Documentatie proiect programare in C

Moglan Calin-Stefan

Cerinta al 2 lea proiect:

An accounting project involves managing financial and accounting data, performing financial calculations, generating reports, and ensuring data security. Here are some common requirements for such a project:

1. User Interface: Develop a user interface (UI) to allow users to input and view financial data. This can be a command-line program or a graphical user interface (recommended). 10 points /20 points

2. Account Management: Allow users to create, edit, and delete financial accounts, such as savings accounts, checking accounts, credit accounts, etc. 10 points

3. Transaction Recording: Implement functionality to record financial transactions, including deposits, withdrawals, transfers, payments, and more. 10 points

4. Balance Calculation: Automatically calculate account balances based on recorded transactions. 10 points

5. Financial Reporting: Provide options for generating financial reports, such as account statements, balances, transaction registers, and expense reports. 5 points

6. Security and Authentication: Ensure that financial data is protected, and users must authenticate to access the accounting system. 5 points

7. Customer Data Management: Allow users to input and manage information about customers or entities with whom they do business. 5 points

8. Data Export and Import: Offer options for exporting and importing financial data in and out of various file formats (e.g., CSV, Excel, PDF export). 5 points

9. Input Data Validation: Perform checks on user-entered data to prevent errors or incorrect data. 10 points

10. Error and Exception Handling: Implement a system for proper error handling and handling exceptional situations that may arise during the accounting process. 15 points

11. Audit and Logging: Maintain an activity log and provide audit capabilities to track changes and activities related to accounts. 5 points

12. Documentation and Comments: Document the code (PDF) and add comments to make it easier to understand. 10 points

13. Compliance with Specific Project Requirements: Ensure that the project meets the specific requirements set by the instructor.

14. Presentation and Explanations: Ensure that you can explain and present the project to the instructor or peers, highlighting key features and design decision

Lista de functionalitati:

-adaugare user

-adaugare cont financiar

-adaugare tranzactii

-adaugare transfere intre conturi

-adaugare review

-intrare in mod admin

-editare user(nume , parola)

-editare cont financiar(nume, prenume, parola)

-stergere cont financiar

-calculare sold

-facut rapoarte pentru incasari, cheltuieli, account statement si transaction register

-scriere date in fisier .txt

-scriere date in fisier csv

-validare date introduse de utilizator

Biblioteca functii:

void citeste\_utilizator(struct Users \*user, int id\_cont, int id); - citeste datele unui utilizator

int add\_utilizator(struct Users \*user, int id,int id\_cont, int type, char nume[], char prenume[], char parola[]); - adauga utilizatorul in structura de tip Users

int edit\_utilizator\_nume(struct Users \*user); - editeaza numele utilizatorului

int edit\_utilizator\_prenume(struct Users \*user); - editeaza prenumele utilizatorului

int edit\_utilizator\_parola(struct Users \*user); - editeaza parola utilizatorului

void citeste\_cont(struct Account \*cont, int id); - citeste datele contului

int add\_account(struct Account \*cont,int id, char username[], char parola[]); - adauga contul in structura de conturi

int edit\_cont\_username(struct Account \*cont); - editeaza username ul contului

int edit\_cont\_parola(struct Account \*cont); - editeaza parola contului

void afisare\_cont(struct Account \*cont, int nr\_conturi); - afiseaza conturile

void afisare\_utilizatori(struct Users \*user, int id, int nr\_utilizatori); - afiseaza utilizatorii asociati unui cont

void afisare\_cont\_admin(struct Account \*cont, int nr\_conturi); - functie pentru afisarea conturilor in modul de admin

void afisare\_utilizatori\_admin(struct Users \*user, int nr\_utilizatori); - functie pentru a afisa toti utilizatorii in modul admin

int authentication\_account(struct Account \*cont,int id\_citit, int nr\_utilizatori); - functie care autentifica conturile

int authentication\_fin\_acc(struct Users \*user, int nr\_fin\_acc, int id\_citit, int id\_cont); - functie care autentifica utilizatorii

void sterge\_utilizator(struct Users \*user, int \*nr\_fin\_acc); - functie care sterge un utilizator

void citeste\_tranzactie(struct Transactions \*transaction, int id\_cont, int id\_utilizator, int type); functie cu care se citesc tranzactiile

int adauga\_tranzactie(struct Transactions \*transaction, int type, float amount, char description[], int id\_cont, int id\_utilizator, int day, int month, int year); - functie care adauga datele tranzactiei in structura de tip transfer

int adauga\_transfer(struct Transactions \*transaction, int id\_cont, int id\_tranzactie, int id\_dest\_cont, int id\_dest\_utilizator,int type, char description[], float amount, int day, int month, int year ); - adauga transferal in structura de tip transfer

void citeste\_transfer(struct Transactions \*transaction, int id\_cont, int id\_utilizator, int type); - functie care citeste datele unui transfer

float acc\_balance(struct Transactions \*transaction, int id\_cont, int id\_utilizator, int nr\_tranzactii); - functie care calculeaza soldul unui utilizator

void afiseaza\_tranzactii(struct Transactions \*transaction, int id\_cont, int id\_utilizator, int nr\_tranzactii); - functie care afiseaza toate tranzactiie

void afiseaza\_transfer(struct Transactions \*transaction, int id\_cont, int id\_utilizator, int nr\_tranzactii); - functie care afiseaza toate transferurile

void raport\_expense(struct Transactions \*transaction, int id\_cont, int id\_utilizator, int nr\_tranzactii); - functie care face raportul cheltuielilor

void raport\_income(struct Transactions \*transaction, int id\_cont, int id\_utilizator, int nr\_tranzactii); - functie care face raportul incasarilor

void citeste\_un\_cuvant(char \*cuvant, int lungime\_maxima); - functie folosita pentru a asigura corectitudinea datelor introduce de utilizator. In cazul de fata date precum numele sau parolele nu pot contine maim ult de un cuvan. Aceasta functie ne garanteaza ca datele vor fi corecte

void afiseaza\_fin\_acc(struct Users \*user, int id\_cont, int nr\_fin\_acc); - afiseaza toti utilizatorii

bool citeste\_numar\_intreg(int \*numar); - functie care garanteaza ca data introdusa de utilizator este un numar intreg

int citeste\_admin(char parola\_admin[255]); - functie care citeste utilizatorul de tip admin

void afiseaza\_tranzactii\_admin(struct Transactions \*transaction, int nr\_tranzactii); - functie care afiseaza toate tranzactiile in modul admin

void scrieConturiInFisier(struct Account \*cont, int nr\_conturi, const char \*numeFisier); - functie care scrie datele din structura de tip Account in fisier

int citesteConturi(const char \*numeFisier, struct Account \*cont, int dimensiuneVector, int frecvcont[]); - functie care citeste datele structurii de tip account din fisier

void scrieUseriInFisier(struct Users \*user, int nr\_useri, const char \*numeFisier2);

int citesteUseri(const char \*numeFisier2, struct Users \*user, int dimensiuneVector, int frecvuser[]);  
  
int citesteTranzactii(const char \*numeFisier3, struct Transactions \*transaction, int dimensiuneVector);  
  
void scrieTRanzactiiInFisier(struct Transactions \*transaction, int nr\_tranzactii, const char \*numeFisier3);  
  
void scrieConturiInFisierCSV(struct Account \*cont, int nr\_conturi, const char \*numeFisier);  
  
int citesteConturiCSV(const char \*numeFisier, struct Account \*cont, int dimensiuneVector);  
  
void scrieUseriInFisierCSV(struct Users \*user, int nr\_useri, const char \*numeFisier2);  
  
int citesteUseriCSV(const char \*numeFisier2, struct Users \*user, int dimensiuneVector);  
  
int citesteTranzactiiCSV(const char \*numeFisier3, struct Transactions \*transaction, int dimensiuneVector);

void scrieTRanzactiiInFisierCSV(struct Transactions \*transaction, int nr\_tranzactii, const char \*numeFisier3);

Toate aceste functii citesc si scriu date in fisier!!!

void scrieReview(struct Reviews \*tranzactie, int \*numarReviews, int frecvcont[]); - functie care ajuta la citirea unei recenzii

void scrieReviewsinFisier(struct Reviews \*recenzie, int nr\_recenzii, const char \*numeFisier4);  
  
int citesteReviews(const char \*numeFisier4, struct Reviews \*recenzie, int dimensiuneVector);

Functii care scriu si citesc recenziile din fisierul predestinat recenziilor!

void afiseaza\_all\_review(struct Reviews \*recenzie, int nr\_recenzii); - functie care afiseaza toate recenziile!