Kenneth Robinson

MIS 320 - Assignment 2

February 1, 2011

# CONTROLS LIST

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FORM** | **CONTROL** | **EVENT** | **PSUEDOCODE** | |
| Login | lblName |  |  |  |
|  |  |  |  |  |
|  | lblAddress |  |  |  |
|  |  |  |  |  |
|  | lblEmail |  |  |  |
|  |  |  |  |  |
|  | lblProgramStatus |  |  |  |
|  |  |  |  |  |
|  | lblPhone |  |  |  |
|  |  |  |  |  |
|  | tbxName |  |  |  |
|  |  |  |  |  |
|  | tbxAddress |  |  |  |
|  |  |  |  |  |
|  | tbxEmail |  |  |  |
|  |  |  |  |  |
|  | tbxPhone |  |  |  |
|  |  |  |  |  |
|  | btnSubmit | click | create MyCustomer class | |
|  |  |  | check phone number | |
|  |  |  | if phone number is in valid format (numeric only) | |
|  |  |  |  | saveCustomer |
|  |  |  |  | clear textboxes |
|  |  |  | else |  |
|  |  |  |  | display error message box |

# CLASS DESIGN

FormCustomerEntry

* Submit
* Check Phone
* Class MyCustomer

CustomerDataAccess

* filePath
* customerNameDAO
* customerAddressDAO
* customerEmailDAO
* Committed
* Gets/sets
* Save Customer

MyCustomer

* customerName
* customerAddress
* customerEmail
* customerPhone
* committed
* errorInfo
* gets/sets
* save Customer
* class CustomerDataAccess

checkPhoneNumber

* phoneNumber
* validSubmission
* errorMessage
* new()
* checkPhoneNumber()
* getValidSubmission()
* getErrorMessage()

# CLASS DEFINITIONS

1. Class FormCustomerEntry (User Interface layer)
   1. Description
      1. User Interface layer; does error checking on form
   2. Members
      1. none
   3. sub btnSubmit\_Click
      1. create new instances of Customer Information class
      2. Set variables in formCustomerEntry to variables in CustomerInformation class
      3. Check phone field
      4. If there are no errors with phone field, then call CustomerInformation.SaveCust
      5. If there are errors do not save customer
   4. checkphone
      1. get the length of the phone number; if too short or too long or not numeric, then display error message
2. class CustomerInfo (Business Logic layer)
   1. Definition
      1. Gets customer info from the UI moves it to the DA
   2. Members
      1. CustomerName
      2. CustomerAddress
      3. CustomerEmail
      4. CustomerPhone
      5. Committed
      6. errorInfo
   3. Methods
      1. saveCust
         1. set variables in CustomerInfo class to CustomerDAO class
         2. call CustomerDAO.saveCustDAO
         3. if the commit was not successful, display error message; otherwise display saved message
3. Class DataAccess (Data Access Layer)
   1. Definition
      1. Saves the customer info in the BL to a CSV file.
   2. Members
      1. filePath
      2. customerNameDAO
      3. customerAddressDAO
      4. customerEmailDAO
      5. customerPhoneDAO
      6. committed
   3. Methods
      1. saveCustDAO
         1. check if file already exists
         2. try to create new file (StreamWriter)
            1. if the file doesn’t exist write the header; then write the form information to the text file.
            2. Change the committed status
            3. Close the file
         3. If there was an exception, throw the message and report commitment error
4. Class **checkPhoneNumber**
   1. phoneNumber - variable that holds the phone number that is entered
   2. validSubmission – variable that holds whether the submission is valid
   3. errorMessage – variable containing the error message text
   4. new(inputPhoneNumber) – overloaded class constructor
      1. sets the argument to the phoneNumber
   5. checkPhoneNumber() - checks the phone number string
      1. removes spaces out of the phone number
      2. If the phone number length is not zero and if it is not ten-digits or is not numeric
         1. validSubmission = false and add error message about invalid phone number
   6. getValidSubmission() – returns the validSubmission variable
   7. getErrorMessage() – returns the error message