**SCI – Proyecto Utilitario de Transacciones**

**Objetivo:** SCI es una compania que se dedica al proceso de ordenes en linea (Shopping Carts), uno de sus clientes requiere que las transacciones sean enviadas cada noche por correo indicando cuantas ordenes y los particulares de cada orden.

**El Problema:** es que la informacion llega de dos diferentes sistemas (uno el shopping cart) y el otro el proceso de Autorizacion de Compras (Authorize.Net).

Cada noche un empleado debe de colectar todas las ordenes del shopping cart y con la informacion recibida por email de Authorize.Net, proceder a compilar la informacion en un archivo de texto que debe ser enviado al cliente.

**De la informacion:**

1. Las transacciones (informacion del comprador, codigo de autorizacion, monto pagado, tarjeta de credito usada), llega por email a SCI, un email por cada transaccion diariamente. (ver archivo 1 adjunto)
2. Las ordenes (con informacion del comprador, informacion de los productos comprados y otros) se encuentra en la base de datos del shopping cart y es extraida diariamente como archivo de texto. (ver archivo 2 adjunto)

**Del proceso manual:**

Cada noche un empleado compila todos los emails llegados y extrae en un archive de texto la informacion necesaria (transacciones). Abre el archive de texto extraido del shopping cart (conteniendo las ordenes), procede a comparar el nombre del comprador, monto y direccion, encuentra la informacion en las transacciones y procede a copiar la informacion de transacciones en el archive de texto de las ordenes. Asi continua hasta que termina con todas las transacciones y ordenes, no debe quedar ninguna de ella huerfana. Es decir no transacciones sin ordenes o viceversa.

Una vez que cheque por ultima vez el document generado procede a enviarlo por email al cliente (cada noche). El proceso dependiendo del numero de ordenes en el dia puede tomar horas.

**Del proceso automatizado (requerimientos):**

1. El sistema debe ser capaz de contener la informacion de las transacciones extraidas de cada email recibido.
2. El sistema debe ser capaz de contener la informacion de las ordenes extraidas del sistema shopping cart.
3. El sistema debe ser capaz de leer ambas piezas de informacion y hacer un cross – reference (referencia cruzada) de las ordenes con respecto a las transacciones y poner ambas piezas de informacion (multiples campos) en un nuevo archivo que sera enviado por email eventualmente.
4. El sistema debe ser capaz de determinar que no existen ordenes o transaciones que no hayan hecho match (huerfanas), cada dia # de ordenes igaul # de transacciones.
5. El sistema debe ser capaz de presentar una vista preliminary del archive resultado para una inspeccion visual por el empleado de la compania.
6. El sistema debe ser desarrollado en .NET (C#) en modo web para poder ser publicado en los servidores web de SCI.
7. El sistema debe ser entregado en codigo fuente para poder ser modificado de ser necesario.
8. La presentacion del sistema debe ser basica, no se necesita graficos or disenos especiales debido a que es una herramienta interna.
9. Las etiquetas del sistema deben estar en Idioma Ingles. Los comentarios insertos en el codigo fuente pueden estar en Ingles o Espanol.
10. La definicion de los controles del sistema (cajas de texto, botones, etiquetas, etc) se dejan al criterio del desarrollador, sin embargo deben ser entendidos y entendibles (self explanatories).

**Se adjunta archivos de referencia:**

Archivo 1 (email recibido cada dia por cada orden)

Archivo 2 (ordenes extraidas del shopping cart system)

Archivo 3 (resultado a ser enviado al cliente)

Una vez que se encuentre al posible desarrollador, necesito tener una conversacion para explicarle los campos para referencia cruzada.