

Projet : Structures de données



L2 CUPGE

Table des matières

I.	Introduction :	4
II.	Organisation du projet :	5
1)	Architecture du projet :	5
2)	Découpage du programme :	6

I. Introduction :

Dans le cadre du module de structures de données, nous étions amenés à réaliser un projet consistant à modéliser un programme qui intègre toutes les fonctionnalités d'une banque, de la gestion des comptes bancaires (compte courant, livret A, PEL, etc...), jusqu'à la sécurisation des données des clients, en utilisant bien évidemment le langage C sous un environnement Linux.

Le projet en question sert à nous familiariser plus avec le langage C et le système d'exploitation Linux ainsi que les systèmes VCS (GIT, BITBUCKET, MERCURIAL, ETC...) en général et l'implémentation des structures de données plus sophistiquées ainsi que l'abstraction de ces objets dans un contexte abstrait en particulier.

Nous sommes, MOUDDENE Hamza et KANE Nouhou, les réalisateurs de ce travail que nous vous avons transmis par voie électronique.

II. Explication programme

Notre programme est une application bancaire assez complète. Elle est composée de deux parties : une partie administration et une partie clientèle toutes deux protégées par un mot de passe, ainsi qu'un identifiant pour les clients.

L'administration se gère de créer les clients. Chaque client se voit assigner un ID unique lors de son implémentation au sein de la banque ainsi qu'un compte courant. Ses coordonnées doivent être renseignées (numéro de téléphone, mail etc...), cela peut être fait par l'administrateur ou par le client lui-même. Le client doit choisir un mot de passe confidentiel lors de la création de son compte.

Une fois le compte client créé, il est possible d'ouvrir plusieurs comptes de différents types (livret A, compte joints etc...).

L'administrateur se charge de la création et suppression des comptes, l'ajout d'un client, l'affichage de la liste des titulaires d'un compte ou l'affichage des titulaires des comptes par type.

Le programme permet au client de consulter son solde, de faire des transactions d'un compte à un autre ainsi que de regarder ses dépenses au cours d'une période choisie.

Maintenant que nous avons parlé du programme, passons à l'explication du code.

III. Organisation du projet :

1) Architecture du projet :

L'ensemble de notre projet est organisé selon une structure conventionnelle d'un projet en langage C, avec plusieurs dossiers et fichiers répertoriés ci-dessous:

Bin : Est le dossier qui contient le fichier exécutable après la compilation sous le nom « main ».

Data : Est le dossier contenant tous les fichiers .csv contenant les opérations effectuées par les clients, ainsi que deux fichiers .json, un contenant la liste des clients avec leurs données, l'autre contenant le mot de passe de l'administrateur du programme.

Doc : Est le dossier contenant l'énoncé de ce projet, ainsi que son rapport.

Include : Est le dossier contenant tous fichiers .h contenant les structures utilisées dans le projet ainsi que les prototypes de toutes les fonctions.

Lib : Est le dossier qui regroupe tous les fichiers .o générés par le Makefile puis produit un fichier une librairie statique qui sert à achever la compilation dans une étape postérieure.

Src : Est le dossier contenant tous fichiers .c contenant l'implémentation des structures et des fonctions écrites dans le dossier include.

Makefile : Est le fichier qui met en place la compilation séparée et facilite le processus de l'exécution.

README : Est le fichier contenant une présentation rapide du projet ainsi que les bibliothèques nécessaires et notamment les commandes pour exécuter le programme.

LICENSE : Est le fichier contenant la licence open de source de ce projet qui garantit les quatre libertés : le droit de **lire**, **d'exécuter**, de **changer** et de **distribuer**.

2) Découpage du programme :

Pour simplifier l'implémentation de ce programme, nous avons créé plusieurs fichiers contenant chacun plusieurs fonctions, ces fichiers sont des .c et se trouvent dans le dossier src.

1- Admin.c

Admin.c est un fichier répertoriant toutes les fonctions liées à l'administrateur et toutes les actions qui lui sont possibles d'exécuter à partir du programme.

Add_client : la fonction permettant de créer un client au sein de la banque (lors de la création d'un client, un compte courant est créé automatiquement.).

Create_account : la fonction créant un compte d'un client donné.

Edit client : la fonction servant à éditer et modifier les coordonnées d'un client donné.

Delete_account : la fonction supprimant un compte choisi d'un client donné.

Display_account_list_by_type : la fonction affichant la liste des comptes d'un type donné.

Display_entitled_list : la une fonction servant à afficher la liste des titulaires d'un comptes donné.

Change_admin_passwd : la fonction qui permet de changer le mot de passe de l'administrateur.

2- Client.c

Client.c est un fichier répertoriant toutes les fonctions liées au client.

Consult_balance : permet au client de consulter son solde.

Transaction_list : permet au client de consulter sa liste de transactions sur une période choisie.

Transfer_money : permet au client de transférer de l'argent d'un compte à un autre.

Pay_by_card : Permet au client d'utiliser son compte bancaire afin de payer avec sa carte bleue.

Make_deposit : Permet au client de faire un dépôt sur un compte.

Client_cretae_account : Permet au client de créer un compte bancaire lui-même du type qu'il souhaite.

Client_delete_account : Permet au client de supprimer lui-même un compte.

3- displayShell.c

DisplayShell.c est un fichier contenant toutes les fonctions **d'affichage** du programme. Ce fichier est composé en plusieurs parties, la première pour afficher le menu principal, et deux pour afficher les sous-menus clients et administrateurs et enfin une dernière affichant les différents messages d'erreurs. Les affichages sont colorés pour la plupart afin de faciliter la compréhension du programme.

Init_bar / End_bar : Affiche la barre d'initialisation au début et à la fin du programme.

Display_menu : Affiche le menu principal avec les choix d'accéder aux menus client et administrateur ou de quitter le programme.

Display_client : Affiche le menu client divisé en sections : **Gestion de compte et Administration**.

Display_client_account_management : Affiche toutes les options que le client peut gérer en lien avec la gestion de compte.

Display_client_administration : Affiche toutes les options que le client peut gérer en lien avec l'administration.

Display_administrator : Affiche le menu administrateur divisé en sections : **Gestion de comptes , Gestion et client et Administration**.

Display_admin_account_management : Affiche toutes les options que l'administrateur peut gérer en lien avec la gestion de compte.

Display_admin_client_management : Affiche toutes les options que l'administration peut gérer en lien avec la gestion de client.

Display_admin_administration : Affiche un sous menu de l'administration leur permettant de changer de mot de passe.

Display_error_flag : Affiche un message d'erreur lorsque l'utilisateur tape un mauvais numéro dans le menu

Display_error_connexion : Affiche un message d'erreur lorsque le client rentre un mauvais identifiant ou mot de passe.

Display_error_connexion_admin : Affiche un message d'erreur lorsque l'administrateur rentre un mauvais mot de passe.

Display_account_type : Affiche les comptes par types

4- input.c

Input.c est un fichier répertoriant toutes les fonctions d'entrées du programme.

Hide_passwd : Affiche des étoiles lors de l'entrée du mot de passe afin de le garder confidentiel.

Generate_unique_id : Fonction qui génère automatiquement un identifiant unique lors de l'ajout d'un client.

Create_passwd : Permet au client de se créer un mot de passe lors de son ajout dans la banque.

Input_id / Input_passwd : Permet au client de rentrer son identifiant et son mot de passe afin de se connecter à son compte.

Input_last_name	}	Demande les coordonnées d'un client et les enregistre
Input_first_name		
Input_birthday		
Input_mail		
Input_phone		

Choose_feature : Permet de choisir à l'utilisateur l'option affichée dans le menu qu'il souhaite exécuter.

Input_type : Permet à l'utilisateur de choisir le type de compte qu'il souhaite créer.

Input_entitled : Enregistre le nom des titulaires d'un compte

Input_perso_info : Demande les coordonnées d'un client et les enregistre

5- json.c

Le fichier json.c manipule la structure de donnée json.

6- security.c

Security.c est le fichier regroupant les fonctions s'occupant du cryptage des mots de passes, les identifiants, sécurisant les connexions etc...

Encrypt_passwd : La fonction permettant de crypter les mots de passe.

Valid_client : Permet au client de se connecter avec l'identifiant et le mot de passe car cette fonction les vérifie.

Connect_admin : Permet à l'administrateur de se connecter avec le mot de passe car cette fonction le vérifie.

Change_client_passwd : Permet au client de changer son mot de passe à tout moment.

Change_administration_passwd : Permet à un administrateur de changer le mot de passe.

7- struct.c

struct.c est un fichier regroupant toutes les structures du programme.

8- Main.c

Main.c est un fichier regroupant l'appel de toutes les fonctions, c'est la fonction principale du programme.

IV. Problèmes rencontrés

Le principal problème rencontré est l'organisation au sein du groupe. Certes nous étions seulement deux mais le travail s'est tout de même retrouvé mal répartie avec une non-cohésion du groupe.

Les problèmes concernant le code ont été nombreux mais pour la plupart, résolue.

Principalement, nous avons eu des problèmes concernant la bibliothèque **json** qui n'est pas stable.

Aussi, au tout début du projet, il a été compliqué de s'organiser dans le programme, nous ne savions pas vraiment comment l'aborder et sommes souvent revenus sur nos décisions pour modifier la structure de notre programme (comme la suppression des scanf par exemple).

La plupart des problèmes restent des problèmes de réflexion que nous avons finalement réussi à résoudre.

V. Mode d'emploi

Comme dit plus haut, notre programme est divisé en deux parties. Une pour la clientèle et l'autre pour l'administration.

Sur le menu principal, il vous sera demandé si vous êtes un client ou un administrateur. Tapez 1 si vous êtes un client un identifiant et un mot de passe vous sera demandé (en cas d'oubli veuillez contacter l'administration) et tapez 2 si vous êtes un administrateur, seul un mot de passe vous sera demandé.

Si vous êtes client :

Un menu va s'afficher vous proposant de gérer votre compte ou de gérer votre profil.

Tapez 1 si vous souhaitez exécuter une des actions suivantes :

- Consulter votre solde
- Consulter vos transactions effectuées sur une période donnée
- Transférer de l'argent entre vos comptes ou sur un compte d'une autre personne
- Payer par carte bancaire
- Faire un dépôt

Tapez 2 si vous souhaitez exécuter une des actions suivantes :

- Changer votre mot de passe
- Créer un compte du type que vous souhaitez
- Supprimer un compte

Si vous êtes administrateur:

Tapez 1 si vous souhaitez exécuter une des actions suivantes :

- Créer un compte
- Supprimer un compte
- Afficher la liste des comptes par type

Tapez 2 si vous souhaitez exécuter une des actions suivantes :

- Ajouter un client
- Modifier les coordonnées d'un client
- Afficher la liste des titulaires des comptes

Tapez 3 si vous souhaitez exécuter une des actions suivantes :

- Changer le mot de passe administrateur