

# **Rapport de Projet**

Oussama saad / Marwane jouad/ El mehdi laaroussi

**Titre :**

Gestion des clubs de football

**Encadré par :**

MR NADIF Sami

**Soutenu le : 22/06/2023**

## **Remerciements**

Nous tenons à exprimer notre profonde gratitude envers notre cher professeur, M. NADIF, dont le dévouement sans faille a grandement contribué à la réalisation de ce projet. Nous sommes véritablement reconnaissants pour votre soutien inconditionnel, vos conseils précieux et votre présence constante. Votre passion pour l'enseignement a laissé une empreinte significative et indéniable sur notre parcours académique, en plus de favoriser notre développement personnel.

Nous souhaitons également exprimer notre sincère reconnaissance pour votre disponibilité lors des séances de consultation. Votre patience et votre clarté dans vos réponses à nos questions ont été d'une aide inestimable. Votre expertise et votre expérience ont enrichi notre apprentissage, nous permettant d'accroître notre compréhension du sujet de manière significative.

Enfin, nous aimerions adresser nos remerciements, notre respect et notre profonde gratitude à tous les intervenants qui ont contribué à ce projet. Votre implication et votre soutien ont été d'une importance cruciale et nous vous en sommes infiniment reconnaissants.

Merci du fond du cœur pour votre précieuse contribution à notre réussite académique.

## **Sommaire :**

- I. Chapitre 1 : introduction**
- II. Chapitre 2 : conception**
- III. Chapitre 3 : réalisation :**
  - section 1 : Fonctions générales.**
  - section 2 : Fonction Principale main.**
- IV. Chapitre 4 : Conclusion**

## **Chapitre 1 : introduction**

Le football est l'un des sports les plus populaires et passionnants à travers le monde. Les clubs de football, qu'ils soient amateurs ou professionnels, font face à de nombreux défis lorsqu'il s'agit de gérer leurs effectifs de joueurs. La gestion efficace des joueurs est essentielle pour assurer le bon fonctionnement du club, le développement des talents et la réussite sur le terrain. Dans cette optique, le projet de gestion de club de football vise à fournir une solution logicielle complète pour faciliter cette tâche complexe.

Le projet de gestion de club de football constitue une solution logicielle complète pour faciliter la gestion des joueurs au sein d'un club. Ce rapport présente une analyse détaillée de l'implémentation de ce programme, en mettant en évidence les différentes étapes de conception, de réalisation et les choix effectués tout au long du processus de développement.

Le chapitre 1 introduit le projet en mettant en évidence l'importance de la gestion des joueurs dans un club de football. Il présente les objectifs spécifiques du projet, qui consistent à créer un système permettant d'ajouter, de supprimer, de rechercher et d'afficher les informations des joueurs de manière efficace et pratique. Cette introduction souligne également les défis auxquels sont confrontés les responsables de club dans leur gestion quotidienne des effectifs de joueurs.

Le chapitre 2 se concentre sur la conception du système de gestion de club de football. Il décrit en détail les choix architecturaux et les modèles de conception utilisés pour structurer le programme. Des explications claires et des diagrammes sont fournis pour illustrer la modélisation des joueurs, la gestion des listes et des éléments, ainsi que les différentes fonctionnalités mises en œuvre.

Le chapitre 3 aborde la réalisation du projet, en se concentrant sur les fonctionnalités spécifiques implémentées. La section 1 présente les fonctions générales du système, telles que l'ajout, la suppression, la recherche et l'affichage des informations des joueurs. Des extraits de code significatifs sont fournis pour illustrer la mise en œuvre de ces fonctionnalités. Dans la section 2, la fonction principale "main" est détaillée, montrant comment les différentes fonctionnalités sont utilisées dans un contexte opérationnel.

Enfin, le chapitre 4 conclut le rapport en résumant les réalisations du projet. Il met en évidence les points forts du système de gestion de club de football, les défis rencontrés lors du développement et les améliorations possibles pour les futures itérations. Une réflexion sur les leçons apprises tout au long du processus de développement est également présentée.

Ce rapport offre ainsi une vue d'ensemble complète du projet de gestion de club de football, en mettant en avant les différentes étapes de conception et de réalisation du système. Il fournit une base solide pour comprendre les choix de conception effectués et les fonctionnalités implémentées, tout en proposant des pistes d'amélioration pour le futur développement du programme.

## **Chapitre 2 : conception :**

### **2.1 Diagramme de Structure**

Dans cette section, nous présentons le diagramme de structure utilisés pour modéliser le programme de gestion de club de football. Un diagramme de classe est utilisé pour représenter la structure de données des joueurs et des éléments de liste. Un diagramme de flux est également inclus pour illustrer le flux d'exécution des opérations principales telles que l'ajout, la suppression et la recherche de joueurs.

## Programme

```
|  
|-- Liste  
|   |-- Tete: Element  
|   |-- Fin: Element  
|   |-- Taille: Entier  
|  
|-- Element  
|   |-- Joueur  
|       |-- Nom: Chaine de caractère  
|       |-- Prenom: Chaine de caractère  
|       |-- DateNaissance: Chaine de caractère  
|       |-- Poste: Chaine de caractère  
|       |-- Salaire: Entier  
|       |-- ValeurMarchande: Entier  
|       |-- Transfert: Entier  
|       |-- Annee: Entier
```

## 2.2 Fonctionnalités Générales

Le programme de gestion de club de football offre les fonctionnalités suivantes :

Ajouter un joueur : Permet à l'utilisateur de saisir les informations d'un joueur et de l'ajouter à la liste.

Supprimer un joueur : Permet à l'utilisateur de supprimer un joueur de la liste en utilisant son nom.

Rechercher un joueur : Permet à l'utilisateur de rechercher un joueur dans la liste en utilisant son nom.

Afficher la liste des joueurs : Affiche toutes les informations des joueurs présents dans la liste.

Calculer la moyenne des salaires : Calcule et affiche la moyenne des salaires de tous les joueurs.

Vérifier la disponibilité des joueurs pour un transfert : Affiche les joueurs disponibles pour un transfert en indiquant leur statut de transfert.

## **Chapitre 3 : Réalisation**

### **3.1 Fonctions Générales**

Dans cette section, nous décrivons les principales fonctions implémentées dans le code. Cela inclut les fonctions pour l'ajout, la suppression, la recherche et l'affichage des joueurs, ainsi que la fonction pour calculer la moyenne des salaires et la fonction pour vérifier la disponibilité des joueurs pour un transfert.

#### **Appel des bibliothèques:**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

**Ajout d'un joueur :** L'utilisateur peut ajouter un nouveau joueur en fournissant les informations nécessaires telles que le nom, le prénom, la date de naissance, le poste, le salaire, la valeur marchande et le statut de transfert du joueur. Les joueurs sont ajoutés au début de la liste des joueurs.

```
void ajouter_debut(Liste *L) {
    if (L->Taille == 100) {
        printf("Liste des joueurs pleine\n");
        return;
    }

    Element *E = malloc(sizeof(Element));
    printf("Donner le nom de joueur: ");
    scanf("%s", E->joueur.Nom);
    printf("Donner le prenom de joueur: ");
    scanf("%s", E->joueur.Prenom);
    printf("Donner la date de naissance du joueur: ");
    scanf("%s", E->joueur.date_naissance);
    printf("Poste de joueur: ");
    scanf("%s", E->joueur.poste);
    printf("Saisir le salaire de joueur: ");
    scanf("%d", &E->joueur.salaire);
    printf("Saisir la valeur marchande du joueur: ");
    scanf("%d", &E->joueur.valeur_marchande);
    printf("Saisir le transfert de joueur: ");
    scanf("%d", &E->joueur.transfert);

    if (L->Tete == NULL) {
        L->Tete = L->Fin = E;
        E->suiv = E->prec = NULL;
    } else {
        E->suiv = L->Tete;
        E->prec = NULL;
        L->Tete->prec = E;
        L->Tete = E;
    }

    L->Taille++;
}
```

**Suppression d'un joueur :** L'utilisateur peut supprimer un joueur en spécifiant son nom. Le joueur est recherché dans la liste des joueurs et supprimé de manière appropriée, en mettant à jour les références de la liste.



```
void supp_joueur(Liste *L) {
    char Nom[20];
    int cpt = 1;
    Element *tmp = L->Tete;
    if (tmp == NULL) {
        printf("Liste de joueurs vide\n");
    } else {
        printf("\nDonner le nom du joueur a supprimer: ");
        scanf("%s", Nom);
        while (tmp != NULL) {
            if (strcmp(Nom, tmp->joueur.Nom) == 0) {
                if (cpt == 1)
                    supp_debut(L);
                else if (cpt == L->Taille)
                    supp_fin(L);
                else
                    supp_milieu(L, cpt);
                break;
            }
            tmp = tmp->suiv;
            cpt++;
        }
    }
}
```

**Recherche d'un joueur :** L'utilisateur peut rechercher un joueur en spécifiant son nom. Le programme parcourt la liste des joueurs et affiche les détails du joueur correspondant.

```
char *chercher_joueur(Liste L) {
    char Nom[20];
    Element *tmp = L.Tete;
    if (tmp == NULL) {
        printf("Liste de joueurs vide\n");
    } else {
        printf("\nDonner le nom du joueur a chercher: ");
        scanf("%s", Nom);
        while (tmp != NULL) {
            if (strcmp(Nom, tmp->joueur.Nom) == 0) {
                return tmp->joueur.salaire;
            }
            tmp = tmp->suiv;
        }
    }
    return NULL;
}
```

**Affichage des joueurs :** L'utilisateur peut afficher tous les joueurs du club, avec leurs informations détaillées.

```
void afficher_joueur(Joueur j) {
    printf("Nom de joueur: %s\n", j.Nom);
    printf("Prenom de joueur: %s\n", j.Prenom);
    printf("Date de naissance de joueur: %s\n", j.date_naissance);
    printf("Poste de joueur: %s\n", j.poste);
    printf("Salaire de joueur: %d\n", j.salaire);
    printf("Valeur marchande de joueur: %d\n", j.valeur_marchande);
    printf("Transfert: %d\n", j.transfert);
}
```

**Calcul de la moyenne des salaires :** Le programme calcule et affiche la moyenne des salaires des joueurs du club.

```
//fonction du moyenne de salaire des joueurs
float calculer_moyenne_salaire(Liste L) {
    if (L.Tete == NULL) {
        printf("Liste de joueurs vide\n");
        return 0.0;
    }

    Element* tmp = L.Tete;
    int somme_salaires = 0;
    int nombre_joueurs = 0;

    while (tmp != NULL) {
        somme_salaires += tmp->joueur.salaire;
        nombre_joueurs++;
        tmp = tmp->suiv;
    }

    if (nombre_joueurs > 0) {
        float moyenne = (float)somme_salaires / nombre_joueurs;
        return moyenne;
    } else {
        printf("La liste de joueurs est vide.\n");
        return 0.0;
    }
}
```

**Affichage des joueurs en transfert :** Le programme affiche les joueurs marqués comme étant en transfert, avec leurs informations complètes.

```

void afficher_joueurs_transfert(Liste L) {
    Element* tmp = L.Tete;
    int c = 0;

    printf("Joueurs en transfert:\n");
    while (tmp != NULL) {
        if (tmp->joueur.transfert) {
            printf("Joueur %d:\n", ++c);
            afficher_joueur(tmp->joueur);
            printf("\n");
        }
        tmp = tmp->suiv;
    }

    if (c == 0) {
        printf("Aucun joueur en transfert.\n");
    }
}

```

### 3.2 Fonction Principale "main"

La fonction principale "main" gère le flux global du programme. Elle présente un menu interactif qui permet à l'utilisateur de choisir parmi les différentes fonctionnalités. La boucle principale permet à l'utilisateur d'exécuter plusieurs opérations tant qu'il n'a pas choisi de quitter.

```

int main() {
    Liste L;
    L.Tete = L.Fin = NULL;
    L.Taille = 0;

    int choix;
    do {
        system("cls");
        printf("// GESTION DE CLUB//\n\n");
        printf("1. Acheter un joueur\n");
        printf("2. Vendre un joueur\n");
        printf("3. Chercher un joueur\n");
        printf("4. Afficher liste des joueurs\n");
        printf("5. Moyenne des salaires\n");
        printf("6. disponibilite de transfert\n");
        printf("7. Quitter\n\n");
        printf("Donner votre choix : ");
        scanf("%d", &choix);

        switch (choix) {
            case 1:
                ajouter_debut(&L);
                break;
            case 2:
                supp_joueur(&L);

```

```
case 2:
    supp_joueur(&L);
    // Code pour vendre un joueur
    break;
case 3:
    chercher_joueur(L);
    // Code pour rechercher un joueur
    break;
case 4:
    afficher_LISTE(L);
    break;
case 5:
    calculer_moyenne_salaire(L);
    break;
case 6:
    afficher_joueurs_transfert(L);
    break;
case 7:
    printf("Au revoir!\n");
    break;
default:
    printf("Choix invalide. Veuillez reessayer.\n");
    break;
}
```

```
    printf("\n");
    system("pause");
} while (choix != 7);

return 0;
}
```

## **Chapitre 4 : Conclusion**

Le projet de gestion de club de football a été réalisé avec succès. Le programme offre des fonctionnalités complètes pour gérer les joueurs d'un club, y compris l'ajout, la suppression, la recherche et l'affichage des informations. Il fournit également des fonctionnalités supplémentaires telles que le calcul de la moyenne des salaires et la vérification de la disponibilité des joueurs pour un transfert. L'implémentation a été réalisée en suivant une approche structurée et en utilisant des structures de données appropriées. Des améliorations futures peuvent inclure l'ajout de fonctionnalités avancées telles que la gestion des équipes et des matchs.

En conclusion, ce projet a permis de développer un programme fonctionnel et utile pour la gestion de club de football. Il démontre également l'application des concepts de programmation et de la conception de logiciels dans un contexte réel