

CHAIRE INTERNATIONALE DATA SCIENCE INSTITUTE (IDSI)



Année académique : 2020-2021

PROJET DE FIN DE COURS DE LANGAGE C++

APPLICATION DE GESTION DE CHAMPIONNAT

Présentée par :

**KONGO Richmond
ODOUNHARO Ella
DAH SIE Marius**

Enseignant:

Dr. ASSALE Adjé Luc

Table des figures

1	Modèle Entité Association	5
2	Table "championnat"	6
3	Table "equipe"	6
4	Table "rencontrer"	7
5	Table "Structure du projet"	9
6	Page de d'accueil	10
7	Menu principal	11
8	Insertion d'un championnat	11
9	Menu d'un championnat	12
10	Modification d'un championnat	12
11	Supression d'un championnat	12
12	Classement d'un championnat	13
13	Calendrier d'un championnat	13
14	Equipes d'un championnat	14

Table des matières

Table des Figures	2
INTRODUCTION	3
CHAPITRE 1 : ETUDE CONCEPTUELLE	4
1.1. Identification des besoins	4
1.2. Analyse conceptuelle	5
CHAPITRE 2 : ETUDE TECHNIQUE	8
2.1. Environnement de développement	8
2.1.1. Logiciels utilisés	8
2.1.2. Organisation du projet dans codeblock	9
2.2. Présentation de la solution	10
2.2.1. Page d'accueil de l'application	10
2.2.2. Page d'accueil de l'application	10
2.2.3. Menu d'un championnat	12
2.2.4. Menu du calendrier du championnat	13
2.2.5. Menu des équipes du championnat	14
CONCLUSION	15

INTRODUCTION

Chaque année les étudiants de master 1 de l'IDSI suivent des cours de langage C++ où ils apprennent tous les concepts relatifs à la programmation orientée objet.

Afin d'asseoir nos connaissances dans ce langage et d'aller au delà de ce que notre professeur nous a enseigné, un projet de fin de cours nous a été donné et dans le cadre de ce projet il nous est demandé de réaliser une application qui gère les informations relatives à plusieurs championnats de football.

Ce document présente les études conceptuelle et technique qui ont guidé la réalisation de ce projet.

CHAPITRE 1

Etude conceptuelle

Ce chapitre présente l'identification des besoins et une analyse conceptuelle relatives au problème présenté.

1.1. Identification des besoins

La solution que nous proposons est une application qui comporte différentes fonctionnalités pour une bonne gestion des championnats. Les principales fonctions sont :

- ☞ la gestion des championnats ;
- ☞ la gestion des équipes d'un championnat ;
- ☞ la gestion du calendrier d'un championnat ;
- ☞ la gestion des résultats pour chaque journée ;
- ☞ la gestion du classement d'un championnat ;
- ☞ la gestion des résultats pour chaque équipe.

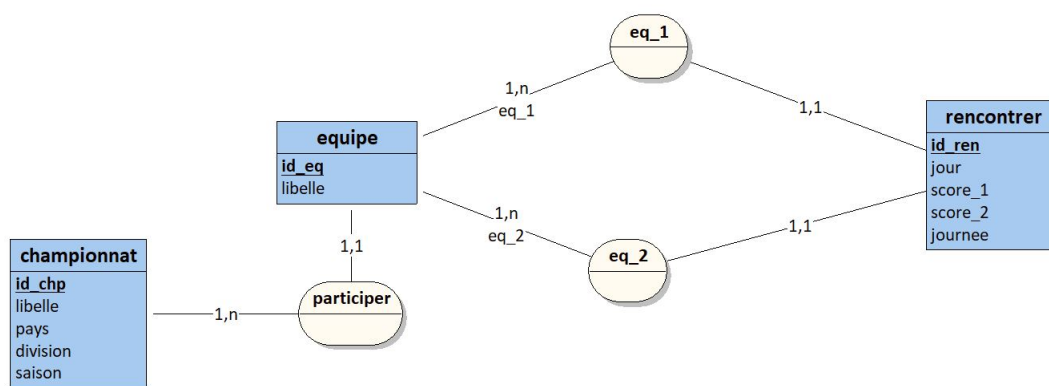
Par ailleurs, l'application doit respecter certaines contraintes énumérées ci-dessous :

- * Plusieurs championnats peuvent être gérés en même temps (de différents pays, de différentes divisions et différentes saisons). L'utilisateur peut créer un nouveau championnat, le supprimer ou modifier ces attributs.
- * Pour un championnat donné, l'utilisateur peut créer, modifier et supprimer des équipes.
- * Initialement, le programmeur crée un calendrier pour un championnat donné. Il peut toutefois le modifier au cours de la saison. Il pourra également ajouter une nouvelle journée, mettre à jour une journée ou supprimer une journée.
- * La gestion des résultats devra permettre de saisir les résultats d'une journée, d'afficher les résultats d'une journée et de mettre à jour les résultats d'une journée
- * La gestion du classement permettra d'afficher un tableau qui contiendra les nombres de points, les classements et les autres informations (matches gagnés, nuls, perdus, écart de buts) pour chaque équipe.

1.2. Analyse conceptuelle

La modélisation de notre base de données s'est axée sur la méthode MERISE. Nous avons choisis cette méthode en raison du fait que c'est elle qui a été vue cette année en cours de Base de Données 1 ; elle était par conséquent bien comprise. Après l'analyse du problème, nous avons proposé le modèle entité-association ci-dessous :

FIGURE 1 – Modèle Entité Association



Le modèle logique des données (MLD) suivant à été déduit de ce MEA.

- ☞ **championnat** : (id_chp, libelle, pays, division, saison) ;
- ☞ **equipe** : (id_eq, libelle, #id_chp) ;
- ☞ **rencontrer** : (id_ren, jour, score1, score2, journee, #eq2, #eq1)) ;

Nous aurons donc trois (03) tables dans notre base de données : championnat, equipe, et rencontrer.

La table "championnat" stockera toutes les informations de bases relatives à un championnat, à savoir le libelle, le pays, la division et la saison. La figure ci-dessous présente une vue de la table dans DBBrowser.



La table "equipe" contiendra le nom de l'équipe que nous avons nommé libelle. La figure suivante nous présente cette table dans la base de données.

FIGURE 2 – Table "championnat"

Table :  championnat   

	id_chp	libelle	pays	division	saison
	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	ligua	espagne	1	2021

FIGURE 3 – Table "equipe"

Table :  equipe 

	id_eq	libelle	id_chp
	Filtre	Filtre	Filtre
1	1	eq_A	1
2	2	eq_B	1
3	3	eq_C	1
4	4	eq_D	1

La table "rencontrer" quant à elle présentera les différents matchs à travers leurs jours (date de la rencontre), leurs journées, les scores enregistrés lors de ces matchs et les équipes qui se sont affrontées.

Cette dernière table permettra de voir le calendrier du championnat et les résultats. Remarquons que sur le tableau contenu dans la figure ci dessous plus précisément à la dernière ligne que les scores sont négatifs. Ceci est tout à fait normal car ce match n'a pas encore été joué.

C'est également à partir des données de cette table que nous obtenons le clas-

FIGURE 4 – Table "rencontrer"

Table : rencontrer							
	id_ren	jour	score_1	score_2	eq_2	eq_1	num_journee
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	19/04/2021	0	0	2	1	1
2	2	20/04/2021	5	3	1	3	1
3	3	21/04/2021	0	2	4	1	1
4	4	22/04/2021	0	1	2	3	1
5	5	23/04/2021	0	0	2	4	1
6	6	24/04/2021	-1	-1	3	4	1

sement progressif des différents championnats.

L'implémentation de la base de données a été réalisé avec SQLite 3.

CHAPITRE 2

Etude technique

Ce chapitre présente l'environnement de développement de la solution proposée au problème, les librairies utilisées et la présentation de la solution proprement dite.

2.1. Environnement de développement

2.1.1. Logiciels utilisés

Pour développer notre application, nous avons utilisé cinq (05) logiciels ci-dessous présentés.

Looping

Ce logiciel nous à servir à faire notre modélisation ; aussi bien le Modèle Entité Association (MEA) que le Modèle Logique des Données (MLD).

La version que nous avons utilisé est disponible via le lien :

<https://www.looping-mcd.fr/Looping.exe>

Codeblock

Il a été utilisé en tant que IDE pour l'écriture du programme de notre application

Il est téléchargeable via le lien : <https://sourceforge.net/projects/codeblocks/files/Binaries/20.03/Windows/codeblocks-20.03mingw-setup.exe/download>

DBBrowser

Nous l'avons utiliser pour visualiser le contenu de notre base de données SQLite.

La version que nous avons utilisé est disponible via le lien :

<https://github.com/sqlitebrowser/sqlitebrowser/releases/download/v3.12.1/DB.Browser.for.SQLite-3.12.1-win64.zip>

Mingw

Utilisé pour la génération du fichier exécutable du programme, il est disponible sur le lien suivant :

<https://sourceforge.net/projects/mingw-w64/files/latest/download>

Innosetup

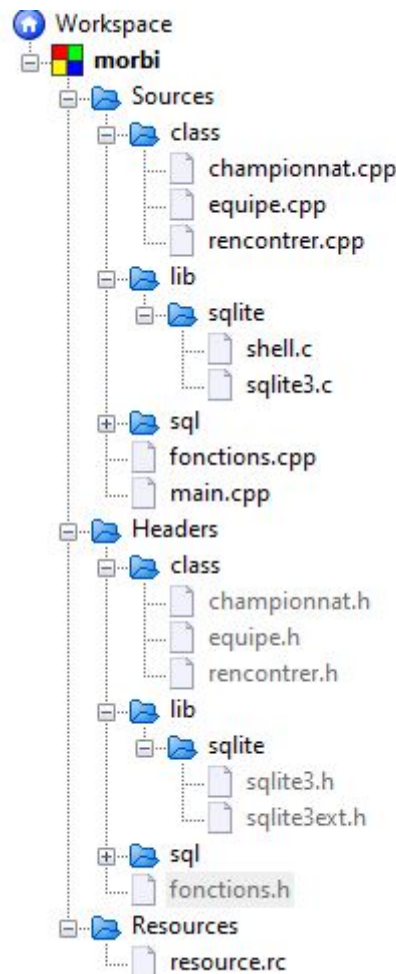
Innosetup a été utilisé pour l’empaquetage du fichier exécutable de l’application généré grâce à Mingw dans un installable. Il a été téléchargé via ce lien :

<https://jrsoftware.org/download.php/is.exe>

2.1.2. Organisation du projet dans codeblock

A ce niveau nous avons créé trois (03) dossiers nommés **lib**, **class** et **sql** contenant respectivement toutes les librairies utilisées, les classes, les fonctions en rapport avec la base données. La figure 5 présente mieux la structure du projet.

FIGURE 5 – Table "Structure du projet"



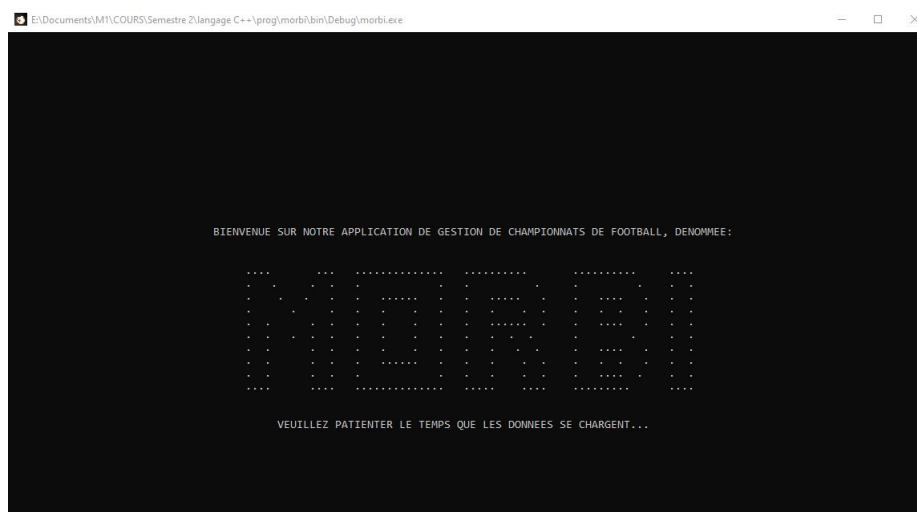
Il est à noter que nous avons travaillé avec les fichiers de type header d'extension ".h". Le fichier de type resource d'extension ".rc" contient des liens vers des fichiers ressources. Dans notre cas on a qu'un fichier ressource qui contient le lien vers l'icône de l'application.

2.2. Présentation de la solution

2.2.1. Page d'accueil de l'application

La figure ci-dessous présente la page d'accueil de l'application. Celle-ci redirige après quelques secondes vers un menu présentant la liste des championnats ainsi qu'une liste d'actions pouvant être effectuées.

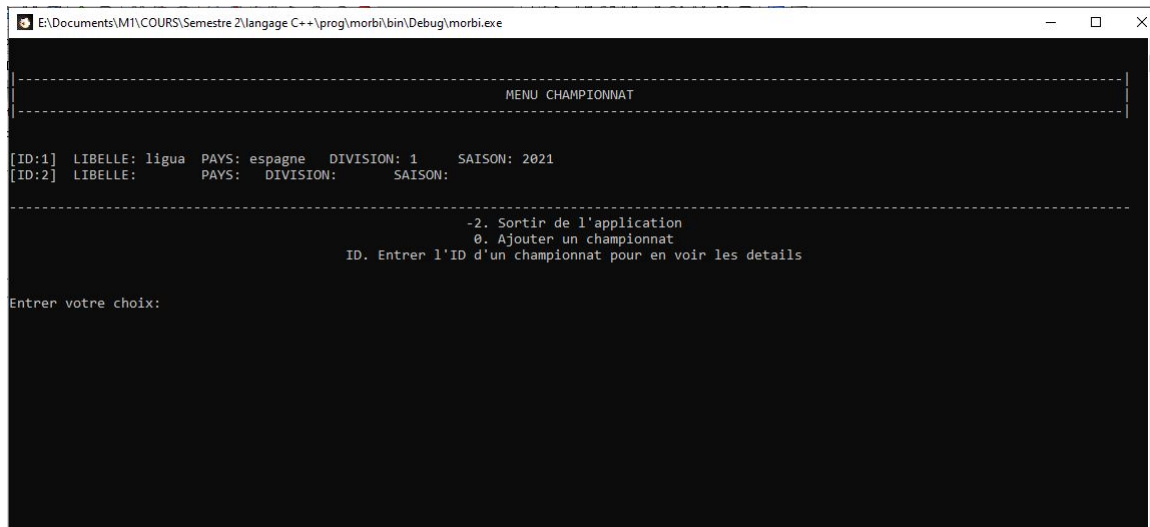
FIGURE 6 – Page de d'accueil



2.2.2. Menu principal de l'application

Comme l'illustre la figure suivante, le menu principale de l'application offre la possibilité de choisir un championnat par son id. On peut également sortir de l'application, ajouter un championnat.

FIGURE 7 – Menu principal



Si l'option de l'insertion d'un nouveau championnat est choisie (option 0), on obtient un petit formulaire pour renseigner les détails du championnat (libelle, pays, saison, division) (cf figure 8)

FIGURE 8 – Insertion d'un championnat

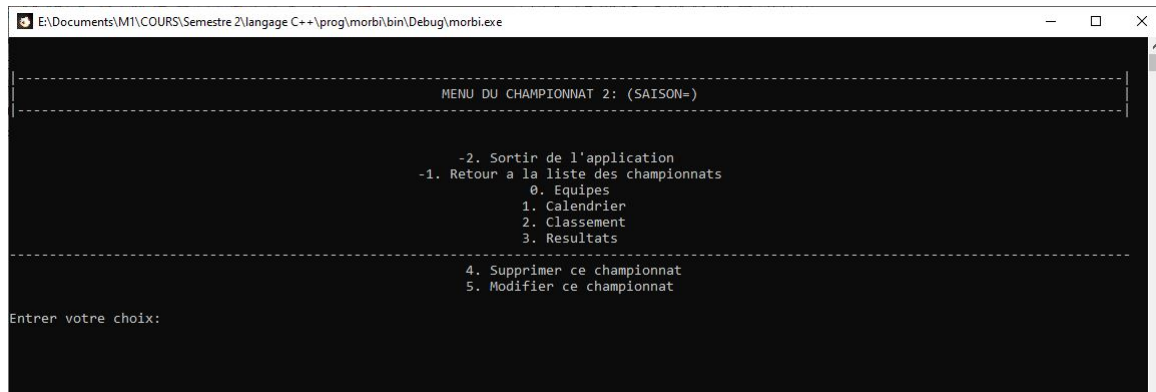


Dès que ces informations sont renseignées puis validées, le nouveau championnat est créé.

Si c'est plutôt l'id d'un championnat qui est choisie, après validation, on obtient un nouveau menu (cf figure 9).

2.2.3. Menu d'un championnat

FIGURE 9 – Menu d'un championnat



Ce menu offre la possibilité de modifier ou supprimer le championnat courant. Les figures 10 et 11 montrent les pages de suppression et de modification.

FIGURE 10 – Modification d'un championnat

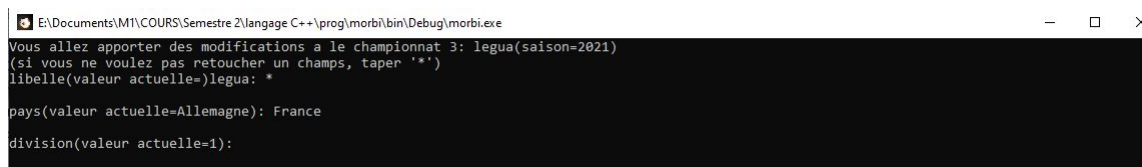
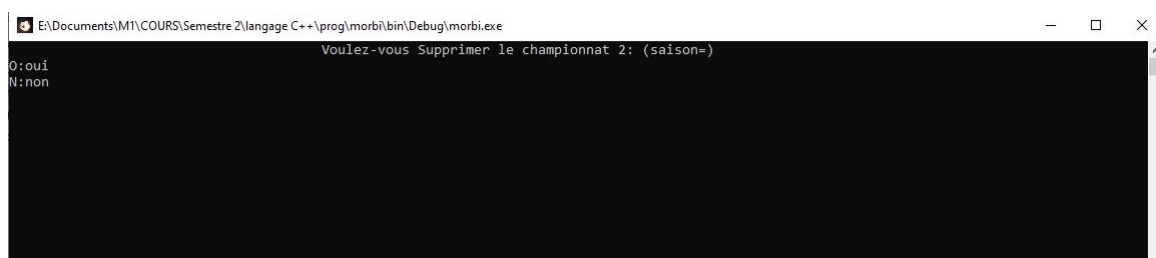
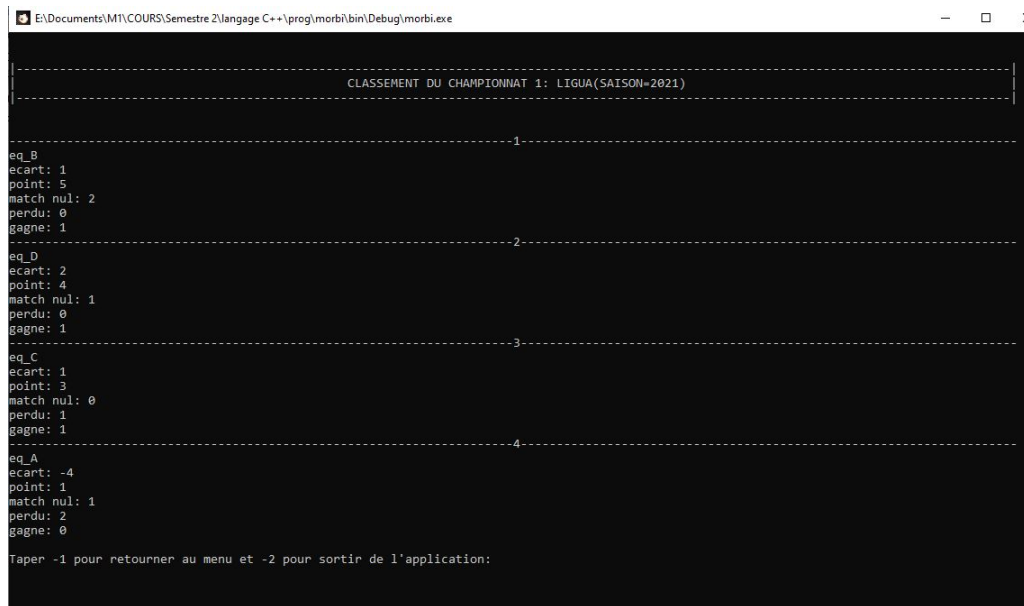


FIGURE 11 – Supression d'un championnat



L'option 2 qui est le classement présente le classement actuel des équipes du championnat comme on peut le voir sur la figure 12.

FIGURE 12 – Classement d'un championnat



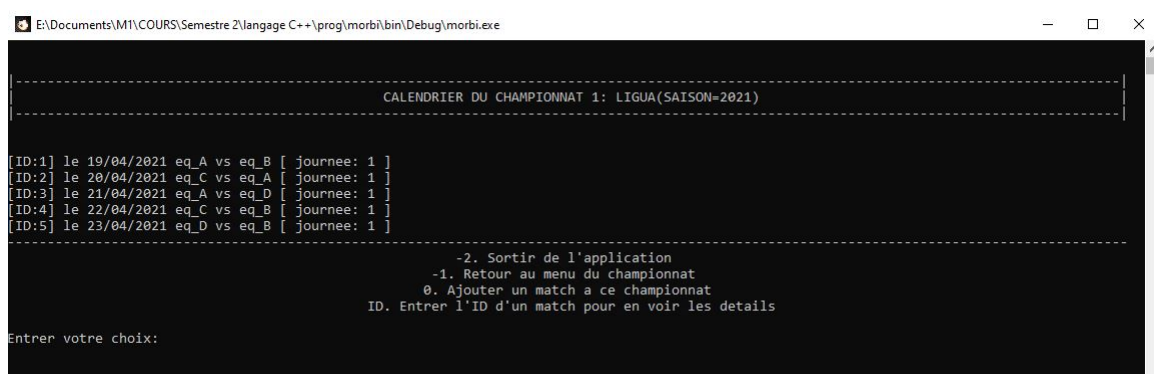
```
E:\Documents\M1\COURS\Semestre 2\langage C++\prog\morbi\bin\Debug\morbi.exe
-----
CLASSEMENT DU CHAMPIONNAT 1: LIGUA(SAISON=2021)
-----
1-----
eq_B
ecart: 1
point: 5
match nul: 2
perdu: 0
gagne: 1
-----
2-----
eq_D
ecart: 2
point: 4
match nul: 1
perdu: 0
gagne: 1
-----
3-----
eq_C
ecart: 1
point: 3
match nul: 0
perdu: 1
gagne: 1
-----
4-----
eq_A
ecart: -4
point: 1
match nul: 1
perdu: 2
gagne: 0
-----
Taper -1 pour retourner au menu et -2 pour sortir de l'application:
```

Quant aux options 0 et 3 (respectivement Equipe et calendrier), elles renvoient à de nouveaux menus

2.2.4. Menu du calendrier du championnat

Sur la figure 13, on peut voir que le menu du calendrier permet d'effectuer des actions telles que l'ajout d'un match. On y voit aussi la liste des rencontres (matches) programmées. Le choix de l'id d'un match permet d'en consulter les détails. Comme dans le cas d'un championnat, on pourra ainsi supprimer ou modifier le match dont l'id a été choisis.

FIGURE 13 – Calendrier d'un championnat

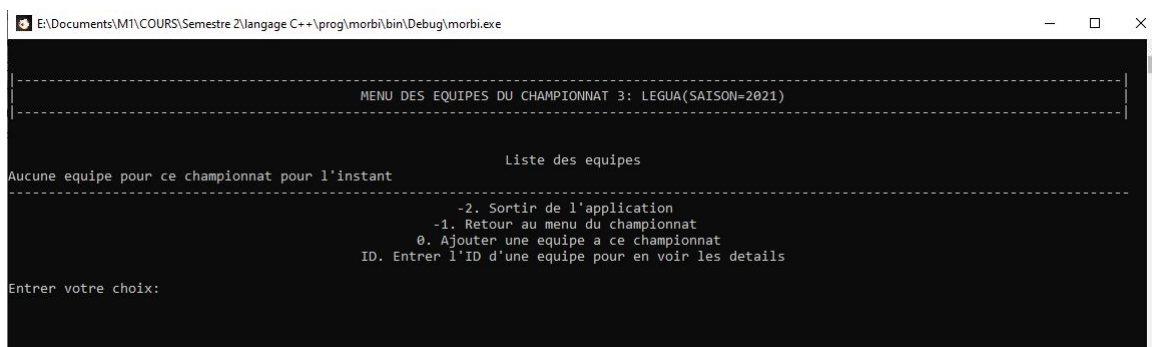


```
E:\Documents\M1\COURS\Semestre 2\langage C++\prog\morbi\bin\Debug\morbi.exe
-----
CALENDRIER DU CHAMPIONNAT 1: LIGUA(SAISON=2021)
-----
[ID:1] le 19/04/2021 eq_A vs eq_B [ journee: 1 ]
[ID:2] le 20/04/2021 eq_C vs eq_A [ journee: 1 ]
[ID:3] le 21/04/2021 eq_A vs eq_D [ journee: 1 ]
[ID:4] le 22/04/2021 eq_C vs eq_B [ journee: 1 ]
[ID:5] le 23/04/2021 eq_D vs eq_B [ journee: 1 ]
-----
-2. Sortir de l'application
-1. Retour au menu du championnat
0. Ajouter un match a ce championnat
ID. Entrer l'ID d'un match pour en voir les details
-----
Entrer votre choix:
```


2.2.5. Menu des équipes du championnat

La figure 14 présente le menu relatif aux équipes du championnat. On y dispose de la liste des équipes participant au championnat et d'une liste d'action pouvant être effectuée. Le choix de l'id d'une équipe permet d'en consulter les détails. Comme dans le cas d'un championnat et d'un match, on pourra ainsi supprimer ou modifier les détails de l'équipe dont l'id a été choisi.

FIGURE 14 – Equipes d'un championnat



```
E:\Documents\M1\COURS\Semestre 2\langage C++\prog\morbi\bin\Debug\morbi.exe
-----
MENU DES EQUIPES DU CHAMPIONNAT 3: LEGUA(SAISON=2021)
-----
Aucune equipe pour ce championnat pour l'instant
Liste des equipes
-----
-2. Sortir de l'application
-1. Retour au menu du championnat
0. Ajouter une equipe a ce championnat
ID. Entrer l'ID d'une equipe pour en voir les details
Entrer votre choix:
```

CONCLUSION

Au terme de ce projet, nous pouvons dire que l'objectif poursuivi a été atteint : réaliser une application avec le langage de programmation C++. Mais au delà de ce fait, nous avons non seulement pu asseoir nos connaissances théoriques acquises en classe mais nous avons également pu acquérir de nouvelles telles que l'intégration d'une base de données et d'une librairie dans un projet C++ pour ne citer que celles là.

Par ailleurs, notons que ce projet est potentiellement améliorable par l'ajout d'une interface graphique grâce à des logiciels spécifiques (QT designer par exemple).