# Rapport Technique :

# Système de Gestion de la Réputation en Ligne

**Auteur :**

**Thierno Idrissa Diallo**

**Edit : Décembre 2024**

1- Introduction

2- Function

3- Execution

# Rapport Technique : Système de Gestion de la Réputation en Ligne

## Introduction :

Ce projet vise à créer un système pour évaluer la réputation de produits en fonction des avis donnés par des utilisateurs. Ce système repose sur des structures de données dynamiques pour modéliser utilisateurs et produits, et sur des algorithmes permettant de calculer la réputation des produits en fonction des avis positifs et négatifs.

Ce document détaille l'architecture logicielle, les différentes fonctionnalités et leur implémentation. Il inclut des instructions pour la compilation et l'exécution, ainsi que des captures d'écran des tests.

### Structure du Projet

Pour une meilleure organisation, le projet est structuré de la manière suivante :

```
project_root/
online-reputation-management-system/
├── src
│   ├── main.c          # Point d'entrée de l'application
│   ├── reputation.c    # Fonctions pour la gestion de la réputation
│   └── utilisateur.c   # Fonctions pour la gestion des utilisateurs
│   └── produit.c       # Fonctions pour la gestion des produits
├── include
│   ├── reputation.h     # En-tête pour les fonctions de réputation
│   └── utilisateur.h    # En-tête pour les fonctions d'utilisateur
│   └── produit.h        # En-tête pour les fonctions de produit
├── Makefile            # Instructions de compilation
├── ressource            # logiciel nécessaire
```

```
└──── README.md              # Documentation du projet
```

## Fonctionnalités Principales

### 1. Gestion des Produits

Les produits sont modélisés comme des éléments de liste chaînée (Graphe) avec des attributs tels que :

- **id** : Identifiant unique du produit.
- **nom** : Nom du produit.
- **score_reputation** : Score calculé basé sur les avis.

```c
#ifndef PRODUIT_H
#define PRODUIT_H
/**
 * Structure de données pour un produit
 */
typedef struct Produit {
    int id;
    char nom[50];
    int score_reputation;
    struct Produit *suivant;
} Produit;

/**
 * Fonction pour créer un nouveau produit
 * @param nom Nom du produit
 * @return ID du produit créé ou -1 en cas d'erreur
 */
int creer_produit(const char *nom);

/**
 * Fonction pour obtenir un produit à partir de son ID
 * @param id ID du produit
 * @return Pointeur vers le produit ou NULL si l'ID est invalide
 */
Produit* obtenir_produit(int id);

/**
 * Fonction pour imprimer les informations d'un produit
 * @param id ID du produit
 */
```

```c
void imprimer_info_produit(int id);

/**
 * Fonction pour imprimer tous les produits enregistrés
 */
void imprimer_tous_les_produits();

#endif // PRODUIT_H
```

## Fonctions Implémentées

```c
#include "produit.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

#define MAX_PRODUITS 100

static Produit *tete_produits = NULL;
static int nombre_produits = 0;
/**
 * Fonction pour créer un nouveau produit
 * @param nom Nom du produit
 * @return ID du produit créé ou -1 en cas d'erreur
 */
int creer_produit(const char *nom) {
    if (nombre_produits >= MAX_PRODUITS) {
        return -1; // Limite de produits atteinte
    }
    Produit *nouveau_produit = (Produit *)malloc(sizeof(Produit));
    if (!nouveau_produit) {
        return -1; // Erreur d'allocation mémoire
    }
    nouveau_produit->id = nombre_produits + 1;
    strncpy(nouveau_produit->nom, nom, sizeof(nouveau_produit->nom) - 1);
    nouveau_produit->score_reputation = 0;
    nouveau_produit->suivant = tete_produits;
    tete_produits = nouveau_produit;
    nombre_produits++;
    return nouveau_produit->id; // Retourne l'ID du nouveau produit
}

/***
 * Fonction pour obtenir un produit à partir de son ID
 * @param id ID du produit
 * @return Pointeur vers le produit ou NULL si l'ID est invalide
 */
Produit* obtenir_produit(int id) {
```

```c
    Produit *courant = tete_produits;
    while (courant) {
        if (courant->id == id) {
            return courant;
        }
        courant = courant->suivant;
    }
    return NULL; // ID produit invalide
}

/**
 * Fonction pour imprimer les informations d'un produit
 * @param id ID du produit
 */
void imprimer_info_produit(int id) {
    Produit *produit = obtenir_produit(id);
    if (produit) {
        printf("ID Produit: %d\n", produit->id);
        printf("Nom: %s\n", produit->nom);
        printf("Score de Reputation: %d\n", produit->score_reputation);
    } else {
        printf("Produit non trouve.\n");
    }
}

/**
 * Fonction pour imprimer tous les produits enregistrés
 */
void imprimer_tous_les_produits() {
    if (nombre_produits == 0) {
        printf("Aucun produit enregistre.\n");
        return;
    }
    Produit *courant = tete_produits;
    while (courant) {
        printf("ID Produit: %d\n", courant->id);
        printf("Nom: %s\n", courant->nom);
        printf("Score de Reputation: %d\n", courant->score_reputation);
        printf("--------------------------\n");
        courant = courant->suivant;
    }
}
```

## 2. Gestion des Utilisateurs

Les utilisateurs sont modélisés avec les attributs suivants :

- **id** : Identifiant unique de l'utilisateur.
- **nom** : Nom de l'utilisateur.
- **email** : Adresse email.

```c
#ifndef UTILISATEUR_H
#define UTILISATEUR_H
/**
 * Structure de données pour un utilisateur
 */
typedef struct Utilisateur {
    int id;
    char nom[50];
    char email[100];
    struct Utilisateur *suivant;
} Utilisateur;

typedef struct Utilisateur Utilisateur;

/**
 * Fonction pour créer un nouvel utilisateur
 * @param nom Nom de l'utilisateur
 * @param email Email de l'utilisateur
 * @return ID de l'utilisateur créé ou -1 en cas d'erreur
 */
int creer_utilisateur(const char *nom, const char *email);

/***
 * Fonction pour obtenir un utilisateur à partir de son ID
 * @param id ID de l'utilisateur
 * @return Pointeur vers l'utilisateur ou NULL si l'ID est invalide
 */
Utilisateur* obtenir_utilisateur(int id);

/**
 * Fonction pour imprimer les informations d'un utilisateur
 * @param id ID de l'utilisateur
 */
void imprimer_info_utilisateur(int id);

/**
 * Fonction pour imprimer tous les utilisateurs enregistrés
 */
void imprimer_tous_les_utilisateurs();

#endif // UTILISATEUR_H
```

## Fonctions Implémentées

```c
#include "utilisateur.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

#define MAX_UTILISATEURS 100


static Utilisateur *tete_utilisateurs = NULL;
static int nombre_utilisateurs = 0;

/**
 * Fonction pour créer un nouvel utilisateur
 * @param nom Nom de l'utilisateur
 * @param email Email de l'utilisateur
 * @return ID de l'utilisateur créé ou -1 en cas d'erreur
 */
int creer_utilisateur(const char *nom, const char *email) {
    if (nombre_utilisateurs >= MAX_UTILISATEURS) {
        return -1; // Limite d'utilisateurs atteinte
    }
    Utilisateur *nouveau_utilisateur = (Utilisateur
*)malloc(sizeof(Utilisateur));
    if (!nouveau_utilisateur) {
        return -1; // Erreur d'allocation mémoire
    }
    nouveau_utilisateur->id = nombre_utilisateurs + 1;
    strncpy(nouveau_utilisateur->nom, nom, sizeof(nouveau_utilisateur->nom) -
1);
    strncpy(nouveau_utilisateur->email, email, sizeof(nouveau_utilisateur-
>email) - 1);
    nouveau_utilisateur->suivant = tete_utilisateurs;
    tete_utilisateurs = nouveau_utilisateur;
    nombre_utilisateurs++;
    return nouveau_utilisateur->id; // Retourne l'ID du nouvel utilisateur
}

/***
 * Fonction pour obtenir un utilisateur à partir de son ID
 * @param id ID de l'utilisateur
 * @return Pointeur vers l'utilisateur ou NULL si l'ID est invalide
 */
Utilisateur* obtenir_utilisateur(int id) {
    Utilisateur *courant = tete_utilisateurs;
    while (courant) {
        if (courant->id == id) {
```

```
            return courant;
        }
        courant = courant->suivant;
    }
    return NULL; // ID utilisateur invalide
}

/**
 * Fonction pour imprimer les informations d'un utilisateur
 * @param id ID de l'utilisateur
 */
void imprimer_info_utilisateur(int id) {
    Utilisateur *utilisateur = obtenir_utilisateur(id);
    if (utilisateur) {
        printf("ID Utilisateur: %d\n", utilisateur->id);
        printf("Nom: %s\n", utilisateur->nom);
        printf("Email: %s\n", utilisateur->email);
    } else {
        printf("Utilisateur non trouve.\n");
    }
}

/**
 * Fonction pour imprimer les informations de tous les utilisateurs
 */
void imprimer_tous_les_utilisateurs() {
    if (nombre_utilisateurs == 0) {
        printf("Aucun utilisateur enregistre.\n");
        return;
    }
    Utilisateur *courant = tete_utilisateurs;
    while (courant) {
        printf("ID Utilisateur: %d\n", courant->id);
        printf("Nom: %s\n", courant->nom);
        printf("Email: %s\n", courant->email);
        printf("--------------------------\n");
        courant = courant->suivant;
    }
}
```

### 3. Gestion de la Réputation

Un algorithme simple permet de calculer un score de réputation basé sur les avis positifs et négatifs donnés à un produit.

#### Fonctions Implémentées

```c
#include "reputation.h"
#include <stdio.h>

/**
 * Fonction pour calculer le score de réputation d'un produit
 * @param avis_positifs Nombre d'avis positifs
 * @param avis_negatifs Nombre d'avis négatifs
 * @return Score de réputation calculé
 */
int calculer_score_reputation(int avis_positifs, int avis_negatifs) {
    if (avis_positifs + avis_negatifs == 0) {
        return 0; // Aucun avis signifie une réputation neutre
    }
    return (avis_positifs * 100) / (avis_positifs + avis_negatifs);
}

/**
 * Fonction pour ajouter un avis sur un produit
 * @param utilisateur Utilisateur donnant l'avis
 * @param produit Produit sur lequel l'avis est donné
 * @param avis_positifs Nombre d'avis positifs
 * @param avis_negatifs Nombre d'avis négatifs
 */
void ajouter_avis(Utilisateur *utilisateur, Produit *produit, int
avis_positifs, int avis_negatifs) {
    produit->score_reputation = calculer_score_reputation(avis_positifs,
avis_negatifs);
    printf("Score de reputation mis a jour pour le produit %s: %d\n", produit-
>nom, produit->score_reputation);
}
```

## Instructions de Compilation et d'Exécution

### Prérequis

- **Compilateur** : GCC ou tout autre compilateur C compatible.
- **IDE** : Visual Studio Code, CLion ou tout éditeur de votre choix.

### Compilation avec Makefile

1. Naviguez dans le répertoire racine du projet :
   cd online-reputation-management-system

2. Compilez le projet :
   mingw32-make

```
PS D:\Algo3\graphe\Projet_C\online-reputation-management-system> mingw32-make
gcc -o main src/main.o src/reputation.o src/utilisateur.o src/produit.o
PS D:\Algo3\graphe\Projet_C\online-reputation-management-system>
```

3. Exécutez l'application :
   ./main

```
PS D:\Algo3\graphe\Projet_C\online-reputation-management-system> ./main

        ========================================
        ||           MENU PRINCIPAL           ||
        ========================================
        ||  1. Creer un utilisateur           ||
        ||  2. Creer un produit               ||
        ||  3. Ajouter un avis sur un produit ||
        ||  4. Afficher infos utilisateur     ||
        ||  5. Afficher infos produit         ||
        ||  6. Afficher tous les utilisateurs ||
        ||  7. Afficher tous les produits     ||
        ||  8. Quitter                        ||
        ========================================
        >> Veuillez entrer votre choix :
        Entrez votre choix:
```

**Teste :**

```
PS D:\Algo3\graphe\Projet_C\online-reputation-management-system> ./main

        ========================================
        ||           MENU PRINCIPAL           ||
        ========================================
        ||  1. Creer un utilisateur           ||
        ||  2. Creer un produit               ||
        ||  3. Ajouter un avis sur un produit ||
        ||  4. Afficher infos utilisateur     ||
        ||  5. Afficher infos produit         ||
        ||  6. Afficher tous les utilisateurs ||
        ||  7. Afficher tous les produits     ||
        ||  8. Quitter                        ||
        ========================================
        >> Veuillez entrer votre choix :
        Entrez votre choix: 1
Entrez le nom de l'utilisateur: diallo
Entrez l'email de l'utilisateur: diallo@gmail.com
Utilisateur cree avec l'ID: 1
```

```
          ========================================
          ||           MENU PRINCIPAL           ||
          ========================================
          ||  1. Creer un utilisateur           ||
          ||  2. Creer un produit               ||
          ||  3. Ajouter un avis sur un produit ||
          ||  4. Afficher infos utilisateur     ||
          ||  5. Afficher infos produit         ||
          ||  6. Afficher tous les utilisateurs ||
          ||  7. Afficher tous les produits     ||
          ||  8. Quitter                        ||
          ========================================
          >> Veuillez entrer votre choix :
          Entrez votre choix: 2
 Entrez le nom du produit: Ordi_HP
 Produit cree avec l'ID: 1
```

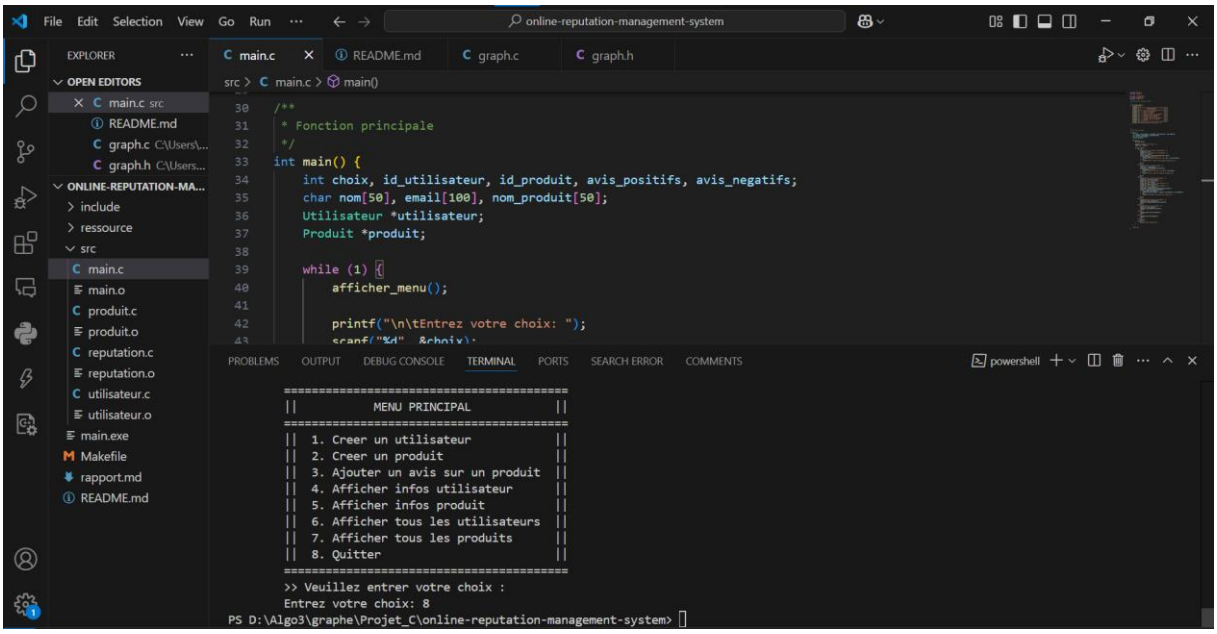
```
        ==========================================
        ||            MENU PRINCIPAL             ||
        ==========================================
        ||  1. Creer un utilisateur              ||
        ||  2. Creer un produit                  ||
        ||  3. Ajouter un avis sur un produit    ||
        ||  4. Afficher infos utilisateur        ||
        ||  5. Afficher infos produit            ||
        ||  6. Afficher tous les utilisateurs    ||
        ||  7. Afficher tous les produits        ||
        ||  8. Quitter                           ||
        ==========================================
        >> Veuillez entrer votre choix :
        Entrez votre choix: 3
Entrez l'ID de l'utilisateur: 1
Entrez l'ID du produit: 1
Entrez le nombre d'avis positifs: 100
Entrez le nombre d'avis negatifs: 50
Score de reputation mis a jour pour le produit Ordi_HP: 66
```

```
        ==========================================
        ||           MENU PRINCIPAL             ||
        ==========================================
        ||  1. Creer un utilisateur             ||
        ||  2. Creer un produit                 ||
        ||  3. Ajouter un avis sur un produit   ||
        ||  4. Afficher infos utilisateur       ||
        ||  5. Afficher infos produit           ||
        ||  6. Afficher tous les utilisateurs   ||
        ||  7. Afficher tous les produits       ||
        ||  8. Quitter                          ||
        ==========================================
        >> Veuillez entrer votre choix :
        Entrez votre choix: 4
Entrez l'ID de l'utilisateur: 1
ID Utilisateur: 1
Nom: diallo
Email: diallo@gmail.com
```

```
        ==========================================
        ||           MENU PRINCIPAL             ||
        ==========================================
        ||  1. Creer un utilisateur             ||
        ||  2. Creer un produit                 ||
        ||  3. Ajouter un avis sur un produit   ||
        ||  4. Afficher infos utilisateur       ||
        ||  5. Afficher infos produit           ||
        ||  6. Afficher tous les utilisateurs   ||
        ||  7. Afficher tous les produits       ||
        ||  8. Quitter                          ||
        ==========================================
        >> Veuillez entrer votre choix :
        Entrez votre choix: 5
Entrez l'ID du produit: 1
ID Produit: 1
Nom: Ordi_HP
Score de Reputation: 66
```

```
          ==========================================
          ||            MENU PRINCIPAL              ||
          ==========================================
          ||  1. Creer un utilisateur               ||
          ||  2. Creer un produit                   ||
          ||  3. Ajouter un avis sur un produit     ||
          ||  4. Afficher infos utilisateur         ||
          ||  5. Afficher infos produit             ||
          ||  6. Afficher tous les utilisateurs     ||
          ||  7. Afficher tous les produits         ||
          ||  8. Quitter                            ||
          ==========================================
          >> Veuillez entrer votre choix :
          Entrez votre choix: 6
ID Utilisateur: 2
Nom: thierno
Email: thierno@gmail.com
-------------------------
ID Utilisateur: 1
Nom: diallo
Email: diallo@gmail.com
-------------------------
```

```
          ==========================================
          ||            MENU PRINCIPAL              ||
          ==========================================
          ||  1. Creer un utilisateur               ||
          ||  2. Creer un produit                   ||
          ||  3. Ajouter un avis sur un produit     ||
          ||  4. Afficher infos utilisateur         ||
          ||  5. Afficher infos produit             ||
          ||  6. Afficher tous les utilisateurs     ||
          ||  7. Afficher tous les produits         ||
          ||  8. Quitter                            ||
          ==========================================
          >> Veuillez entrer votre choix :
          Entrez votre choix: 7
ID Produit: 2
Nom: Ordi_DEL
Score de Reputation: 0
-------------------------
ID Produit: 1
Nom: Ordi_HP
Score de Reputation: 66
-------------------------
```


File Edit Selection View Go Run
online-reputation-management-system
EXPLORER
OPEN EDITORS
main.c src
README.md
graph.c C:\Users\...
graph.h C:\Users...
ONLINE-REPUTATION-MA...
include
ressource
src
main.c
main.o
produit.c
produit.o
reputation.c
reputation.o
utilisateur.c
utilisateur.o
main.exe
Makefile
rapport.md
README.md
main.c
README.md
graph.c
graph.h
src > main.c > main()
/**
* Fonction principale
*/
int main() {
int choix, id_utilisateur, id_produit, avis_positifs, avis_negatifs;
char nom[50], email[100], nom_produit[50];
Utilisateur *utilisateur;
Produit *produit;
while (1) {
afficher_menu();
printf("\n\tEntrez votre choix: ");
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR COMMENTS
powershell
MENU PRINCIPAL
1. Creer un utilisateur
2. Creer un produit
3. Ajouter un avis sur un produit
4. Afficher infos utilisateur
5. Afficher infos produit
6. Afficher tous les utilisateurs
7. Afficher tous les produits
8. Quitter
>> Veuillez entrer votre choix :
Entrez votre choix: 8
PS D:\Algo3\graphe\Projet_C\online-reputation-management-system>