**A PROJECT REPORT**

**ON**

**Product Sales Dashboard**

**Submitted to**



**Project Report**

**on**

**Product Sales Dashboard**

**In the partial fulfillment of the requirement for the Degree of**

**Master of Computer Application**

**Submitted by**

**Vivek Dhakre (1405006452)**

[OBJECTIVE 1](#_Toc222596668)

[Project Scope 2](#_Toc222596669)

[Customer, Products, Billing Generation 2](#_Toc222596670)

[Report Generation System 2](#_Toc222596671)

[Out of Scope 3](#_Toc222596672)

[Goal 3](#_Toc222596673)

[Solution Concept 3](#_Toc222596674)

[User Profiles 5](#_Toc222596675)

[Usage Summary 6](#_Toc222596676)

[Requirements Summary 7](#_Toc222596677)

[Business Requirements 7](#_Toc222596678)

[User Requirements 7](#_Toc222596679)

[Operational Requirements 8](#_Toc222596680)

[System Requirements 9](#_Toc222596681)

[Success Criteria 9](#_Toc222596682)

[Sales Automation 9](#_Toc222596683)

[Design Goals and Constraints 10](#_Toc222596684)

[Performance 10](#_Toc222596685)

[Availability 10](#_Toc222596686)

[Reliability 10](#_Toc222596687)

[Scalability 10](#_Toc222596688)

[Security 10](#_Toc222596689)

[Interoperability 10](#_Toc222596690)

[Location 10](#_Toc222596691)

[Setup/Installation 10](#_Toc222596692)

[Features/Functionality 10](#_Toc222596693)

[Software and Hardware Requirement 13](#_Toc222596694)

[General Requirement for Server/Client: 13](#_Toc222596695)

[Development Tools and Technologies 13](#_Toc222596696)

[Development Tools and Technologies 14](#_Toc222596697)

[Front End 14](#_Toc222596698)

[Back End 14](#_Toc222596699)

[Development Environment 15](#_Toc222596700)

[Use Case Summary 16](#_Toc222596701)

[Manage Orders 16](#_Toc222596702)

[Description 16](#_Toc222596703)

[Business Need 16](#_Toc222596704)

[Priority 16](#_Toc222596705)

[Manage Products 16](#_Toc222596706)

[Description 16](#_Toc222596707)

[Business Need 16](#_Toc222596708)

[Priority 16](#_Toc222596709)

[Manage Contacts 16](#_Toc222596710)

[Description 16](#_Toc222596711)

[Business Need 16](#_Toc222596712)

[Priority 17](#_Toc222596713)

[Analyze Customers 17](#_Toc222596714)

[Description 17](#_Toc222596715)

[Business Need 17](#_Toc222596716)

[Priority 17](#_Toc222596717)

[Forecast Sales 17](#_Toc222596718)

[Description 17](#_Toc222596719)

[Business Need 17](#_Toc222596720)

[Priority 17](#_Toc222596721)

[Establish Sales Goals 17](#_Toc222596722)

[Description 17](#_Toc222596723)

[Business Need 17](#_Toc222596724)

[Priority 17](#_Toc222596725)

[Usage Scenarios 18](#_Toc222596726)

[Use Case ID: UC 4.7.1 Retrieve Customer Data 18](#_Toc222596727)

[Intent 18](#_Toc222596728)

[Scenario Narrative 18](#_Toc222596729)

[Assumptions/Preconditions 18](#_Toc222596730)

[Actors 18](#_Toc222596731)

[Basic Course 18](#_Toc222596732)

[Alternate Course 18](#_Toc222596733)

[Uses/Extends 18](#_Toc222596734)

[User Implementation Requests 18](#_Toc222596735)

[Frequency 18](#_Toc222596736)

[Authority 18](#_Toc222596737)

[Associated Specifications 19](#_Toc222596738)

[Issues 19](#_Toc222596739)

[Decision Points 19](#_Toc222596740)

[Future Requirements 19](#_Toc222596741)

[Modification 19](#_Toc222596742)

[Use Case Model: Retrieve Customer Data 20](#_Toc222596743)

[Use Case ID: UC 04.7.2 Manage Products 21](#_Toc222596744)

[Intent 21](#_Toc222596745)

[Scenario Narrative 21](#_Toc222596746)

[Assumptions/Preconditions 21](#_Toc222596747)

[Actors 21](#_Toc222596748)

[Basic Course: 21](#_Toc222596749)

[Alternate Course: 21](#_Toc222596750)

[Uses/Extends 21](#_Toc222596751)

[User Implementation Requests 21](#_Toc222596752)

[Frequency 21](#_Toc222596753)

[Authority 21](#_Toc222596754)

[Associated Specifications 21](#_Toc222596755)

[Issues 22](#_Toc222596756)

[Decision Points 22](#_Toc222596757)

[Future Requirements 22](#_Toc222596758)

[Modification 22](#_Toc222596759)

[Use Case: Manage Products 23](#_Toc222596760)

[Use Case ID: UC 04.7.3 Manage Orders 24](#_Toc222596761)

[Intent 24](#_Toc222596762)

[Scenario Narrative 24](#_Toc222596763)

[Assumptions/Preconditions 24](#_Toc222596764)

[Actors 24](#_Toc222596765)

[Basic Course: 24](#_Toc222596766)

[Uses/Extends 24](#_Toc222596767)

[User Implementation Requests 24](#_Toc222596768)

[Frequency 24](#_Toc222596769)

[Authority 24](#_Toc222596770)

[Associated Specifications 25](#_Toc222596771)

[Issues 25](#_Toc222596772)

[Decision Points 25](#_Toc222596773)

[Future Requirements 25](#_Toc222596774)

[Modification 25](#_Toc222596775)

[Use Case ID: UC 04.7.3 Manage Contacts 27](#_Toc222596776)

[Intent 27](#_Toc222596777)

[Scenario Narrative 27](#_Toc222596778)

[Assumptions/Preconditions 27](#_Toc222596779)

[Actors 27](#_Toc222596780)

[Basic Course: 27](#_Toc222596781)

[Uses/Extends 27](#_Toc222596782)

[User Implementation Requests 27](#_Toc222596783)

[Frequency 27](#_Toc222596784)

[Authority 27](#_Toc222596785)

[Associated Specifications 27](#_Toc222596786)

[Issues 28](#_Toc222596787)

[Decision Points 28](#_Toc222596788)

[Future Requirements 28](#_Toc222596789)

[Modification 28](#_Toc222596790)

[Solution Architecture 30](#_Toc222596792)

[Infrastructure 30](#_Toc222596793)

[Visual Design 30](#_Toc222596794)

[Conceptual Design 30](#_Toc222596795)

[Logical Design 32](#_Toc222596796)

[Physical Design 34](#_Toc222596797)

[eBilling and Invoice System Data Dictionary 35](#_Toc222596798)

[eBilling and Invoice System Data Dictionary 36](#_Toc222596799)

[Table : product\_master 36](#_Toc222596800)

[Table : bill 36](#_Toc222596801)

[Table : bill\_details 37](#_Toc222596802)

[Table : Company\_master 37](#_Toc222596803)

[Table : user\_master 38](#_Toc222596804)

[eBilling and Invoice System – Data Schema ER Diagram 39](#_Toc222596805)

[eBilling and Invoice System DFD (Data Flow Diagram) 40](#_Toc222596806)

[Form Design and Coding 41](#_Toc222596807)

[FrmSplash 41](#_Toc222596808)

[FrmLogin 43](#_Toc222596809)

[FrmMain 46](#_Toc222596810)

[FrmProducts 48](#_Toc222596811)

[FrmBill 54](#_Toc222596812)

[FrmReports 71](#_Toc222596814)

[FrmPrintBill 73](#_Toc222596815)

[ProductSummary Report 75](#_Toc222596816)

[FrmBillSummary 76](#_Toc222596817)

[FrmExportData 78](#_Toc222596819)

[FrmUtlity 84](#_Toc222596820)

[FrmCompactAndRepairDatabase 86](#_Toc222596821)

[FrmBackupDatabase 88](#_Toc222596822)

[FrmCompanyInfo 91](#_Toc222596823)

[FrmClientInfo 96](#_Toc222596824)

[FrmChangePassword 101](#_Toc222596825)

[ModGen 103](#_Toc222596826)

[ModBrowse 105](#_Toc222596827)

[ModRsToWord 107](#_Toc222596828)

[eBilling and Invoice System Test Plan 111](#_Toc222596829)

[Introduction 111](#_Toc222596830)

[Test Scope 111](#_Toc222596831)

[Test Strategy 111](#_Toc222596832)

[Preconditions 112](#_Toc222596833)

[Test Priorities 112](#_Toc222596834)

[Test Techniques 112](#_Toc222596835)

[Test Organization 113](#_Toc222596836)

[Roles and Responsibilities 113](#_Toc222596837)

[Deliverables 113](#_Toc222596838)

[Test Environment 113](#_Toc222596839)

[Hardware and Software 113](#_Toc222596840)

[Testing Automation Software 114](#_Toc222596841)

[Application Configuration 114](#_Toc222596842)

[Test Management 114](#_Toc222596843)

[Testing Schedules 114](#_Toc222596844)

[Threats to Testing 115](#_Toc222596845)

[Conclusion and future enhancement 116](#_Toc222596846)

[Bibliography 117](#_Toc222596847)

[Websites 117](#_Toc222596848)

[Books 117](#_Toc222596849)

# OBJECTIVE

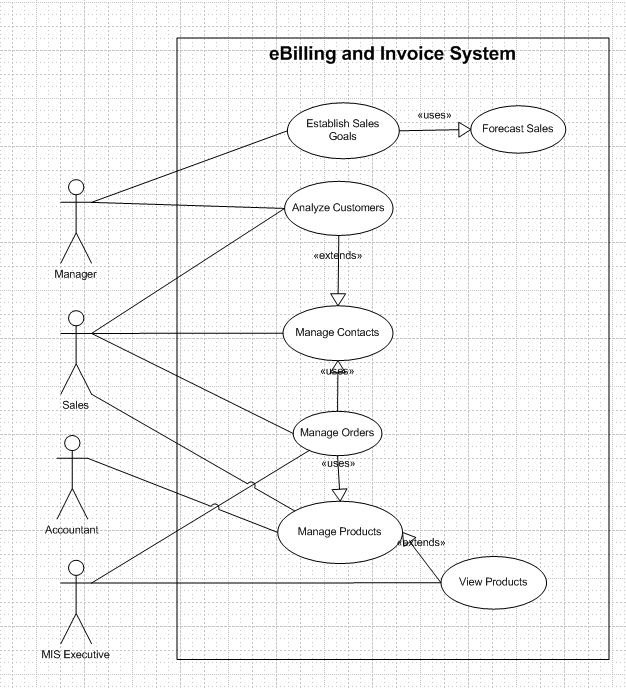
Product Sales Dashboard is basically based on web based application. In this dashboard we can see Specific product sale report between from date to to date. In this System you can see report in Chart Form. Product Sold statistics from date to to date of multiple store in column chart, You can see city wise statistic in column chart but no date wise statistics,

Sales and Marketing personnel are interested in viewing sales performance information, such as what is the sales status from selected date a day wise and also want to see statistics from selected date by city wise but group with stores.

We have decided to implement a Sales dashboard to provide specific product sales statistics across the organization with the information they need. The Sales dashboard includes a variety of chart, arranged in two or more dashboard pages.

## Usage Summary

Product Sales Dashboard will address the following use cases. The complete usage scenarios will be completed during the information-gathering process. Use cases will be created and prioritized. Selected use cases will be expanded into usage scenarios and features that are derived from both use cases and the usage scenarios, as represented in the following diagram:



Usage summary use case

## Requirements Summary

The following preliminary lists are based on initial interviews

### Business Requirements

The goal for the application is to support an management guy to see the statistics of specific product according to the date wise from date to to date by a daywise and then group by stores and you also see the statistics of citywise from date to to date but group in cities and then stores .

Business requirements are discussed in the Scope section, with the following additional detail:

* Sales representatives need to login a our application and after that they need to select a from date, to date and campaign which statistics you want to see and then you can see statistics

* Each sales representative should have a login credentials and then he can login and use this statistics.
* Sales or Marketing representative don’t need to update or upload sold data.
* Only when product billing will be done at that time billing application need to send that update.
* This application helps Sales and Marketing representative to see statistics.
* Sales and Marketing representative don’t need to do excel work to see these statistics.

Sales or Marketing representative wants to improve their current ability to analyze their product sale. In particular, they want to focus on identifying their best product sales and concentrate to sale that product efficiently. To enable them to accomplish this goal, they want to extract meaningful data that easily answers the following questions:

* How to get Product Sales Data?
* What is the product product sales?
* Which product sales we need to concentrate?
* Geographically, See the best product sales in the city?
* Which product sales are too much and what is the price of that product?

### User Requirements

User requirements are categorized by user type.

**Sales Staff (Representatives and Managers)**

* View the data in various ways, for example:
  + Customers who are the top buyers of specific items
  + Best customers based on criteria to be determined
  + Best customers based on geographical analysis
  + Drops in a customer's sales
* Store multilingual and multiregional information in the database rather than relying on the sales staff to translate the information
* Identify which product prices have been modified, especially on current orders in progress
* Use opportunity rules, which are statements that help the sales representative convert an opportunity into a sale
* Add third-party data sources and financial evaluation tools
* Identify where promotions and programs would be the most beneficial
* Apply discounts to customer orders:
  + Sales representatives can offer discounts up to 15 percent, or up to 20 percent with authorization.
  + Sales managers can offer discounts up to 20 percent
* Enable capture, analysis, and sharing of data about a customer across the company
* Use forecasts to establish sales goals
* Allow all staff to view all contacts, but allow each contact to be assigned to a sales representative

##### Accountant

* Update product details, including price, photo, and description
* Add, delete, and update product specifications
* Generate stock reports and check the availability, order the item as and when needed.

##### MIS Executive

* Generate Reports as per requirement.
* Share the data, and convert reports to different format like Text, CSV.

### Operational Requirements

The following requirements provide a high-level view of how the system will run:

* Processor usage should not exceed 80 percent during concurrent uses.
* Backups will occur incrementally throughout the day.
* A full weekly backup is required to WORM drives.
* Ensure that information is easy to access either, and meaningful for the sales representative and the company.
* Minimize the technical knowledge that sales and marketing staff need to access the data, generate ad hoc queries, track promotions, and view customer segmentation information.
* Any change to information must be reflected immediately, and the changes must be propagated to the search engine so that employees that perform searches see this new information.
* The application should work with the existing communications and networking infrastructure.
* The application should deploy with a minimum of additional operational processes, manual or otherwise.

### System Requirements

These are additional constraints from a system perspective:

* Previous data of customer, product details must be imported in the new system.
* The administrator must be able to monitor everything from the IT department.
* The information must be accessible by everyone in the company as per the rights specify.

## Success Criteria

To determine the success of this project, the following metrics can be quantified and used to analyze success factors.

### Sales Automation

* **Problem customer identification**. Ability to identify top 10 developing problem customers by sales representative.
* **Identification of best customers across product lines and regions**. All customers can be sorted by the factors (to be determined) that rank them qualitatively. Variable sorts and rankings can be chosen by the sales representative.
* **Identification of issues across groups**. All customers can be sorted by the issues (to be determined) within group categories that can be updated. Variable sorts and rankings can be chosen by the sales representative or other user.
* **Analysis of quantities and rates of products**. Products can be sorted by the quantity and dates sold. Variable sorts and rankings can be chosen by the user.

# Design Goals and Constraints

## Performance

No more than a 5-percent degradation in average query response is allowed while all concurrent user are using the system.

Processor utilization should not exceed 80 percent during all concurrent users are using the system.

## Availability

Because the system is accessed by sales representative to book the order and print the invoice, and their should not be any single point of failure.

## Reliability

Because of the need no single point failure, automatic failover will be required. In addition, existing disaster recovery and backup plans and procedures must be revised to incorporate the eBilling and Invoicing System.

## Scalability

eBilling and Invoice System an average load of 50 concurrent users after the system is fully operational, and expects that to grow by 5 percent each year for the next five years.

## Security

For the sensitive customer and order data, all users will need to log on the system with their user id and password.

Every resource in the system are defined by the role and privileged. System administrator assigned user role and privileged for their access rights.

## Interoperability

In Version 1.0 of the eBilling and Invoice System, there are no requirements for interoperability with other systems.

## Location

The eBilling and Invoice System are used in the company office only.

## Setup/Installation

Setup and installation must not interrupt the sales staff’s daily tasks and work flow.

# Use Case Summary

## Manage Orders

### Description

This function will enable sales representative to place orders for eBilling and Invoice System products, track submitted orders, and view completed orders and print the invoice.

### Business Need

This function will enable sales representative to interact with eBilling and Invoice System catalog directly without the intervention of any other employees.

### Priority

High

## Manage Products

### Description

This function allows user of eBilling and Invoice System to create, maintain, and delete information about products in the product database that can then be viewed and ordered by customers, and accessed and updated by the sales staff.

### Business Need

This function will support the sales staff with accurate and relevant information, and will also support the Manage Orders function by ensuring up-to-date product information.

### Priority

High

## Manage Contacts

### Description

This function enables the sales staff to create and manage contacts as well as share contact information with the rest of the company.

### Business Need

This function enables user of eBilling and Invoice System to have accurate and up-to-date contact information when working with customers.

### Priority

Medium

## Analyze Customers

### Description

This function allows management to analyze the customer database and find out information, such as the identities of the best customers, the top buyers, and the most popular products.

### Business Need

This function will allow management to determine the most (and least) profitable customers and product lines, enabling better decision making in the running of the business.

### Priority

Medium

## Forecast Sales

### Description

This function allows management to plan and track production costs, track sales results, and plan sales staffing needs.

### Business Need

This function will allow management to manage the revenue streams of the organizations and control costs.

### Priority

Medium

## Establish Sales Goals

### Description

This function allows management to view current sales trends and employee performance and to determine sales goals for the sales staff within the organization.

### Business Need

This function will allow management to set goals for staff, review the performance of staff members, and track employee performance.

### Priority

Medium

## Usage Scenarios

Usage scenarios that describe how users will use the solution within the context of the business, See following usage scenarios.

### Use Case ID: UC 4.7.1 Retrieve Customer Data

Title: UC 04.7.1 Retrieve Customer Data

Abbreviated Title: Retrieve Customer Data

Requirement ID: 21

### *Intent*

Return the selected information about one or more customers.

### *Scenario Narrative*

**A sales representative may want to search for and retrieve information about one or more customers for use in analysis and forecasting activities.**

### *Assumptions/Preconditions*

1. Sales representative has access rights to view customer data.

### *Actors*

1. Sales Representative

### *Basic Course*

1. Use case begins when the sales representative decides to retrieve customer information.
2. System prompts for the type of customer information that is required.
3. Sales representative selects the type of customer information to be retrieved.
4. System presents the appropriate customer information to the sales representative.
5. Use case ends when the selected information is passed to a method of delivery.

### *Alternate Course*

1. If no appropriate customer information can be found, system reports this fact.
2. Use case restarts to enable sales representative to update the type of customer information.

### *Uses/Extends*

1. None

### *User Implementation Requests*

1. None

### *Frequency*

Quite frequent

### *Authority*

1. Not applicable

### Associated *Specifications*

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Name** | **Category** | **Type** |
|  |  |  |  |
|  |  |  |  |

### *Issues*

1. None identified

### *Decision Points*

1. None

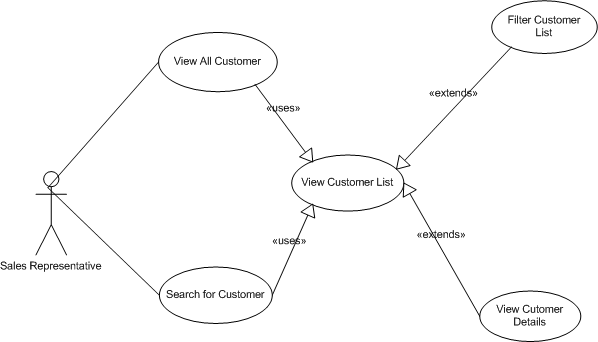
### *Future Requirements*

1. The sales representative may want to apply filters to the retrieved data.
2. The sales representative may want to sort the data. For example, the sales representative may want to list the names of customers in chronological order based on the sale date.

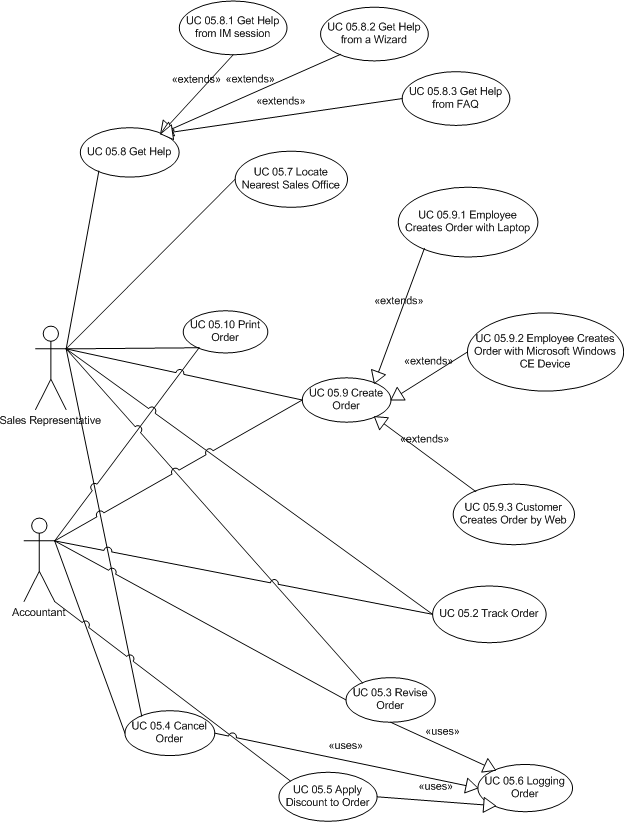
### *Modification*

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |

### *Use Case Model: Retrieve Customer Data*



***Use Case: Manage Orders***



## Use Case ID: UC 04.7.3 Manage Contacts

Title: UC 04.7.2 Manage Contacts

Abbreviated Title: Manage Contacts

Requirement ID: 19

## Intent

Enable actor to create, view, modify of the information about customer contacts.

## Scenario Narrative

**An actor wants to create view, modify, and delete the customer contacts.**

## Assumptions/Preconditions

* The actor has rights to access to the eBilling and Invoice System, and having rights to view, ads, edit and delete the customer contacts.

## Actors

* Sales Representative
* Manager
* Accountant

## Basic Course:

* Use case begins when actor decides to search customer.
* Actor searches the customer.
* System displays the customer details.
* Actor view the customer contact details:
* Actor add, modify the contact details.

## Uses/Extends

1. None

## User Implementation Requests

1. None

## Frequency

Frequent

## Authority

1. Not applicable

## Associated Specifications

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Name** | **Category** | **Type** |
|  |  |  |  |
|  |  |  |  |

## Issues

1. None

## Decision Points

1. None

## Future Requirements

1. None

## Modification

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |

# Solution Architecture

## Infrastructure

New servers must be dual processor 1.26 GHz computers, with 1 GB of RAM. The operating system software will be Microsoft Windows 2000 with the latest service pack, although use of Windows Server 2003 will be considered when it is available.

These servers will be installed on existing network.

To help provide security of the application, both an internal and external firewall will be installed.

## Visual Design

See later on the visual interface design..

## Conceptual Design

For information about the conceptual design see the use case of previous chapters of the eBilling and Invoice System,

Business Rule Catalog of eBilling and Invoice System

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BR ID** | **Business Rule Title** | **Description** | **Authority** | **Current Process** |
| **1** | Timely Product Information Adjustments | The product information, price in the system must be current 100% of the time. Adjustments to this information must be made at any time. | Mr. Ajay Kumar, Accountant. Interview dated May 6, 2008 | 1. Product is identified by Manager 2. Current information is reviewed and modifications are identified. 3. Accountant notes changes to information and obtains approval from Manager. 4. Accountant makes the changes to the product information. 5. After approval is given by all owners, Accountant saves the changes. |
| **2** | Reactivating Deleted Products | A product can be "deleted," meaning that it is no longer sold by eBilling and Invoice System. However, the product information should be available in case the product is reactivated at a future time. | Mr. Ajay Kumar, Accountant. Interview dated May 6, 2008 | 1. Product is identified by Manager. 2. Accountant is notified that the product should be removed from the catalog. Notification comes from Manager. 3. Accountant archives product information and saves the information in text and graphics files on an internal business server. 4. If a product is reactivated, the Accountant receives notification from Manager. 5. Accountant creates a new product record in the database and copies the information from the files on the internal business server to the database record. 6. Accountant notifies product information "owners" in Manager to review the new product record. 7. After approval is given by all owners, Accountant makes the product record available to the catalog. |
| **3** | Authorization for Catalog Changes | No change to product information can be made publicly available without the approval from the information "owners," usually Manager. | Mr. Ajay Kumar, Accountant. Interview dated May 6, 2008 | Derived from BR1 |
| **4** | Delivery of Daily Pricing Updates to Sales | The Sales force needs to use the most up-to-date information about product pricing and related information, such as availability. Otherwise, sales representatives miss receiving changes that do affect them. This results in the sales representatives having to redo orders and recalculate pricing. | Mr. Ajay Kumar, Accountant. Interview dated May 6, 2008 | Every morning of every day:  1. Accountant Update the pricing from the vendor. |
| **5** | Authorization for Information Access | Certain roles have access to certain information types and functions. Current roles include Sales Representative, Sales Manager, Accountant, MIS Executive | Mr. Joydeep Das, IT Manager. Interview dated May 12, 2008 | 1. A suitable authority is requested for creation of a role.  2. The role is created and authorized by the authorization authority. |

## Logical Design

***EBilling and Invoice System Service Matrix***

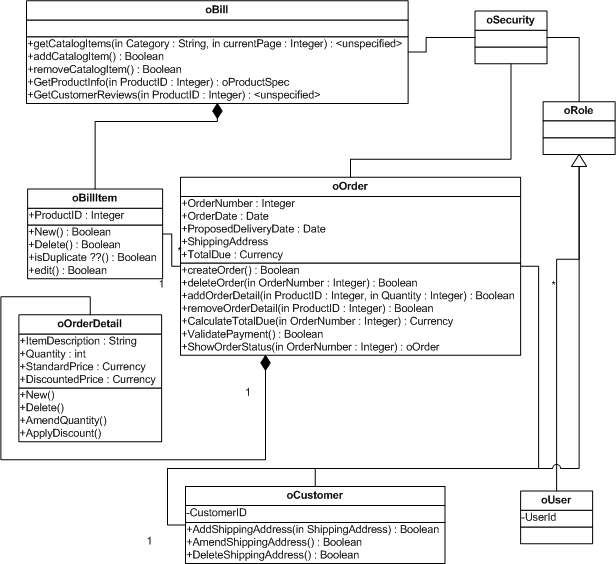
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Business Objects** | **Services** | **Actors (Roles)** | **Responsibilities** | **Collaborations** | **Use Case Reference** |
| Products | Manage Products | * Sales representative | * Set filtering information * Get price modifications | * Select a catalog item from catalog item | Use Case : Manage Product |
| * Sales representative | * Select a catalog item * Get details of catalog item * Get product specs | * None | Use Case : Manage Product |
| * Sales representative | * Set address details * Get address list * Validate address * Confirm delivery | * Set delivery address from order | Use Case : Manage Product |
| * Sales representative | * Print product specs | * None | Use Case : Manage Product |
| * Sales representative | * View product specs | * None | Use Case : Manage Product |
| * Customer * Sales representative | * View summary information | * None | Use Case : Manage Product |
| * Sales representative | * View detailed information | * None | Use Case : Manage Product |
| Edit Products | * Accountant | * Add item to catalog | * Browse catalog items * Set item details from catalog item | Use Case : Edit Product |
| * Accountant | * Get editable items * Set item details * Validate item details * Save item details | * Get details of catalog item from catalog item | Use Case : Edit Product |
| Search Products | * Sales representative * Accountant * Sales manager | * List categories * Browse to category * List catalog items | * View summary information from catalog item | Use Case : Manage Product |
| * Sales representative * Accountant * Sales manager | * Set search criteria * Search catalog | * None | Use Case : Manage Product |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Order | Manage an order | * Sales representative | * Set dispatch mechanism * Enter dispatch details * Validate dispatch details | * Get details of catalog item from catalog item | Use Case : Manage Orders |
| * Sales representative | * Set delivery address | * None | Use Case : Manage Orders |
| * Sales representative | * Search for an order * Get order details * Set order details | * Get order details from order | Use Case : Manage Orders |
| * Sales representative | * Search for an order * Get order details * Cancel order | * Get order details from order | Use Case : Manage Orders |
| * Sales representative | * Create Order * Set order details * Add catalog item to order * Set delivery address * Set payment details * Confirm order * Synchronize order with CE device | * Browse catalog items | Use Case : Manage Orders |
| * Sales representative | * Synchronize order with laptop * Set customer signature | * Browse catalog items | Use Case : Manage Orders |
| * Sales representative * Sales manager | * Apply a discount | * None | Use Case : Manage Orders |
| Track an order | * Sales representative | * Get order status | * None | Use Case : Manage Orders |
| * Sales representative | * Search for an order * Get order details * Set discount amount * Validate discount amount * Set order details | * Get details of catalog item from catalog item | Use Case : Manage Orders |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Customer | Track a customer | * Sales representative | * Get customer details * Search for a customer | * None | Use Case : Retrieve Customer Data |

## Physical Design

***eBilling and Invoice System Physical Object Model***



***Add new product to product master sequence diagram***

# 

eBilling and Invoice System Data Dictionary

## Table : product\_master

**Owner:** dbo

**Destination DB name:** ebill

**Number of columns:** 5

**Number of indexes:** 2

**Number of foreign keys:** 0

**Extended attributes:**

**OnFileGroup** PRIMARY

**Clustered PK** Yes

|  |  |  |  |
| --- | --- | --- | --- |
| **Columns** | **Data type** | **Allow NULLs** | **Value/range** |
| **Sno** | int | Not allowed |  |
| **ProdType** | nvarchar(60) | Not allowed |  |
| **ProdSubType** | nvarchar(60) | Not allowed |  |
| **Rate** | int | Not allowed |  |
| **isactive** | nvarchar(30) | Not allowed |  |

|  |  |  |
| --- | --- | --- |
| **Indexes** | **Type** | **Columns** |
| **PK\_ProductMaster\_sno** | clustered, unique, primary key located on PRIMARY | sno |

## Table : bill

**Owner:** dbo

**Destination DB name:** ebill

**Number of columns:** 15

**Number of indexes:** 3

**Number of foreign keys:** 0

**Extended attributes:**

**OnFileGroup** PRIMARY

**Clustered PK** Yes

|  |  |  |  |
| --- | --- | --- | --- |
| **Columns** | **Data type** | **Allow NULLs** | **Value/range** |
| **sno** | int | Not allowed |  |
| **invoiceno** | nvarchar(60) | Not allowed |  |
| **canme** | nvarchar(60) | Not allowed |  |
| **Invoice\_date** | datetime | Not allowed |  |
| **location** | nvarchar(60) | Not allowed |  |
| **chalan\_no** | int | Not allowed |  |
| **customer\_name** | nvarchar(100) | Not allowed |  |
| **Customer\_address1** | nvarchar(100) | Not allowed |  |
| **Customer\_address2** | nvarchar(100) | Allowed |  |
| **Total\_amt** | int | Not allowed |  |
| **discount** | int | Not allowed |  |
| **Amt\_words** | nvarchar(100) | Not allowed |  |
| **Paid\_type** | nvarchar(60) | Not allowed |  |
| **Cheque\_no** | nvarchar(60) | Allowed |  |
| **Entry\_date** | datetime | Not allowed |  |

|  |  |  |
| --- | --- | --- |
| **Indexes** | **Type** | **Columns** |
| **PK\_Bill\_SNO** | clustered, unique, primary key located on PRIMARY | sno |
| **IX\_bill\_cname** | nonclustered located on PRIMARY | cname |
| **IX\_bill\_invoiceno** | nonclustered, unique located on PRIMARY | invoiceno |

## Table : bill\_details

**Owner:** dbo

**Destination DB name:** ebill

**Number of columns:** 6

**Number of indexes:** 3

**Number of foreign keys:** 2

**Extended attributes:**

**OnFileGroup** PRIMARY

**Clustered PK** Yes

|  |  |  |  |
| --- | --- | --- | --- |
| **Columns** | **Data type** | **Allow NULLs** | **Value/range** |
| **sno** | int | Not allowed |  |
| **Bill\_sno** | int | Not allowed |  |
| **Prod\_sno** | nvarchar(60) | Not allowed |  |
| **qty** | datetime | Not allowed |  |
| **rate** | nvarchar(60) | Not allowed |  |
| **amt** | int | Not allowed |  |

|  |  |  |
| --- | --- | --- |
| **Indexes** | **Type** | **Columns** |
| **PK\_Bill\_details\_SNO** | clustered, unique, primary key located on PRIMARY | sno |
| **IX\_bill\_details\_bill\_sno** | nonclustered located on PRIMARY | Bill\_sno |
| **IX\_bill\_deatils\_prod\_sno** | nonclustered, unique located on PRIMARY | Prod\_sno |

## Table : Company\_master

**Owner:** dbo

**Destination DB name:** ebill

**Number of columns:** 7

**Number of indexes:** 2

**Number of foreign keys:** 0

**Extended attributes:**

**OnFileGroup** PRIMARY

**Clustered PK** Yes

|  |  |  |  |
| --- | --- | --- | --- |
| **Columns** | **Data type** | **Allow NULLs** | **Value/range** |
| **Company\_name** | nvarchar(60) | Not allowed |  |
| **Address1** | nvarchar(100) | Not allowed |  |
| **Address2** | nvarchar(100) | Allowed |  |
| **city** | nvarchar(60) | Not allowed |  |
| **pin** | int | Not allowed |  |
| **telephone** | nvarchar(60) | Not allowed |  |
| **vatno** | nvarchar(60) | Not allowed |  |

|  |  |  |
| --- | --- | --- |
| **Indexes** | **Type** | **Columns** |
| **PK\_comapny\_master\_company\_name** | clustered, unique, primary key located on PRIMARY | Comapnay\_name |
| **IX\_ comapny\_master\_pin** | nonclustered located on PRIMARY | pin |

## Table : user\_master

**Owner:** dbo

**Destination DB name:** ebill

**Number of columns:** 3

**Number of indexes:** 2

**Number of foreign keys:** 0

**Extended attributes:**

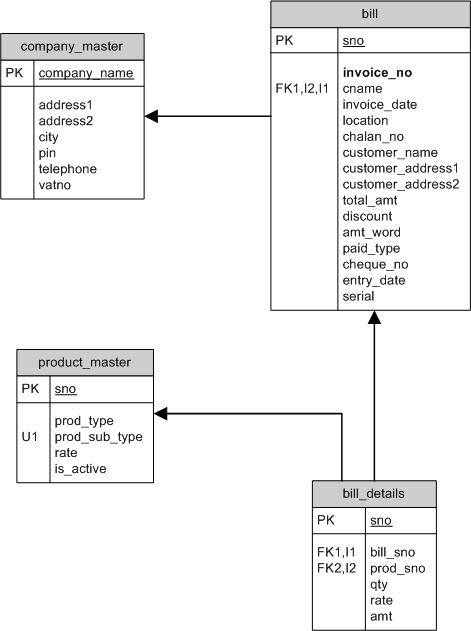
**OnFileGroup** PRIMARY

**Clustered PK** Yes

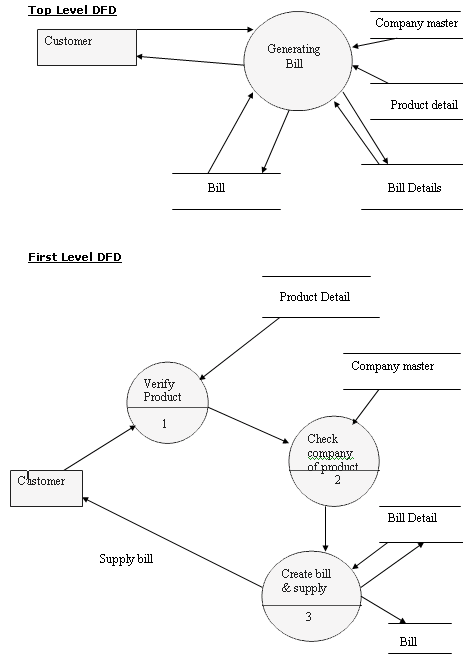
|  |  |  |  |
| --- | --- | --- | --- |
| **Columns** | **Data type** | **Allow NULLs** | **Value/range** |
| **user\_name** | nvarchar(60) | Not allowed |  |
| **User\_password** | nvarchar(60) | Not allowed |  |
| **User\_type** | nvarchar(60) | Allowed |  |

|  |  |  |
| --- | --- | --- |
| **Indexes** | **Type** | **Columns** |
| **PK\_user\_master\_user\_name** | clustered, unique, primary key located on PRIMARY | user\_name |
| **IX\_ user\_master\_user\_type** | nonclustered located on PRIMARY | User\_type |

# eBilling and Invoice System – Data Schema ER Diagram



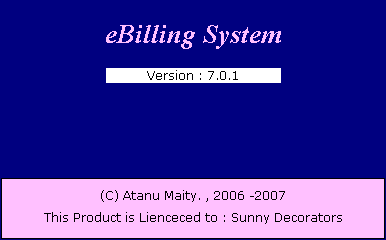
# eBilling and Invoice System DFD (Data Flow Diagram)



# Html Design and Coding

### Login Page

(Module to show startup screen)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Splash Screen

' Used Table : NA

'Module to show startup screen

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Dim r As Integer

Dim i As Integer

Private Sub Form\_Load()

'>>> center the form

Me.Left = (Screen.Height - Me.Height) / 2

Me.Top = (Screen.Width - Me.Width) / 2

'>>> get a random value to decide how many seconds

'>>> startup screen should be displayed

r = Rnd \* 5 + 1

End Sub

Private Sub Form\_Unload(ByVal Cancel As Integer)

'>>> release all the references

FrmSplash = Nothing

End Sub

Private Sub Timer1\_Timer()

'>>> check the ellapsed time

'>>> if the ellapsed time greater then random value

'>>> stored in form load, stop the timer

'>>> show main from and close the startup screen

i = i + 1

If r >= i Then

i = 0

Timer1.Interval = 0

Unload(Me)

Load(FrmMain)

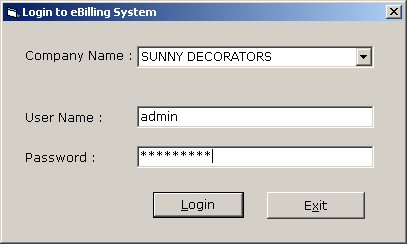
FrmMain.Show()

End If

End Sub

### FrmLogin

(Module to Authenticate User)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Login Module

' Used Table : user\_master

'Module to check user login and load

'user rights as per user type.

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Dim Rs As New ADODB.Recordset

Private Sub Command1\_Click()

'>>> check wheather user name and password are blank

'>>> if its is blan warn user to enter

If TxtUserName.Text = "" Or TxtPassword.Text = "" Then

MsgBox("Enter user name and password ...", vbExclamation)

TxtUserName.SetFocus()

Exit Sub

End If

'>>> check for entered company

'>>> query to database and if no record found warn user to select company from the list.

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select \* from company\_master where company\_name='" & CmbCompanyName.Text & "'", Cn, adOpenStatic, adLockReadOnly)

If Rs.RecordCount > 0 Then

CompanyName = CmbCompanyName.Text

Else

MsgBox("Select company name from the list", vbExclamation)

CmbCompanyName.SetFocus()

Exit Sub

End If

'>>> check for username and password

'>>> query to user\_master with user\_name and password

'>>> if no record found check warn user for enter valid user namne and password

'>>> if record found store user\_nmae, user\_type in global variable for future use.

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select \* from user\_master where USER\_name ='" & TxtUserName.Text & "' and user\_password ='" & TxtPassword & "'", Cn, adOpenStatic, adLockReadOnly)

If Rs.RecordCount > 0 Then

CheckLogin = True

UserName = IIf(IsNull(Rs("USER\_name").Value) = True, "NA", Rs("USER\_name").Value)

UserType = IIf(IsNull(Rs("user\_type").Value) = True, "NA", Rs("user\_type").Value)

Unload(Me)

Else

MsgBox("Invalid User Name and Password ... ", vbExclamation, "Login Error ")

TxtPassword.Text = ""

TxtUserName.SetFocus()

Exit Sub

End If

End Sub

Private Sub Command2\_Click()

'>>> close the application

End

FrmLogin = Nothing

End Sub

Private Sub Form\_Load()

'>>> open the global connection

If Cn.State = 1 Then Cn.Close()

OpenCon()

'>>> center the form

Me.Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2

'>>> fill the combo box with all company\_name from company master

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select \* from company\_master ", Cn, adOpenStatic, adLockReadOnly)

CmbCompanyName.Clear()

If Rs.RecordCount > 0 Then

While Rs.EOF = False

CmbCompanyName.AddItem(Rs("company\_name"))

Rs.MoveNext()

End While

End If

If Rs.State = adStateOpen Then Rs.Close()

End Sub

Private Sub Form\_QueryUnload(ByVal Cancel As Integer, ByVal UnloadMode As Integer)

'>>> release all the object variable used by form

FrmLogin = Nothing

End Sub

### FrmMain

(Give user access to different module)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Main Screen

' Used Table : NA

'Module to show diffrent options

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Private Sub MDIForm\_Activate()

'>>> show the login form as modulas for first time

'>>> when main form activate first time

If CheckLogin = False Then

FrmLogin.Show(1)

End If

LblCompanyName = CompanyName

End Sub

Private Sub MDIForm\_Resize()

'>>> reposition the picturebox as per form size

'>>> to display the selected company name

Picture1.Left = Me.Width - Picture1.Width - 400

End Sub

Private Sub Toolbar1\_ButtonClick(ByVal Button As MSComctlLib.Button)

'>>> load appropriate form as per user click on toolbar buttons

'>>> in design time we set key value for each toolbar button

If Button.Key = "exit" Then

End

ElseIf Button.Key = "product\_master" Then

FrmProducts.Show(1)

ElseIf Button.Key = "bill" Then

FrmBill.Show(1)

ElseIf Button.Key = "utility" Then

FrmUtility.Show(1)

ElseIf Button.Key = "reports" Then

FrmReports.Show(1)

ElseIf Button.Key = "company" Then

FrmChangeCompany.Show(1)

End If

End Sub

### FrmProducts

(Maintain Product master)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Maintain Product Master

' Used Table : product\_master

'

'Module to allow user to select product

'add/modify product details

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

'>> decalre form level valriable

Dim Rs As New ADODB.Recordset

Dim AddEdit As String

Private Sub CmbProductType\_Change()

'>>> as per product type fill the product list

Dim QrStr As String

If CmbProductType.Text = "ALL" Then

QrStr = "select prod\_sub\_type from product\_master order by prod\_sub\_type"

Else

QrStr = "select prod\_sub\_type from product\_master where prod\_type='" & CmbProductType.Text & "' order by prod\_sub\_type"

End If

LstProductSubType.Clear()

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open(QrStr, Cn, adOpenStatic, adLockReadOnly)

While Rs.EOF = False

LstProductSubType.AddItem(Rs("prod\_sub\_type"))

Rs.MoveNext()

End While

'>>> select the first product

If LstProductSubType.ListCount > 0 Then

LstProductSubType.ListIndex = 0

DisplayRecord()

End If

End Sub

Private Sub CmbProductType\_Click()

'>>> call change event

CmbProductType\_Change()

End Sub

Private Sub CmdCancel\_Click()

'>>> cancel update

ED(False, True)

DisplayRecord()

End Sub

Private Sub CmdClose\_Click()

'>>> close the fron

Unload(Me)

End Sub

Private Sub CmdEdit\_Click()

'>>> set flag to edit

ED(True, False)

AddEdit = "EDIT"

End Sub

Private Sub CmdNew\_Click()

'>>> set the flag to add

'>>> claer text box

LblSr.Caption = 0

CmbType.Text = ""

TxtName.Text = ""

TxtRate.Text = 0

ChkActive.Value = 1

ED(True, False)

AddEdit = "ADD"

End Sub

Private Sub CmdSave\_Click()

'>>> validate the entry

If Trim(CmbType.Text) = "" Then

MsgBox("Select or Enter product type.", vbExclamation)

CmbType.SetFocus()

Exit Sub

End If

If Trim(TxtName.Text) = "" Then

MsgBox("Enter product name.", vbExclamation)

TxtName.SetFocus()

Exit Sub

End If

If InStr(1, TxtName.Text, Chr(34)) > 0 Then

MsgBox("Don't use double qoute in product name.", vbExclamation)

TxtName.SetFocus()

Exit Sub

End If

If IsNumeric(TxtRate.Text) = False Then

MsgBox("Enter rate, numeric only", vbExclamation)

TxtRate.SetFocus()

Exit Sub

End If

'>>> check the flag from add/edit

If AddEdit = "ADD" Then

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select max(sno) +1 from product\_master ", Cn, adOpenStatic, adLockReadOnly)

Dim sno As Integer

sno = Rs(0)

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select \* from product\_master where 1=2", Cn, adOpenDynamic, adLockOptimistic)

Rs.AddNew()

Rs("sno") = sno

Rs("prod\_type") = CmbType.Text

Rs("prod\_sub\_type") = TxtName.Text

Rs("rate") = Val(TxtRate.Text)

Rs("is\_active") = Val(ChkActive.Value)

Rs.Update()

Rs.Close()

Else

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select \* from product\_master where sno=" & Val(LblSr.Caption), Cn, adOpenDynamic, adLockOptimistic)

Rs("prod\_type") = CmbType.Text

Rs("prod\_sub\_type") = TxtName.Text

Rs("rate") = Val(TxtRate.Text)

Rs("is\_active") = Val(ChkActive.Value)

Rs.Update()

Rs.Close()

End If

'>>> dispaly and update lists

Dim OldPType As String

OldPType = CmbType.Text

Dim OldPName As String

OldPName = TxtName.Text

CmbType.Clear()

CmbProductType.Clear()

CmbProductType.AddItem("ALL")

CmbProductType.Text = OldPType

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select distinct prod\_type from product\_master order by prod\_type", Cn, adOpenStatic, adLockReadOnly)

While Rs.EOF = False

CmbProductType.AddItem(Rs("prod\_type"))

CmbType.AddItem(Rs("prod\_type"))

Rs.MoveNext()

End While

'>>> fill the product list again with updated/inserted records

Dim QrStr As String

If CmbProductType.Text = "ALL" Then

QrStr = "select prod\_sub\_type from product\_master order by prod\_sub\_type"

Else

QrStr = "select prod\_sub\_type from product\_master where prod\_type='" & CmbProductType.Text & "' order by prod\_sub\_type"

End If

LstProductSubType.Clear()

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open(QrStr, Cn, adOpenStatic, adLockReadOnly)

While Rs.EOF = False

LstProductSubType.AddItem(Rs("prod\_sub\_type"))

Rs.MoveNext()

End While

'>>> show the first record

If LstProductSubType.ListCount > 0 Then

LstProductSubType.Text = OldPName

DisplayRecord()

End If

'>>> enable/diable button

ED(False, True)

End Sub

Private Sub Form\_Load()

'>>> center the form

Me.Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2

'>>> reset connection

OpenCon()

ED(False, True)

CmbType.Clear()

'>>> fill the product type

CmbProductType.Clear()

CmbProductType.AddItem("ALL")

CmbProductType.Text = "ALL"

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select distinct prod\_type from product\_master order by prod\_type", Cn, adOpenStatic, adLockReadOnly)

While Rs.EOF = False

CmbProductType.AddItem(Rs("prod\_type"))

CmbType.AddItem(Rs("prod\_type"))

Rs.MoveNext()

End While

'>>> fill the product sub type

Dim QrStr As String

If CmbProductType.Text = "ALL" Then

QrStr = "select prod\_sub\_type from product\_master order by prod\_sub\_type"

Else

QrStr = "select prod\_sub\_type from product\_master where prod\_type='" & CmbProductType.Text & "' order by prod\_sub\_type"

End If

LstProductSubType.Clear()

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open(QrStr, Cn, adOpenStatic, adLockReadOnly)

While Rs.EOF = False

LstProductSubType.AddItem(Rs("prod\_sub\_type"))

Rs.MoveNext()

End While

'>>> select the first record

If LstProductSubType.ListCount > 0 Then

LstProductSubType.ListIndex = 0

DisplayRecord()

End If

End Sub

Private Sub DisplayRecord()

'>>> display record as per selected product name

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select \* from product\_master where prod\_sub\_type=" & Chr(34) & LstProductSubType.Text & Chr(34), Cn, adOpenStatic, adLockReadOnly)

If Rs.RecordCount > 0 Then

LblSr.Caption = Rs("sno")

CmbType.Text = Rs("prod\_type")

TxtName.Text = Rs("prod\_sub\_type")

TxtRate.Text = Rs("rate")

ChkActive.Value = Rs("is\_active")

Else

LblSr.Caption = ""

CmbType.Text = ""

TxtName.Text = ""

TxtRate.Text = ""

ChkActive.Value = 1

End If

End Sub

Private Sub ED(ByVal T1 As Boolean, ByVal T2 As Boolean)

'>>> enable/disable button

CmdSave.Visible = T1

CmdCancel.Visible = T1

CmdFind.Visible = T2

CmdNew.Visible = T2

CmdEdit.Visible = T2

CmdClose.Visible = T2

CmbType.Locked = T2

TxtName.Locked = T2

TxtRate.Locked = T2

ChkActive.Enabled = T1

End Sub

Private Sub LstProductSubType\_Click()

DisplayRecord()

End Sub

Private Sub TxtName\_KeyDown(ByVal KeyCode As Integer, ByVal Shift As Integer)

'>>> SELECT FROM LIST

If CmbProductType.Text <> "ALL" Then

CmbProductType.Text = "ALL"

End If

If KeyCode = vbKeyDown Then

If LstProductSubType.ListIndex < LstProductSubType.ListCount - 1 Then

LstProductSubType.ListIndex = LstProductSubType.ListIndex + 1

End If

End If

If KeyCode = vbKeyUp Then

If LstProductSubType.ListIndex > 0 Then

LstProductSubType.ListIndex = LstProductSubType.ListIndex - 1

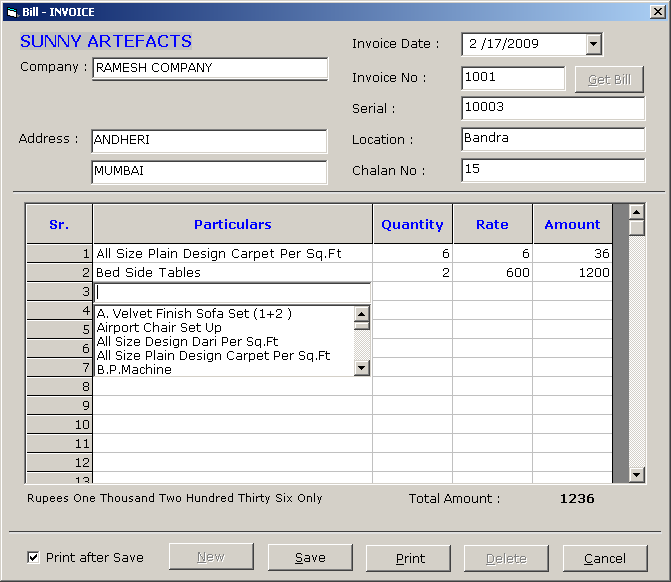
End If

End If

End Sub

### FrmBill

(Create and modify bill and print the saved bill)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' New/Modify Bill Module

' Used Table : Bill

' : Bill Details

' : Product

' : Company

'Module to make new or modify bill,

'Print bill after save the record

'also we make editable flexgrid with

'text box and combobox

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

'>>> declare form global variable

Dim AddEdit As String

Dim Rs As New ADODB.Recordset

Dim sno As Integer

Dim SavePrint As String

Private Sub CmdClose\_Click()

'>>> check the caption of the button

'>>> close the form or cancel the save

If CmdClose.Caption = "&Close" Then

Unload(Me)

Else

DE(True, False)

End If

End Sub

Private Sub CmdDelete\_Click()

'>>> confirm for deletion of bill

'>>> if user select YES delete the record from bill and bill\_details

'>>> clear the seleted data from the screen for deleted bill

If MsgBox("Record will delete permantley ?", vbYesNo + vbCritical) = vbYes Then

Cn.Execute("delete from bill\_details where bill\_sno =" & sno)

Cn.Execute("delete from bill where sno =" & sno)

AddEdit = ""

CmdGetBill.Caption = "&Find"

CmdDelete.Enabled = False

Call ClearField()

If TxtInvoiceNo.Enabled = True Then

TxtInvoiceNo.SetFocus()

End If

End If

End Sub

Private Sub CmdGetBill\_Click()

'>>> find the bill details

'>>> find the bill by invoice no

AddEdit = ""

If CmdGetBill.Caption = "&Get Bill" Then

TxtInvoiceNo.Enabled = True

Call ClearField()

TxtInvoiceNo.BackColor = vbYellow

TxtInvoiceNo.SetFocus()

CmdGetBill.Caption = "&Find"

Else

Dim RS1 As New ADODB.Recordset

RS1.Open("select \* from bill where invoice\_no =" & Val(TxtInvoiceNo.Text) & " and cname='" & CompanyName & "'", Cn, adOpenStatic, adLockReadOnly)

If RS1.RecordCount > 0 Then

'>>> show details from bill table

DtDate.Value = RS1("invoice\_date")

TxtLocation.Text = RS1("location")

TxtChalanNo.Text = RS1("chalan\_no")

TxtCompany.Text = RS1("customer\_name")

TxtAddress1.Text = RS1("customer\_address1")

TxtAddress2.Text = RS1("customer\_address2")

LblTotalAmount = RS1("total\_amt")

LblRsWord.Caption = RS1("amt\_word")

TxtSerial.Text = RS1("serial")

'>>> show data from bill\_details

Dim Rs2 As New ADODB.Recordset

If Rs2.State = adStateOpen Then Rs2.Close()

Dim Rs3 As New ADODB.Recordset

Rs2.Open("select \* from bill\_details where bill\_sno=" & RS1("sno") & " order by sno ", Cn, adOpenStatic, adLockReadOnly)

If Rs2.RecordCount > 0 Then

Dim i As Integer

Rs2.MoveFirst()

For i = 0 To Rs2.RecordCount - 1

If Rs3.State = adStateOpen Then Rs3.Close()

Rs3.Open("select \* from product\_master where sno =" & Rs2("prod\_sno"), Cn, adOpenStatic, adLockReadOnly)

If Rs3.RecordCount > 0 Then

Mf1.TextMatrix(i + 1, 1) = Rs3("prod\_sub\_type")

End If

If Rs3.State = adStateOpen Then Rs3.Close()

Mf1.TextMatrix(i + 1, 2) = Rs2("qty")

Mf1.TextMatrix(i + 1, 3) = Rs2("rate")

Mf1.TextMatrix(i + 1, 4) = Rs2("amt")

Rs2.MoveNext()

Next

End If

If Rs2.State = adStateOpen Then Rs2.Close()

CmdGetBill.Caption = "&Get Bill"

CmdDelete.Enabled = True

cmbval.Visible = False

CmdPrint.Enabled = True

CmdSave.Enabled = True

'>>> locak the buttons

Mf1.Enabled = True

cmbval.Enabled = True

TXTVAL.Enabled = True

TxtCompany.Locked = False

TxtAddress1.Locked = False

TxtAddress2.Locked = False

DtDate.Enabled = True

TxtLocation.Locked = False

TxtChalanNo.Locked = False

Else

MsgBox("No Previous Details found for invoice..." & TxtInvoiceNo.Text, vbExclamation)

CmdGetBill.Caption = "&Find"

TxtInvoiceNo.Enabled = True

TxtInvoiceNo.SetFocus()

CmdDelete.Enabled = False

CmdPrint.Enabled = False

CmdSave.Enabled = False

Mf1.Enabled = False

cmbval.Enabled = False

TXTVAL.Enabled = False

TxtCompany.Locked = True

TxtAddress1.Locked = True

TxtAddress2.Locked = True

DtDate.Enabled = False

TxtLocation.Locked = True

TxtChalanNo.Locked = True

End If

If RS1.State = adStateOpen Then RS1.Close()

End If

End Sub

Private Sub CmdNew\_Click()

'>>> claer the screen for entering data for new bill

'>>> enable/disable buttons

DE(False, True)

AddEdit = "ADD"

Call ClearField()

'>>> get the new system id from bill

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select max(sno) from bill ", Cn, adOpenStatic, adLockReadOnly)

If Rs.RecordCount > 0 Then

sno = IIf(IsNull(Rs(0)) = True, 0, Rs(0)) + 1

End If

If Rs.State = adStateOpen Then Rs.Close()

TxtCompany.SetFocus()

End Sub

Private Sub CmdPrint\_Click()

'NOTE : it is not the right solution to call crystal report by temp using temp table

'some time it is a good practice for complecated databse relation table

'This may not run properly in multi user environment

'Better approch is passing value by SelectionFormula in crystal report

'but anyway it is a working solution

'>>> find the bill sno from seleted invoice no

'>>> if record found

'>>> delete temp bill na dbill\_details

'>>> insert from bill,bill\_details to temp\_bill, teemp\_bill\_details

Dim RS1 As New ADODB.Recordset

If RS1.State = 1 Then RS1.Close()

RS1.Open("select sno from bill where invoice\_no=" & Val(TxtInvoiceNo.Text) & " and cname ='" & CompanyName & "'", Cn, adOpenStatic, adLockReadOnly)

If RS1.RecordCount > 0 Then

Cn.Execute("delete from temp\_bill\_details")

Cn.Execute("delete from temp\_bill")

Cn.Execute("insert into temp\_bill select \* from bill where sno=" & RS1("sno"))

Cn.Execute("insert into temp\_bill\_details select \* from bill\_details where bill\_sno=" & RS1("sno"))

Call OpenCon()

'>>> call crystal report

Cr1.WindowState = crptMaximized

Cr1.ReportFileName = App.Path & "\reports\bill.rpt"

Cr1.DataFiles(0) = App.Path & "\data.mdb"

Cr1.Action = 1

Else

MsgBox("No Bill found select/enter invoice no for print", vbExclamation)

Exit Sub

End If

End Sub

Private Sub CmdSave\_Click()

'>>> validation

'>>> check the required field

If Trim(TxtCompany.Text) = "" Then

MsgBox("Enter Company Name...", vbExclamation)

TxtCompany.SetFocus()

Exit Sub

End If

If IsNumeric(TxtInvoiceNo.Text) = False Then

MsgBox("Enter only Numeric Invoice No...", vbExclamation)

TxtInvoiceNo.SetFocus()

Exit Sub

End If

If Trim(TxtLocation.Text) = "" Then

MsgBox("Enter Location...", vbExclamation)

TxtLocation.SetFocus()

Exit Sub

End If

If Trim(TxtChalanNo.Text) = "" Then

MsgBox("Enter Chalan No...", vbExclamation)

TxtChalanNo.SetFocus()

Exit Sub

End If

If IsNumeric(TxtChalanNo.Text) = False Then

MsgBox("Enter only Numeric Chalan No...", vbExclamation)

TxtChalanNo.SetFocus()

Exit Sub

End If

'>>> reset the transaction

Call OpenCon()

'>>> create transaction for insert bill and bil details

Cn.BeginTrans()

Dim RS1 As New ADODB.Recordset

'>>> check wheather we need to insert or edit the record

'>>> if it is edit, then delete the old bill and insert new record

If AddEdit <> "ADD" Then

If RS1.State = adStateOpen Then RS1.Close()

RS1.Open("select \* from bill where invoice\_no =" & Val(TxtInvoiceNo.Text) & " and cname='" & CompanyName & "'", Cn, adOpenStatic, adLockReadOnly)

If RS1.RecordCount > 0 Then

sno = RS1("sno")

End If

If RS1.State = adStateOpen Then RS1.Close()

Cn.Execute("delete from bill\_details where bill\_sno =" & sno)

Cn.Execute("delete from bill where sno =" & sno)

AddEdit = ""

End If

'>>> check for product master

'>>> check the grid

'>>> wheather there is a product or not

'>>> wheater they enter any quantity or not

'>>> wheatehr there is any price or not

'>>> if any thing goes wrong show message

Dim cc As Integer

Dim i As Integer

For i = 1 To Mf1.Rows - 1

Dim Ch As Boolean

Ch = False

If Trim(Mf1.TextMatrix(i, 1)) = "" Then

Ch = True

End If

If RS1.State = adStateOpen Then RS1.Close()

RS1.Open("select sno from product\_master where prod\_sub\_type =" & Chr(34) & Mf1.TextMatrix(i, 1) & Chr(34), Cn, adOpenStatic, adLockReadOnly)

If RS1.RecordCount <= 0 Then

Ch = True

End If

If RS1.State = adStateOpen Then RS1.Close()

If Val(Mf1.TextMatrix(i, 2)) = 0 Then

Ch = True

End If

If Val(Mf1.TextMatrix(i, 3)) = 0 Then

Ch = True

End If

If Val(Mf1.TextMatrix(i, 4)) = 0 Then

Ch = True

End If

If Ch = False Then

cc = cc + 1

End If

Next

If cc = 0 Then

MsgBox("No Bill details found for save", vbExclamation)

Exit Sub

End If

'>>> check for duplicate invoice no

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select \* from bill where invoice\_no=" & Val(TxtInvoiceNo.Text), Cn, adOpenStatic, adLockReadOnly)

If Rs.RecordCount > 0 Then

MsgBox("Invalid Invoice No cannot save..", vbExclamation)

TxtInvoiceNo.SetFocus()

Exit Sub

End If

If Rs.State = adStateOpen Then Rs.Close()

'>>> save record in bill table

RS1.Open("select \* from bill where 1=2", Cn, adOpenDynamic, adLockOptimistic)

RS1.AddNew()

RS1("sno") = sno

RS1("invoice\_no") = Val(TxtInvoiceNo.Text)

RS1("cname") = CompanyName

RS1("invoice\_date") = Format(DtDate.Value, "dd-mmm-yy")

RS1("location") = Trim(TxtLocation.Text)

RS1("chalan\_no") = Val(TxtChalanNo.Text)

RS1("customer\_name") = Trim(TxtCompany.Text)

RS1("customer\_address1") = Trim(TxtAddress1.Text)

RS1("customer\_address2") = Trim(TxtAddress2.Text)

RS1("total\_amt") = Val(LblTotalAmount.Caption)

RS1("discount") = 0

RS1("amt\_word") = LblRsWord

RS1("paid\_type") = "NA"

RS1("cheque\_no") = "NA"

RS1("entry\_date") = Now

RS1("serial") = TxtSerial.Text

RS1.Update()

If RS1.State = 1 Then RS1.Close()

'>>> vaildate each row before save in details

Dim LastSno As Integer

Dim ProdSno As Integer

Dim Rs2 As New ADODB.Recordset

If Rs2.State = 1 Then Rs2.Close()

Rs2.Open("select max(sno) from bill\_details", Cn, adOpenStatic, adLockReadOnly)

If Rs2.RecordCount > 0 Then

LastSno = IIf(IsNull(Rs2(0)) = True, 0, Rs2(0)) + 1

End If

For i = 1 To Mf1.Rows - 1

Ch = False

If Trim(Mf1.TextMatrix(i, 1)) = "" Then

Ch = True

End If

If RS1.State = adStateOpen Then RS1.Close()

RS1.Open("select sno from product\_master where prod\_sub\_type =" & Chr(34) & Mf1.TextMatrix(i, 1) & Chr(34), Cn, adOpenStatic, adLockReadOnly)

If RS1.RecordCount > 0 Then

ProdSno = RS1(0)

Else

Ch = True

End If

If RS1.State = adStateOpen Then RS1.Close()

If Val(Mf1.TextMatrix(i, 2)) = 0 Then

Ch = True

End If

If Val(Mf1.TextMatrix(i, 3)) = 0 Then

Ch = True

End If

If Val(Mf1.TextMatrix(i, 4)) = 0 Then

Ch = True

End If

If Ch = False Then

'>>> insert in bill details for each validated grid row

If Rs2.State = 1 Then Rs2.Close()

Rs2.Open("select \* from bill\_details where 1=2", Cn, adOpenDynamic, adLockOptimistic)

Rs2.AddNew()

Rs2("sno") = LastSno

Rs2("bill\_sno") = sno

Rs2("prod\_sno") = ProdSno

Rs2("qty") = Val(Mf1.TextMatrix(i, 2))

Rs2("rate") = Val(Mf1.TextMatrix(i, 3))

Rs2("amt") = Val(Mf1.TextMatrix(i, 4))

Rs2.Update()

If Rs2.State = 1 Then Rs2.Close()

LastSno = LastSno + 1

End If

Next

'>>> commit the transaction

Cn.CommitTrans()

MsgBox("Bill Saved", vbInformation)

If ChkPrint.Value = 1 Then

'>>> call the report for print of the saved bill

CmdPrint\_Click()

End If

'>>> prepare for new bill entry

AddEdit = ""

Call ClearField()

Call CmdNew\_Click()

End Sub

Private Sub DtDate\_KeyDown(ByVal KeyCode As Integer, ByVal Shift As Integer)

'>>> move the cursor to new field

If KeyCode = 13 Then

SendKeys("{tab}")

End If

End Sub

Private Sub DtDate\_KeyPress(ByVal KeyAscii As Integer)

'>>> move the cursor to new field

If KeyAscii = 13 Then

SendKeys("{tab}")

End If

End Sub

Private Sub Form\_Load()

'>>> cnter the form

Me.Left = (Screen.Width - Me.Width) / 2

Me.Top = (Screen.Height - Me.Height) / 2

'>>> reset connection

Call OpenCon()

LblCompanyName.Caption = CompanyName

DtDate.Value = Now

'>>> clear the form for new bill entry

Call ClearField()

'>>> enable/diable buttons

DE(True, False)

'>>> load last status for bill print after saved

Dim ChkV As Integer

ChkV = Val(GetSetting("billsystem", "print", "checkprint", "1"))

ChkPrint.Value = ChkV

'>>> load clent name from the table

Rs.Open("select client\_name from client\_master where client\_name is not null order by client\_name", Cn, adOpenStatic, adLockReadOnly)

While Not Rs.EOF

TxtCompany.AddItem(Rs(0))

Rs.MoveNext()

End While

Picture1.Visible = True

End Sub

Private Sub DE(ByVal T1 As Boolean, ByVal T2 As Boolean)

'>>> enable/disable buttons

CmdGetBill.Enabled = T1

CmdNew.Enabled = T1

CmdSave.Enabled = T2

CmdPrint.Enabled = T2

TXTVAL.Enabled = T2

cmbval.Enabled = T2

Mf1.Enabled = T2

If T1 = True Then

CmdClose.Caption = "&Close"

Else

CmdClose.Caption = "&Cancel"

End If

TxtCompany.Locked = T1

TxtAddress1.Locked = T1

TxtAddress2.Locked = T1

DtDate.Enabled = T2

TxtLocation.Locked = T1

TxtChalanNo.Locked = T1

TxtInvoiceNo.BackColor = vbWhite

CmdDelete.Enabled = False

End Sub

Private Sub ClearField()

'>>> clear the fields

TxtCompany.Text = ""

TxtAddress1.Text = ""

TxtAddress2.Text = ""

TxtInvoiceNo.Text = ""

TxtLocation.Text = ""

LblRsWord.Caption = ""

LblTotalAmount.Caption = ""

TxtChalanNo.Text = ""

TxtSerial.Text = ""

'>>> for grid edit

Call set\_heading()

Call move\_textbox()

End Sub

Private Sub Form\_QueryUnload(ByVal Cancel As Integer, ByVal UnloadMode As Integer)

'>>> save last status for bill print after saved in the registry

If ChkPrint.Value = 1 Then

SaveSetting("billsystem", "print", "checkprint", "1")

Else

SaveSetting("billsystem", "print", "checkprint", "0")

End If

End Sub

Private Sub TxtAddress1\_KeyPress(ByVal KeyAscii As Integer)

'>>> move the focus to next control

If KeyAscii = 13 Then

SendKeys("{tab}")

End If

End Sub

Private Sub TxtAddress2\_KeyPress(ByVal KeyAscii As Integer)

'>>> move the focus to next control

If KeyAscii = 13 Then

SendKeys("{tab}")

End If

End Sub

Private Sub TxtChalanNo\_KeyPress(ByVal KeyAscii As Integer)

'>>> move the focus to next control

If KeyAscii = 13 Then

cmbval.SetFocus()

End If

End Sub

Private Sub TxtCompany\_GotFocus()

'>>> make listbox as editable textbox

TxtCompany.Height = 1020

Picture1.Visible = False

End Sub

Private Sub TxtCompany\_KeyDown(ByVal KeyCode As Integer, ByVal Shift As Integer)

'>>> show the address of the seleted company

Dim RS1 As New ADODB.Recordset

If RS1.State = adStateOpen Then RS1.Close()

RS1.Open("select \* from client\_master where client\_name='" & TxtCompany.Text & "'", Cn)

If RS1.RecordCount > 0 Then

TxtAddress1.Text = IIf(IsNull(RS1("address1")) = True, "", RS1("address1"))

TxtAddress2.Text = IIf(IsNull(RS1("address2")) = True, "", RS1("address2"))

Else

TxtAddress1.Text = ""

TxtAddress2.Text = ""

End If

If RS1.State = adStateOpen Then RS1.Close()

End Sub

Private Sub TxtCompany\_KeyPress(ByVal KeyAscii As Integer)

'>>> move the focus to next control

If KeyAscii = 13 Then

SendKeys("{tab}")

End If

End Sub

Private Sub TxtCompany\_LostFocus()

'>>> move the focus to next control

TxtCompany.Height = 330

Picture1.Visible = True

End Sub

Private Sub TxtInvoiceNo\_KeyDown(ByVal KeyCode As Integer, ByVal Shift As Integer)

'>>> call getbill by enter key in invoice no text box

If KeyCode = 13 And CmdGetBill.Caption = "&Find" Then

CmdGetBill\_Click()

End If

End Sub

Private Sub TxtLocation\_KeyPress(ByVal KeyAscii As Integer)

'>>> move the focus to next control

If KeyAscii = 13 Then

SendKeys("{tab}")

End If

End Sub

Private Sub TxtSerial\_KeyPress(ByVal KeyAscii As Integer)

'>>> move the focus to next control

If KeyAscii = 13 Then

SendKeys("{tab}")

End If

End Sub

Private Sub TXTVAL\_KeyPress(ByVal KeyAscii As Integer)

'>>> make flex gird editable move th textbox in gid cell for entering value

'>>> check wheathe we press enter key

'>>> if yes..move the control to next grod

Dim t As Integer

If KeyAscii = 13 And Mf1.Col = 2 Then

If IsNumeric(TXTVAL.Text) = False Then

KeyAscii = 0

Exit Sub

End If

Mf1.Text = TXTVAL.Text

'>>> show total

Mf1.TextMatrix(Mf1.Row, Mf1.Col + 2) = Val(Mf1.TextMatrix(Mf1.Row, Mf1.Col + 1)) \* TXTVAL.Text

Dim x As Integer

Dim T1 As Double

For x = 1 To Mf1.Rows - 1

T1 = T1 + Val(Mf1.TextMatrix(x, 4))

Next

LblTotalAmount.Caption = T1

LblRsWord = RsWord(LblTotalAmount.Caption)

If Mf1.Col <> Mf1.Cols - 3 Then

Mf1.Col = Mf1.Col + 1

Else

If Mf1.Row <> Mf1.Rows - 2 Then

'>>> go to next row

Mf1.Row = Mf1.Row + 1

Else

'>>> add new rows

Mf1.Rows = Mf1.Rows + 1

'>>> set the current row

Mf1.Row = Mf1.Row + 1

'>>> set sr no

Mf1.TextMatrix(Mf1.Row, 0) = Val(Mf1.TextMatrix(Mf1.Row - 1, 0)) + 1

End If

Mf1.Col = 1

End If

move\_textbox()

'>>> select the text in text box

TXTVAL.SelStart = 0

TXTVAL.SelLength = Len(TXTVAL.Text)

End If

End Sub

Private Sub cmbval\_KeyPress(ByVal KeyAscii As Integer)

'>>> make flex gird editable move th combobox in gid cell for seleting value

'>>> check wheathe we press enter key

'>>> if yes..move the control to next grod

Dim t As Integer

If KeyAscii = 13 Then

'>>> find prod rate

Dim RS1 As New ADODB.Recordset

If RS1.State = adStateOpen Then RS1.Close()

RS1.Open("select \* from product\_master where prod\_sub\_type=" & Chr(34) & cmbval.Text & Chr(34), Cn, adOpenStatic, adLockReadOnly)

If RS1.RecordCount > 0 Then

Mf1.TextMatrix(Mf1.Row, Mf1.Col + 2) = RS1("rate")

Else

KeyAscii = 0

Exit Sub

End If

Mf1.Text = cmbval.Text

'>>> show total

Mf1.TextMatrix(Mf1.Row, 4) = Val(Mf1.TextMatrix(Mf1.Row, 2)) \* Val(Mf1.TextMatrix(Mf1.Row, 3))

Dim x As Integer

Dim T1 As Double

For x = 1 To Mf1.Rows - 1

T1 = T1 + Val(Mf1.TextMatrix(x, 4))

Next

LblTotalAmount.Caption = T1

LblRsWord = RsWord(LblTotalAmount.Caption)

If Mf1.Col <> Mf1.Cols - 2 Then

Mf1.Col = Mf1.Col + 1

Else

If Mf1.Row <> Mf1.Rows - 1 Then

Mf1.Row = Mf1.Row + 1

Else

'>>> add new rows

Mf1.Rows = Mf1.Rows + 1

'>>> set the current row

Mf1.Row = Mf1.Row + 1

'>>> set sr no

Mf1.TextMatrix(Mf1.Row, 0) = Val(Mf1.TextMatrix(Mf1.Row - 1, 0)) + 1

End If

Mf1.Col = 1

End If

move\_textbox()

cmbval.SelStart = 0

cmbval.SelLength = Len(cmbval.Text)

End If

End Sub

Public Sub set\_heading()

'>>> creating for the grid

Dim K As Integer

Dim t As Integer

Mf1.Clear()

Mf1.Refresh()

Mf1.Rows = 30

Mf1.Cols = 5

Mf1.Row = 0

Mf1.RowHeight(0) = 600

Mf1.Col = 0

Mf1.ColWidth(0) = 1000

Mf1.CellForeColor = vbBlue

Mf1.CellFontBold = True

Mf1.CellAlignment = 4

Mf1.Text = "Sr."

Mf1.Col = 1

Mf1.ColWidth(1) = 4200

Mf1.CellForeColor = vbBlue

Mf1.CellFontBold = True

Mf1.CellAlignment = 4

Mf1.Text = "Particulars"

Mf1.Col = 2

Mf1.ColWidth(2) = 1200

Mf1.CellForeColor = vbBlue

Mf1.CellFontBold = True

Mf1.CellAlignment = 4

Mf1.Text = "Quantity"

Mf1.Col = 3

Mf1.ColWidth(3) = 1200

Mf1.CellForeColor = vbBlue

Mf1.CellFontBold = True

Mf1.CellAlignment = 4

Mf1.Text = "Rate"

Mf1.Col = 4

Mf1.ColWidth(4) = 1200

Mf1.CellForeColor = vbBlue

Mf1.CellFontBold = True

Mf1.CellAlignment = 4

Mf1.Text = "Amount"

Mf1.TextMatrix(1, 0) = "1"

Mf1.Row = 0

For K = 0 To Mf1.Cols - 1

Mf1.Col = K

Mf1.CellFontBold = True

Next

Mf1.Row = 1

Mf1.Col = 1

'>>> set serial from 1.2...

For K = 1 To Mf1.Rows - 1

Mf1.TextMatrix(K, 0) = K

Next

Mf1.Row = 1

End Sub

Private Sub MF1\_EnterCell()

'>>> call appropriate control for edit the grid

If Mf1.Col = 1 Then

'>>> visble combo box for select product

cmbval.Visible = True

TXTVAL.Visible = False

If cmbval.Visible = True Then

If cmbval.Enabled = True Then

cmbval.SetFocus()

End If

End If

cmbval.Clear()

Dim Rs As New ADODB.Recordset

If Rs.State = 1 Then Rs.Close()

'>>>Fill item

If Mf1.Col = 1 Then

Rs.Open("select prod\_sub\_type from product\_master order by prod\_sub\_type", Cn, adOpenStatic, adLockReadOnly)

While Not Rs.EOF

cmbval.AddItem(Rs(0))

Rs.MoveNext()

End While

ElseIf Mf1.Col = 3 Then

cmbval.AddItem("")

End If

Else

'>>> visble text box for entring quantity

cmbval.Visible = False

TXTVAL.Visible = True

If TXTVAL.Visible = True Then

If TXTVAL.Enabled = True Then

TXTVAL.SetFocus()

End If

End If

End If

Call move\_textbox()

End Sub

Public Sub move\_textbox()

'>>align textbox as per grid cell and set text

TXTVAL.Left = Mf1.CellLeft + Mf1.Left

TXTVAL.Top = Mf1.CellTop + Mf1.Top

TXTVAL.Width = Mf1.CellWidth

TXTVAL.Height = Mf1.CellHeight

TXTVAL.Text = Mf1.Text

'>>align combo box as per grid cell and set text

cmbval.Left = Mf1.CellLeft + Mf1.Left

cmbval.Top = Mf1.CellTop + Mf1.Top

cmbval.Width = Mf1.CellWidth

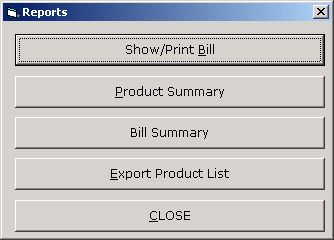
cmbval.Text = Mf1.Text

End Sub

(Print out of the bill)

### FrmReports

(Show Different Report Options)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Show report options

'

'

'allow user to select diffrent report

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Private Sub CmdBillSummary\_Click()

'>>> show bill summary

FrmBillSummary.Show(1)

End Sub

Private Sub CmdClose\_Click()

'>>> cloase the form

Unload(Me)

End Sub

Private Sub CmdExport\_Click()

'>>> show export product form

FrmExportData.Show(1)

End Sub

Private Sub CmdPrintBill\_Click()

'>> show print bill

FrmPrintBill.Show(1)

End Sub

Private Sub CmdProductSummary\_Click()

'>>> show all product list report

Cr1.WindowState = crptMaximized

Cr1.ReportFileName = App.Path & "\reports\products.rpt"

Cr1.DataFiles(0) = App.Path & "\data.mdb"

Cr1.Action = 1

End Sub

Private Sub Form\_Load()

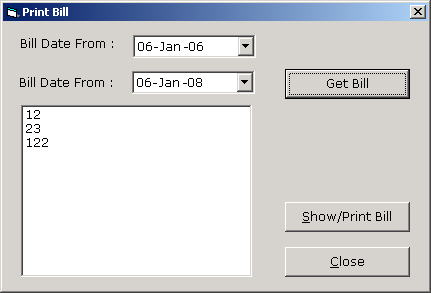
'>>> center the form

Me.Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2

End Sub

### FrmPrintBill

(Print already saved bill)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Print already saved bill

' Used Table : bill

' : bill\_details

'Module to allow user to select the invoice

'no from seleted date range

'and open the bill by its invoice/bill no

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Dim Rs As New ADODB.Recordset

Private Sub CmdClose\_Click()

'>>> close the bill

Unload(Me)

End Sub

Private Sub CmdGetBill\_Click()

'>>> get bill no as per selected date range

'>>> fill the list box from record set

LstInvoiceNo.Clear()

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select invoice\_no from bill where invoice\_date>=#" & Format(DTPicker1.Value, "dd-mmm-yy") & "# and invoice\_date<=#" & Format(DTPicker2.Value, "dd-mmm-yy") & "# and cname='" & CompanyName & "' order by invoice\_no", Cn, adOpenStatic, adLockReadOnly)

While Rs.EOF = False

LstInvoiceNo.AddItem(Rs(0))

Rs.MoveNext()

End While

End Sub

Private Sub CmdShowBill\_Click()

'>>> get teh bill details from bill table as per selected bill no

If LstInvoiceNo.ListIndex < 0 Then Exit Sub

Dim RS1 As New ADODB.Recordset

If RS1.State = 1 Then RS1.Close()

RS1.Open("select sno from bill where invoice\_no=" & Val(LstInvoiceNo.Text) & " and cname ='" & CompanyName & "'", Cn, adOpenStatic, adLockReadOnly)

If RS1.RecordCount > 0 Then

'>>> delete old data from temp table and insert from bill, bill\_deatils table

Cn.Execute("delete from temp\_bill\_details")

Cn.Execute("delete from temp\_bill")

Cn.Execute("insert into temp\_bill select \* from bill where sno=" & RS1("sno"))

Cn.Execute("insert into temp\_bill\_details select \* from bill\_details where bill\_sno=" & RS1("sno"))

'>>> reset the connection

Call OpenCon()

'>>> open the crystal report

Cr1.DataFiles(0) = App.Path & "\data.mdb"

Cr1.WindowState = crptMaximized

Cr1.ReportFileName = App.Path & "\reports\bill.rpt"

Cr1.Action = 1

Else

MsgBox("No Bill found select/enter invoice no for print", vbExclamation)

Exit Sub

End If

End Sub

Private Sub Form\_Load()

'>>> center form

Me.Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2

LstInvoiceNo.Clear()

'>>> set datepicker to current date

DTPicker1.Value = Date

DTPicker2.Value = Date

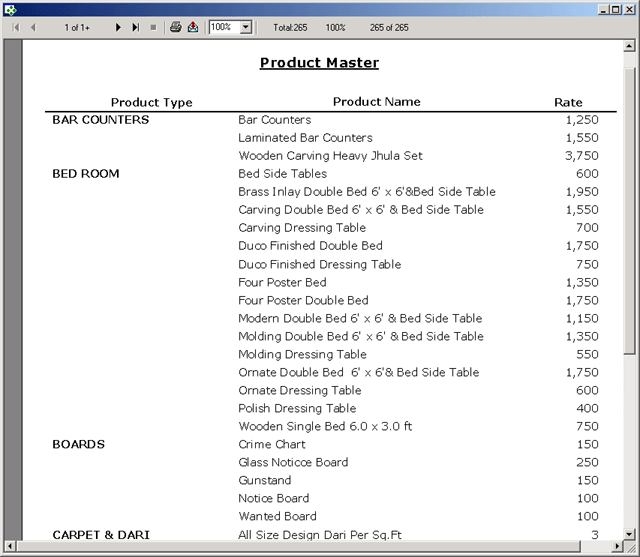
'>>> reset connection

Call OpenCon()

End Sub

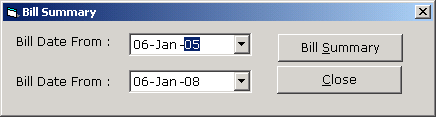
### ProductSummary Report

(Show Product Summary report in crystal report)



### FrmBillSummary

(Show bill summary for particular date range)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Show summery of bill

' Used Table : bill

' : bill\_details

'show bill summary for seleted date

'range, show report in crystal report

'move data into temp table and show

'report from temp table

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Private Sub CmdClose\_Click()

'>>> close the form

Unload(Me)

End Sub

Private Sub CmdGetBill\_Click()

'NOTE : it is not the right solution to call crystal report by temp using temp table

'some time it is a good practice for complecated databse relation table

'This may not run properly in multi user environment

'Better approch is passing value by SelectionFormula in crystal report

'but anyway it is a working solution

'>>> find the bill sno from seleted invoice no

'>>> if record found

'>>> delete temp bill na dbill\_details

'>>> insert from bill,bill\_details to temp\_bill, teemp\_bill\_details

Cn.Execute("delete from temp\_bill\_details")

Cn.Execute("delete from temp\_bill")

Cn.Execute("insert into temp\_bill select \* from bill where invoice\_date>=#" & Format(DTPicker1.Value, "dd-mmm-yy") & "# and invoice\_date<=#" & Format(DTPicker2.Value, "dd-mmm-yy") & "# and cname='" & CompanyName & "' ")

Cn.Execute("insert into temp\_bill\_details select \* from bill\_details where bill\_sno in ( select sno from bill where invoice\_date >=#" & Format(DTPicker1.Value, "dd-mmm-yy") & "# and invoice\_date<=#" & Format(DTPicker2.Value, "dd-mmm-yy") & "# and cname='" & CompanyName & "')")

Call OpenCon()

'>>> open crystal report

Cr1.DataFiles(0) = App.Path & "\data.mdb"

Cr1.WindowState = crptMaximized

Cr1.ReportFileName = App.Path & "\reports\billsummary.rpt"

Cr1.Action = 1

End Sub

Private Sub Form\_Load()

'>>> cnter the form

Me.Left = (Screen.Width - Me.Width) / 2

Me.Top = (Screen.Height - Me.Height) / 2

'>>> show the current date

DTPicker1.Value = Date

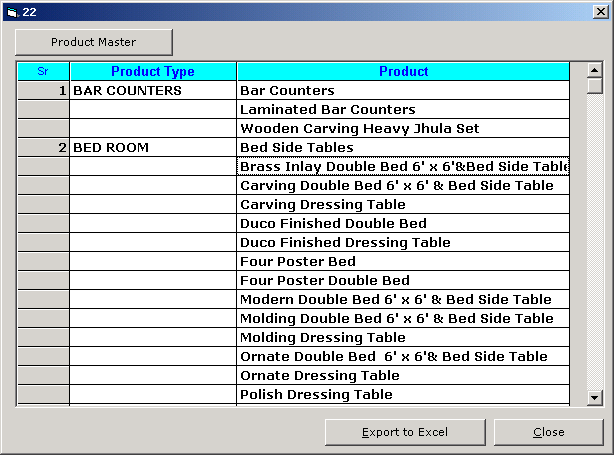
DTPicker2.Value = Date

End Sub

Bill Summary Report

### FrmExportData

(Export product Data in FlexGrid and MS-Excel with formatting)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Export product\_master in grid and ms-excel

' Used Table : product\_master

'open the product\_master in grid

'display record in flex grid with gropu by product type

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Private Sub CmdClose\_Click()

'>>> close the form

Unload(Me)

End Sub

Private Sub CmdExcel\_Click()

'>>>export data into ms excel from grid with formatting

'>>> check the grid

If Mf1.TextMatrix(0, 0) = "" Then

MsgBox("No Records Available for Exporting ... ", vbExclamation)

Exit Sub

End If

Label1.Caption = "WAIT ... Generate Excel "

Label1.Refresh()

'>>> creating excel object variable

Dim ex As New Excel.Application

Dim wb As New Workbook

Dim Es As New Worksheet

wb = ex.Workbooks.Add

Es = wb.Worksheets(1)

Dim i As Integer

Dim j As Integer

'>>> set excel columns width as per flex grid columns width

For i = 0 To Mf1.Cols - 1

Mf1.Row = 1

Mf1.Col = i

Es.Columns(ReturnAlphabet(i + 1) & ":" & ReturnAlphabet(i + 1)).ColumnWidth = Mf1.CellWidth / 110

Next

'>>> set data from grid to excel row, column wise

Dim K As Integer

For i = 0 To Mf1.Rows - 1

For j = 0 To Mf1.Cols - 1

ex.Cells(i + 1, j + 1) = Mf1.TextMatrix(i, j)

Next

Next

Dim R1 As String

Dim R2 As String

R2 = ReturnAlphabet(Mf1.Cols) & "1"

'>>> formatting excel

Dim x As Range

'>>>head

x = Es.Range("A1:" & R2)

x.Font.Bold = True

x.Font.ColorIndex = 40

x.Interior.ColorIndex = 9

x.Interior.Pattern = xlSolid

x.HorizontalAlignment = xlCenter

x.VerticalAlignment = xlBottom

'>>>border

R2 = ReturnAlphabet(Mf1.Cols) & Mf1.Rows - 1

x = Es.Range("A1:" & R2)

With x.Borders(xlEdgeLeft)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With x.Borders(xlEdgeTop)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With x.Borders(xlEdgeBottom)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With x.Borders(xlEdgeRight)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With x.Borders(xlInsideVertical)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With x.Borders(xlInsideHorizontal)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

'>>> FILL DETAILS

x = Es.Range("A2:" & R2)

x.Interior.ColorIndex = 40

'>>TOTAL

R2 = ReturnAlphabet(Mf1.Cols) & Mf1.Rows

x = Es.Range("A" & Mf1.Rows & ":" & R2)

x.Font.Bold = True

x.Font.ColorIndex = 9

'

Es.Name = "Report"

ex.Visible = True

ex.Quit()

wb = Nothing

Es = Nothing

ex = Nothing

'>>> process complete

Label1.Caption = "Ready"

Label1.Refresh()

End Sub

Private Sub CmdProductMaster\_Click()

'>>> reset the grid

Mf1.Rows = 2

Mf1.Cols = 3

Mf1.Clear()

Mf1.Refresh()

Mf1.Row = 0

Mf1.Col = 0

Mf1.ColWidth(0) = 800

Mf1.Text = "Sr"

Mf1.CellAlignment = 4

Mf1.CellFontName = "Arial"

Mf1.Font.Size = 10

Mf1.Font.Bold = True

Mf1.CellForeColor = vbBlue

Mf1.CellBackColor = vbCyan

Mf1.Col = 1

Mf1.ColWidth(1) = 2500

Mf1.Text = "Product Type"

Mf1.CellAlignment = 4

Mf1.CellFontName = "Arial"

Mf1.Font.Size = 10

Mf1.Font.Bold = True

Mf1.CellForeColor = vbBlue

Mf1.CellBackColor = vbCyan

Mf1.Col = 2

Mf1.ColWidth(2) = 5000

Mf1.Text = "Product"

Mf1.CellAlignment = 4

Mf1.CellFontName = "Arial"

Mf1.Font.Size = 10

Mf1.Font.Bold = True

Mf1.CellForeColor = vbBlue

Mf1.CellBackColor = vbCyan

'>>> find distinct product type from product master

'>>> loop all product type

Dim RS1 As New ADODB.Recordset

Dim Rs2 As New ADODB.Recordset

RS1.Open("select distinct prod\_type from product\_master", Cn, adOpenStatic, adLockReadOnly)

Dim i As Integer

Dim j As Integer

For i = 0 To RS1.RecordCount - 1

Me.Caption = i + 1

Mf1.Row = Mf1.Rows - 1

Mf1.Col = 0

Mf1.Text = i + 1

Mf1.Col = 1

Mf1.Text = RS1("prod\_type")

'>>> query product master for each prod type from outer loop

If Rs2.State = adStateOpen Then Rs2.Close()

Rs2.Open("select prod\_sub\_type from product\_master where prod\_type ='" & RS1("prod\_type") & "' order by prod\_sub\_type", Cn, adOpenStatic, adLockReadOnly)

For j = 0 To Rs2.RecordCount - 1

Mf1.Row = Mf1.Rows - 1

Mf1.Col = 2

Mf1.Text = Rs2(0)

Mf1.Rows = Mf1.Rows + 1

Rs2.MoveNext()

Next

RS1.MoveNext()

Next

End Sub

Private Sub Form\_Load()

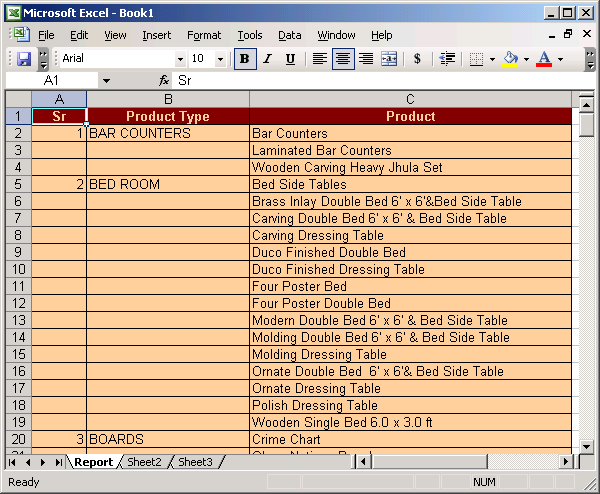
'>>> cnter the form

Me.Left = (Screen.Width - Me.Width)

Me.Top = (Screen.Height - Me.Height)

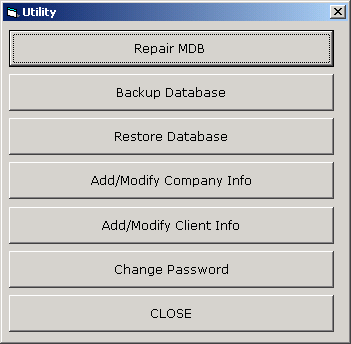
End Sub

Export product Data MS-Excel with formatting



### FrmUtlity

(To Access different utility program like compact, backup, change password etc.)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Show options for utilities

'

'

'allow user to select diffrent options

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Private Sub CmdAddModifyClient\_Click()

FrmClientInfo.Show(1)

End Sub

Private Sub CmdAddModifyCompany\_Click()

FrmCompanyInfo.Show(1)

End Sub

Private Sub CmdBackUp\_Click()

FrmBackupDatabase.Show(1)

End Sub

Private Sub CmdChangePassword\_Click()

FrmChangePassword.Show(1)

End Sub

Private Sub CmdClose\_Click()

Unload(Me)

End Sub

Private Sub CmdRepair\_Click()

FrmCompactAndRepairDatabase.Show(1)

End Sub

Private Sub Command1\_Click()

End Sub

Private Sub Command2\_Click()

End Sub

Private Sub Command4\_Click()

Unload(Me)

End Sub

Private Sub CmdRestore\_Click()

MsgBox("Copy last backup file, rename it to data.mdb and then paste it in application folder." & vbCrLf & vbCrLf & "Contact program vendor..it may cause lost of data, if backup is not current", vbCritical)

End Sub

Private Sub Form\_Load()

'>>> center the form

Me.Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2

'>>> check user type and show diffrent options as per user rights

If LCase(UserType) <> "admin" Then

CmdRepair.Enabled = False

CmdBackUp.Enabled = False

CmdAddModifyCompany.Enabled = False

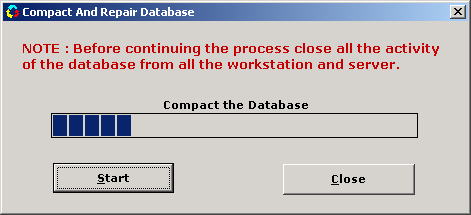
CmdRestore.Enabled = False

End If

End Sub

### FrmCompactAndRepairDatabase

(To shrink and compact main database data.mdb.)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' comapct/shrink the access database

' Used Table : NA

'check the repairdb.mdb file in application path

'if it is already their delete the file

'use DBENGINE CompactDatabase function to comapct the access database

'create new compacted tempdb.mdb from data.mdb.

'delete old data.mdb and rename tempdb.mdb to data.mdb

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Dim dbE As New DAO.DBEngine

Private Sub Command1\_Click()

Unload(Me)

End Sub

Private Sub Command2\_Click()

'>>> if any connection open close all the connection

If Cn.State = 1 Then Cn.Close()

Dim x As String

'>>> check allready file is there or not

x = Dir(App.Path & "\repairedDB.mdb")

'>>> if file present delete the file

If x <> "" Then Kill(App.Path & "\repairedDB.mdb")

Timer1.Enabled = True

'>>> compact teh database

dbE.CompactDatabase(App.Path & "\data.mdb", App.Path & "\RepairedDB")

'>>> delete old database

Kill(App.Path & "\data.mdb")

'>>> rename the new database to old database

Name App.Path & "\repairedDB.mdb" As App.Path & "\data.mdb"

'>>> open connection

Call OpenCon()

End Sub

Private Sub Form\_Load()

'>>> center the form

Me.Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2

'>>> set the progress bar initial value

ProgressBar1.Min = 0

ProgressBar1.Max = 100

End Sub

Private Sub Timer1\_Timer()

'>> show the progress of compact process

If ProgressBar1.Value < 100 Then

ProgressBar1.Value = ProgressBar1.Value + 10

Else

MsgBox("Process Complete Successfully ..", vbInformation)

ProgressBar1.Value = 0 'Reset the min value

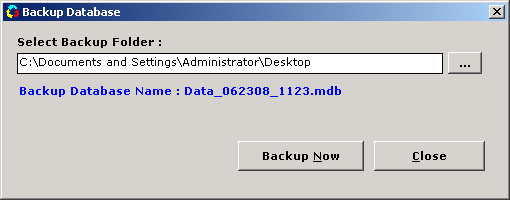
Timer1.Enabled = False 'Disable the Timer

End If

End Sub

### FrmBackupDatabase

(To take backup of main database data.mdb.)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Backup the Database (data.mdb)

' Used Table : NA

'Module to take a copy of data.mdb in

'diffrent location with timestramp

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

'>>> Declare File System variable

Dim Fs As New FileSystemObject

Dim NewFile As String

Private Sub Command1\_Click()

'>>> open folder browser dialog

'>>> select the folder path

'>>> store the path in text box text1

Dim S As String

S = BrowseFolders(hWnd, "Select Folder for Creating Backup file ... ", BrowseForEverything, CSIDL\_DESKTOP)

If S = "" Then

MsgBox("Select Valid Folder for Creating Dump File.", vbInformation, "Creating Dump")

Command1.SetFocus()

Exit Sub

Else

If Fs.FolderExists(S) = False Then

MsgBox("Invalid Folder,Select Valid Folder. ", vbInformation, "Creating Dump")

Command1.SetFocus()

Exit Sub

End If

Text1.Text = S

End If

End Sub

Private Sub Command2\_Click()

On Error GoTo myer1

'>>> check the selected folder wheather

'>>> it is exist or not

If Fs.FolderExists(Text1) = False Then

MsgBox("Invalid Folder,Select Valid Folder. ", vbInformation, "Creating Dump")

Command1.SetFocus()

Exit Sub

End If

'>>> save the settings in registry

SaveSetting("BILLING\_SOFTWARE", "BACKUP\_DATABASE", "BACKUP\_PATH", Text1)

'>>> copy the database file in selected folder for backup

Fs.CopyFile(App.Path & "\data.mdb", Text1.Text & "\" & NewFile, False)

MsgBox("Backup Process Complete.", vbInformation)

Exit Sub

myer1:

'>> check the folder if the file alreday exist warn for overwrite

If Err.Number = 58 Then

If MsgBox("File Allready exist in same name , do you want to overwrite the existing file ..", vbExclamation + vbYesNo + vbDefaultButton1) = vbYes Then

Fs.CopyFile(App.Path & "\data.mdb", Text1.Text & "\" & NewFile, True)

MsgBox("Backup Process Complete", vbInformation)

End If

Else

MsgBox("Can not complete backup following error occured : " & Err.Description, vbCritical)

End If

End Sub

Private Sub Command3\_Click()

Unload(Me)

End Sub

Private Sub Form\_Load()

'>>> center the form

Me.Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2

'>>> new backup file name like data\_040108\_1745.mdb

NewFile = "Data\_" & Format(Now, "ddnnyy\_hhnn") & ".mdb"

Label2.Caption = "Backup Database Name : " & NewFile

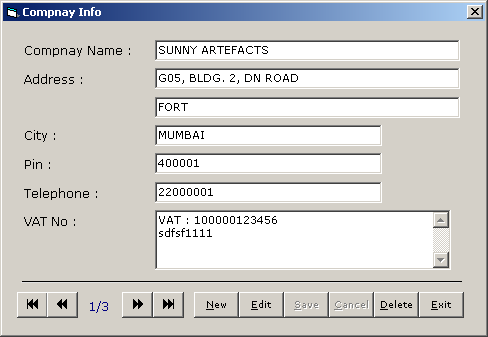
'>>> load the last saved settings from registry

Text1.Text = GetSetting("BILLING\_SOFTWARE", "BACKUP\_DATABASE", "BACKUP\_PATH", "")

End Sub

### FrmCompanyInfo

(Add/Modify Company Info)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' add/edit/delete company details

' Used Table : company\_master

'open the company\_master

'display first record in form load

'add edit save delete and navigation

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Dim RS1 As New ADODB.Recordset

Dim AddEdit As String

Private Sub Command1\_Click()

'>>> close the form

Unload(Me)

End Sub

Private Sub Command11\_Click()

'>>> delete the record

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

On Error GoTo myer1

'>>> confirm before delete

If MsgBox("Delete the Record ? ", vbCritical + vbYesNo) = vbYes Then

RS1.Delete()

Call ClearText()

Command4\_Click()

End If

Exit Sub

myer1:

MsgBox("Error Occured : " & Err.Description, vbCritical)

End Sub

Private Sub Command2\_Click()

'>>> move record ponter to first record

'>>> display first record

On Error Resume Next

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

RS1.MoveFirst()

Call DisplayRecord()

End Sub

Private Sub Command3\_Click()

'>>> move back the record pointer and display current record

On Error Resume Next

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

If RS1.AbsolutePosition > 1 Then

RS1.MovePrevious()

Else

MsgBox("First Record ..", vbInformation)

RS1.MoveFirst()

End If

Call DisplayRecord()

End Sub

Private Sub Command4\_Click()

'>>> move next the record pointer and display current record

On Error Resume Next

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

If RS1.AbsolutePosition < RS1.RecordCount Then

RS1.MoveNext()

Else

MsgBox("Last Record ..", vbInformation)

RS1.MoveLast()

End If

Call DisplayRecord()

End Sub

Private Sub Command5\_Click()

'>>> move last the record pointer and display current record

On Error Resume Next

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

RS1.MoveLast()

Call DisplayRecord()

End Sub

Private Sub Command6\_Click()

'>>> prepare for add record, clear all text box, set flag to ADD

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

AddEdit = "ADD"

Call ClearText()

DE(False, True)

TxtCompanyName.SetFocus()

End Sub

Private Sub Command7\_Click()

'>>> prepare for edit record, set flag to EDIT

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

AddEdit = "EDIT"

DE(False, True)

TxtCompanyName.SetFocus()

End Sub

Private Sub Command8\_Click()

'>>> save the record

'>>> check for validation

'>>> check the flag for ADD/Edit

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

On Error GoTo myer1

If Trim(TxtCompanyName.Text) = "" Then

MsgBox("Enter Company Name ", vbCritical)

TxtCompanyName.SetFocus()

Exit Sub

End If

If AddEdit = "ADD" Then

RS1.AddNew()

RS1("company\_name") = TxtCompanyName.Text

RS1("Address1") = TxtAddress1.Text

RS1("Address2") = TxtAddress2.Text

RS1("city") = TxtCity.Text

RS1("pin") = TxtPin.Text

RS1("telephone") = TxtTelephone.Text

RS1("vatno") = TxtVatNo.Text

RS1.Update()

RS1.MoveLast()

Call DisplayRecord()

Else

RS1("Address1") = TxtAddress1.Text

RS1("Address2") = TxtAddress2.Text

RS1("city") = TxtCity.Text

RS1("pin") = TxtPin.Text

RS1("telephone") = TxtTelephone.Text

RS1("vatno") = TxtVatNo.Text

RS1.Update()

'>>> if it is edit after requery show the edited record

Dim p As Integer

p = RS1.AbsolutePosition

RS1.Requery()

RS1.MoveFirst()

RS1.Move(p - 1)

Call DisplayRecord()

End If

DE(True, False)

Exit Sub

myer1:

MsgBox("Error Occured : " & Err.Description, vbCritical)

End Sub

Private Sub Command9\_Click()

'>>> cancel save

DE(True, False)

End Sub

Private Sub Form\_Load()

'>>> center the form

Me.Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2

'>>> reset connection

'>>> clear all text

ClearText()

OpenCon()

'>>> load alreday saved clent data

'>>> and show the first record

If RS1.State = adStateOpen Then RS1.Close()

RS1.Open("select \* from company\_master order by company\_name ", Cn, adOpenDynamic, adLockOptimistic)

If RS1.RecordCount > 0 Then

RS1.MoveFirst()

Call DisplayRecord()

End If

DE(True, False)

End Sub

Private Sub ClearText()

'>>> clear all text box in the form

Dim Ctl As Control

For Each Ctl In Me.Controls

If TypeOf Ctl Is TextBox Then

Ctl.Text = ""

End If

Next

End Sub

Private Sub DisplayRecord()

'>>> display current record

On Error Resume Next

Call ClearText()

TxtCompanyName.Text = IIf(IsNull(RS1("company\_name")) = True, "", RS1("company\_name"))

TxtAddress1.Text = IIf(IsNull(RS1("Address1")) = True, "", RS1("Address1"))

TxtAddress2.Text = IIf(IsNull(RS1("Address2")) = True, "", RS1("Address2"))

TxtCity.Text = IIf(IsNull(RS1("city")) = True, "", RS1("city"))

TxtPin.Text = IIf(IsNull(RS1("pin")) = True, "", RS1("pin"))

TxtTelephone.Text = IIf(IsNull(RS1("telephone")) = True, "", RS1("telephone"))

TxtVatNo.Text = IIf(IsNull(RS1("vatno")) = True, "", RS1("vatno"))

Label17.Caption = RS1.AbsolutePosition & "/" & RS1.RecordCount

End Sub

Private Sub DE(ByVal T1 As Boolean, ByVal T2 As Boolean)

'>>> enable disable buttons

Command2.Enabled = T1

Command3.Enabled = T1

Command4.Enabled = T1

Command5.Enabled = T1

Command6.Enabled = T1

Command7.Enabled = T1

Command11.Enabled = T1

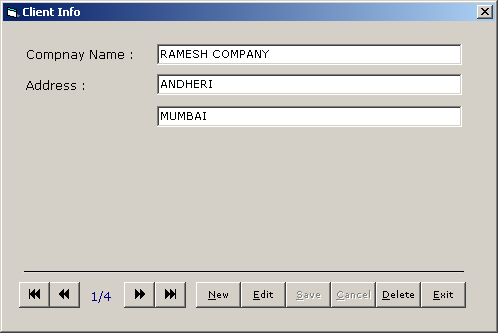
Command8.Enabled = T2

Command9.Enabled = T2

End Sub

### FrmClientInfo

(Add/Modify Client Info)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' add/edit/delete client details

' Used Table : client\_master

'open the client\_master

'display first record in form load

'add edit save delete and navigation

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Dim RS1 As New ADODB.Recordset

Dim AddEdit As String

Private Sub Command1\_Click()

'>>> close the form

Unload(Me)

End Sub

Private Sub Command11\_Click()

'>>> delete the record

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

On Error GoTo myer1

'>>> confirm before delete

If MsgBox("Delete the Record ? ", vbCritical + vbYesNo) = vbYes Then

RS1.Delete()

Call ClearText()

Command4\_Click()

End If

Exit Sub

myer1:

MsgBox("Error Occured : " & Err.Description, vbCritical)

End Sub

Private Sub Command2\_Click()

'>>> move record ponter to first record

'>>> display first record

On Error Resume Next

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

RS1.MoveFirst()

Call DisplayRecord()

End Sub

Private Sub Command3\_Click()

'>>> move back the record pointer and display current record

On Error Resume Next

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

If RS1.AbsolutePosition > 1 Then

RS1.MovePrevious()

Else

MsgBox("First Record ..", vbInformation)

RS1.MoveFirst()

End If

Call DisplayRecord()

End Sub

Private Sub Command4\_Click()

'>>> move next the record pointer and display current record

On Error Resume Next

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

If RS1.AbsolutePosition < RS1.RecordCount Then

RS1.MoveNext()

Else

MsgBox("Last Record ..", vbInformation)

RS1.MoveLast()

End If

Call DisplayRecord()

End Sub

Private Sub Command5\_Click()

'>>> move last the record pointer and display current record

On Error Resume Next

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

RS1.MoveLast()

Call DisplayRecord()

End Sub

Private Sub Command6\_Click()

'>>> prepare for add record, clear all text box, set flag to ADD

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

AddEdit = "ADD"

Call ClearText()

DE(False, True)

TxtCompanyName.SetFocus()

End Sub

Private Sub Command7\_Click()

'>>> prepare for edit record, set flag to EDIT

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

AddEdit = "EDIT"

DE(False, True)

TxtCompanyName.SetFocus()

End Sub

Private Sub Command8\_Click()

'>>> save the record

'>>> check for validation

'>>> check the flag for ADD/Edit

If RS1.State = adStateClosed Then Exit Sub

If RS1.RecordCount <= 0 Then Exit Sub

On Error GoTo myer1

If Trim(TxtCompanyName.Text) = "" Then

MsgBox("Enter Company Name ", vbCritical)

TxtCompanyName.SetFocus()

Exit Sub

End If

If AddEdit = "ADD" Then

RS1.AddNew()

RS1("client\_name") = TxtCompanyName.Text

RS1("Address1") = TxtAddress1.Text

RS1("Address2") = TxtAddress2.Text

RS1.Update()

RS1.MoveLast()

Call DisplayRecord()

Else

RS1("Address1") = TxtAddress1.Text

RS1("Address2") = TxtAddress2.Text

RS1.Update()

'>>> if it is edit after requery show the edited record

Dim p As Integer

p = RS1.AbsolutePosition

RS1.Requery()

RS1.MoveFirst()

RS1.Move(p - 1)

Call DisplayRecord()

End If

DE(True, False)

Exit Sub

myer1:

MsgBox("Error Occured : " & Err.Description, vbCritical)

End Sub

Private Sub Command9\_Click()

'>>> cancel save

DE(True, False)

End Sub

Private Sub Form\_Load()

'>>> center the form

Me.Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2

ClearText()

'>>> reset connection

OpenCon()

'>>> load alreday saved clent data

'>>> and show the first record

If RS1.State = adStateOpen Then RS1.Close()

RS1.Open("select \* from client\_master order by client\_name ", Cn, adOpenDynamic, adLockOptimistic)

If RS1.RecordCount > 0 Then

RS1.MoveFirst()

Call DisplayRecord()

End If

DE(True, False)

End Sub

Private Sub ClearText()

'>>> clear all text box in the form

Dim Ctl As Control

For Each Ctl In Me.Controls

If TypeOf Ctl Is TextBox Then

Ctl.Text = ""

End If

Next

End Sub

Private Sub DisplayRecord()

'>>> display current record

Call ClearText()

TxtCompanyName.Text = IIf(IsNull(RS1("client\_name")) = True, "", RS1("client\_name"))

TxtAddress1.Text = IIf(IsNull(RS1("Address1")) = True, "", RS1("Address1"))

TxtAddress2.Text = IIf(IsNull(RS1("Address2")) = True, "", RS1("Address2"))

Label17.Caption = RS1.AbsolutePosition & "/" & RS1.RecordCount

End Sub

Private Sub DE(ByVal T1 As Boolean, ByVal T2 As Boolean)

'>>> enable disable buttons

Command2.Enabled = T1

Command3.Enabled = T1

Command4.Enabled = T1

Command5.Enabled = T1

Command6.Enabled = T1

Command7.Enabled = T1

Command11.Enabled = T1

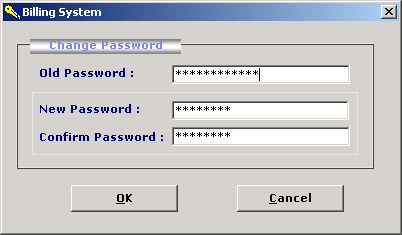
Command8.Enabled = T2

Command9.Enabled = T2

End Sub

### FrmChangePassword

(Change password for currently logged user)



'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' change the current company

' Used Table : company\_master

'Module to allow user to change the

'current comopany from the list

'set company name to global variable

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Dim Rs As New ADODB.Recordset

Private Sub Command1\_Click()

'>>> check the company nmae select by user

'>>> frm the comapny\_master table

'>>> if record found set global variable

'>>> otherwise warn user to select from the list.

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select \* from company\_master where company\_name='" & CmbCompanyName.Text & "'", Cn, adOpenStatic, adLockReadOnly)

If Rs.RecordCount > 0 Then

CompanyName = CmbCompanyName.Text

FrmMain.LblCompanyName = CompanyName

Unload(Me)

Else

MsgBox("Select company name from the list", vbExclamation)

CmbCompanyName.SetFocus()

Exit Sub

End If

End Sub

Private Sub Command2\_Click()

'>>> cloase the form

Unload(Me)

End Sub

Private Sub Form\_Load()

'>>> reset the database connection

If Cn.State = 1 Then Cn.Close()

OpenCon()

'>>> center the form

Me.Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2

'>>> fill the combo box with company name from company\_master

'>>> open record from company\_master

'>>> loop throgh recordset and add each company\_name into combo box

If Rs.State = adStateOpen Then Rs.Close()

Rs.Open("select \* from company\_master ", Cn, adOpenStatic, adLockReadOnly)

CmbCompanyName.Clear()

If Rs.RecordCount > 0 Then

While Rs.EOF = False

CmbCompanyName.AddItem(Rs("company\_name"))

Rs.MoveNext()

End While

End If

If Rs.State = adStateOpen Then Rs.Close()

'>>> set the already selected company name from login form

CmbCompanyName.Text = CompanyName

End Sub

### ModGen

(Module to store global variable and procedure)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'

'declare global variable and procedure

'

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Public Cn As New ADODB.Connection

Public CheckLogin As Boolean

Public UserName As String

Public UserType As String

Public CompanyName As String

Public Sub OpenCon()

'>>> open connction

If Cn.State = 1 Then Cn.Close()

Cn.ConnectionString = "provider=microsoft.jet.oledb.4.0; data source= " & App.Path & "\data.mdb"

Cn.CursorLocation = adUseClient

Cn.Open()

End Sub

Public Function newsno(ByVal table As String) As Integer

'>>> find max sno for passing table

Dim Rs As New ADODB.Recordset

If Rs.State = 1 Then Rs.Close()

Rs.Open("select max(sno) from " & table, Cn, adOpenStatic, adLockReadOnly)

If IsNull(Rs(0)) = True Then

newsno = 1

Else

newsno = Val(Rs(0)) + 1

End If

End Function

Public Function ReturnAlphabet(ByVal n As Integer) As String

'>>> return alphabel as per supplied no

'>>> like 1 - A,2-B, 26-Z, 27-AA, 256-IV

'>>> this function is used to excel formatting to set column value in range

If n < 0 Or n > 256 Then

MsgBox("Invalid Invalid range is 1-256", vbQuestion)

Exit Function

End If

Dim i As Integer

Dim r As Integer

Dim S As String

Dim R1 As Integer

If n <= 26 Then

S = Chr(n + 64)

Else

r = n Mod 26

R1 = n / 26

S = Chr(R1 + 64) & Chr(r + 64)

End If

ReturnAlphabet = S

End Function

### ModBrowse

(Show Folder Browser)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'

'declare global variable and procedure

'it is used to show folder browser

'

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Private Type BrowseInfo

hWndOwner As Long

pIDLRoot As Long

pszDisplayName As Long

lpszTitle As Long

ulFlags As Long

lpfnCallback As Long

lParam As Long

iImage As Long

End Type

'>>> Browsing type.

Public Enum BrowseType

BrowseForFolders = &H1

BrowseForComputers = &H1000

BrowseForPrinters = &H2000

BrowseForEverything = &H4000

End Enum

'>>> Folder Type

Public Enum FolderType

CSIDL\_BITBUCKET = 10

CSIDL\_CONTROLS = 3

CSIDL\_DESKTOP = 0

CSIDL\_DRIVES = 17

CSIDL\_FONTS = 20

CSIDL\_NETHOOD = 18

CSIDL\_NETWORK = 19

CSIDL\_PERSONAL = 5

CSIDL\_PRINTERS = 4

CSIDL\_PROGRAMS = 2

CSIDL\_RECENT = 8

CSIDL\_SENDTO = 9

CSIDL\_STARTMENU = 11

End Enum

Private Const MAX\_PATH = 260

Private Declare Sub CoTaskMemFree Lib "ole32.dll" (ByVal hMem As Long)

Private Declare Function lstrcat Lib "kernel32.dll" Alias "lstrcatA" (ByVal lpString1 As String, ByVal lpString2 As String) As Long

Private Declare Function SHBrowseForFolder Lib "shell32.dll" (ByVal lpbi As BrowseInfo) As Long

Private Declare Function SHGetPathFromIDList Lib "shell32.dll" (ByVal pidList As Long, ByVal lpBuffer As String) As Long

Private Declare Function SHGetSpecialFolderLocation Lib "shell32.dll" (ByVal hWndOwner As Long, ByVal nFolder As Long, ByVal ListId As Long) As Long

'>>>> main function for broser

Public Function BrowseFolders(ByVal hWndOwner As Long, ByVal sMessage As String, ByVal Browse As BrowseType, ByVal RootFolder As FolderType) As String

Dim Nullpos As Integer

Dim lpIDList As Long

Dim res As Long

Dim sPath As String

Dim BInfo As BrowseInfo

Dim RootID As Long

SHGetSpecialFolderLocation(hWndOwner, RootFolder, RootID)

BInfo.hWndOwner = hWndOwner

BInfo.lpszTitle = lstrcat(sMessage, "")

BInfo.ulFlags = Browse

If RootID <> 0 Then BInfo.pIDLRoot = RootID

lpIDList = SHBrowseForFolder(BInfo)

If lpIDList <> 0 Then

sPath = String(MAX\_PATH, 0)

res = SHGetPathFromIDList(lpIDList, sPath)

Call CoTaskMemFree(lpIDList)

Nullpos = InStr(sPath, vbNullChar)

If Nullpos <> 0 Then

sPath = Left(sPath, Nullpos - 1)

End If

End If

BrowseFolders = sPath

End Function

### ModRsToWord

(Convert Rs to Word)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' eBilling System

' Version 1.0.0

' Created by Mr. Atanu Maity

' Date : 21-Aug-2006

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'

'Module to convert Rs to Word

'123.50 = One Hundred Twent Three and Ffty Paise Only

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

'display digit to words

'123 -- > One hundred and twentythree only.

Public Function RsWord(ByVal t As String) As String

On Error Resume Next

Dim la As Boolean

Dim th As Boolean

Dim l As Integer

Dim l1 As Integer

Dim i As String

Dim r As String

Dim p1 As String

Dim NL As Integer

Dim p As Boolean

Dim pl As Integer

Dim a As Integer

Dim b As Integer

Dim c As Integer

Dim z As Boolean

NL = InStr(t, ".")

p = True

pl = Len(t) - NL

If NL = 0 Then

NL = Len(t) : p = False

l = Len(t)

Else

l = NL - 1

End If

l1 = l

While l > 0

i = Mid(t, l1 - (l - 1), 1)

If l = 7 Then la = True

If l = 7 And i = "0" And Mid(t, l1 - (l - 2), 1) <> "0" Then

r = r & SingleDigit(Mid(t, l1 - (l - 2), 1)) & " Lackhs "

End If

If l = 7 And (i <> "0" And i <> "1") Then

r = r & Tenths(i) & " " & SingleDigit(Mid(t, l1 - (l - 2), 1)) & " Lackhs "

End If

If l = 7 And i = "1" Then

r = r & TwoDigit(Mid(t, l1 - (l - 2), 1)) & " Lackhs "

End If

If l = 6 And i <> "0" And la = False Then

r = r & SingleDigit(i) & " Lackhs "

End If

If l = 5 Then th = True

If l = 5 And i = "0" And Mid(t, l1 - (l - 2), 1) <> "0" Then

r = r & SingleDigit(Mid(t, l1 - (l - 2), 1)) & " Thousand "

End If

If l = 5 And (i <> "0" And i <> "1") Then

r = r & Tenths(i) & " " & SingleDigit(Mid(t, l1 - (l - 2), 1)) & " Thousand "

End If

If l = 5 And i = "1" Then

r = r & TwoDigit(Mid(t, l1 - (l - 2), 1)) & " Thousand "

End If

If l = 4 And i <> "0" And th = False Then

r = r & SingleDigit(i) & " Thousand "

End If

If l = 3 And i <> "0" Then

r = r & SingleDigit(i) & " Hundred "

End If

If l = 2 And (i <> "0" And i <> "1") Then

r = r & Tenths(i)

End If

If l = 2 And i = "1" Then

r = r & TwoDigit(Mid(t, l1 - (l - 1), 1)) & " "

End If

If l = 1 And Mid(t, l1 - (l - 2), 1) <> "1" Then

r = r & " " & SingleDigit(i)

End If

l = l - 1

End While

If p = True Then

l1 = NL + 1

While pl > 0

i = Mid(t, Len(t) - pl + 1, 1)

If pl = 2 And (i <> "0" And i <> "1") Then

p1 = p1 & Tenths(i)

End If

If pl = 2 And i = "1" Then

p1 = p1 & TwoDigit(Mid(t, Len(t) - pl + 2, 1)) & " "

End If

If pl = 1 And Mid(t, Len(t) - pl, 1) <> "1" Then

p1 = p1 & " " & SingleDigit(i)

End If

pl = pl - 1

End While

End If

If r <> "" And p1 <> "" Then

RsWord = "Rupees " & r & " and " & p1 & " Paise Only"

End If

If p1 = "" And r <> "" Then

RsWord = "Rupees " & r & " Only"

End If

If r = "" And p1 <> "" Then

RsWord = p1 & " Paise Only"

End If

If r = "" And p1 = "" Then

RsWord = "Nil"

End If

End Function

Private Function TwoDigit(ByVal d As String) As String

Dim S As String

If d = "1" Then S = "Eleven"

If d = "2" Then S = "Twelve"

If d = "3" Then S = "Thirteen"

If d = "4" Then S = "Forteen"

If d = "5" Then S = "Fifteen"

If d = "6" Then S = "Sixteen"

If d = "7" Then S = "Seventeen"

If d = "8" Then S = "Eighteen"

If d = "9" Then S = "Nineteen"

If d = "0" Then S = "Ten"

TwoDigit = S

End Function

Private Function Tenths(ByVal d As String) As String

Dim S As String

If d = "2" Then S = "Twenty"

If d = "3" Then S = "Thirty"

If d = "4" Then S = "Forty"

If d = "5" Then S = "Fifty"

If d = "6" Then S = "Sixty"

If d = "7" Then S = "Seventy"

If d = "8" Then S = "Eighty"

If d = "9" Then S = "Ninety"

Tenths = S

End Function

Private Function SingleDigit(ByVal d As String) As String

Dim S As String

If d = "1" Then S = "One"

If d = "2" Then S = "Two"

If d = "3" Then S = "Three"

If d = "4" Then S = "Four"

If d = "5" Then S = "Five"

If d = "6" Then S = "Six"

If d = "7" Then S = "Seven"

If d = "8" Then S = "Eight"

If d = "9" Then S = "Nine"

If d = "0" Then S = ""

SingleDigit = S

End Function

# eBilling and Invoice System Test Plan

## Introduction

This document describes the user acceptance test plan for the EBilling and Invoice System. The complete test strategy for the EBilling and Invoice System is to perform the following kinds of tests, in sequence:

1. **Component testing** of each component that makes up the EBilling and Invoice System
2. **Integration testing** of the EBilling and Invoice System, to ensure the correct interworking of its components
3. **Validation testing** of the EBilling and Invoice System, to ensure that it works correctly in a pseudo-live environment
4. **User acceptance testing** of the EBilling and Invoice System, to ensure that its function is acceptable to its users

Acceptance testing is the last set of tests to be performed before the application goes officially live.

## Test Scope

The scope of the user acceptance testing covers:

* Version 1 of the EBilling and Invoice System
* User-facing functionality defined by a set of use cases
* Administrator-facing functionality defined by a set of use cases

The aim of the testing is to determine how well the application meets its functional requirements from the perspective of the user, and to identify any issues so they can be resolved. Also, the testing serves to compile a set of test data and results that can be used during subsequent test cycles, to test for non-regression of the software in later releases or after the application is in maintenance.

Working practices might vary from user to user and are considered outside the scope of the testing.

## Test Strategy

The basis of user acceptance testing is that other tests were completed successfully, so the application and its required infrastructure are considered to be stable and reliable. Acceptance testing concentrates on the application from the user’s perspective, that is, how the application is used and whether it meets the necessary quality criteria.

Change requests will be sent to the development team as the actionable documentation. Change criteria will be determined by the Test team and the Development team prior to the beginning of testing. For instance, criteria may include *impact to desired functionality, amount of code impacted by proposed change,* and *design required by proposed change*. The tester will evaluate the criteria. The test lead will determine Change Required or not. Once a bug has been determined as Change Required, the bug report will be translated into a Change Request and passed on to development.

The customer of the acceptance testing is the System Users, Supervisor and Adminstratorfor EBilling and Invoice System. The progress of the acceptance testing will be reported to the customer, together with any issues that are discovered and their planned resolutions. Sign-off of the tests, and therefore the acceptance of the application, will be performed by the customer or a selected representative.

## Preconditions

The following items are required before testing can take place:

* A complete and coherent functional specification of the EBilling and Invoice System expressed as use cases and usage scenarios
* A complete and validation-tested release of EBilling and Invoice System, delivered according to the delivery plan
* An agreed-upon procedure for dealing with any anomalies that are discovered during the testing process
* A set of test specifications describing how each functional area of the EBilling and Invoice System is to be acceptance tested
* An implemented test environment for the testing
* Sufficient, suitable resources to carry out the testing
* Available standards for the acceptance testing

## Test Priorities

During testing of the EBilling and Invoice System, the following qualities will be tested in order of priority:

* Functionality—whether the required functions are available and working as expected
* Usability—how user-friendly and intuitive the EBilling and Invoice System is
* Security—how well-protected and guaranteed corporate and user data is
* Performance—whether the response times are within acceptable limits
* Customization—how straightforward it is to use the application in new, unpredicted ways

## Test Techniques

The following techniques will be applied:

* Scripted tests—sequences of user interactions (based on the use case and usage scenarios) using predefined data sets against predicted results
* Unscripted tests—based on scripted tests, the tester tries to modify the scenarios to explore what-if possibilities
* Penetration tests—scripted tests to attempt unauthorized entry into the system
* Usability checklists—tests to determine the complexity of interactions
* Performance statistics—generation of performance information to check against desired performance criteria

## Test Organization

## Roles and Responsibilities

The following roles are defined:

* QA lead/test manager—responsible for planning and ensuring the smooth running of the test process
* Tester—carries out the tests according to the test plan, and then reports the results
* Product manager—ensures that the tests are carried out successfully from a user perspective
* Project sponsor/client—acts as main stakeholder, and ensures that the needs of the customer community as a whole are considered
* Test support—provides technical assistance, such as test environment configuration, and non-technical assistance, such as methodological support

Weekly team meetings will be held involving the test manager, testers, and product managers. At these meetings, the progress of the testing process will be reported, any issues will be discussed, and actions will be agreed upon.

# Conclusion and future enhancement

This project was developed to fulfill user and business requirement; however there are lots of scope to improve the performance of the eBilling and Invoice System in the area of user interface, database performance, and query processing time. Etc.

So there are many things for future enhancement of this project. The future enhancements that are possible in the project are as follows.

* Linking and integration of any legacy system for accounting.
* Integration with travel agent through Web Services
* Connection to third-party OLAP applications
* Electronic Data Interchange (EDI) system between banks, other credit verification agency and their vendors
* In the area of data security and system security.
* Provide more online tips and help.
* To optimize the query which is embedded in the system.