Requirement Analysis

The aim of this project is to create a software system to be used on an Automatic Teller Machine (ATM) to allow bank customers to make basic transactions with their bank accounts. Although this type of software already exists, my goal is to make something that is flexible so that this software system could be used in many different applications.

This system needs to support the following types of transactions:

- Inquiry: Customers should be able to determine how much money is in a specified account.
- Transfer: Customers should be able to transfer money between accounts.
- Deposit: Customers should be able to deposit money into a specified account.
- Withdrawal: Customers should be able to withdraw money from a specified account.

This type of system requires a high level of security and robustness. It needs to extremely reliable so that when a customer uses this system, they can trust that their banking information will be kept safe. Also, from the banks point of view, when a withdrawal is made, only the correct amount of money is dispensed. If too much money is dispensed, the bank could potentially be losing money.

I decided to only design and implement the back end of this system to keep it flexible. The system is very pluggable, meaning different types of implementations are possible for the front end of the system. Many different front end implementations are possible with this design.

Possible front end implementations include:

- GUI Simulation (or for testing/debugging)
- CLI Simulation (or for testing/debugging)
- A physical ATM. (an ATM with physical devices such as the display, keyboard, cash dispenser, etc.)
- Different types of physical ATM's (i.e. different banks might use different hardware)

Keeping with the theme of flexibility, Java was used to implement this system. Because Java is used, this system is platform independent.

It is standard for ATM's to use a network connection to communicate with the bank, to obtain bank account information such as the customer's account balances. My system allows for different ways of implementing how the ATM communicates with the bank. It was designed this way because different banks may have different standards regarding network communication.

The devices that must be implemented with this system include:

- Card Reader: a device to read a customer's bank card.
- Cash Dispenser: a device to dispense cash. (for withdrawals)
- Console: Includes a display and keyboard.
- Envelope Receiver: a device to accept envelopes. (for deposits)
- Maintenance Panel: a device that allows the ATM to be switched on or off and allows maintenance staff to get/set information about the ATM.
- Receipt Printer: a device that prints a receipt for the customer. Records all transaction details.

A physical ATM would also have a physical network device to allow it to connect to a network.

This system should also support multiple currencies so that it could be used anywhere in the world. The system is designed to only allow one type of bank note. (i.e. 20 CAD). This could be changed in the front end implementation to handle say 20 ZAR bills.