

# Spécifications techniques

Application de gestion de Clubs, d'adhérents et d'évènements  
(Desktop, C#)

## Sommaire

Introduction.....	3
Base de données.....	4
Environnement de développement .....	5
Logiciels utilisés.....	5
Langages utilisés.....	5
Architecture de travail.....	6
Gestion des bugs.....	7
Gestion des versions.....	7
Tests .....	7

## **Introduction**

La mise en place de la solution applicative de gestion des Adhérents, Clubs et Événements est soumise à des contraintes et des spécifications techniques : langages utilisés, logiciels utilisés, structure de la base de données, moyens de gestions de bugs et de versions.

Ce document a pour but de lister les éléments ci-dessus et d'expliquer les différents choix techniques mis en place dans le cadre de la réalisation de la solution applicative.

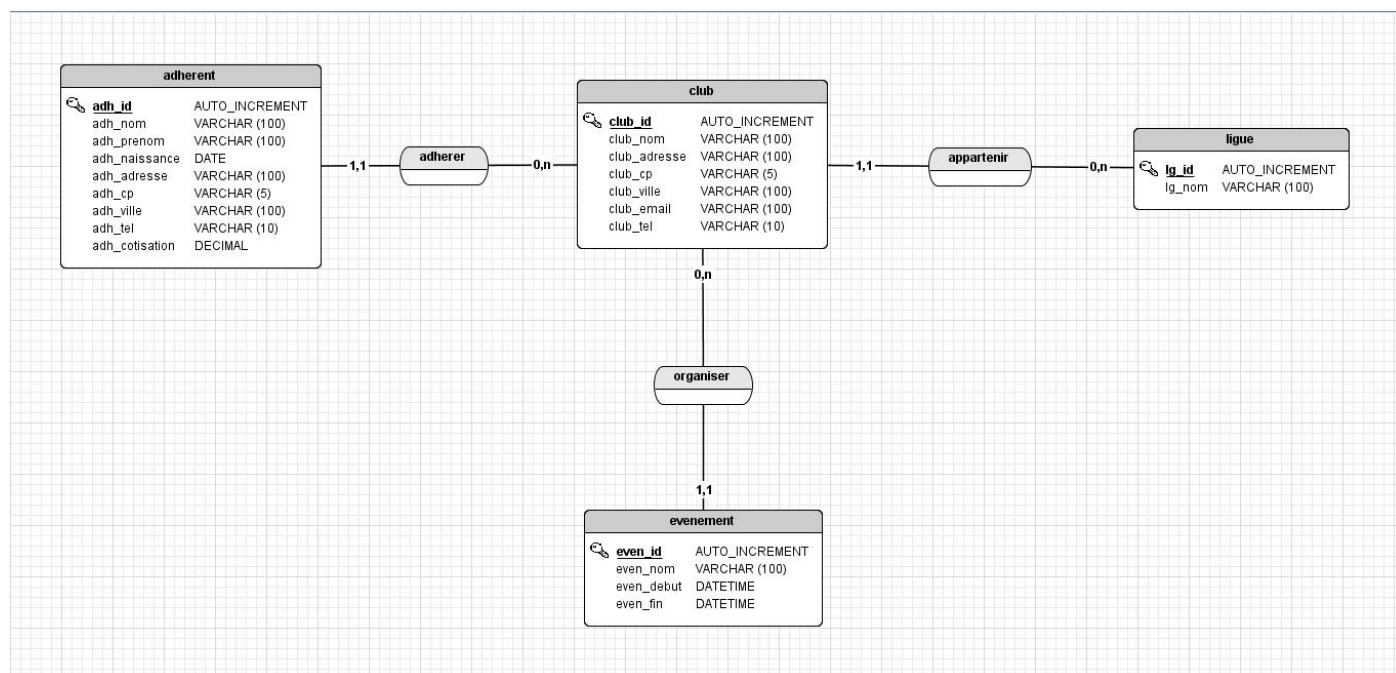
De plus, un plan de test et une liste non exhaustive des tests fonctionnels réalisés sont inclus dans ces spécifications.

## Base de données

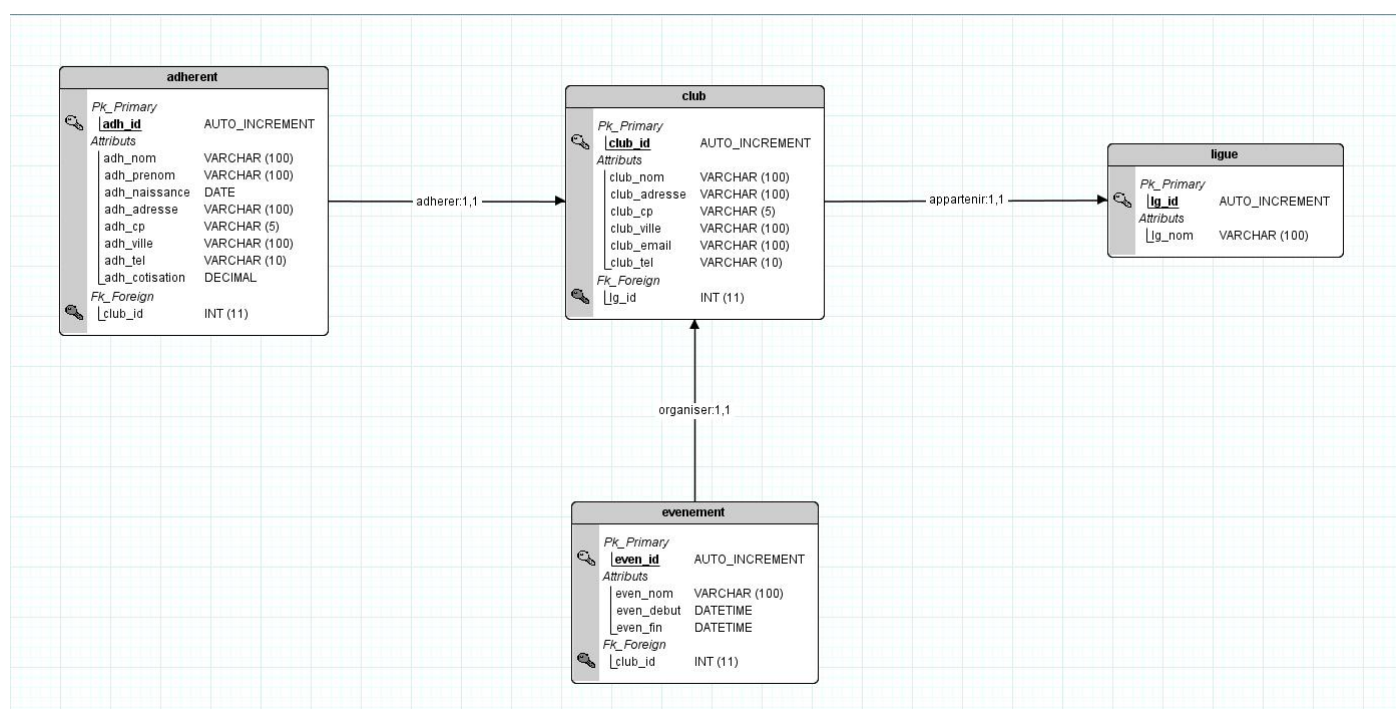
Dans cette partie sera mis en avant les choix techniques liés à la base de données.

Tout d'abord, le logiciel Jmerise a été utilisé pour réaliser le schéma entité-association qui représente la structure de la base de données. Ensuite, on génère le schéma relationnel et le script Sql.

### Schéma entité-association



### Schéma relationnel



## **Environnement de développement**

Dans cette partie, les langages utilisés ainsi que les frameworks associés, les logiciels utilisés, les outils de gestion de bugs et de versions seront explicités.

Le projet et la réalisation de la base de données sont gérés en local sur l'ordinateur du développeur. Après une phase de tests et de gestions d'incidents, le projet est placé sur le serveur de production. Toute modification apportée au projet est réalisée dans un premier temps en local.

### *Logiciels utilisés*

#### **Editeur de code : Visual Studio entreprise 2017**

Visual Studio est un éditeur gratuit développé par Microsoft, il donne la possibilité de développement des programmes dans plusieurs langages dont le C#. Visual Studio a été utilisé pour éditer et gérer le code de l'application Bureau.

#### **Modélisation : Jmerise**

JMerise est un logiciel gratuit permettant de faire de la modélisation de projet informatique grâce à la méthode Merise.

#### **Scripts SQL : MySqlWorkbench (phase de développement)**

MySqlWorkbench est un logiciel gratuit développé par Oracle, il permet la gestion et l'administration de bases de données MySQL.

### *Langages utilisés*

#### **C#**

Application graphique réalisée en C# avec Visual Studio.

#### **SQL**

Langage relatif aux bases de données, le type de base de données utilisé est MySQL.

## **Architecture de travail**

Une architecture spécifique a été utilisée dans le cadre de ce projet.

Elle est composée de différentes couches permettant de travailler de façon indépendante sur les différentes étapes du projets :

- Base de données (Model)
- Classes métier (Controller)
- Vues (Views)

## **Gestion des bugs**

La gestion des bugs est réalisée par des réunions hebdomadaires entre développeurs, les versions de test des applications étant disponibles librement. La gestion des bugs est supervisée via l'outil GitHub, avec le système d'issues.

## **Gestion des versions**

La gestion des versions de l'application bureau se fait avecGitHub.

## **Tests**

### *Objectifs*

L'intérêt de réaliser des tests est de mesurer le bon fonctionnement de notre application sur le plan technique et fonctionnel.

## Plan de tests

Nous procéderont à 2 types de tests, les tests unitaires (utilisation de MSTest) et les tests fonctionnels. Les tests concerneront la version finale de l'application en local.

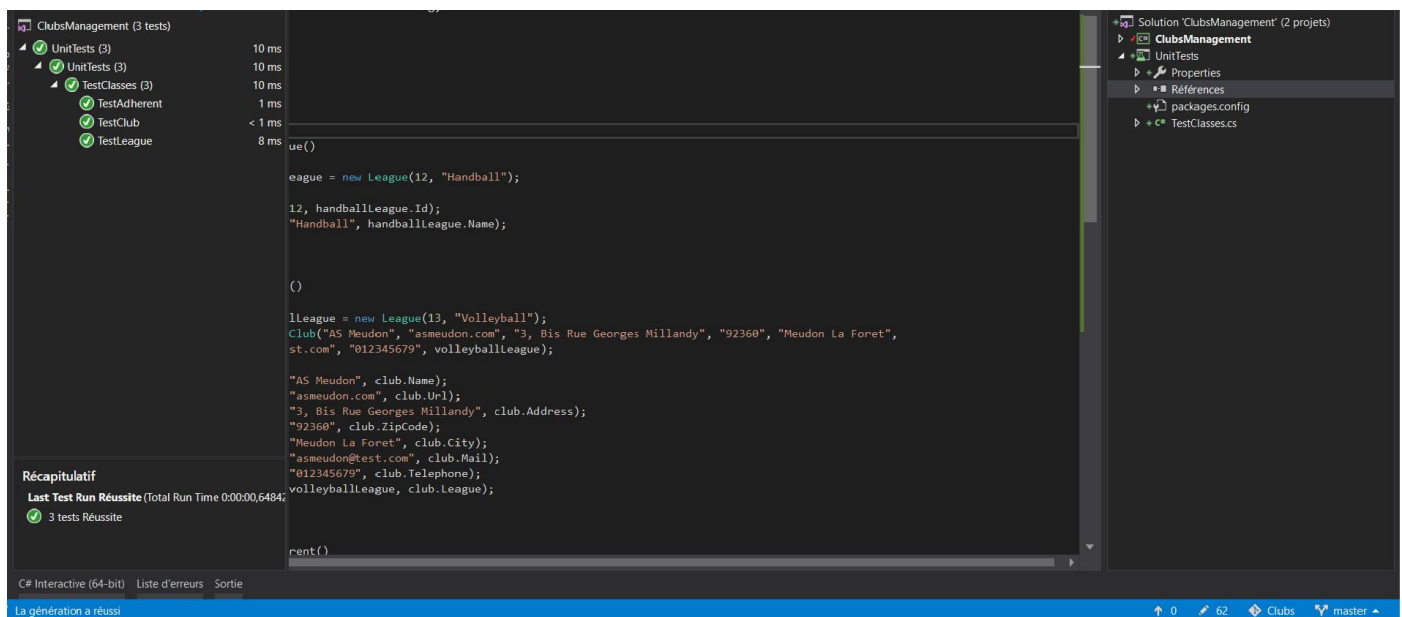
## Tests unitaires

Ils ont été réalisés via l'outil MSTest, un environnement de test a été créé et des tests unitaires ont été réalisés afin de vérifier le bon fonctionnement de l'accès aux classes, des méthodes, etc.

### Exemple de tests unitaires

```
[TestMethod]
public void TestClub()
{
    League volleyballLeague = new League(13, "Volleyball");
    Club club = new Club("AS Meudon", "3, Bis Rue Georges Millandy", "92360", "Meudon La Foret",
        "asmeudon@test.com", "012345679", volleyballLeague);

    Assert.AreEqual("AS Meudon", club.Name);
    Assert.AreEqual("3, Bis Rue Georges Millandy", club.Address);
    Assert.AreEqual("92360", club.ZipCode);
    Assert.AreEqual("Meudon La Foret", club.City);
    Assert.AreEqual("asmeudon@test.com", club.Mail);
    Assert.AreEqual("012345679", club.Telephone);
    Assert.AreEqual(volleyballLeague, club.League);
}
```



### *Tests fonctionnels*

Les tests fonctionnels sont établis et réalisés par le développeur aux moyens de fiches de tests.

Ici on testera :

- La création, la modification et la suppression d'un adhérent
- Le bon affichage de la liste des adhérents
- L'affectation d'un adhérent à un club
- L'affectation d'un adhérent à un autre club
- La création et la modification d'un club
- Le bon affichage de la liste des clubs
- La bonne affectation d'un type de club pour un club
- Le bon affichage des statistiques liées au nombre d'adhérents par club
- Le bon affichage des statistiques liées au nombre d'événements par club
- La création, la modification et la suppression d'un événement
- La bonne affectation d'un événement à un club
- Le bon affichage de la liste des événements

Voici quelques exemples de fiches de tests fonctionnels :



# SPECIFICATIONS TECHNIQUES APPLI BUREAU

## Création d'un club

Fonctionnalité à tester : Création d'un club

Version : Version finale Projet

ClubsManagement Période mai 2019 Date : 20/05/2019

Procédure de test :

- Accéder à l'application depuis le bureau
- Une fois sur l'écran d'accueil, appuyer sur le bouton club
- Une fois sur la gestion des clubs appuyer sur Ajouter
- Remplir tous les champs disponibles
- Affecter un type de club
- Appuyer sur Ajouter

Résultat(s) attendu(s) :

- Ajout d'un club dans la base de données
- Message de confirmation
- Retour à la gestion des clubs

Résultat(s) obtenu(s) :

Conclusion : Réussite du test.

# SPECIFICATIONS TECHNIQUES APPLI BUREAU

## Modification d'un adhérent

Fonctionnalité à tester : Modification d'un adhérent

Version : Version finale Projet

ClubsManagement Période mai 2019 Date : 24/05/2019

Procédure de test :

- Accéder à l'application depuis le bureau
- Une fois sur l'écran d'accueil, appuyer sur le bouton Adhérent
- Une fois sur la gestion des clubs double cliquer sur l'entête d'une des lignes de la liste des adhérents
- Modifier les champs voulus
- Appuyer sur Modifier

Résultat(s) attendu(s) :

- Modification d'un adhérent dans la base de données
- Message de confirmation de modification
- Retour à la gestion des adhérents

Résultat(s) obtenu(s) :

Conclusion : Réussite du test.