



ESCUELA DE INGENIERÍA  
Práctica Profesional

**Migración de aplicaciones**

**PBC (Power Builder Conversion )**

Empresa LLBean Costa Rica

Proyecto Final

Pablo Rodríguez González

III CUATRIMESTRE 2007

## Índice

Perfil de la empresa:.....	3
Descripción del proyecto .....	4
Justificación del Proyecto .....	5
Objetivo General.....	6
Objetivos específicos .....	6
Resultados esperados .....	7
Estrategia Metodológica .....	8
Interfaces.....	8
Requisitos de desarrollo .....	8
Requisitos tecnológicos .....	9
Seguridad .....	9
Características generales.....	10
Plan de Trabajo .....	11
Cronograma de Actividades.....	12
Población Beneficiaria .....	13
Análisis de Riesgos .....	14
Modelo de diseño.....	15
Clases.....	15
Aplicación PCM.....	15
Aplicación DNIS.....	19
<i>Interfaces de Usuario</i> .....	22
Aplicación PCM.....	22
Aplicación DNIS.....	24
Componentes.....	26
Relación Capa Sesión – Capa Servicio.....	26
Relación Capa Interfaz – Capa Sesión.....	27
Diseño de Base de Datos.....	28
Aplicación PCM.....	28
Aplicación DNIS.....	28
Diagrama de Clases de PCM.....	29
Diagrama de Clases de DNIS.....	29
Conclusiones .....	30
ANEXOS .....	31
Anexo 1 .....	31
Historias de PCM.....	31
Anexo 2.....	57
Glosario de términos.....	57
Anexo 3.....	59
Metodología XP (Extreme Programing).....	59

**Perfil de la empresa:**

LLBean Inc es una empresa que se dedica a la producción y venta de ropa e implementos para actividades al aire libre durante mas de 90 años. Fundada en 1912 por Leon Leonwood Bean la compañía pasó de ser operada por una sola persona a convertirse en una organización global con ventas anuales por mas de \$1.5 billones. Las oficinas centrales se encuentran en Freeport Maine USA.

La oficina en Costa Rica se inicio con un departamento de control de proveedores hasta crecer a control de calidad, mercadeo, análisis de datos y actualmente también hay uno de software. El departamento de sistemas de la empresa emplea a mas de 300 personas, de las cuales 15 trabajamos en Costa Rica en el desarrollo de aplicaciones Web. La oficina se ubica en el Centro Corporativo Plaza Roble en Guachipelín de Escazu.

## **Descripción del proyecto**

El proyecto consiste en la migración de dos aplicaciones desarrolladas previamente en Power Builder a un ambiente Web utilizando el lenguaje y tecnología Java, además de DB2 como sistema de base de datos.

Las aplicaciones que se migraran son:

**DNIS (Dial Number Identification System)** esta aplicación realiza una asociación entre los códigos de promoción y el número de teléfono que brinda L.L. Bean Inc. en el catálogo al cliente. Por medio de este sistema, brinda información a la sección de Advertising como: cuantos catálogos son requeridos, por códigos de promoción, mediante los teléfonos. Estudiando la respuesta de las promociones, le permite a la empresa a determinar, cuál promoción fue más solicitada por el cliente; cuales publicaciones de catálogos son mejores que los otros publicados; y entre otros. Toda esta información brindada por el sistema, ayuda a la empresa, a determinar la mejor manera posible de asignar dinero aquellas campañas de promociones, que obtuvieron un mejor desempeño.

**PCM (Product Maintenance Category)** permite la asignación o reasignación de Grupos de Mercadeo, Centros de Demanda, Clasificaciones de mercadeo a las categorías de productos. El usuario es el responsable de decidir cual categoría de producto debe ser asignada a cada combinación.

Las funciones principales dentro de este proyecto serán las de Ingeniero de Configuración (Configuration Engineer ), las cuales serán preparar y soportar toda la plataforma tecnológica a nivel local, base de datos, herramientas de Integración Continua utilizando herramientas como SVN Tortoise, Clear Case de IBM, Ant, WAS (Websphere Application Server de IBM), colaboración en los procesos de calidad , configuración de los ambientes y verificación y revisión del código fuente cuando la implementación al ambiente de calidad y producción así lo ameriten.

## **Justificación del Proyecto**

Este proyecto representa un paso más para la empresa en la migración de una aplicación hacia una plataforma Web robusta, la cuál permite que pueda ser mejora con mayor facilidad en el futuro dentro del corredor tecnológico de la compañía.

El problema consiste en tener dos aplicaciones (PCM y DNIS) implementadas con Power Builder, la cual es una tecnología con la que en el pasado se implementaron sistemas internos de L.L. Bean Inc.

Los impactos que surgen al tener estas aplicaciones desarrolladas con Power Builder son:

- a. Power Builder, es una tecnología que desde el punto de vista de los intereses de L.L. Bean Inc. se ha vuelto obsoleta.
- b. Power Builder no se ajusta a los estándares de la empresa actuales. Hoy en día el estándar de la empresa es Java/Web.
- c. Ambos programas son aplicaciones de escritorio, y por ello no aprovechan todo el potencial que puede derivarse de una aplicación orientada a Web.

Este problema impacta a L.L. Bean Inc. en términos generales y en específico al departamento de marketing, que será el usuario directo del mismo.

Adicionalmente me permite desarrollar mis conocimientos y adquirir experiencia en un ambiente laboral muy competitivo y utilizando herramientas de ultima tecnología.

## **Objetivo General**

Realizar la migración de una aplicación desarrollada en Power Builder a un ambiente Web, abarcando además un cambio completo en el sistema de base de datos.

## **Objetivos específicos**

- Utilizar la metodología del XP Programming para realizar la migración de las aplicaciones, por medio de iteraciones , programación en parejas y pruebas continuas bajo un ambiente de integración continua.
- Desarrollar el sistema sobre el lenguaje Java, cumpliendo así el objetivo de que sea un ambiente Web robusto.
- Instalar y configurar un ambiente de pruebas local con el nuevo sistema de base de datos DB2, requerido por el nuevo sistema.
- Mejorar el modelo de datos actual con los cambios que se requieran y adecuen para el sistema en desarrollo.

## **Resultados esperados**

Para que el proyecto se considere exitoso se han impuesto los siguientes objetivos a cumplir:

### *Metas del negocio*

- El nuevo sistema debe tener la misma funcionalidad que el sistema existente, con la excepción de las funciones que no se utilizan.
- La experiencia del usuario en el nuevo sistema deber ser muy similar al del ya existente.

### *Metas de IS*

- El usuario debe notar que la aplicación le esta dando valor agregado a su trabajo.
- Se toma como tiempo máximo de carga de las páginas 3 segundos.
- El proyecto debe ser implementado para producción en 16 semanas

## **Estrategia Metodológica**

Este proyecto se lleva a cabo utilizando la técnica de XP Programming, para el desarrollo directo de las aplicaciones se llevara el control por medio de iteraciones donde en cada iteración se asigna el trabajo y el objetivo que se debe cumplir, que se conoce como historia. Las historias las crean el Analista del Negocio y el cliente, estas deben pasar por un proceso de pre-análisis por parte de lo desarrolladores para determinar el alcance de la historia y del tiempo que se va a llevar terminarla.

Una vez que este tiempo se ha establecido se inicia el desarrollo utilizando las herramientas dispuestas para ese fin, durante este tiempo también se realizan las pruebas de calidad a nivel local, por medio de scripts, las revisiones del negocio se hacen cada vez que una historia se termina y se da por aprobada cuando el grupo de calidad confirma que no posee errores y que el negocio asegura que cumple con sus requerimientos.

## **Interfaces**

Las interfaces gráficas, que en ambas aplicaciones son de tipo Web, deben hacer énfasis en el hecho de ser amigables con el usuario y estéticamente agradables. La ubicación de los diferentes componentes debe permitir el despliegue completo y cómodo de la información, así como una navegación sencilla a través de la misma. Los colores y estilos de las interfaces deben ser tomados de las clases de estilos ya definidos previamente, ello para que se ajusten a los estándares de la empresa.

## **Requisitos de desarrollo**

Las páginas web deben ser JSPs y la comunicación con los servlets la deben realizar por medio de AJAX, ello para lograr la mayor eficiencia posible.

Los SQL deben estar contenidos en Stored Procedures para DB2. Todo Stored Procedure debe ser evaluado para ver su eficiencia con Explains.

El desarrollo debe estar hecho en capas según los estándares de L.L. Bean Inc.



En la capa de servicio (la encargada de la conexión con la base de datos) se debe usar el patrón de diseño Factory. A la hora de instanciar esta capa debe utilizarse la tecnología Reflection de Java.

Deben utilizarse Servlets para realizar la comunicación entre el cliente y el servidor. Siempre toda petición al servidor debe pasar por un Servlet central llamado el controller, éste recibe toda petición y la re-direcciona a otro Servlet, quien es el verdadero destinatario.

Se debe contemplar el manejo de Soft Exceptions, lo cual es un estándar de la compañía, para todos aquellos datos que requieran validación según las reglas del negocio.

Se deben manejar todas las excepciones de tipo grave (llamadas en la empresa Hard Exceptions), de tal manera que se muestre una pantalla específica cuando una de estas suceda. Esta pantalla y la información que muestra está definida en los estándares de L.L. Bean.

Las clases deben contar con testing de JUnit. Estas clases de testing han de ser programadas por los desarrolladores. Se exige que mínimo un 93% del código cuente con clases de testing.

### **Requisitos tecnológicos**

El motor de base de datos a utilizar es DB2, de IBM; con Quest Central v. 4.9. El buscador Web a utilizar es el Internet Explorer de Windows. Como lenguaje de programación se va a utilizar Java v. 1.4., con el entorno de desarrollo MyEclipse 5.5.1.

### **Seguridad**

Las aplicaciones deben estar integradas a la infraestructura de aplicaciones Web de la compañía, y por lo tanto únicamente deben poder ser accesadas por medio de la misma. El acceso a las aplicaciones es posible únicamente si se cuenta con el debido

nombre de usuario y password, los cuales son asignados a los empleados que cuenten con los permisos debidos.

### **Características generales.**

Las dos aplicaciones a desarrollar deben seguir todos los estándares de la compañía. En concreto los estándares referentes a la manera de hacer el código: nombres de variables, clases, comentarios, etc. Para comentar cada método debe usarse JavaDoc, de manera que se dé una breve descripción de cada método y se enuncien sus parámetros y el valor de retorno.

Las aplicaciones deben estar integradas a la infraestructura de aplicaciones Web de la compañía, por lo tanto No deben funcionar por separado como aplicaciones aparte, sino como parte de un gran conjunto.

Todo el código, comentarios y texto de la interfaz debe estar en idioma inglés.

La interacción con la base de datos siempre se va a hacer por medio de Stored Procedures.

Todo el desarrollo de las aplicaciones es tipo Web, y específico para que las aplicaciones se usen con el Internet Explorer de Windows.

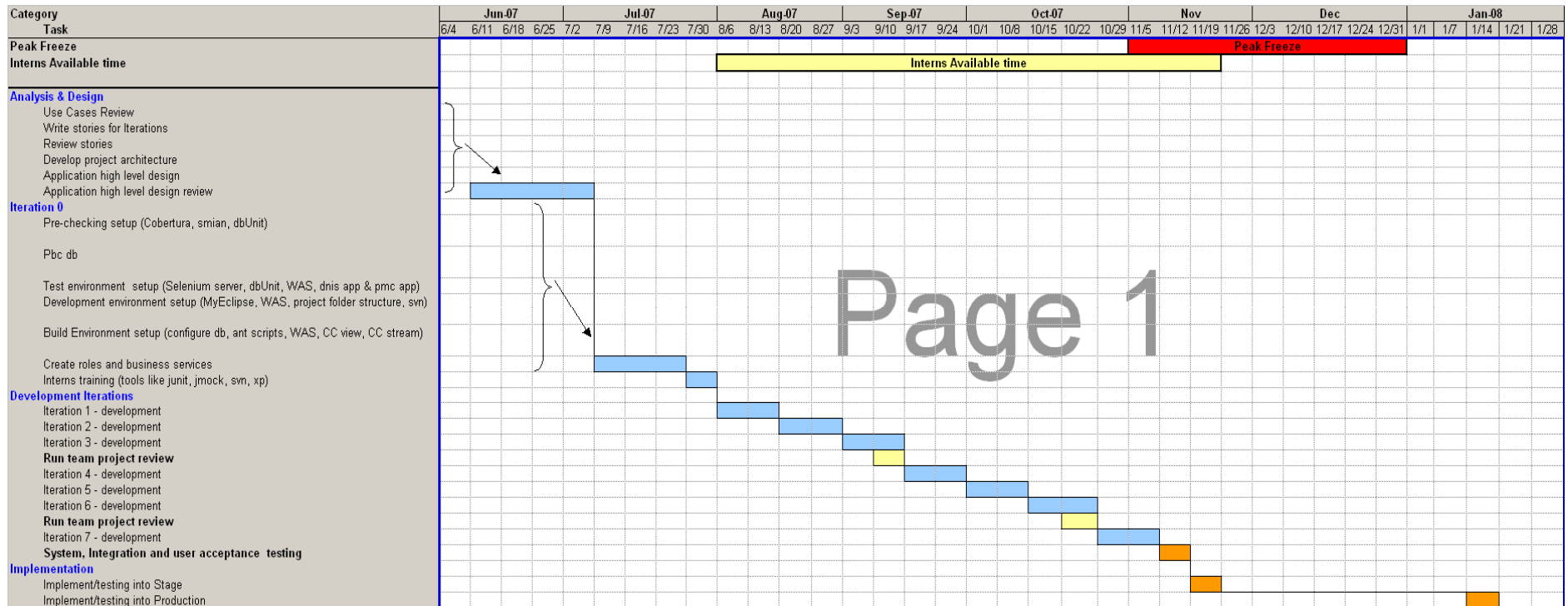
Ambos programas deben haber sido sometidos a control de calidad. Esto implica que hayan superado las pruebas a las que son sometidos con Selenium y contar así mismo con testing de JUnit.

## Plan de Trabajo

El proyecto se realizará en las oficinas de LLBean Costa Rica y la empresa se encarga de proveer el equipo técnico necesario así como las herramientas de desarrollo. El recurso humano que se empleará en el proyecto es el siguiente:

<b>Rol</b>	<b>Ubicación</b>	<b>Responsabilidades</b>
Patrocinador	Freeport	<ul style="list-style-type: none"> <li>- Aprueba el alcance, dirección y objetivos del proyecto</li> <li>- Revisa, aprueba y soporta los procesos del negocio.</li> <li>- Participa en las reuniones</li> <li>- Resolver los problemas que requieren asistencia del patrocinador.</li> <li>- Comunicar claramente las expectativas del proyecto.</li> </ul>
Contacto Experto del Negocio,	Freeport	<ul style="list-style-type: none"> <li>- Desarrollo de procesos para el acceso a los sistemas.</li> <li>- Resolver problemas relacionados al negocio.</li> <li>- Aprueba las historias a nivel de Negocio.</li> <li>- Participa en las pruebas de los usuarios.</li> <li>- Aprobación final de la aplicación antes de la implementación</li> <li>- Participante activo durante la implementación</li> </ul>
Líder de Proyecto	Costa Rica	<ul style="list-style-type: none"> <li>- Desarrollo de los documentos iniciales del proyecto.</li> <li>- Dirige y actualiza el plan del proyecto.</li> <li>- Resolver los problemas del proyecto.</li> <li>- Fortalece la comunicación entre los miembros del equipo</li> <li>- Crear el plan para la implementación.</li> </ul>
Líder Técnico	Costa Rica	<ul style="list-style-type: none"> <li>- Definir la arquitectura técnica de los dos sistemas.</li> <li>- Crear las historias y su respectiva explicación.</li> <li>- Líder Técnico de los desarrolladores</li> <li>- Colaborar con el equipo de calidad en la traducción de los requerimientos para las pruebas de los casos</li> </ul>
Desarrolladores	Costa Rica	<ul style="list-style-type: none"> <li>- Desarrollar las aplicaciones DNIS &amp; PCM de manera iterada.</li> <li>- Seguir todos los estándares de programación y las prácticas de XP</li> <li>- Participar en la integración de las pruebas y en la implementación de los sistemas</li> </ul>
Analista del Negocio	Costa Rica	<ul style="list-style-type: none"> <li>- Definir el estado actual y los requerimientos por medio de casos de uso.</li> <li>- Aprobación de las historias</li> <li>- Participar en las pruebas de los usuarios.</li> <li>- Participar en la implementación.</li> <li>- Participar en la revisión de los punto que se requieran.</li> </ul>
Analista de Calidad	Costa Rica	<ul style="list-style-type: none"> <li>- Desarrollar un plan de pruebas para las aplicaciones DNIS &amp; PCM</li> <li>- Definir todos las pruebas de los casos.</li> <li>- Desarrollar todos los scripts de pruebas.</li> <li>- Proveer aprobación de calidad a todas las historias.</li> <li>- Capturar errores y dar retroalimentación a los desarrolladores.</li> <li>- Participar en la implementación.</li> </ul>
Ingeniero de Configuración	Costa Rica	<ul style="list-style-type: none"> <li>- Establecer el software de control de las configuraciones (SVN)</li> <li>- Desarrollar los scripts para el trabajo del checkin y pre-checking</li> <li>- Establecer y soportar el Wiki Interno como centro de documentación y además el ambiente de captura de problemas o acontecimientos.</li> <li>- Desarrollar la estructura de integración continua de las aplicaciones (Continues Integration)</li> <li>- Establecer las relaciones en el ambiente de desarrollo local y los estándares y aplicaciones en casa matriz.</li> <li>- Instalar las aplicaciones en el ambiente de pruebas diarias.</li> <li>- Preparar las aplicaciones para su instalación en el ambiente de calidad y producción.</li> </ul>
Seguridad	Freeport	<ul style="list-style-type: none"> <li>- Colaboración y soporte en la definición de la seguridad del proyecto dentro de la infraestructura de la compañía</li> </ul>

## Cronograma de Actividades



## **Población Beneficiaria**

Los beneficiarios directos son los usuarios del departamento de marketing, para quienes el desarrollo de aplicaciones bajo el estándar de la empresa les permitirá tener a disposición un mejor soporte y una aplicación más robusta.

## **Análisis de Riesgos**

Es importante destacar que este proyecto no es tan amplio, por este motivo, no fue necesario realizar un análisis de riesgos. Estas aplicaciones son necesarias para el uso interno de la empresa, sin embargo no comprometen de una forma negativa la operación adecuada de la empresa L.L. Bean Inc.

## Modelo de diseño

Esta sección de diseño se subdivide en dos secciones, la primera corresponde al primer proyecto realizado, este es PCM; la segunda corresponde al segundo proyecto, este es DNIS.

### Clases.

Aplicación PCM.

El diagrama, de tipo UML de todas las clases de la aplicación, nos muestra una diagrama a tomar en cuenta, en el momento de iniciar a codificar el proyecto, para respetar la arquitectura establecida por la compañía. La imagen de este se encuentra en el anexo 1.3.

A continuación, se realizará una descripción breve de las capas, con sus clases.

#### a. Plain Old Java Objects

ProductCategoryColumnInfo: esta clase es simplemente encargada de contener y mostrar información que representa una fila de la tabla en la base de datos. Contiene ocho variables que representan cada una de las columnas de la base de datos, y una última variable que para indicar si el estado de los datos ha sido modificado o no.

Además, posee métodos simples que permiten fijar o obtener los valores de cada una de esas variables.

ProductCategoryInfo: esta clase es responsable de contener la información de un par de instancias de la clase de ProductCategoryColumnInfo, y representa la información que el usuario de la aplicación puede ver al mismo tiempo en una pantalla. En concreto contiene una columna llamada “frozen” y otra de tipo “primary”.

## **b. Capa de Servicio**

ProductCategorySvc: es la más importante en la capa de servicio. En la misma se encuentran los métodos que permiten realizar operaciones relacionadas con la base de datos, no obstante la conexión a la base de datos no se realiza directamente en esta clase.

La clase contiene dos métodos principales, que a su vez llaman a los métodos que finalmente se conectan en la base de datos, los cuales se encuentran en la clase llamada UnitOfWork.

El primer método notable es “save”, este método es responsable de salvar los cambios existentes en la colección.

El Segundo método de importancia es “find”, el mismo se encarga de retornar una colección cuyos valores satisfacen las restricciones de búsqueda que reciba.

PCMUnitOfWork: esta clase extiende de la clase llamada UnitOfWork, y es la responsable de llamar a los stored procedures que permiten realizar las actualizaciones y las búsquedas. En esta clase los nombres de los stored procedures son conocidos.

Dentro de las clases implementadas es la de más bajo nivel en relación con la base de datos.

ProductCategoryServiceFactory: esta es una clase muy peculiar, la misma hace uso de una tecnología de Java llamada Reflection. Por medio de Reflection es posible crear instancias de una clase sin usar la palabra reservada “new”.

Esta clase sólo posee un método, éste método es responsable de instanciar a la clase llamada ProductCategorySvc. De esta manera cuando desde cualquier punto de la aplicación se desea crear una instancia de la clase de servicio, se llama a este método. De ahí el nombre de “Factory”.

PCMDataSourceFactory: esta es una clase pequeña y con una función específica. El objetivo de haber creado esta clase es poder obtener “DataSources”. Con esta clase es posible crear “DataSources” de dos tipos: uno que es para testing, y el otro que es para la aplicación real.

En el caso de los “DataSources” para testing se crean utilizando la clase “JdbcConnectionFactory”. Por otro lado, en el caso de la creación del otro tipo de



“DataSources” estos se crean con la clase “LLBJdbcConnection”, la cual es parte de la infraestructura empresarial de L.L. Bean.

JdbcConnectionFactory: esta clase es utilizada para definir los valores de algunas variables muy importantes que se necesitan para lograr una conexión exitosa con la base de datos. Obtiene los valores de las variables sacándolos de un archivo específico, con esto se logra otorgarle una mayor versatilidad a la aplicación. Esta clase es usada para crear los “DataSources” que se utilizan en los test.

### **c. Capa de Sesión**

ProductCategorySessionFactory: al igual que la clase homóloga en la capa de servicio, esta es una clase muy peculiar, la misma hace uso de una tecnología de Java llamada Reflection. Por medio de Reflection es posible crear instancias de una clase sin usar la palabra reservada “new”.

Esta clase sólo posee un método, éste método es responsable de instanciar a la clase llamada ProductCategory. De esta manera cuando desde cualquier punto de la aplicación se desea crear una instancia de la clase de servicio, se llama a este método. De ahí el nombre de “Factory”.

ProductCategory: es encargada de llamar los métodos llamados “find” y “save”, encargados en último término de conectarse a la base de datos. Asimismo, en esta clase se encuentran los métodos encargados de realizar validaciones a los valores que ingrese el usuario a la hora de realizar modificaciones en los datos que se muestran en la pantalla.

Todas estas validaciones que se realizan en la capa de servicio se realizan por medio del uso de “soft exceptions”, este es un estándar de la compañía para manejar todo tipo de validaciones que se refieran a reglas del negocio.

### **d. Servlets**

PCMControllerServlet: este servlet es necesario para poder cumplir con las prácticas de la empresa. Tiene la función de recibir todas las peticiones de comunicación hacia otros servlets y redirigirlas a estos.

PCMSaveServlet: este servlet se encarga de realizar las acciones referentes a guardar cambios que hayan sucedido en la colección a la base de datos. Es además responsable de validar que los datos de la página actual sean correctas y puedan ser guardados en la base de datos. También tiene la función de construir un mensaje con formato JSON que le va a informar a la página Web sobre el resultado de la inserción.

PCMShowServlet: este servlet es el encargado de realizar todas las funciones referentes a mostrar datos en la pantalla. Cada vez que el usuario desea ver los resultados de una nueva búsqueda, o ver el anterior o previo registro este es el servlet que se llama.

Asimismo, este servlet tiene la capacidad para validar los datos que el usuario modifique en la pantalla. Y siempre va a construir un mensaje con formato JSON que contiene los nuevos datos a mostrar o el mensaje que indica la validación que no se cumplió.

ServletUtils: esta clase directamente relacionada a los demás servlets, contiene métodos que son utilizados en más de un servlet. Esta es una clase estática, por lo tanto nunca es instanciada.

Contiene métodos que son utilizados para construir mensajes con resultados de las validaciones, mensajes con resultados de error, y verificaciones sobre los datos de la colección.

#### **e. Utilitarios**

Constants: por razones de orden y conveniencia se cuenta con esta clase, la cual contiene una gran cantidad de variables de tipo “final”, que son obtenidas en múltiples puntos del programa.

Existen variables que, por ejemplo, corresponden a parámetros que son enviados a los servlets, o a nombres de datos que se van a mostrar en pantalla, o a mensajes que se muestran como resultado de las validaciones.

## **Aplicación DNIS.**

El diagrama, de tipo UML de todas las clases de la aplicación, nos muestra una diagrama a tomar en cuenta, en el momento de iniciar a codificar el proyecto, para respetar la arquitectura establecida por la compañía.

A continuación, se realizará una descripción breve de las capas, con sus clases.

### *a. Plain Old Java Objects*

DnisInfo: esta clase contiene las variables necesarias para representar un registro de la base de datos en la pantalla que puede ver el usuario; un detalle importante es que cuenta con una variable que indica el estado de registro, el estado del mismo puede ser modificado o no modificado por el usuario.

FilterInfo: esta clase cuenta con cuatro variables, cada una de ellas representa un filtro de búsqueda en la pantalla del usuario.

JspButtonsInfo: esta clase tiene cuatro variables, cada una de ellas es un va a representar el estado del botón que representa en la pantalla principal. El estado de estas variables puede ser falso o verdadero, significando si está activo o desactivado.

JspInfo: esta clase contiene toda la información importante para el manejo de la funcionalidad de la aplicación. Por ejemplo, contiene la colección con los resultados, así como el estado de la bandera para el gemelo, también contiene el estado de los botones en la interface, así como el contenido de los filtros de búsqueda.

### *b. Capa de Servicio*

DnisSvc: esta clase es la más importante en la capa de servicio. En la misma se encuentran los métodos que permiten realizar operaciones relacionadas con la base de datos, no obstante la conexión a la base de datos no se realiza directamente en esta capa.

La clase contiene dos métodos principales, que a su vez llaman a los métodos que finalmente se conectan en la base de datos, los cuales se encuentran en la clase llamada UnitOfWork.

El primer método notable de esta clase es “find”, este se encarga de retornar una colección con los resultados de la búsqueda.

El siguiente método de importancia es “update”, el mismo recibe una colección que ha sido modificada por el usuario y la salva en la base de datos, llamando al respectivo método en la clase de nombre UnitOfWork.

También contiene un método llamado “save”, que recibiendo una colección de valores los inserta por completo en la base de datos.

De la misma manera es importante mencionar al método de nombre “delete”, este método recibe una colección con valores que debe eliminar de la base de datos y llamando al método correspondiente en la clase UnitOfWork los elimina.

DnisUnitOfWork: esta clase extiende de la clase llamada UnitOfWork, y es la responsable de llamar a los stored procedures que permiten realizar las actualizaciones y las búsquedas. En esta clase los nombres de los stored procedures son conocidos.

Dentro de las clases implementadas es la de más bajo nivel en relación con la base de datos.

DnisServiceFactory: esta clase sólo posee un método, éste método es responsable de instanciar a la clase llamada DnisSvc. De esta manera cuando desde cualquier punto de la aplicación se desea crear una instancia de la clase de servicio, se llama a este método.

DnisDataSourceFactory: esta es una clase pequeña y con una función específica. El objetivo de haber creado esta clase es poder obtener “DataSources”. En la aplicación se crean dos tipos de “DataSources”, el tipo que se utiliza en las clases de test, y los que se utilizan en la aplicación real.

JdbcConnectionFactory: esta clase es utilizada para definir los valores de algunas variables muy importantes que se necesitan para lograr una conexión exitosa con la base

de datos. Obtiene los valores de las variables sacándolos de un archivo específico. Estos valores son utilizados en la creación de los “DataSources” que se usan para los test.

*c. Capa de Sesión*

DnisSessionFactory: esta clase sólo posee un método, éste método que es responsable de instanciar a la clase llamada DnisSession. De esta manera cuando desde cualquier punto de la aplicación se desea crear una instancia de la clase de servicio, se llama a este método.

DnisSession: esta es una clase muy importante, es encargada de llamar los métodos llamados “find”, “save”, “delete” y “update”, encargados en último término de conectarse a la base de datos.

También, esta clase realiza las validaciones usando “soft exceptions” (estándar de la compañía). En específico esta clase valida que en el caso de una nueva inserción la llave de la nueva fila a insertar no esté repetida, esto implica verificar que el “Dnis\_Id” sea único, y si no es único que no tenga conflictos con los rangos de fechas de las filas con el mismo “Dnis\_Id”.

*d. Servlets*

DNISControllerServlet: este servlet es necesario para poder cumplir con las prácticas de la empresa. Tiene la función de recibir todas las peticiones de comunicación hacia otros servlets y redirigirlas a estos.

DNISSaveServlet: este servlet se encarga de realizar las acciones referentes a guardar cambios que hayan sucedido en la colección a la base de datos.

DNISShowServlet: este servlet es el encargado de realizar todas las funciones referentes a mostrar datos en la pantalla.

DNISInsertServlet: este servlet es el que maneja las acciones necesarias cuando el usuario decide insertar un nuevo registro

ServletUtils: esta clase directamente relacionada a los demás servlets, contiene métodos que son utilizados en más de un servlet. Esta es una clase estática, por lo tanto nunca es instanciada.

*e. Utilitarios*

Constants: por razones de orden y conveniencia se cuenta con esta clase, la cual contiene una gran cantidad de variables de tipo “final”, que son obtenidas en múltiples puntos del programa.

Existen variables que, por ejemplo; corresponden a parámetros que son enviados a los servlets, o a nombres de datos que se van a mostrar en pantalla, o a mensajes que se muestran como resultado de las validaciones.

***Interfaces de Usuario.***

Aplicación PCM.

Este sistema contiene una interface de usuario principal llamada pcmMainScreen y una interface para mostrar errores de nombre LLBError. Ambas interfaces son de tipo JSP. A continuación se hace una breve descripción de dichas pantallas.

*a. “pcmMainScreen”:*

Esta interface es la página principal, en ella el usuario puede realizar búsquedas, navegar a través de los resultados, realizar modificaciones en los datos y salvarlos en la base de datos.

La página también posee código en JavaScript para el manejo de lógica de validaciones, comunicación con los servlets, manejo de los mensajes que se obtienen como respuesta con formato JSON, etc.

Esta es una imagen que muestra la interface:

PCM Application - Microsoft Internet Explorer provided by L.L.Bean

### Product Category Maintenance

MG	DC	SC	CL	Retrieve
10				Retrieve
<input type="button" value="Save"/>				

Record 1 from 42.

Prod Ctgry Grp Id: 2	Prod Ctgry Grp Id: 1
Prod Ctgry Grp Desc: FROZEN	Prod Ctgry Grp Desc: PRIMARY
Prod Ctgry Id: 8	Prod Ctgry Id: 35
Prod Ctgry Desc: WMS FTWR CSL	Prod Ctgry Desc: LL KID'S INFANT/TODDLER
Merch Grp Id: 10	Merch Grp Id: 10
Demand Ctr Id: 10	Demand Ctr Id: 10
Sub Ctr Id: 10	Sub Ctr Id: 10
Merch Class Id: 10	Merch Class Id: 10

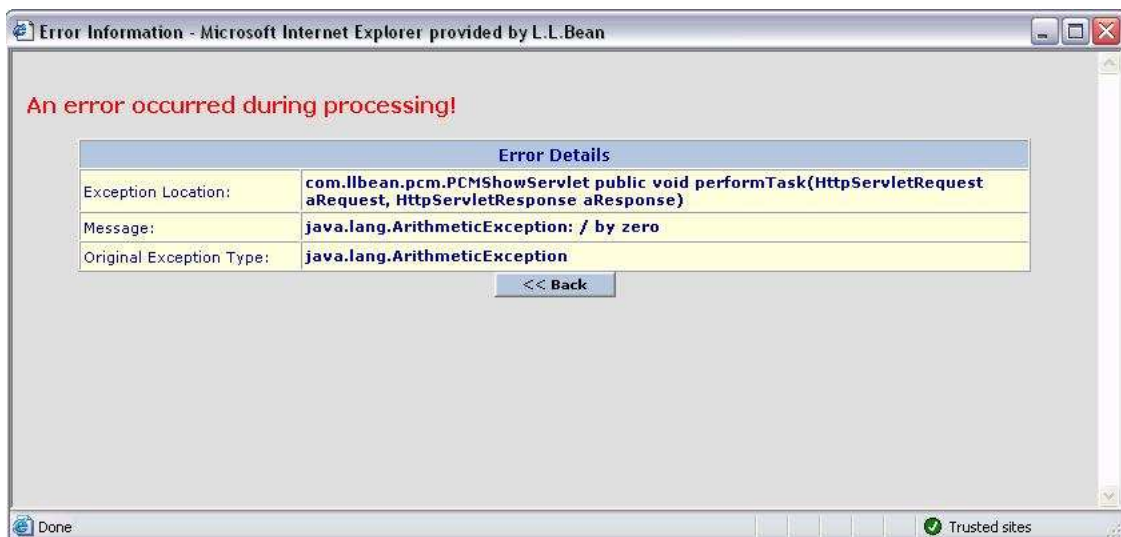
Navigation:

b. “*LLBErrorPage*”:

Esta es la página que se muestra cuando sucede un error en la ejecución de la aplicación, en la misma se muestra el tipo de error, la clase donde sucedió y el nombre del método.

Esta página corresponde a un estándar de la empresa para el manejo de errores.

Esta es una imagen que muestra cómo se vería un error causado por una división entre cero:



## Aplicación DNIS.

Este sistema contiene una interface de usuario principal llamada dnisMainScreen y una interface para mostrar errores de nombre LLBError. Ambas interfaces son de tipo JSP. A continuación se hace una breve descripción de dichas pantallas.

a. “*dnisMainScreen*”:

Esta interface es la página principal, en ella el usuario puede realizar búsquedas, ver los resultados, y al estilo de una hoja de Excel modificarlos.



La página también posee código en JavaScript para el manejo de lógica de validaciones, comunicación con los servlets, manejo de los mensajes que se obtienen como respuesta con formato JSON, etc.

Esta es una imagen que muestra, la interface:



*b. "LLBErrorPage":*

Esta es la página que se muestra cuando sucede un error en la ejecución de la aplicación, en la misma se muestra el tipo de error, la clase donde sucedió y el nombre del método.

Esta página corresponde a un estándar de la empresa para el manejo de errores.

Esta es una imagen que muestra cómo se vería un error causado por una división entre cero:



## Componentes.

Para el diseño de los componentes se estableció una categorización por capas. Cada componente ofrece una interfaz con funcionalidad para la capa inmediatamente superior, en donde a su vez, los componentes internos se encargan de proveer una funcionalidad específica y el acceso a las exteriores es mediante los contratos de la interfaz.

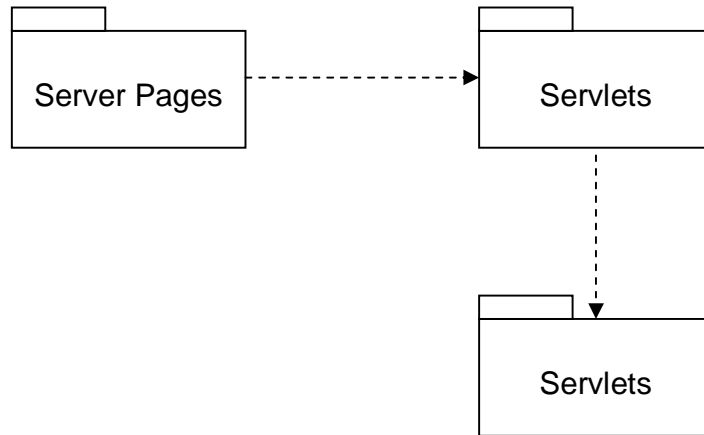
Estos componentes contienen todos los paquetes o clases necesarias, para ser efectivas las funcionalidades correctas de cada capa.

A continuación, se presenta el diagrama de componentes de las dos aplicaciones realizadas PCM y DNIS; ya que las dos soluciones cuentan con la misma estructura de componentes:

### Relación Capa Sesión – Capa Servicio



## Relación Capa Interfaz – Capa Sesión



## Diseño de Base de Datos.

### Aplicación PCM.

La base de datos, ya había sido diseñada previamente en la empresa, y cuenta únicamente con una tabla. Por el motivo, de que es cuanta con solo una tabla se proveerá, de la descripción de los campos de esta.

Los campos de esta tabla son:

<b>Nombre</b>	<b>Tipo</b>
PROD_CTGY_GRP_ID	smallint
PROD_CTGY_GRP_DESC	character(40)
PROD_CTGY_ID	smallint
PROD_CTGY_DESC	character(45)
MERCH_GRP_ID	smallint
DEMAND_CTR_ID	smallint
SUB_CTR_ID	smallint
MERCH_CLASS_ID	smallint

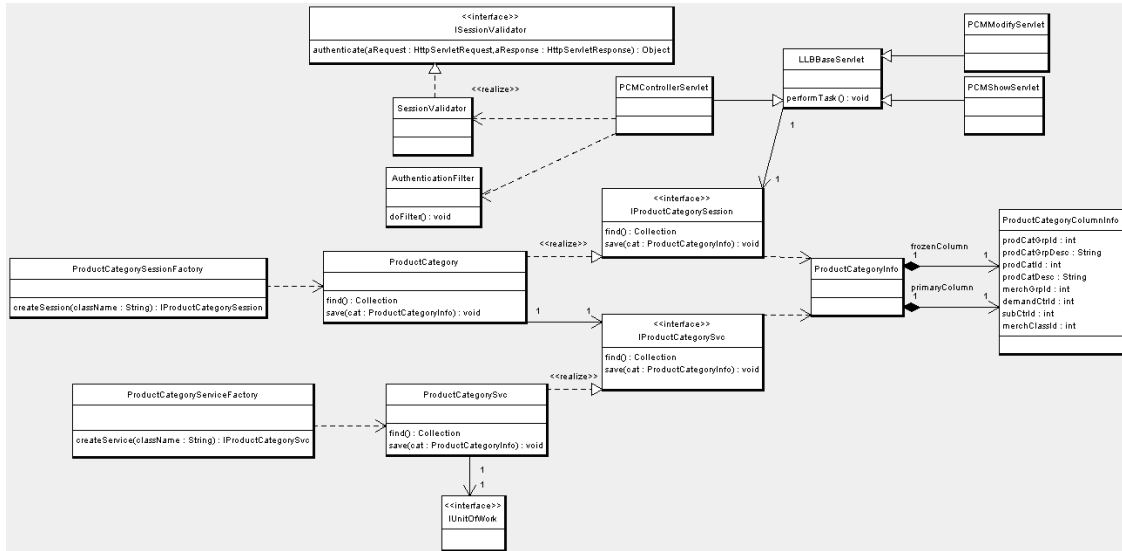
### Aplicación DNIS.

La base de datos ya había sido diseñada previamente en la empresa, y cuenta únicamente con una tabla. Por el motivo, de que es cuanta con solo una tabla se proveerá, de la descripción de los campos de esta.

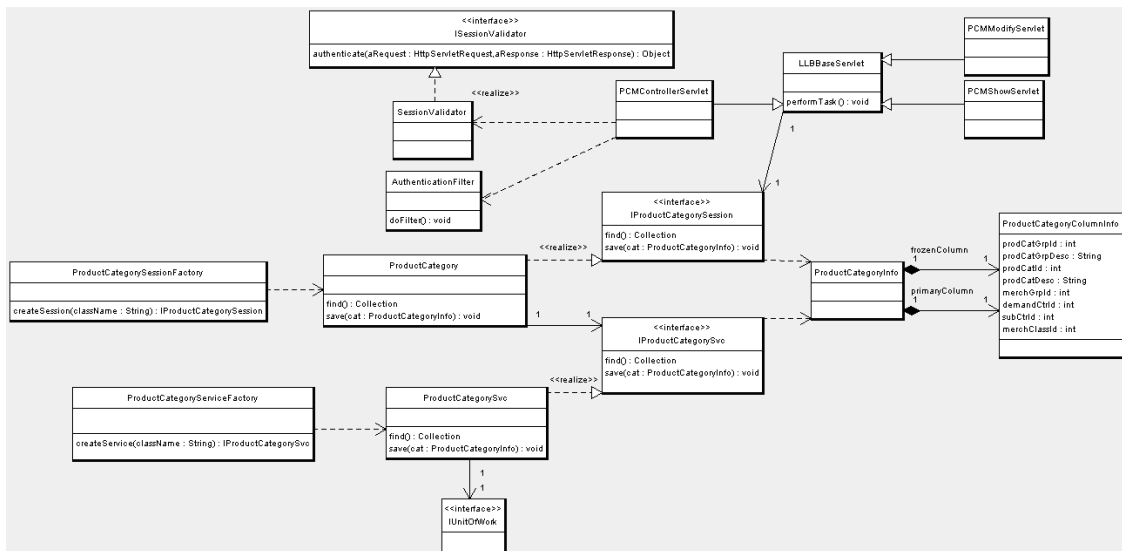
Los campos de esta tabla son:

<b>Nombre</b>	<b>Tipo</b>
DNIS_ID	character(10)
DIALED_PHONE_NBR	character(35)
SRC_CD	character(7)
SRC_CD_DESC	character(60)
EFF_DT	date
EXP_DT	date

## Diagrama de Clases de PCM.



## Diagrama de Clases de DNIS.



## **Conclusiones**

El desarrollo de este proyecto incluyó la experiencia de trabajar bajo la metodología de XP, donde la calidad de las aplicaciones se aplica en casi todas las etapas del desarrollo además de concentrarse en la comunicación con el cliente y estar listos para cualquier cambio, la documentación no es tan extensa lo que facilita cambios por parte del usuario, sin invertir tanto tiempo en el mantenimiento de documentación, la cual es poco consultada por los usuarios usualmente.

En las etapas de análisis y diseño se realizaron cuidadosamente con el fin de reducir los cambios en el sistema y el impacto que provocan los mismos. La etapa de configuración del ambiente de los desarrolladores, de la base de datos y del ambiente en general de trabajo se basó en los Standard de la compañía

## **ANEXOS**

### **Anexo 1**

#### **Historias de PCM**

##### Story 1 Display PCM Screen

###### **PLANNED ITERATION: Iteration 1**

###### **SCREEN: Product Category Maintenance**

###### **STORY TITLE**

Display PCM main screen

###### **DESCRIPTION**

As Marketing Analyst I need to enter the PCM main screen and view the header information to clearly identify the filters. Also, I need to have a message area defined for messages to be displayed and button with the options of the application.

###### **LIMITATIONS**

No changes should be persisted to the database at this point.

**ACCEPTANCE CRITERIA DESCRIPTION:**

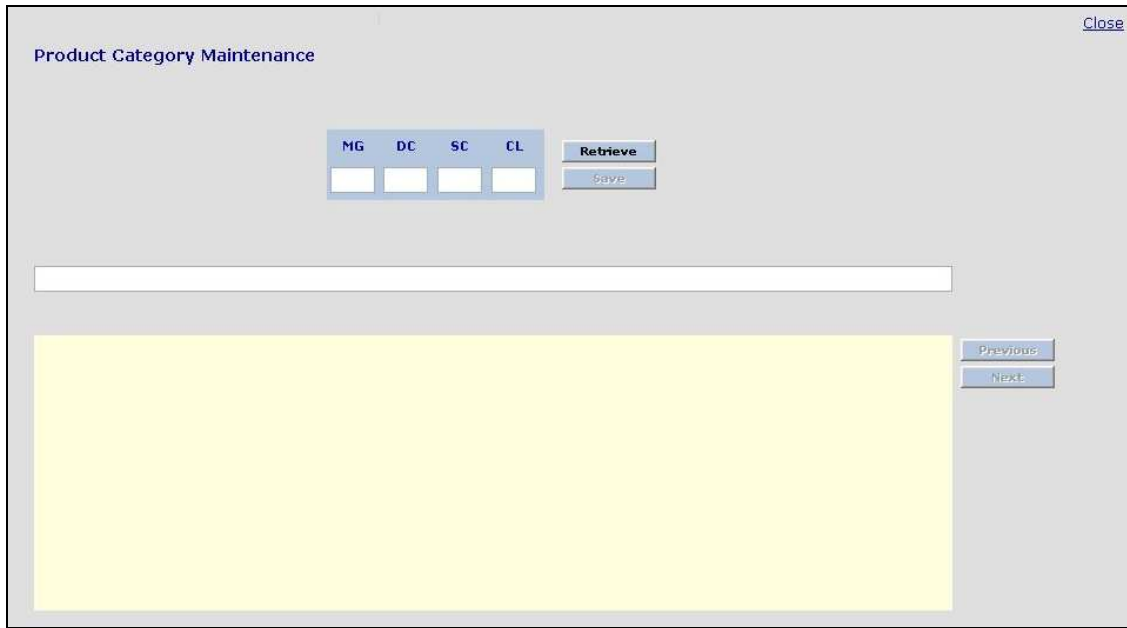
Criteria #	Criteria Description	Selenium Test
1.1	When the Marketing Analyst first logs into PCM application through the application portal, he /she is driven to the PCM main screen. See the image below.	test1_Display_PCM_main_screen
1.2	PCM main screen is divided into four sections: <ul style="list-style-type: none"> <li>• Title: “Product Category Maintenance”, aligned to the left</li> <li>• Filters section</li> <li>• Message Area section, is empty when the user logs in for the first time</li> <li>• Buttons Area</li> <li>• Results area, is empty when the user logs in for the first time.</li> </ul>	Manual, test1_Display_PCM_main_screen
1.3	The Filters section of the main screen displays: <ul style="list-style-type: none"> <li>▪ MG label.</li> <li>▪ MG text field (numeric values).</li> <li>▪ DC label.</li> <li>▪ DC text field (numeric values).</li> <li>▪ SC label.</li> <li>▪ SC text field (numeric values).</li> <li>▪ CL label.</li> <li>▪ CL text field (numeric values).</li> </ul>	test1_Display_PCM_main_screen
1.4	The Buttons section of the main screen displays: <ul style="list-style-type: none"> <li>• ‘Retrieve’ button.</li> <li>• ‘Save’ button.</li> <li>• ‘Previous’ button.</li> <li>• ‘Next’ button.</li> <li>• ‘Close’ link (this link is displayed on the top of the page)</li> </ul>	Manual, test1_Display_PCM_main_screen
1.5	If no records are displayed, “Save”, “Previous” and “Next” buttons are disabled.	test1_Display_PCM_main_screen
1.6	By default, focus is attached to MG Filter when the main page is displayed and no records exist.	Manual



**PERFORMANCE CONSIDERATIONS**

There should be no major impact to performance. Performance should be comparable to previous screens (Less than a 3 second response time).

**Rough Screen Mockup**



## Story 2 – Retrieve unassigned rows

### PLANNED ITERATION: Iteration 1

SCREEN: **Product Category Maintenance**

### STORY TITLE

Retrieve unassigned rows

### DEFINITIONS

Row: a MG/DC/SC/CL merchandising hierarchy combination

### DESCRIPTION

As Marketing Analyst I need to display merchandising combinations that are not assigned to a product category or change an already existing association.

### ENHANCEMENTS / DEPENDENCIES

This story is dependant on story 1

### LIMITATIONS

No changes should be persisted to the database at this point.

### ACCEPTANCE CRITERIA DESCRIPTION:

Criteria #	Criteria Description	Selenium Test
2.1	<p>If no values are set in MG or DC or MG and DC and SC filters, “Retrieve” button will be disabled. In other words, all the filters are optional, but at least one of them has to have a value to enable “Retrieve” button.</p> <p>For instance, if MG and DC and SC filters are empty and CL filter is set to 1 and the user empties the CL filter deleting all the characters inside, the “Retrieve button” will be disabled.</p> <p>If MG and DC filters are empty and DC and SC filters are non-empty and the user empties DC filter, “Retrieve” button will not be disabled because SC filter has a value.</p>	<b>Manual</b>
2.2	<p>The user can enter search criteria to identify which rows should be displayed:</p> <ul style="list-style-type: none"> <li>• Only positive integer values are allowed.</li> <li>• MG, DC, SC and CL filters are optional but mandatory from left to right. For instance, if the user set only SC=10, then automatically the system assume MG='-1' and DC='-1' as default values because are higher levels in hierarchy. If the user set criteria MG=10 and DC=12 then no assumptions are included and the query is performed as is.</li> <li>• If no errors are detected, the search will be performed.</li> </ul>	<b>Test2_Retrieve _Unassigned_Ro ws.rb</b>
2.3	<ul style="list-style-type: none"> <li>• Upon returning from search, every record displays the following fields in two columns (see the image below):</li> </ul>	<b>Manual, Test2_Retrie</b>

	<ul style="list-style-type: none"> <li>○ Prod Ctgry Grp Id</li> <li>○ Prod Ctgry Grp Desc</li> <li>○ Prod Ctgry Id</li> <li>○ Prod Ctgry Desc</li> <li>○ Merch Grp Id</li> <li>○ Demand Ctr Id</li> <li>○ Sub Ctr Id</li> <li>○ Merch Class Id</li> </ul> <ul style="list-style-type: none"> <li>• Two columns are displayed per record, Frozen at left and Primary at right.</li> <li>• No multiple records are displayed at the same time.</li> <li>• Both columns in figure below correspond to the same record</li> <li>• Both columns display the same fields with the same format.</li> <li>• Results are ordered by MG and DC and SC and CL fields in ASC order.</li> <li>• Negative numbers should be displayed ‘as is’. User is not allowed to enter negative values but seed data from unknown source can enter arbitrary values.</li> <li>• Records with Prod Ctgry Id = 9999 are <b>unassigned</b> records, otherwise are already assigned to a Product Category.</li> </ul>	ve _Unassigned_ Rows.rb
2.4	A label should be displayed on the header region. The message is:  ‘Record X from Y’  where X: current record and Y: total number of records.	Test2_Retrie ve _Unassigned_ Rows.rb
2.5	Information entered as search criteria in Filter region is maintained between searches. E.g. If the user entered ‘01’ as search criteria in MG filter, when the application returns from the search process, the page will display ‘01’ on the MG filter.	Test2_Retrie ve _Unassigned_ Rows.rb
2.6	If a search returns no matching records, the results area should be set to blank, as entering the application for the first time (see mockup in story #1) and in addition the message area displays the message preceded by an ‘i’ in a blue circle:  ‘No records found, please review your search criteria’.	Test2_Retrie ve _Unassigned_ Rows.rb

#### PERFORMANCE CONSIDERATIONS

There should be no major impact to performance. Performance should be comparable to previous screens (Less than a 3 second response time).

## Rough Screen Mockup

Product Category Maintenance [Close](#)

MG	DC	SC	CL	Retrieve
12	<input type="text"/>	<input type="text"/>	<input type="text"/>	Save

---

<p>Prod Ctgry Grp Id: <b>2</b> Prod Ctgry Grp Desc: <b>FROZEN</b> Prod Ctgry Id: <input type="text" value="9999"/> Prod Ctgry Desc: <b>MEN'S FOOTWEAR</b> Merch Grp Id: <b>30</b> Demand Ctr Id: <b>18</b> Sub Ctr Id: <b>95</b> Merch Class Id: <b>10</b></p>	<p>Prod Ctgry Grp Id: <b>1</b> Prod Ctgry Grp Desc: <b>PRIMARY</b> Prod Ctgry Id: <input type="text" value="9999"/> Prod Ctgry Desc: <b>MEN'S FOOTWEAR</b> Merch Grp Id: <b>30</b> Demand Ctr Id: <b>18</b> Sub Ctr Id: <b>95</b> Merch Class Id: <b>10</b></p>	<p><a href="#">Previous</a> <a href="#">Next</a></p>
--	---	--

## Story 3 – Navigate through search results

### PLANNED ITERATION: Iteration 1

### SCREEN: Product Category Maintenance

#### STORY TITLE

Navigate through search results

#### DESCRIPTION

After searching, the Marketing Analyst should be able to visit each retrieved row using the next and previous buttons in order to assign product category to unassigned rows.

#### ENHANCEMENTS / DEPENDENCIES

This story is dependant on story 2

#### LIMITATIONS

None

#### ACCEPTANCE CRITERIA DESCRIPTION:

Criteria #	Criteria Description	Selenium Test
3.1	<b>Next</b> and <b>Previous</b> buttons should be disabled when either the screen has no search results or has exactly one result.	Test3_Navigate_Through_Search_Results.rb
3.2	<b>Previous</b> button should be disabled after a successful search when the user is located on the first row.	Test3_Navigate_Through_Search_Results.rb
3.3	<b>Next</b> button should be disabled after a successful search when the user is located on the last row.	Test3_Navigate_Through_Search_Results.rb
3.4	<b>Next</b> and <b>Previous</b> buttons should be enabled whenever a valid record exists after or before the current record.	Test3_Navigate_Through_Search_Results.rb
3.5	Information entered as search criteria in Filter region is maintained when the user moves back and forth in the result list using the <b>Next</b> and <b>Previous</b> buttons.	Test3_Navigate_Through_Search_Results.rb
3.6	When the user moves back and forth in the record list, the screen looks the same as the previous page, populated with the information of the current record.  For instance if the user moves ahead to record 3 the label says: "Record 3 of 5"	Manual, Test3_Navigate_Through_Search_Results.rb
3.7	When each record is displayed on the screen, the cursor position is set to the Product Category Id entry on the left column. See the image below	Manual

3.8	Regardless the current position mouse pointer position, <Enter> key should be defaulted to "Next" Button. If no records were returned during the search, this event should be ignored.	Manual, Test3_Navigate_Through_Search_Results.rb
-----	--	--

**PERFORMANCE CONSIDERATIONS**

There should be no major impact to performance. Performance should be comparable to previous screens (Less than a 3 second response time).

**Rough Screen Mockup**



## Story 4 – Assign Product Category to New Rows

**PLANNED ITERATION: Iteration 1**

**SCREEN: Product Category Maintenance**

**STORY TITLE**

Assign Product Category to New Rows

**DESCRIPTION**

After searching, the Marketing Analyst picks an unselected row and set a new value for Product Category ID in both columns.

**ENHANCEMENTS / DEPENDENCIES**

This story is dependant on story 3

**LIMITATIONS**

If database already contains records including product category ID fields with non-allowed values such as 19 or out of range values entered by external applications, PCM application will not validate this specific field if the user does not change the product category ID value. If the user modifies this field, then all the validations described in stories are applied.

**ACCEPTANCE CRITERIA DESCRIPTION:**

Criteria #	Criteria Description	Selenium Test
4.1	In both columns all the fields, except Product Category Id should be disabled under any circumstance.	<b>Manual,</b> <b>Test4_Assign_Product_Category_to_New_Rows.rb</b>
4.2	User is only allowed to enter positive integer values in Frozen Product Category Id field. Any other value will be ignored.	<b>Test4_Assign_Product_Category_to_New_Rows.rb</b>
4.3	The input field associated with the FROZEN category allows the entry of categories between 1 and 22, but not 19.	<b>Test4_Assign_Product_Category_to_New_Rows.rb</b>
4.4	If the user sets 19 as Product Category in FROZEN column, and clicks on Next, Previous or Save buttons, the message area displays an error message preceded by an 'x' in a red circle:  ' 19 is not a valid value for Frozen Product Category Id, please use a different value'.	<b>Test4_Assign_Product_Category_to_New_Rows.rb</b>
4.5	If the user sets a value out of 1-22 range as Product Category in FROZEN column, the message area displays an error message preceded by an 'x' in a red circle:  'Entered value is out of range for FROZEN Product Category, please use a value between 1 and 22'.	<b>Test4_Assign_Product_Category_to_New_Rows.rb</b>

4.6	If the user sets a blank or erases the value that was at Product Category in FROZEN column, and clicks on Next, Previous or Save buttons the message area displays an error message preceded by an 'x' in a red circle:  'The following required field(s) are empty: Frozen Product Category Id'	Test4_Assign_Product_Category_to_New_Rows.rb
4.7	User is only allowed to enter positive integer values in Primary Product Category Id field. Any other value will be ignored.	Test4_Assign_Product_Category_to_New_Rows.rb
4.8	The input field associated with the PRIMARY category will allow the entry of categories between 18 and 56, but not 19.	Test4_Assign_Product_Category_to_New_Rows.rb
4.9	If the user sets 19 as Product Category in PRIMARY column, and clicks on Next, Previous or Save buttons, the message area displays an error message preceded by an 'x' in a red circle:  '19 is not a valid value for Primary Product Category Id, please use a different value'.	Test4_Assign_Product_Category_to_New_Rows.rb
4.10	If the user sets a value out of 18-56 range as Product Category in PRIMARY column, the message area displays an error message preceded by an 'x' in a red circle:  'Entered value is out of range for PRIMARY Product Category, please use a value between 18 and 56'.	Test4_Assign_Product_Category_to_New_Rows.rb
4.11	If the user sets a blank or erases the value that was at Product Category in PRIMARY column, and clicks on Next, Previous or Save buttons, the message area displays an error message preceded by an 'x' in a red circle:  'The following required field(s) are empty: Primary Product Category Id'	Test4_Assign_Product_Category_to_New_Rows.rb
4.12	If the user sets a blank or erases the value that was in both Product Category in PRIMARY and Product Category in PRIMARY columns, the message area displays an error message preceded by an 'x' in a red circle:  'The following required field(s) are empty: Frozen Product Category Id, Primary Product Category Id'	Test4_Assign_Product_Category_to_New_Rows.rb
4.13	The Product Category Id field value can be different in PRIMARY and FROZEN columns.	Test4_Assign_Product_Category_to_New_Rows.rb
4.14	If the user updates several rows: <ul style="list-style-type: none"> <li>the system should keep modifications until the user hits the save button.</li> <li>Moving back and forth in the result list and altering data in several rows should not lose changes.</li> <li>If a row is modified and the user moves to another row and goes back to the modified row, the system should display the modified data.</li> <li>After saving data, the message area displays an info message preceded by an 'i' in a blue circle:</li> </ul>	Manual, Test4_Assign_Product_Category_to_New_Rows.rb



	‘Data saved successfully’.	
4.15	The PCM Main screen displays a ‘Reset’ button. On 'Reset' the filter section and the Frozen or Primary Product Category Id are cleared and the initial values of these records are returned.	Test4_Assign Product_Cat egory_to_New Rows.rb
4.16	<p>If there are unsaved changes on the Main screen and the user clicks either the “close” or “retrieve” buttons, the following message is displayed in a pop up window:</p> <p>“Are you sure you want to either select a different Row or navigate away? Your changes will not be saved. Press OK to continue or Cancel to stay on current page.”</p> <ul style="list-style-type: none"> <li>On OK, the Main screen for the new row is displayed <b>and</b> the unsaved changes are lost.</li> </ul> <p>On Cancel, the user remains on the Main screen for the same row with the unsaved changes still on the screen.</p>	Test4_Assign Product_Cat egory_to_New Rows.rb
4.17	“Save” button is enabled only if any change is performed, otherwise the button is disabled.	Test4_Assign Product_Cat egory_to_New Rows.rb

### PERFORMANCE CONSIDERATIONS

There should be no major impact to performance. Performance should be comparable to previous screens (Less than a 3 second response time).

### Rough Screen Mockup

[Close](#)

**Product Category Maintenance**

MG	DC	SC	CL	Retrieve
12				Save

---

Prod Ctgry Grp Id: **2**

Prod Ctgry Grp Desc: **ERZEN**

Prod Ctgry Id:

Prod Ctgry Desc: **MEN'S FOOTWEAR**

Merch Grp Id: **30**

Demand Ctr Id: **18**

Sub Ctr Id: **95**

Merch Class Id: **10**

Prod Ctgry Grp Id: **1**

Prod Ctgry Grp Desc: **PRIMARY**

Prod Ctgry Id:

Prod Ctgry Desc: **MEN'S FOOTWEAR**

Merch Grp Id: **30**

Demand Ctr Id: **18**

Sub Ctr Id: **95**

Merch Class Id: **10**

Previous

Next

## **Historias de DNIS**

### **Story 1 – Display DNIS Main Screen**

PLANNED ITERATION: **Iteration 1**

SCREEN: **DNIS**

#### **STORY TITLE**

Display DNIS main screen

#### **DESCRIPTION**

As Advertising Analyst I need to enter the DNIS main screen and view the header information to clearly identify the headers. Also, I need to have a message area defined for messages to be displayed and button with the options of the application.

#### **LIMITATIONS**

No changes should be persisted to the database at this point.

**ACCEPTANCE CRITERIA DESCRIPTION:**

<b>Criteria #</b>	<b>Criteria Description</b>	<b>Selenium Test</b>
<b>1.1</b>	When the Advertising Analyst first logs into DNIS application through the application portal, he /she is driven to the DNIS main screen. See the image below.	
<b>1.2</b>	DNIS main screen is divided into four sections: <ul style="list-style-type: none"><li>• Filters section</li><li>• Message Area section, is empty when the user logs in for the first time</li><li>• Buttons Area</li><li>• Results area, is empty when the user logs in for the first time</li></ul>	
<b>1.3</b>	The Filters section of the main screen displays: <ul style="list-style-type: none"><li>▪ DNIS ID label.</li><li>▪ DNIS ID text field (only numbers, 5 digits length starting with 3 o 5).</li> <li>▪ Phone Number label</li><li>▪ Phone Number text field (alphanumeric values)</li> <li>▪ Source label.</li><li>▪ Source Code text field (numeric values)</li> <li>▪ Desc label</li><li>▪ Desc text field (alphanumeric values)</li></ul>	

1.4	<p>The Buttons section of the main screen displays:</p> <ul style="list-style-type: none"> <li>• Twins checkbox (marked by default).</li> <li>• Label 'Automatically Update Twin DNIS Records'. This label is directly associated to the checkbox mentioned above and together represent only one piece of functionality.</li> <li>• 'Retrieve' button.</li> <li>• 'Save' button.</li> <li>• 'Insert' button.</li> <li>• 'Delete' button (disabled by default).</li> <li>• 'Close' link (this link is displayed on the top of the page)</li> </ul>	
1.5	<p>The Results section of the main screen is empty, but displays the following headers:</p> <ul style="list-style-type: none"> <li>• Dnis id</li> <li>• Dialed Phone Nbr</li> <li>• Src Cd</li> <li>• Source Code Description</li> <li>• Eff Dt</li> <li>• Exp Dt</li> </ul>	
1.6	<ul style="list-style-type: none"> <li>• If no records are displayed, "Save" and "Delete" buttons are disabled.</li> <li>• These buttons can be enabled only after a retrieve operation with one or more matching records.</li> </ul>	
1.7	<p>By default, focus is attached to DNIS ID Filter when the main page is displayed and no records exist</p>	

**PERFORMANCE CONSIDERATIONS**

There should be no major impact to performance. Performance should be comparable to previous screens (Less than a 3 second response time).

## Rough Screen Mockup

Query By Close

DNIS ID  PhoneNumber  Source  Desc

Automatically Update  
Twin DNIS Records

Dnis Id	Dialed Phone Nbr	Src Cd	Source Code Description	Eff Dt	Exp Dt
---------	------------------	--------	-------------------------	--------	--------

## Story 2 – Navigate through search results

**PLANNED ITERATION: Iteration 1**

**SCREEN: DNIS**

STORY TITLE

Navigate through search results

**DESCRIPTION**

The Advertising Analyst should be able to perform partial and exact searches and visit each retrieved item using the scroll bar.

**ENHANCEMENTS / DEPENDENCIES**

This story is dependant on story 1

**LIMITATIONS**

None

**ACCEPTANCE CRITERIA DESCRIPTION:**

Criteria #	Criteria Description	Selenium Test																					
2.1	<p>In the Main screen, the user can retrieve a subset of DNIS records by selecting one of the search criteria available in the Filters area:</p> <ul style="list-style-type: none"> <li>• DNIS ID</li> <li>• Phone number</li> <li>• Src Code</li> <li>• Description</li> </ul> <p>and click “Retrieve” button. A search is triggered and the list of matching records ordered by the last four digits of the DNIS_ID field is displayed including the following format:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th style="text-align: center;">Column header</th> <th colspan="2" style="text-align: center;">Display Format</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Dnis id</td> <td style="text-align: center;">Text</td> <td style="text-align: center;">Up to 10 Characters</td> </tr> <tr> <td style="text-align: center;">Dialed Phone Nbr</td> <td style="text-align: center;">Text</td> <td style="text-align: center;">Up to 20 Characters</td> </tr> <tr> <td style="text-align: center;">Src Cd</td> <td style="text-align: center;">Text</td> <td style="text-align: center;">7 Characters</td> </tr> <tr> <td style="text-align: center;">Source Code Description</td> <td style="text-align: center;">Text</td> <td style="text-align: center;">Up to 45 Characters</td> </tr> <tr> <td style="text-align: center;">Eff Dt</td> <td style="text-align: center;">Date</td> <td style="text-align: center;">mm/dd/yyyy</td> </tr> <tr> <td style="text-align: center;">Exp Dt</td> <td style="text-align: center;">Date</td> <td style="text-align: center;">mm/dd/yyyy</td> </tr> </tbody> </table>	Column header	Display Format		Dnis id	Text	Up to 10 Characters	Dialed Phone Nbr	Text	Up to 20 Characters	Src Cd	Text	7 Characters	Source Code Description	Text	Up to 45 Characters	Eff Dt	Date	mm/dd/yyyy	Exp Dt	Date	mm/dd/yyyy	
Column header	Display Format																						
Dnis id	Text	Up to 10 Characters																					
Dialed Phone Nbr	Text	Up to 20 Characters																					
Src Cd	Text	7 Characters																					
Source Code Description	Text	Up to 45 Characters																					
Eff Dt	Date	mm/dd/yyyy																					
Exp Dt	Date	mm/dd/yyyy																					

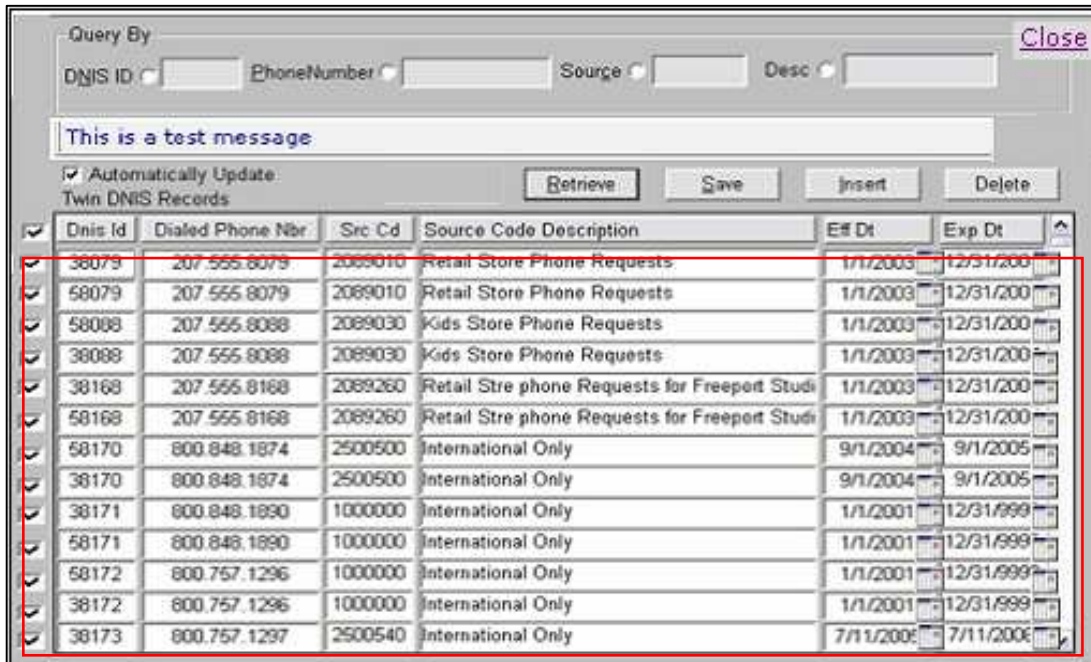
2.2	<p>In the DNIS “Main Screen“, the header region is fixed and appears always on the top of the browser regardless the number of matching records.</p> <p>Only one criteria filter is allowed per search in the Filter area, if the user selects different filter criteria, any other value set on filters is removed. E.g. if the user selects the ‘Src Code’ filter and sets ‘2343’ as value, and then selects the ‘Phone Number’ filter, the application removes the value ‘2343’ from ‘Src Code’ filter.</p>	
2.3	<p>The user can filter out DNIS records using partial searches:</p> <ul style="list-style-type: none"> <li>- last four digits of the five digit DNIS-ID</li> <li>- partial content of the phone number</li> <li>- partial content of the source code</li> <li>- partial content of the description</li> </ul>	
2.4	<p>The user can filter out DNIS records using exact values:</p> <ul style="list-style-type: none"> <li>- complete DNIS-ID</li> <li>- complete content of the phone number</li> <li>- complete content of the source code</li> <li>- complete content of the description</li> </ul>	
2.5	<p>If no records match the query, the message area displays an info message preceded by an ‘i’ in a blue circle:</p> <p style="text-align: center;">‘No records found, please refine your query.’</p>	
2.6	<ul style="list-style-type: none"> <li>• Query results will be constrained to 300 records.</li> <li>• If more than 300 records match the search criteria only the first 300 records will be returned and the message area displays an info message preceded by an ‘i’ in a blue circle:</li> </ul> <p style="text-align: center;">‘Query result was reduced to 300 records, please refine your query.’</p>	
2.7	<ul style="list-style-type: none"> <li>• Regardless the current position mouse pointer position, &lt;Enter&gt; key should be defaulted to “Retrieve” button.</li> <li>• If the user press the &lt;Enter&gt; key, a search based upon the current search criteria is performed.</li> </ul>	
2.8	<p>If the user selects a reduced set of DNIS records that fit the result area:</p> <ul style="list-style-type: none"> <li>• the scroll bar is disabled</li> <li>• all the text fields become editable</li> <li>• checkboxes are unchecked.</li> </ul> <p>The number of records that fit the result area will vary depending on the resolution configured on the user’s monitor.</p>	

2.9	<p>If the user selects a number of records that does not fit the result area:</p> <ul style="list-style-type: none"> <li>• the scroll bar is enabled</li> <li>• all the text fields become editable</li> <li>• checkboxes are unchecked.</li> </ul>	
2.10	<p>The search criteria is remembered between searches, then if the user sets "123" as value for DNIS ID criteria:</p> <ul style="list-style-type: none"> <li>• the application displays the query result</li> <li>• the application keeps "123" as the value in the DNIS ID filter</li> <li>• the value in the remaining filters is cleared out.</li> </ul>	

**PERFORMANCE CONSIDERATIONS**

There should be no major impact to performance. Performance should be comparable to previous screens (Less than a 3 second response time).

**Rough Screen Mockup**





### Story 3 – Insert new DNIS records

**PLANNED ITERATION: Iteration 1**

**SCREEN: DNIS**

**STORY TITLE**

Insert new DNIS records

**DESCRIPTION**

As Advertising Analyst, I should be able to create new DNIS records in order to perform validations.

**ENHANCEMENTS / DEPENDENCIES**

None

**LIMITATIONS**

None

**ACCEPTANCE CRITERIA DESCRIPTION:**

Criteria #	Criteria Description	Selenium Test
3.1	<p>From the “Main Screen”, if the user clicks the “Insert” button:</p> <ul style="list-style-type: none"><li>• A new blank record is displayed on the screen at the beginning of the records area.</li><li>• All the fields in the new records become editable</li><li>• The focus is passed to the ‘DNIS ID’ field in the new record.</li><li>• “delete” checkbox is unmarked by default.</li><li>• “Save” button becomes enabled and “Retrieve”, “Insert”, “Delete” buttons are disabled until the user hits the “Save” button and the new records is persisted successfully, then these buttons will return to its previous state.</li></ul>	
3.2	<p>The “Insert” button is always enabled. The user can add new records, regardless if a query was executed or the results area is empty.</p>	
3.3	<p>The user can enter values in:</p> <ul style="list-style-type: none"><li>• DNIS ID (only numbers, 5 digits length starting with 3 or 5)</li><li>• Dialed phone number (alphanumeric values)</li><li>• Source Code (numeric values)</li><li>• Source Code Description (alphanumeric values)</li><li>• Optionally change the defaulted effective and/or expiration dates.</li></ul>	

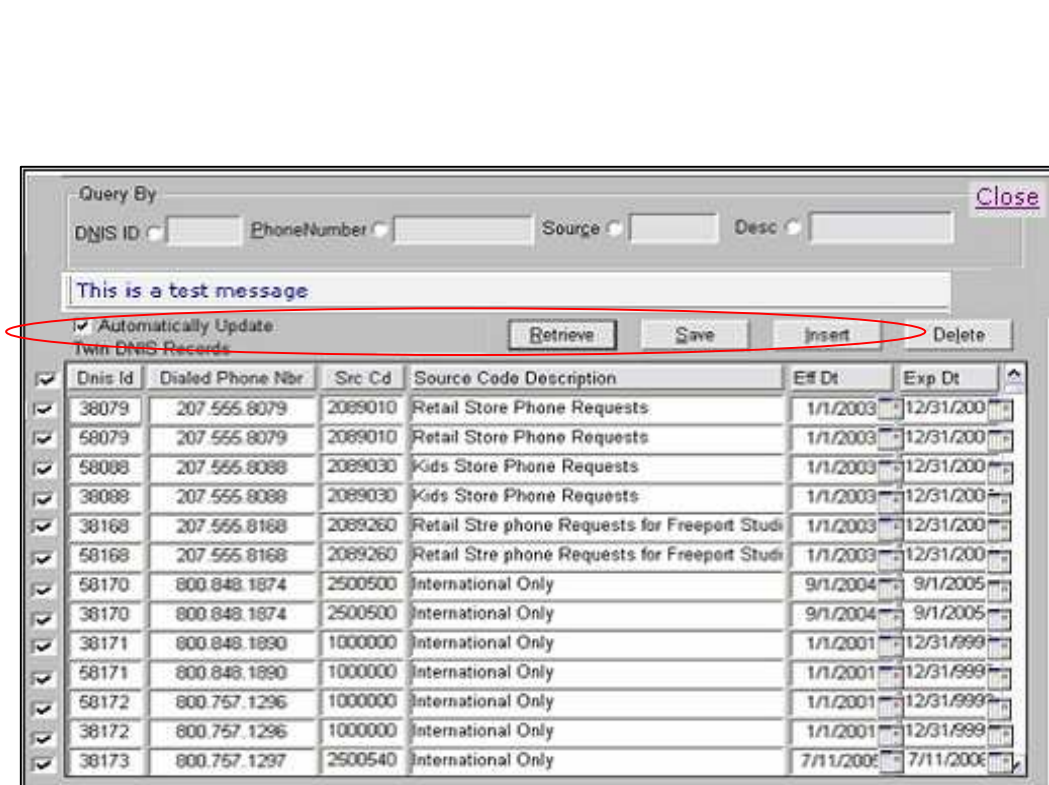
	(mm/dd/yyyy)  If the user enters an invalid value according to the constraints described above, the message area displays an error message preceded by an 'x' in a red circle:  "Invalid values, please review the values"	
3.4	If the twin flag is checked, twin records are created: <ul style="list-style-type: none"> <li>• If DNIS record starts with 3XXXX, a new 5 XXXX record is created.</li> <li>• If DNIS record starts with 5XXXX, a new 3XXXX record is created.</li> </ul>	
3.5	If the new record already exists, the message area displays an error message preceded by an 'x' in a red circle:  'Record already exist, please use a different DNIS Id.'	
3.6	If some mandatory fields are missing when saving, the message area displays an error message preceded by an 'x' in a red circle:  "Required fields are empty, please review the values"	
3.7	In effective date and expiration date fields, the system will display a date picker widget.	
3.8	If there are unsaved changes on the Main screen and the user performs another search, the following message is displayed:  "Are you sure you want to proceed with a new Search? Your changes will not be saved. Press OK to continue or Cancel to stay on current page." <ul style="list-style-type: none"> <li>▪ On OK, the Main screen for the new Item is displayed <b>and</b> the unsaved changes are lost.</li> <li>▪ On Cancel, the user remains on the Main screen with the unsaved changes still on the screen.</li> </ul>	
3.9	If there are unsaved changes on the Main screen and the user clicks the "close" button, the following message is displayed:  "Are you sure you want to close the window? Your changes will not be saved. Press OK to continue or Cancel to stay on current page." <ul style="list-style-type: none"> <li>▪ On OK, the Main screen will be closed <b>and</b> the unsaved changes are lost.</li> </ul>	

	<ul style="list-style-type: none"> <li>On Cancel, the user remains on the Main screen with the unsaved changes still on the screen.</li> </ul>	
--	--	--

**PERFORMANCE CONSIDERATIONS**

There should be no major impact to performance. Performance should be comparable to previous screens (Less than a 3 second response time).

**Rough Screen Mockup**



## Story 4 – Update existing DNIS records

### PLANNED ITERATION: Iteration 1

### SCREEN: DNIS

### STORY TITLE

Update existing DNIS records

### DESCRIPTION

After searching, the Advertising Analyst picks a DNIS record and sets new values for its fields.

### ENHANCEMENTS / DEPENDENCIES

This story is dependant on story 2

### LIMITATIONS

None

### ACCEPTANCE CRITERIA DESCRIPTION:

Criteria #	Criteria Description	Selenium Test
4.1	<ul style="list-style-type: none"><li>The “Save” button is enabled in the “Main Screen” only when one or more records are displayed in the records area.</li><li>When the “Save” button is enabled, the user can modify any record and any field in the displayed rows:<ul style="list-style-type: none"><li>DNIS ID (only numbers, 5 digits length starting with 3 or 5)</li><li>Dialed phone number (alphanumeric values)</li><li>Source Code (numeric values)</li><li>Source Code Description (alphanumeric values)</li><li>Optionally change the defaulted effective and/or expiration dates. (mm/dd/yyyy)</li></ul></li></ul> <p>If the user enters an invalid value according to the constraints described above, the message area displays an error message preceded by an ‘x’ in a red circle:</p> <p>“Invalid values, please review the values”</p>	
4.2	If the user modifies a record with DNIS ID 3XXXXX, then the modifications are also applied to the twin 5XXXXX when the twin flag is checked.	

4.3	If the user modifies a record with DNIS ID 5XXXX, then the modifications are also applied to the twin 3XXXX when the twin flag is checked.	
4.4	<p>If the DNIS-ID field is modified:</p> <ul style="list-style-type: none"> <li>• a new record is created using the remaining fields of the current record and the new DNIS-ID value.</li> <li>• The current record is preserved if no other modification was performed or updated if further fields were modified.</li> </ul>	
4.5	<p>If the modified record already exists, the message area displays an error message preceded by an 'x' in a red circle:</p> <p style="text-align: center;">‘The record already exists, please use a different ID’.</p>	
4.6	<p>If some mandatory fields are missing when updating, the message area displays an error message preceded by an 'x' in a red circle:</p> <p style="text-align: center;">‘Required fields are empty, please review the values’.</p>	
4.7	<p>If there are unsaved changes on the Main screen and the user performs another search, the following message is displayed:</p> <p>“Are you sure you want to proceed with a new Search? Your changes will not be saved. Press OK to continue or Cancel to stay on current page.”</p> <ul style="list-style-type: none"> <li>▪ On OK, the Main screen for the new Item is displayed <b>and</b> the unsaved changes are lost.</li> <li>▪ On Cancel, the user remains on the Main screen with the unsaved changes still on the screen.</li> </ul>	
	<p>If there are unsaved changes on the Main screen and the user clicks the “close” button, the following message is displayed:</p> <p>“Are you sure you want to close the window? Your changes will not be saved. Press OK to continue or Cancel to stay on current page.”</p> <ul style="list-style-type: none"> <li>▪ On OK, the Main screen will be closed <b>and</b> the unsaved changes are lost.</li> <li>▪ On Cancel, the user remains on the Main screen with the unsaved changes still on the screen.</li> </ul>	

## PERFORMANCE CONSIDERATIONS

There should be no major impact to performance. Performance should be comparable to previous screens (Less than a 3 second response time).

## Rough Screen Mockup

Query By  
DNIS ID  PhoneNumber  Source  Desc  [Close](#)

This is a test message

Automatically Update  
Twin DNS Records

<input checked="" type="checkbox"/>	Dnis Id	Dialed Phone Nbr	Src Cd	Source Code Description	Eff Dt	Exp Dt
<input checked="" type="checkbox"/>	38079	207.555.8079	2089010	Retail Store Phone Requests	1/1/2003	12/31/200
<input checked="" type="checkbox"/>	58079	207.555.8079	2089010	Retail Store Phone Requests	1/1/2003	12/31/200
<input checked="" type="checkbox"/>	58008	207.555.8088	2089030	Kids Store Phone Requests	1/1/2003	12/31/200
<input checked="" type="checkbox"/>	38088	207.555.8088	2089030	Kids Store Phone Requests	1/1/2003	12/31/200
<input checked="" type="checkbox"/>	38168	207.555.8168	2089260	Retail Stre phone Requests for Freeport Studi	1/1/2003	12/31/200
<input checked="" type="checkbox"/>	58168	207.555.8168	2089260	Retail Stre phone Requests for Freeport Studi	1/1/2003	12/31/200
<input checked="" type="checkbox"/>	58170	800.848.1874	2500500	International Only	9/1/2004	9/1/2005
<input checked="" type="checkbox"/>	38170	800.848.1874	2500500	International Only	9/1/2004	9/1/2005
<input checked="" type="checkbox"/>	38171	800.848.1890	1000000	International Only	1/1/2001	12/31/999
<input checked="" type="checkbox"/>	58171	800.848.1890	1000000	International Only	1/1/2001	12/31/999
<input checked="" type="checkbox"/>	58172	800.757.1296	1000000	International Only	1/1/2001	12/31/999
<input checked="" type="checkbox"/>	38172	800.757.1296	1000000	International Only	1/1/2001	12/31/999
<input checked="" type="checkbox"/>	38173	800.757.1297	2500540	International Only	7/11/2005	7/11/2006

## Story 5 – Delete existing DNIS records

### PLANNED ITERATION: Iteration 1

### SCREEN: DNIS

### STORY TITLE

Delete existing DNIS records

### DESCRIPTION

After searching, the Advertising Analyst checks the list of records he/she wants to remove.

### ENHANCEMENTS / DEPENDENCIES

This story is dependant on story 2

### LIMITATIONS

None

### ACCEPTANCE CRITERIA DESCRIPTION:

Criteria #	Criteria Description	Selenium Test
5.1	In the “Main Screen”, “Delete” button is enabled only when one or more records are explicitly marked for removal.	
5.2	If the user marks one or more records and hits “Delete” button, the application displays a popup window with a confirmation message:  ‘Are you sure to delete the selected records?’ Yes - No <ul style="list-style-type: none"><li>• If the user clicks ‘No’, then no action is performed and the page remains with no changes.</li><li>• If the user clicks ‘Yes’, changes are persisted in database</li></ul>	
5.3	Upon deletion, the message area displays an info message preceded by an ‘i’ in a blue circle:  ‘ Record deleted successfully’	
5.4	<ul style="list-style-type: none"><li>• If the twin flag not is checked, only selected records are removed</li><li>• If the twin flag is checked, twin records are removed as well, as follows:<ul style="list-style-type: none"><li>• If DNIS record starts with 3XXX, record 5XXX is also removed.</li><li>• If DNIS record starts with 5XXX, record 3XXX is also removed.</li></ul></li></ul>	
5.5	After removal, the new list of records will be displayed keeping the existing search criteria.	

## PERFORMANCE CONSIDERATIONS

There should be no major impact to performance. Performance should be comparable to previous screens (Less than a 3 second response time).

## Rough Screen Mockup

Query By  
DNIS ID  PhoneNumber  Source  Desc  [Close](#)

This is a test message

Automatically Update  
Twin DNS Records

<input checked="" type="checkbox"/>	Dnis Id	Dialed Phone Nbr	Src Cd	Source Code Description	Eff Dt	Exp Dt
<input checked="" type="checkbox"/>	38079	207.555.8079	2089010	Retail Store Phone Requests	1/1/2003	12/31/200
<input checked="" type="checkbox"/>	58079	207.555.8079	2089010	Retail Store Phone Requests	1/1/2003	12/31/200
<input checked="" type="checkbox"/>	58008	207.555.8088	2089030	Kids Store Phone Requests	1/1/2003	12/31/200
<input checked="" type="checkbox"/>	38088	207.555.8088	2089030	Kids Store Phone Requests	1/1/2003	12/31/200
<input checked="" type="checkbox"/>	38168	207.555.8168	2089260	Retail Stre phone Requests for Freeport Studi	1/1/2003	12/31/200
<input checked="" type="checkbox"/>	58168	207.555.8168	2089260	Retail Stre phone Requests for Freeport Studi	1/1/2003	12/31/200
<input checked="" type="checkbox"/>	58170	800.848.1874	2500500	International Only	9/1/2004	9/1/2005
<input checked="" type="checkbox"/>	38170	800.848.1874	2500500	International Only	9/1/2004	9/1/2005
<input checked="" type="checkbox"/>	38171	800.848.1890	1000000	International Only	1/1/2001	12/31/999
<input checked="" type="checkbox"/>	58171	800.848.1890	1000000	International Only	1/1/2001	12/31/999
<input checked="" type="checkbox"/>	58172	800.757.1296	1000000	International Only	1/1/2001	12/31/999
<input checked="" type="checkbox"/>	38172	800.757.1296	1000000	International Only	1/1/2001	12/31/999
<input checked="" type="checkbox"/>	38173	800.757.1297	2500540	International Only	7/11/2005	7/11/2005



## Anexo 2

### Glosario de términos.

En este apartado se incluye las definiciones de aquellos términos y abreviaturas requeridas para la comprensión de este documento, de una forma clara y concisa.

Por lo anterior, se mostrarán las definiciones y abreviaturas, utilizadas a lo largo del documento:

- a. PBC: Power Builder Conversion.
- b. PCM: Producto Category Maintenance.
- c. DNIS: Dial Number Identification System.
- d. Sign-off: es cuando se aprueba una historia. El sign-off es como darle el visto bueno, se da por parte de un experto en el proceso (llamado Bussiness Sign-off) y del responsable de calidad (llamado QA Sign-off). Es decir, esto es dos momentos durante el proceso del negocio.
- e. Stage: es la etapa en que calidad revisa de manera detallada el sistema desarrollado.
- f. JarJar Test: es el punto donde se integra por primera vez el código de los desarrolladores en un ambiente no local.
- g. Business sign-off: es la revisión de la historia por parte del Business Analyst, esta se realiza cuando se concluye la programación de la historia. El Business Analyst, revisa la aplicación, para cerciorarse que se cumpliera con cada criterio de la historia.
- h. QA sign-off: es la revisión de la historia por parte de la Quality Analyst, esta se realiza cuando se concluye con el Business sign-off de la historia. El Quality Analyst, revisa la aplicación, de una forma exhaustiva, para verificar que se cumpla con los criterios de la historia, he incluso que sea una historia robusta y sin errores.
- i. BA: Business Analyst.

- j. QA: Quality Analyst.
- k. Extreme Programing: ver anexo 1.1
- l. Unit Test: pruebas unitarias, ver características de XP (anexo 1.1).

## Anexo 3

### Metodología XP (Extreme Programming).

Es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es la más destacada de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la pre-visibilidad.

Los defensores de XP, consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

Las características fundamentales del método son:

- a. Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- b. Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación. Véase, por ejemplo, las herramientas de prueba JUnit orientada a Java y DUnit orientada a Delphi e inspirada en JUnit.
- c. Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera (el código es revisado y discutido mientras se escribe), es más importante que la posible pérdida de productividad inmediata.

- d. Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- e. Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- f. Refactorización del código, es decir, rescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- g. Propiedad del código compartida, en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- h. Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.