Mini Software Project 1

EEE 111 Introduction to Programming and Computation

1st Semester, AY 2024-2025

THRU2, THVW, WFRU2, WFVW

READ THE COURSE SYLLABUS BEFORE PROCEEDING TO THE SPECIFICATIONS BELOW.

Problem

The banking industry is now adopting a digitalized and automated transaction system to aid the traditional setup where people physically go to the banks before they can proceed with their transactions. You have been tasked to implement such a system using the Python programming language in the Philippine setting where banknotes and coins have denominations of PHP 1,000, PHP 500, PHP 200, PHP 100, PHP 50, PHP 20, PHP 10, PHP 5, and PHP 1. For now, ignore the coins with values less than PHP 1.

The digitalized and automated system must include the following functionalities:

WELCOME

Display a welcome message and ask the user if the user wants to log in, create a new account, or exit the system.

LOG IN

Prompt the user to enter their case-sensitive username, account number, and password.

If the user inputs the three information correctly, then proceed to the transactions page.

If the user inputs at least one information incorrectly, then display a message to inform the user that the inputs are invalid and ask the user to log in again. A user is only allowed a maximum of three attempts (i.e., first attempt and next two attempts) during this stage before the program notifies the user that the program will go back to the welcome page.

CREATE A NEW ACCOUNT

Prompt the user to enter their case-sensitive username and password, which the user will use to be able to log in.

Display an account number that is one more than the most recent account number in the record of existing and old bank users. This record includes those who performed a close account transaction. This new account number will be used by the user to log in.

Require the user a minimum deposit amount of PHP 500. The user can input any combination of denomination. If the user makes a deposit that is less than the required amount, then display a message to inform the user that the minimum amount is not met and ask the user to deposit again. In addition, require the user to input a username that has not been taken (i.e., usernames must be unique). If the user enters a username that has been taken, then display a message to inform the user that the username has been taken and ask the user to enter another username. A user is only allowed a maximum of three attempts during this stage (simultaneously covers both the minimum deposit requirement and unique username requirement) before the program notifies the user that the program will go back to the welcome page.

If the creation of a new account is successful, go to the login page.

EXIT

An exit message is displayed. Close the Python program.

TRANSACTIONS

The user selects among the following transactions: check balance, deposit, withdraw, close account, and log out.

CHECK BALANCE

Display the remaining balance of the account of the user.

After displaying, ask if the user wants to make another transaction or to log out.

DEPOSIT

The user enters the number of banknotes and coins to be deposited, in which the deposit amount will be added to the balance of the user's account. Moreover, the aggregate deposit will be added to the total available banknotes and coins in the system, considering all users.

> Example: The user has an existing balance of PHP 1,500, while the total available amount in the system, considering all users, is PHP 25,000. The user makes a deposit of 2x PHP 200, 3x PHP 20, and 2x PHP 5 coins. This deposit will result to the user's account having a new balance of PHP 1,970, and the total available amount in the system will be PHP 25,470.

After a successful deposit transaction, ask if the user wants to make another transaction or to log out.

WITHDRAW

The user enters the amount to be withdrawn. Calculate and display the number of banknotes and coins to be dispensed by the system. Prioritize dispensing banknotes and coins of higher values.

> Example 1: The total available amount in the system is PHP 1,400, and this is comprised of the following banknotes/coins: 1x PHP 500, 3x PHP 200, 3x PHP 100 (and 0x for all other denominations). A user wants to withdraw PHP 600, so the system will dispense: 1x PHP 500, 1x PHP 100.
>
> Example 2: The total available amount in the system is PHP 8,000, and this is comprised of the following banknotes/coins: 3x PHP 1,000, 10x PHP 500 (and 0x for all other denominations). A user wants to withdraw PHP 4,000, so the system will dispense: 3x PHP 1,000, 2x PHP 500.
>
> Example 3: The total available amount in the system is PHP 1,500, and this is comprised of the following banknotes/coins: 1x PHP 500, 5x PHP 200, 0x PHP 100 (and 0x for all other denominations). A user wants to withdraw PHP 600, so the system will dispense 3x PHP 200.

The withdrawal amount will be deducted from the user's account, total available amount in the system, and available number of banknotes and coins.

> Example: The user has a remaining balance of PHP 5,600. The total available amount in the system is PHP 10,505 with the following banknotes/coins: 5x PHP 1,000, 4x PHP 500, 8x PHP 200, 5x PHP 100, 20x PHP 50, 20x PHP 20, 0x PHP 10, 1x PHP 5, 0x PHP 1. A withdrawal transaction is made by the user, amounting to PHP 1,600. With this, the system will dispense the following banknotes/coins: 1x PHP 1,000, 1x PHP 500, 1x PHP 100. The resulting remaining balance of the user will be PHP 4,000. The remaining total available amount in the system will be PHP 8,905 with the following remaining banknotes/coins: 4x PHP 1,000, 3x PHP 500, 8x PHP 200, 4x PHP 100, 20x PHP 50, 20x PHP 20, 0x PHP 10, 1x PHP 5, 0x PHP 1.

After a successful withdrawal transaction, ask if the user wants to make another transaction or to log out.

If the total available amount in the system is less than the amount to be withdrawn, notify the user of a withdrawal error.

> Example: The total available amount in the system is PHP 5,500, but a user wants to withdraw PHP 6,505. This will trigger a withdrawal error.

If the total available amount in the user's account is less than the amount to be withdrawn, notify the user of a withdrawal error.

> Example: The total available amount in the user's account is PHP 3,515, but a user wants to withdraw PHP 4,000. This will trigger a withdrawal error.

If the total available amount in the system is enough to cover the amount to be withdrawn but the remaining banknotes and coins cannot meet the withdrawal amount exactly, notify the user of a withdrawal error.

> Example: The amount to be withdrawn is PHP 70 but the system only has one PHP 100 banknote, and no other banknotes/coins are available. This will still trigger a withdrawal error.

After a withdrawal error is conveyed to the user, return to the transactions page.

CLOSE ACCOUNT

Prompt the user to enter their password again for verification. A user is only allowed a maximum of three attempts during this stage before the program notifies the user that the program will go back to the transactions page.

Remove the data of the user and withdraw the total amount remaining in the user's account. Follow the rules in performing a withdrawal transaction. Remember to keep track of the account numbers of users who performed a close account transaction since these will be needed when a new user creates a new account.

After a close account transaction, go back to the welcome page.

LOG OUT

Log out the account of the current user and go back to the welcome page.

EXISTING ACCOUNTS

Assume that the users below have already made initial deposits as summarized in the table below. From the values below, the total available system amount is PHP 20,485 initially.

Username	Account Number	Password	Initial Deposits
mdluffy	101	password	4x PHP 1,000 1x PHP 500 2x PHP 200 0x PHP 100 0x PHP 50 7x PHP 20 1x PHP 10 9x PHP 5 14x PHP 1
buggy_d_clown	102	12345	2x PHP 1,000 2x PHP 500 3x PHP 200 0x PHP 100 6x PHP 50 7x PHP 20 7x PHP 10 5x PHP 5 20x PHP 1
pulang.buhok	103	pulang.buhok	5x PHP 1,000 1x PHP 500 1x PHP 200 0x PHP 100 5x PHP 50 5x PHP 20 2x PHP 10 2x PHP 5 7x PHP 1
balbas-ay-black	104	111	4x PHP 1,000 1x PHP 500 1x PHP 200 0x PHP 100 5x PHP 50 7x PHP 20 1x PHP 10 6x PHP 5 4x PHP 1

Additional Requirements

Due to the nature of the program functionalities, this system will loop indefinitely until the user(s) choose(s) to exit the program.

Print a separator for each page for clarity. For example, print dashes (- - - - - - - - - - - - - - -) to separate each page. You can also use other ways to improve the distinction between two consecutive pages.

Program Restriction

Do not create classes.

Deadline

Refer to your respective UVLe groups.

Frequently Asked Questions

Refer to this section for the answers to frequently asked questions.

[Question 1] Are the dashes strictly required to indicate the separation between two consecutive pages?

[Answer 1] No. You may use other techniques.

Update Logs

Some students may be raising concerns that may affect the specifications minimally. Please refer to this section for any minimal updates on the specifications. All requirements stated in this document (i.e., this includes all minimal updates) will be deemed final by 4 October 2024, 5:00 PM.

[Update 1; 27 September 2024]

Old – CLOSE ACCOUNT: “Prompt the user to enter their case-sensitive username, account number, and password again for verification.”

New – CLOSE ACCOUNT: “Prompt the user to enter their password again for verification.”

[Update 2; 27 September 2024]

Old – CLOSE ACCOUNT: There is no maximum number of attempts during a close account transaction.

New – CLOSE ACCOUNT: “A user is only allowed a maximum of three attempts during this stage before the program notifies the user that the program will go back to the transactions page.”

[Update 3; 27 September 2024]

Old – CREATE A NEW ACCOUNT: “Require the user a minimum deposit amount of PHP 500. The user can input any combination of denomination. If the user makes a deposit that is less than the required amount, then display a message to inform the user that the minimum amount is not met and ask the user to deposit again. A user is only allowed a maximum of three attempts during this stage before the program notifies the user that the program will go back to the welcome page.”

New – CREATE A NEW ACCOUNT: “Require the user a minimum deposit amount of PHP 500. The user can input any combination of denomination. If the user makes a deposit that is less than the required amount, then display a message to inform the user that the minimum amount is not met and ask the user to deposit again. In addition, require the user to input a username that has not been taken (i.e., usernames must be unique). If the user enters a username that has been taken, then display a message to inform the user that the username has been taken and ask the user to enter another username. A user is only allowed a maximum of three attempts during this stage (simultaneously covers both the minimum deposit requirement and unique username requirement) before the program notifies the user that the program will go back to the welcome page.”