CS 292: Final Project Summary  
Jason Bernard   
  
 For my final project I will create a bank account simulation. The problem this application will solve is money and account management. This program will contain information about the account such as the account number, checking account, savings account and a trading account. The account number will be randomly generated. The checking account will allow for manual deposit, and also a deposit slip that is read into the program. The account number on the deposit slip much match the account number that it is being deposited in to. The savings account will have a minimum balance of $5.00. The savings account will have a 0.05 APY%. The “Bank Marketplace” will be a form that will include a few CD’s for sale with their own rates. A bonus goal that I have for myself is to include simulated stock trading.   
Multiple accounts can be maintained by storing and reading from a database. Account log in screen will be utilized. Accounts can be given a name which will be used as the username for the log in screen. The passwords will be displayed for project purposes.   
  
The Object for this project will be Account. The attributes of Account include,   
accountName, accountNumber, savingsAccount, and tradingAccount.   
  
  
**OpenSavingsAccount():**

To open a savings account a deposit of $5.00 will need to be made or a transfer from the checking account.   
  
**Deposit():**   
Deposit will either take manual deposits from a text box or the maturing of a CD occurs.   
This will increment the total count  
  
**DepositSlip():**   
This method will read information from a deposit slip and checks if the correct Account Number has been entered. If it contains the correct information, then a deposit() call will be made. Otherwise an error will be displayed.   
  
**DirectDeposit():**   
A direct deposit can be set up with a user defined amount into the checking account each month.   
  
**Transfer():**   
Allow transfer of funds between savings and checking accounts.   
  
**Update():**   
Update will run after a transaction such as a withdraw or deposit has been made. The total amounts will be updated accordingly.   
**Withdraw():**  
Withdraw will subtract an amount from the total, indicating either a cash withdraw or a purchase.   
  
**NewAccount():**

NewAccount will prompt to save the open account. The form will be cleared and a new Account will be started.

**GenerateAccountNumber():**This method will run when a new account has been created. This will be a randomly generated number and the requirements for the deposit slip will be changed accordingly.

**AdvanceMonth():**This will advance time to the next month and increment the total, savings and trading accounts accordingly. The total will be incremented by direct deposit (if set up) savings will be accrue interest every 12 months. The trading account will increment based on the CD purchased.   
  
PrintReceipt():   
  
Allow a receipt to be created as a separate file. A receipt can be requested after a transfer or purchase.

**Exit():**Close the application**.   
  
Requirements:**1. Allow opening of savings account.

2. Allow manual deposits.

3. Read deposit slip document  
4. Write receipt document

5. Calculate interests

6. Advance the date.

7. Tranfer funds

8. Purchase CDs   
9. Mature CDs

10. Store & read account information to database.