**Unit ## Develop: Open ended Program**

***Step 1: Review the Program Specification***

Review the program requirements and how it is supposed to work.

1. What does the program do?

#1 Creating different types of options. #2 Allows the user to deposit or withdraw money.

#3 Save the current balance #4 Load the previous five transactions #5 Changes the card’s pin #6 Gives and keeps track of money deposits and withdrawers.

2. What user inputs does it have?

3. What output does it produce?

4. How does the program end.

#1 When the user enters “Quit” option

***Co-requirements for the program***

1. **What does the program do?**

\*\*1 displays on opening menu

\*\*2 provides options for cash deposits, withdrawers, checking balance as well as changing card pin.

\*\*3 displays opening message

\*\*4 asks the user to deposit or withdraw any amount.

\*\*5 starts the program and leads the user through it with prompts.

\*\*7 displays the user’s available balance.

\*\*8 displays closing message

1. **What user inputs does it have?**

\*\*1 user’s options.

\*\*2 choosing an option

\*\*3 answers to prompts

\*\*4 recording options as per user

1. **What output does it produce?**

\*\*1 menu

\*\*2 opening message

\*\*3 prompts

\*\*4 options

\*\*5 closing message

1. **How does the program end?**

\*\*1. at the end of program, it tells the user the transaction done and shows available balance in case of deposit or withdrawer than also shows the changed pin if the user changes the pin card of the goals completed as for closing message. Returns to the main menu

\*\*2 end of program enter quit to end the program

***Step 5: Determine the classes***

\*\*1) **Menu class**: to display the menu

\*\*2) **Option class**: base class for options as per user.

\*\*3) **Deposit class**: option that allows the user to deposit money.

\*\*4) **Withdraw class**: option that allows the user to withdraw money.

\*\*5) **Balance Check class**: option that allows to user to check balance.

6) **Change Pin:** Allows the user to change the pin

***Steps 6: Define class behaviors, 7: Define class attributes &***

***8: Define Constructors***

#1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class: **Menu (- means private while + means public)**

Attributes:

* \_programList : List<Activity>

Constructors:

+ Menu()

Behaviors:

+ DisplayMenu() : string

How it will work:

The Menu class will display the choices for the user for the program and record that choice as according to what the user chooses.

#2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class: **Option**

Attributes:

* \_name : string
* \_description : string
* ­\_OptionChoosen : bool
* \_possibleOption : int
* \_ChosenOption : int
* \_failed : bool

Constructors:

+ Option(string name, string description, int possibleOption)

Behaviors:

+ ChosenOption (string option) : virtual void

Description: called when an option is choosen uses polymorphism – changes depending on option chosen.

+ GetName() : string

+ SetName() : void

+ GetDescription() : string

+ SetDescription() : void

+ GetChosenOption() : int

+ SetChosenOption() : void

+ SetFailed() : void

+ DisplayMessage(string message) : void

+DisplayChosenOption(): void

How it will work:

It provides the base class for all options the user has to choose.

#3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class: **DepositOption : Option**

Attributes:

* \_depositOption : int
* \_DepositAmount: int

Constructors:

+ DepositOption()(string name, string description, int possibleDeposits)

Behaviors:

+ GetDeposit() : int

+ SetDeposit() : void

How it will work:

It will inherit its attributes and amount of deposited from the deposit base class and change the methods as needed with polymorphism

#4\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class: **WithdrawerOption**

Attributes:

* \_WithdrawerOption : int
* \_WithdrawerAmount: int

Constructors:

+ WithdrawerOption()(string name, string description, int possibleAmount) : base(name, description, possibleAmount

Behaviors:

+ CompleteWithdrawer() : override int

Add/subtract possibleAmount to the totalAmount variable

How it will work:

It will inherit its attributes and methods from the withdrawer base class and change the amount as needed with polymorphism.

#5\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class: **Balance Check Option:Option**

Attributes:

* \_BalanceCheckOption : int
* \_availabilityBalance : int

Constructors:

+ BalanceCheckOption()(int balancecheck, string name, string description, int possibleBalanceAvailable)

Behaviors:

+ Iterate(CheckBalance) : int

+ AvailabilityBlance() : override void

How it will work:

It will inherit its attributes and methods from the CheckBalance base class and record as needed with polymorphism. Has a number of times for a check and displays completion result that is the availability balance.

#6\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class: **Change Pin Option**

Attributes:

* \_name : string
* \_ChangePin : int

Constructors:

+ ChangePinOption()(int ChangePin, string name, string description, int possibleNewPincode)

Behaviors(methods):

+ ChangedPin() : int

+ DisplayNewPin() : void

How it will work:

Allows the user to change the pin and displays the new changed pin.

Here is how my program will flow with this flow chart diagram below::

