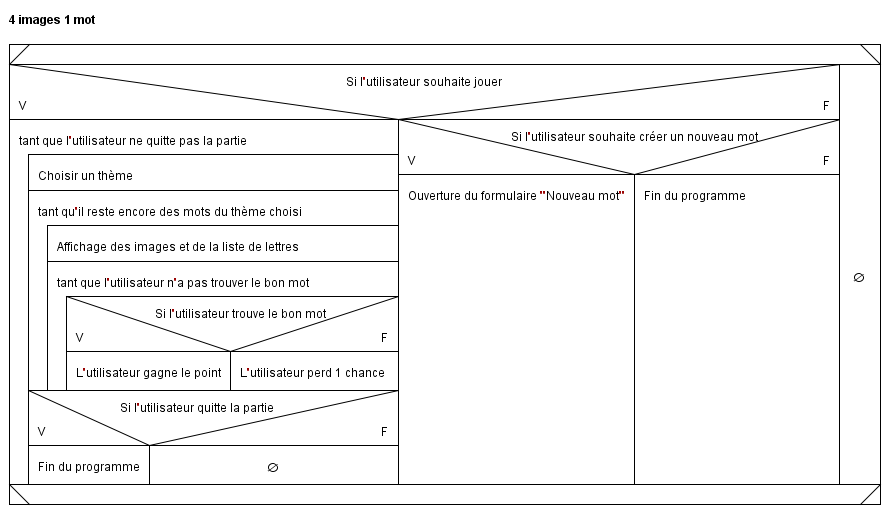
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T\_Decrire | | | | | | |
| Attribut | Type + Taille | Valeur par défaut | contrainte | Indexe | Nul interdit | Divers |
| idimag | Int - 100 | - | Clé étrangère | Non | Oui | NumAuto |
| idmot | Int - 25 | - | Clé étrangère | Non | 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 |
| fktheme | Int - 25 | - | Clé étrangère | Non | Oui | - |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 |

### 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.

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.

Ce paragraphe décrit le fonctionnement de manière détaillée.

Autant que possible de manière graphique, imagée, tableaux, etc.

Tous les cas particuliers devraient y être spécifiés…

S’il y a des fonctionnalités à développer :

Découpage en étapes, en modules, en fonctionnalités, etc.

Formulaires, interfaces graphiques, pages web, etc.

Schémas de navigation, schémas événementiels, structogramme, pseudocode, etc.

Si le projet inclut une base de données :

Dictionnaire des données

Modèle conceptuel des données, modèles logique des données.

Si le projet inclut implémentation système/réseau :

Schéma réseau, plan d’adressage, etc.

Descriptif installation, guide de mise en œuvre, etc.

## Conception des tests

Ce paragraphe permet de spécifier la stratégie de test qui sera menée au point 5.1

Qui, quand, avec quelles données, dans quel ordre, etc.

Avec quels matériels, quels services, etc.

## Planification détaillée

A ce stade, après l’analyse complète du projet, un planning détaillé et complet (avec tâches, sous-tâches, dépendances, durée, …) peut être finalisé.

Le planning détaillé doit s’inscrire dans le planning initial. Il faut que l’on puisse situer cette planification détaillée par rapport à la planification initiale.

# Réalisation

## Dossier de Réalisation

Cette partie permet de reproduire ou reprendre le projet par un tiers.

Pour chaque étape, il faut décrire sa mise en œuvre. Typiquement :

Versions des outils logiciels utilisés (OS, applications, pilotes, librairies, etc.)

Configurations spéciales des outils (Equipements, PC, machines, outillage, etc.)

Code source commenté des éléments logiciels développés.

Modèle physique d’une base de données.

Arborescences des documents produits.

Schémas, plans d’adressages, plan de nommage, etc.

Il faut décrire le parcours de réalisation et justifier les choix.

## Modifications

Historique des modifications demandées (ou nécessaires) aux spécifications détaillées.

Date, raison, description, etc.

# Tests

## Dossier des tests

On dresse le bilan des tests effectués (qui, quand, avec quelles données…) sous forme de procédure. Lorsque cela est possible, fournir un tableau des tests effectués avec les résultats attendus et obtenus, ainsi que les actions à entreprendre en conséquence (et une estimation de leur durée).

Si des tests prévus dans la stratégie n'ont pas pu être effectués :

raison, décisions, etc.

Liste des bugs répertoriés avec la date de découverte et leur état:

Corrigé, date de correction, corrigé par, etc.

# Conclusion

## Bilan des fonctionnalités demandées

Il s’agit de reprendre point par point les fonctionnalités décrites dans les spécifications de départ et de définir si elles sont atteintes ou pas, et pourquoi.

Si ce n’est pas le cas, estimer en « % » ou en « temps supplémentaire » le travail qu’il reste à accomplir pour terminer le tout.

## Bilan de la planification

Distinguer et expliquer les tâches qui ont généré des retards ou de l'avance dans la gestion du projet. Indiquer les différences entre les planifications initiales et détaillées avec le journal de travail.

## Bilan personnel

Si c’était à refaire:

Qu’est-ce qu’il faudrait garder ? Les plus et les moins ?

Qu’est-ce qu’il faudrait gérer, réaliser ou traiter différemment ?

Qu’est que ce projet m’a appris ?

Suite à donner, améliorations souhaitables, …

Remerciements, signature, etc.

# Divers

## Journal de travail

Date, activité (description qui permet de reproduire le cheminement du projet), durée, liens et références sur des documents externes. Lorsqu’une activité de recherches a été entreprise, il convient d’énumérer ce qui a été trouvé, avec les références.

## Bibliographie

Références des livres, revues et publications utilisés durant le projet.

## Webographie

Références des sites Internet consultés durant le projet.

# Annexes

Cahier des charges

Listing du code source (partiel ou, plus rarement complet)

Guide(s) d’utilisation et/ou guide de l’administrateur

Etat ou « dump » de la configuration des équipements (routeur, switch, robot, etc.).

Extraits de catalogue, documentation de fabricant, etc.