ATM CONSOLE-BASED JAVA APPLICATION

Submitted by:

FLORES, Axel John A.

BS Information Technology - NW1H

Submitted to:

THOMAS ANTHONY D. SANCHEZ

Instructor, CTCC0213

I. Background and Description

The ATM Console-based java application is a desktop application that functions similarly to an ATM machine designed in Java to stimulate its process and showcase various Java programming structures. Users are required to input a PIN code to prevent unauthorized access, with the input limited to three tries. If this limit is exceeded, the application halts immediately. The desktop application can perform essential functions, including checking balance, withdrawing, and depositing.

To withdraw money, users must input an amount equal to or less than the current balance. The specified amount is subtracted from the total balance stored in the array, and the array is then updated with the result. Subsequently, the current balance is displayed.

For depositing money, users need to input the desired amount, which is then added to the total balance of the ATM. The array is updated with the result, and the current balance is displayed.

To check the balance, the program prints the user's total balance from the array. Choosing "exit" in a transaction will automatically end the application. Additionally, after each transaction, users have the option to initiate another transaction or exit the ATM application.

However, the desktop application lacks the capability for users to create their accounts and PIN numbers, the default username and PIN number are predetermined. Moreover, the application is restricted to only three transactions (Check balance, Withdraw, and Deposit); other transactions such as transferring money, paying bills, and changing the PIN are not included.

II. Program Logic Flow

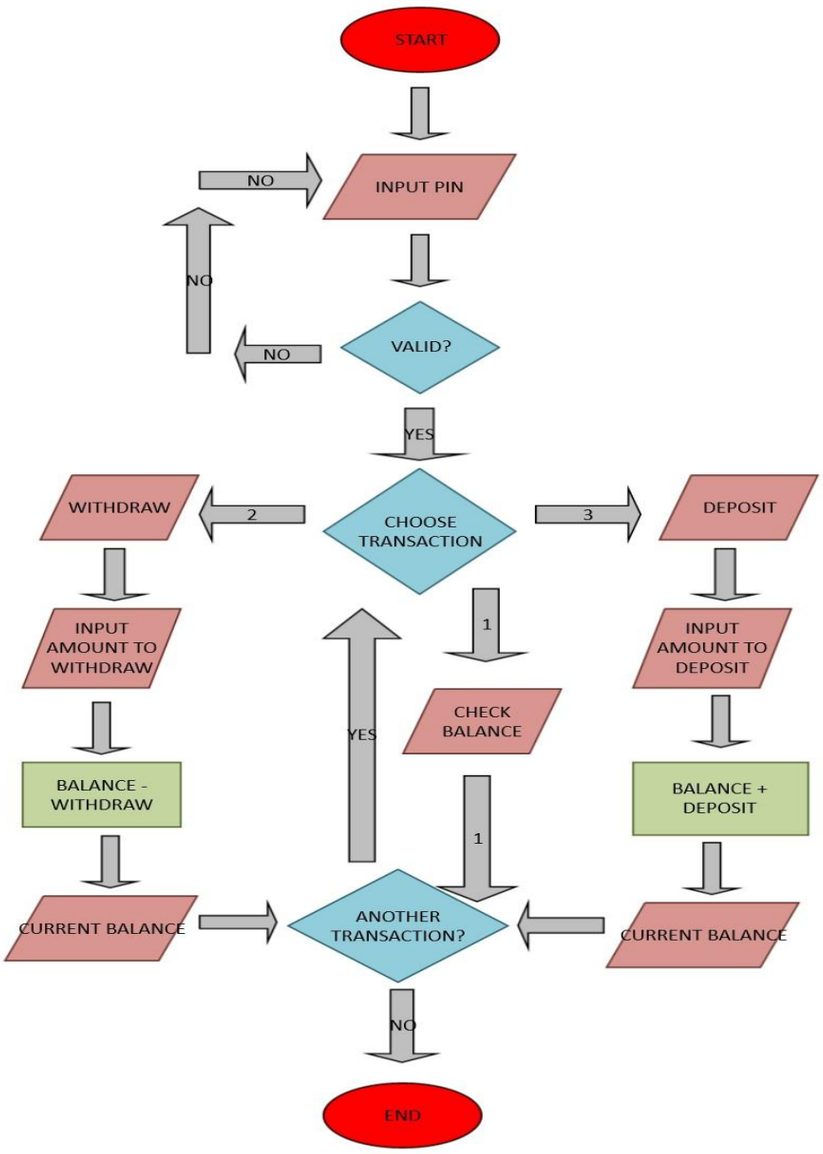

Figure 1. *Program Flow Chart*

Figure 1 illustrates the logical flow of the java console program. The program begins with the input of the PIN number. Using a decision block, the data is evaluated for correctness. If the PIN is incorrect, it will return to the input block; otherwise, it will proceed to the next block. The decision block offers four transaction options: check balance, withdraw, deposit, and exit. If “exit” is chosen, the program will conclude. Opting for “check balance” displays the current balance. Selection of “withdraw” initiates the data through the subtraction block process, while “deposit” subjects the data to the addition block process. Following each option, the current balance is displayed using an input/output block, and a decision block is initiated, providing the option for another transaction or program exit. If “yes” is selected, the process returns to the transaction block; if “no” is chosen, the program concludes.

III. Repository

Instructions: Create a GitHub repository for your desktop application. Ensure that all files (i.e., Java, JPEG/ PNG) are included in the repository. Name your repository in this format: *desktopappname-section-ctcc0213*. For instance, you have created a food delivery application, repository name shall be like this: *fooddelivery-sd1a-ctcc0323*. Make sure it is a *public* repository. Include the username and repository name on the space below. Replace the blue text.

Sample: thommsanchez/foodieapp-sd4a-ictc1117

Repository Name: **user/desktopapp-nw2a-ctcc0513**