Names: Ronald Macmaster, Michael Marino

UT EID: rpm953, mm75343

**PA3: Shopping Cart: OOP**

**1) Analysis**

**Problem Statement:**

Implement a Shopping-Cart mechanism for an online vendor.

Users will add grocery, electronics, and clothing purchases to the shopping cart.

The program shall compute and report the total price of the shopping cart purchases from:

Base price, tax, and shipping cost for each item.

**Input:**

Transactions are input from a given file. The filename is specified through the command line.

General Transaction Format:

<operation> <category> <name> <price> <quantity> <weight> <optional field 1> <optional field2>

Valid operations: **insert, search, delete, update, and print.**

**Input is not necessarily valid!** Report errors using exceptions.

**Output:**

Print command invokes shopping cart statement.

The statement prints all the shopping cart items in order by name with its attributes.

The total charges for the entire cart will follow.

**Questions?**

Should we implement the Item price as two ints for dollars and cents? Can we use a double?

What is a system level use case diagram?

Should we create a separate class or method to read the transactions from a file?

**2) Design**

**Architecture Models:**

1. **System Use-Case Diagram**
2. **UML class diagram**
3. **ADT class description for each class.**
4. **4) functional block diagram**

**OUTPUT**

**Return Account Statement**

* Print out a compiled account statement for each customer

**PROCESS**

**Customer Bank Accounts**

* 4 bank accounts per customer
* Checking, Saving, Auto, and Student Loan

**Perform Transactions**

* Perform Transfers
* Log the action
* Handle and log errors

**Compile Account Statement**

* Put together an account statement based off final account totals
* Label the totals accordingly

**INPUT**

**Bank Account Transactions**

* Customer ID#
* Transaction Type
* [Amount]
* Account Type
* [Account Type 2]

1. **Functional Block Diagram**

getAccountString()

getCustomerName()

getCustomerAddress()

getAccountBalance()

Main

Transaction

ServiceCustomerAccount()

Customer

Withdraw()

Deposit()

depositFunds()

withdrawFunds()

transferFunds()

addAccountInterest()

getAccountBalance()

getTransactionAmount()

getTransactionType()

getAccountType1()

getAccountType2()

BankAccount

SavingsAccount

CheckingAccount

addInterest()

fine()

overDraw()

**Algorithms**

**Driver Algorithm:** (ShoppingCartManager)

1. Create bank account list / database
2. Prompt the bank teller for a transaction
3. Process and log the transaction
4. Perform the bank account transaction
5. Ask to continue. (Y/N)? Yes: repeat from 2)
6. Compile and display a bank account statement for each customer