Tersa Almaw

Professor Pete Tucker

CS 273

October 18, 2016

1. (10 pts) A requirements specification of the software that needs to be created in your own words. This should be a short English narrative describing what you think the software should do and what it will operated like, not how it will implement its functions.

This software will keep a record of the users (which are the customers) that have an account at the bank. It will keep record of the two types of accounts the user will have it can either be a checking and or a savings account at the bank. It will let the user know how much they have left in their account, it will keep track of their deposits and how much they withdrew. In order to have it organized from person to person (senior, adult, student), the users will be organized with their address, name, the phone number, and their account id.

2.     (10 pts) Detailed use cases for all the scenarios you imagine the software will be used. Look at the phone directory example in the text book (section 1.5) to guide you with this.

**When the user wants to add to the checking (deposit)**

User inserts command to deposit

system prompts for the name

user enters name

     system prompts for the account id

user inputs id

system asks for checking (deposit amount) or savings (deposit amount)

user enters checking

system displays the user its deposits

user inputs the amount of money to deposit

     system prompts the user to conform the amount of money they deposit

user enters yes to conform

system will add the transaction, system with then add the amount of money to the checking, it will display the new amount that the user has saved up.

Program ends

**When the user wants to add to the savings (deposit)**

User inserts command to deposit

system prompts for the name

user enters name

     system prompts for the account id

user inputs id

system prompts for the users checking (displays the amount) and or savings (displays the amount)

user inputs saving

     system prompts the user to conform the amount of money they will deposit

user enters yes to conform

system will add the transaction, system with then add the amount of money to the savings account, it will display the new amount that the user has saved up.

Program ends

**When the user wants to withdraw from checking**

 User inserts command to withdraw

system prompts for the name

user enters name

     system prompts for the account id

user inputs id

system prompts for the users checking (displays the amount) and or savings (displays the amount)

user inputs checking

system displays the withdraw

user input the amount of money

system prompts the user to conform the amount of money they will be withdrawn from the checking

user enters yes to conform

system will add the transaction to the old array, system will then display the fee for withdrawing from checking, it will display the new amount that the user has left.

Program ends

**When the user wants to withdraw from savings**

 User inserts command to withdraw

system prompts for the name

user enters name

     system prompts for the account id

user inputs id

system prompts for the users checking (displays the amount) and or savings (displays the amount)

user inputs savings

system displays the withdraw

user input the amount of money

system prompts the user to conform the amount of money they will be withdrawn from the savings

user enters yes to conform

system will add the transaction to the old array, system will then display the fee for withdrawing from savings, it will display the new amount that the user has left.

Program ends

**When a new user wants to add a new account**

user inserts command to add account

            system asks for name

user inputs name

            system ask for phone number

users inputs number

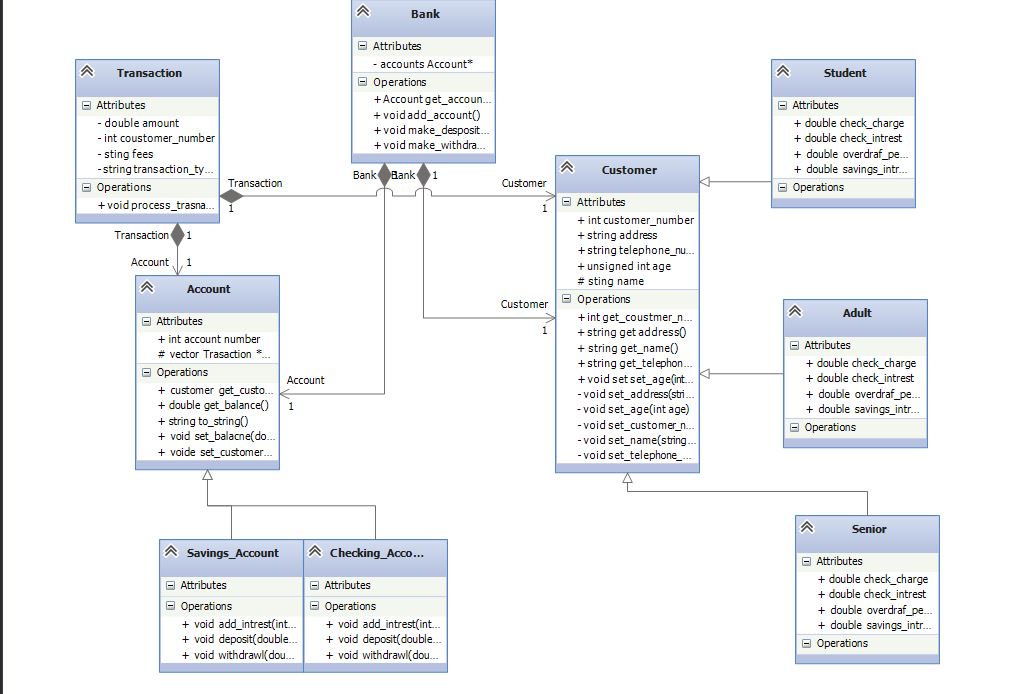
            system ask for address

user inputs address

            system displays users new account number

program ends

3. (10 pts) UML diagrams to describe the relationship between the classes described in the problem. You do not need to describe any sequence diagrams unless you wish too.



4. (10 pts) Pseudocode (see page 106 in your book for an example) to describe how the following methods will be implemented. Note, you will need to implement more code than is described here…

\* Add\_Account in Banking\_Application.cpp

* Asks the user for their information and then it will store it
* It will look through the users information
* It asks the user for the type of customer they are (senior, adult, student)
* It will look through the users information
* It will then create a new customer (calls the constructor)

\* make\_deposit() methods in Bank.h and Banking\_Application.cpp,

* Asks the user for their customer\_number and their account\_number
* Looks through the users information
* Prompts the user how much they want to deposit
* Uses the deposit () function

\* make\_withdrawal() in Banking\_Application.cpp

* Asks the user for their customer\_number and their account\_number
* Looks through the users information
* Prompts the user how much they want to withdraw
* Uses the withdrawl () function

\* Overloaded add\_account() methods in Bank.h

When adding a new user (public):

* Calls a constructor inorder to create a new account
* New account\_number
* Prompts the user if they want a checking or savings account
* Goes through the user’s information
* Balance is set to =0

When adding on top of the existing user(private)

* Askes for account number and type
* Gets the users information
* Prompts the user if they want a checking or savings account
* Uses get\_account() to get the account that is connect to the user to add a new an account
* Sets them a new account number (for either saving or checking which ever one they create )
* Sets balance to zero

\* get\_account() in the Bank.h

* Customer\_number
* Goes though the vector of account to find the account connected to that customer\_number
* Displays the account

5. (10 pts) Bank Data Storage Description – A description of HOW the account numbers for accounts and the customer id numbers will be generated and stored. How will accounts be linked to customers? How will transactions be linked to customers? (see Bank.h)

- Account is linked to the customers by the customers\_number (which is their identification number). When the user inputs their unique/ random number they will be identified because that specific number is linked to their account.

- the users account number will be stored under the transaction so when the specific user types in their unique identification number their transaction history will be associated with their number.;

