My Documentation

Contents

C Account project by Malos Alexandru-Razvan Informatica Engleza

Overview:

The project consists of a C program for managin accounts, transactions, customers. It uses a GUI and a data base

The login window:

The project opens with a login window in which you can either register by making an account or logging in an already existent account, the password is censored and requires at least one special characther and one number and there are no duplicate usernames. After entering into an account a window consisting of 4 buttons and a drop down menu appears.

The Main Menu:

Starting with the drop down menu, there is the Exit option, that has another 2 options, fully exiting or just signing out of this account, going back to the login window. The second option is the data, which is not yet implemented but contains 6 options, CSV Excel PDF Import and Export which will basically either

export some financial reports in the 3 formats, as chosen, or inputs accounts and transaction from one of the 3 formats.

The third option is View, which has 5 options:

1. Accounts. This button opens a dialog which shows all the users accounts by saying the Id, type and the balance

2. Balance. This is also a financial Report, but I chose to put it in the view menu as it made more sense, it shows the accountID followed by the balance and at the end the user general balance

3. Transaction History. This button opens the same dialog as the other 2(all 5 open the same) and it prints the account and then all the transactions done by that account, by showing transactionId, customerID(which can also represent another accountID in case of a transfer between 2 accounts), TransactionType, Amount, Date(DD-MM-YYYY) and Notes

4. Activity Log. This button shows for every account the actions that were done and the time when they were inserted into the application, for example if you record a transaction and you put the date to be 12-12-2021 in the activity log the date will still be the curent one.

5. Customers. This button shows the customers that an account worked with, for every account. An user may only see the customers with which he has worked, as it would be unethical to show him all the customers in the data base

After the view button, the last one is the Financial Reports button, which has 3 options, of which only one implemented for now. Accout Statement, Transaction Register (which are both relatively close to things already implemented) and Expense Reports, which shows only the Payment and Withdrawal Transaction, with the notes and amount, and at the end also shows a Total Amount

The 4 main buttons   The 4 buttons in the main part of the window are used for inputing data: creating, editing and deleting various things.

1. Create account. The Most Important button, as without an account an user can't do anything. When pressing the button, a window is opened with 2 entry widgets, one for the type and one for the initial balance, that is automatically set to 0 if empty. The submit button is sensitive, so until any data in inserted in the type, it won't work. There is also error handling, so if the type contains anything but letters without spaces or the balance is not a positive number, the button won't be accessible and the errors will be shown to the right of the entry

2. Delete/Edit account. Here, the user inputs an accountID, which he can find in the view drop down menu, and maybe a type or balance. After the ID is inserted, the delete button becomes available, if pressed, the program

will first search for the user and give an error next to the ID entry if it's not found, and if it is found, it deletes the account. For the edit functionality, the button will be available only after there is a valid ID and either a valid balance or a valid type, checked the same way as the first button.

3. Record Transactions. The most complex button and perhaps the most useful one. In this button you necessarily insert the AccountId TransactionType(Deposit, Withdrawal, Transfer, Payment), Amount, Date. After that the user can insert notes if he wants, which is also recommended. There is a lot of error checking to be done, first the ID must be a number, then the TransactionType must be one of the 4 specified ones, the amount has to be a positive number, and the date must be before the curent day and actually exist(we check for month, leap years). After this, if the user inputs either Transfer of Payment for the type, he also has to input another AccountID or a CustomerID respectivly. After all these inputs are corect, the button is available and the user can submit this transaction. When pressed the program first checks if the accounts and customer exist(if inputed) and if the transaction is Deposit Withdrawal or Payment it also checks if the account has enough money, if there is any error, it is printed in its respective error widget, if not the balance is updated, the transaction is recorded and so is the audit log.

4. Customer Data Management. The last button. The user can either create or change a customer(but not delete since he might already have transactions with that customer). The user has 6 entries: CustomerID, First Name, Last Name, Address, Email, Phone. For creating a customer the user needs to input all the entries except the ID one, as that one will be automatically incremented. After the data is checked and there are no errors, the button will be available, the customer is created and in the customerID error label the customerID will pe shown, so the user knows what ID to use. For editing the user has to input at least the ID and one of the other 5 and the button unlocks.

Managing the data:

For managing the data I used a data base in sqlite3 made of 5 tables, users, accounts, transactions, customers, activitylog. This way, using the UserID as a foreign key and the accountid aswell, the data is secured. For accessing the data for an account the user needs to login in that account. I used SQLite3 for making the datables and managing all the data. All the data relevant to creating the data base is in the gui.c in src.

GUI:

For making the GUI I used the gtk library in C and the glade program for making the overview of the program. The initialisation of the GUI is present in gui.c.

Functions:

All the functions are made in the function.c file

TODO:

- The 2 financial reports
- the export and import of data