USER FRIENDLY BILLING SYSTEM FOR LANKA ELECTRICITY COMPANY PRIVATE LIMITED

L. M. Anura Kumarasiri
(Reg No 99/AS/018)

This report is written for dissertation submitted in practical fulfillments of the requirement for the degree of
Bachelor of Science in Physical Sciences
Faculty of Applied Sciences Sabaragamuwa University of Sri Lanka

Buttala
2003

DECLARATION

I certify that this dissertation does not incorporate without acknowledgement any material previously submitted for degree or diploma in any university and to the best of my knowledge and belief it does not contain any material previously published or written or orally communicated by another person except where due to reference is made in the text

L. M. Anura Kumarasiri ----------

Signature

29/04/03

To best of my knowledge the above particulars are correct

External Supervisor
Dr. Ravi Corea
Director,
PricewaterhouseCoopers Lanka (Pvt) Ltd.
4/1 Gregory's Road,
Colombo 07.

---

for Signature 25/04/2003

Internal Supervisor
Mr. Jayalath Ekanayaka
Instructor in computer science,
Faculty of Applied Sciences,
Sabaragamuwa University of Sri Lanka.

Signature 05/05/2003

Head of the Department
Dr. Nirmalee Wickramaratne
Head/Dept Physical Sciences,
Faculty of Applied Sciences,
Sabaragamuwa University of Sri Lanka.

Signature

PROJECT REPORT

Affectionately Dedicated to

my parents, teachers

and whoever helped me in my life

ACKNOWLEDGEMENTS

I express my deep gratitude to my internal supervisor Mr. Jayalath Ekanayaka. Instructor in computer science. Department of physical sciences. Faculty of Applied sciences. Sabaragamuwa University of SriLanka for his assistance. encouragement. and guidance throughout this project.

Further I express my sincere gratitude to my external supervisor Dr Ravi Corea. Director Pricewaterhouse Coopers Lanka Limited. who kindly offered me the industrial placement with all the facilities.

Thanks are also due to Dr D.B.M Wickramarathna the Dean. Faculty of Applied sciences Sabaragamuwa University of SriLanka. and Dr Nirmalee Wickramarathna Head of the Department of physical sciences Faculty of Applied sciences Sabaragamuwa University of Sri Lanka for guiding me towards a successful completion.

I heavily thank to Mrs. Florence Fernando. Mr. Priyantha Amarathunga. Mr. Demion Fernando. and all the staff of Pricewaterhouse Coopers Lanka Limited and for the JAVA team of the EWis Company Ltd.

Thanks are also due to Mr. Cyril Karunanayaka Electrical Engineer of Ceylon Electricity Board Pelmadulla Branch.for their cooperation through out my study and this project and my teachers lectures who direct me on right way.

ABSTRACT

This document was submitted to present the detailed description on the project. which I underwent at the work site of Pricewaterhouse Coopers Lanka Private Limited. I was trained under this project as a Trainee Developer about 6 months in practical fulfillment of the requirements for the degree of Bachelor of Science in Physical science. Faculty of Applied Sciences. Sabaragamuwa University of Sri Lanka. This project is a team effort and I took part to design of the technical specification. At the time of preparing this report the project was completed up to the system testing stage.

The objectives of the project to develop a user friendly Billing System for the Lanka Electricity Company Limited (LECO) based on the functional specification of the LECO.
The classical life cycle. model (water fall model) was used and following steps were followed.

1. Requirement definition and got approval from the LECO.

2. Design the technical specifications (Entity Relationship Diagram. Table normalization).

3. Implementation (Actual program in selected language).

4. Test the system according to the testing procedure and get the acceptance of the LECO. (Unit testing. System testing. QA testing. Acceptance testing)

5. Post implementation support (Trained users. Bug fixing. Maintenance).

This LECO Billing system facilitates the creation and maintenance of all relevant to the Billing System. The system was a two-tire application and. implemented in *java* with *Oracle* as the database following the correct software development methodologies can solve most of the real world IT problem.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 PricewaterhouseCoopers Lanka (Pvt) Ltd

1.1.1 Background

The PricewaterhouseCoopers is largest professional services firm in the world. was formed as a result of merging between the former Pricewaterhouse and Coopers & Lybrand. in 1998. Due to some certain regularity and ownership issues. the firm separated two distinct units in Sri Lanka. PricewaterhouseCoopers Lanka (Pvt) Ltd is responsible for Audit. Tax. and financial. Advisory Services. and a limited liability company. The latter it is responsible for Management Consulting Services (MCS) in Sri Lanka. focusing primarily on the Information Technology Consulting and software developing. The former Pricewaterhouse formed as a company in 1995 .The main objectives was to develop IT in Sri Lanka. At the same time non of the big firms had a significant IT presence in Sri Lanka. and PricewaterhouseCoopers Lanka (Pvt) Ltd was the pioneer.

1.1.2 Services

The high comparative cost of IT and the less of experienced professionals in Sri Lanka results the grater risk in the IT projects. Many companies have spent money to purchase software products from the west but most of them fail to support expected benefits due to poor implementation.

PricewaterhouseCoopers Lanka (Pvt) Ltd has built its own reputation in Sri Lanka by addressing these issues. The PricewaterhouseCoopers Lanka (Pvt) Ltd aiming to develop quality software and delivers to International level at local costs. Quality is assured by adopting the methodologies. and practices of the PricewaterhouseCoopers organization world wide. and by recruiting highly qualified professional staff. Cost are kept low by using the firm's excellent training recourses to build on the strengths on the local graduates of high potential rather than relying on international resources wherever possible.

PricewaterhouseCoopers Lanka (Pvt) Ltd has built up specialist software development and integration skills. largely motivated by the difficulties encountered in obtaining quality service from local firms. A small but specialized team trained in networks technologies has undertaken several recent projects. those cover network design.. security. management. as well as implementation large scale networks. As a part of system implementation work the firm also supplies services to. restructure and modernize clients IT management functions to support the demands of new systems.

The client list of PWC mainly consist of top level Sri Lankan multi-nationals companies such as Shell and various government organizations such as Ministry of finance. and major players in energy sector such as Shell gas Lanka Ltd. Lanka Electricity Company.

1.1.3 Skills Profile

PricewaterhouseCoopers Lanka (Pvt) Ltd has a human resource strategy focus on training and development staff with good educational qualifications. Most staff has degree in science. engineering or Commerce from both local and foreign universities

The software development division has many staff with over five years experience in project management. system analysis and design. and implementation. Programming software with *Java. Visual Basic*. Database involved *DB2. SQL server. Oracle. Access*. There is a small lotus Notes development team with Lotus Certification. The network team is trained in *Windows NT. Unix. Linux* operating systems and *Cisco* network hardware.

1.2 Project

1.2.1 Project description

The Billing system developed based on the functional specification of the LECO. The project was previously done by the East West information system (EWis) company using three-tire architecture or web sphere. But it was failed in the acceptance level testing.

The PricewaterhouseCoopers Lanka (Pvt) Ltd (PWCL) then took the responsibility to develop the system again with *Java* team of PWCL and EWis.

The new system consists of two-tier Client-Server architecture where the client does most of the processing(fat client). This architecture was decided after theoretical consideration of LECO requirements. The technology components such as web sphere will not be used for this implementation. The detailed design of the system will mainly concentrate on achieving efficiency within the client application.

1.2.2 Objectives

The main objectives of this system is to carry out the defined set of operations as efficiently as possible while conforming to the required response times.

2. THEORETICAL BACKGROUND

2.1 Software Design

Software design is the process of deriving solutions that satisfies the software. The "solutions" means modeling or describing the problem with sufficient detail. So that it will be easy to implement.

Under the software design there are some topics to consider

1. Typical stages of Design
2. Common Design Phases
3. Top down or Bottom up Design
4. Design Strategies
5. Design quality
6. Architectural design
7. Design Methodology

Under this mentioned only about the design methodology because it is mainly effect to the development of the system. also to select the programming language.

2.2 Design Methodology

The two main forms of design are

1. Function oriented Design
2. Object oriented Design

Function oriented Design

Software design is represented as a set of interacting functions with shared information. The main drawback in function-oriented design is that some functions can change the values of shared data in a manner not expected by other functions. This problem can be minimize by minimizing shearing of data. restricting the impacket of a function to the data pass to it by the calling function. Generally function oriented process has the following stages

a) Identify the data transformations and high-level functions

b) Decompose the high level functions in to hierarchy of sub functions.

c) Describe the operation and interface of each system entity.

d) Document the flow of control in the system.

Object oriented Design

What is an object?

An object is an entity that has a state and defined set of operations. which operates on that state. A set of attributes related with an object is represented the state of an object. The operations related with the object provide services.

Some statements are below that can be describe further about an objects.

a) Objects are abstractions of real world or system entities which are responsible for managing their own private state and offering service to other objects

b) Objects are independent entities that may readily be change. Because state and representation information are held within the objects.

c) System functionality is expressed in terms of operations or service associated with each object.

d) Shared data areas are eliminated by the objects. communicated by calling services offered by other objects rather than shearing variables.

e) Objects may be distributed and may execute either sequent or parallel.

Object oriented Design is design strategy based on the information hiding. It views a software system as a set of interacting objects with their own private state. rather than as a set of function that share a global state. When we define an object it will have its own states or own parameters. When we define an object we have to design its attributes and functions. In Object oriented Design there is no data flowing through each other. Object to object interact passing messages through them. Each object is an independent unit. Other objects do not worry about the data associated with other objects. It is called the information hiding.

Object Identification

In the Object oriented design it is very much necessary to identify the objects clearly. that make up the system and their attributes as well as associated operations with them. Designers use their skills and experience for this task. There have been various proposals made about how to identify objects

Use of grammatical analysis of natural language description of a system objects and Attributes are nouns. Operations or services are verbs

Use of tangible entities in the system

Use scenario base analysis where various scenarios of system use are identified and analyzed interval as each scenario is analysis.

2.3 Theory behind the graphical user interfaces

The goal of user interface design are to define general principle and consider the factors which influence how information should be presented and supported. Users often judge the system by its user interface. Poorly designed user interfaces can make catastrophic errors and reject the system. User interface design must take in to account the needs. experience and capabilities of potential system users. The interface should be based on user-oriented concept rather than computer oriented concept. Considerable factors when design graphical user interfaces.

a) System should display an appropriate level of consistency

b) System should not surprise the user. The effect of an action should predictable as possible

c) The system should provide some rapid recovery to the errors

Example :- understandable error messages

Types of user inter faces

I . Graphical user interfaces

Graphical user interfaces are a long way from completely replacing text/ command base system but are the most common form interface today. Windows. Icons. Pull down menus. dialog boxes. use of mouse are all examples of Graphical User Interfaces.

Advantages:

a) Easy to learn and use.

b) Facilitate switching between tasks and user interfaces.

c) Full screen interaction.

II . Command languages

These types of interfaces used on cheaper hardware and often faster than Graphical user interfaces for experts.

Disadvantages:

- Commands must be learned and remembered.
- Need more error detection and recovery.
- Typing ability is required.

III. Direct manipulation interface

One way to make a system reasonable. user concepts is to directly model with familiar objects and actions. A direct manipulation interfaces the user with a model of their information space that is modified by direct action.

Example: Documents moved in to folders directly using mouse.

Advantages

- User feels more in control.
- Learning time is relatively short.
- Users feel more in control.
- Mistakes can be avoided.

Disadvantages:

- Finding an appropriate model can be difficult.
- Facilitate to navigating around large information space can be risky.

2.3.1 Successful user interface models

Desktops: Users screen represent a desktop entities are represent as the icons on the desktop.

Control panels: Consist of icon representing things like buttons, switches etc.

Menu system: Gives user a choice from a list of option. Menus can be graphical user interface or text base

Advantages:

a) Users don't need remember specific command names.

b) Typing effect is minimal

c) Context dependent helps make more sense

d) Can restrict the availability of some menus according to user

Disadvantages:

a) Experience users find the menus slower than command languages

b) Combined actions are hard to represent

c) Bad for causes where there should be a large numbers of choices

2.3.2 User guidance

User guidance encompasses all information to help a user identify or learn how to correctly use a system. to archive their desired results. This involves

1. System documentation.
2. Online helps on wizards
3. Including messages with errors.

2.3.3 User interface evaluation

NO matter what we do the user is the ultimate proving ground. One cannot assume that the user interfaces we designed will be usable or light without actually it out on real users in real situations Prototyping. video. audio. taping user sessions etc are all valuable in tools in evaluating user interfaces.

User advises should be taken in the following points.

1. How long does a new user take to become productive with the system?
2. How well thus the system responses match the users work practices.
3. How tolerant is the system to use an error.
4. How closely is the system tide to a single work model?

User guidance should be integrated in the design process. not treated as a minor part. once product is completes. Write the use guide first based on the user specifications and then proceed with the design.

2.3.4 Motivation for solid user guidance

1. A thorough and well-organized user guidance system significantly reduces the amount of supports a product requires reduces the amount of training time for new users.

2. Better user guidance makes it easier for users to caught the system encouraging to use our larger scale

3. User guide created in a common easy to use format gives users a grater sense of familiarity with a product faster leading to increased product acceptance.

4. A poorly designed user guidance will lead to frustration and low evaluation of the system by users. even if the features they wants are all available.

3 THE LECO BILLING SYSTEM DEVELOPMENT METHODOLOGY.

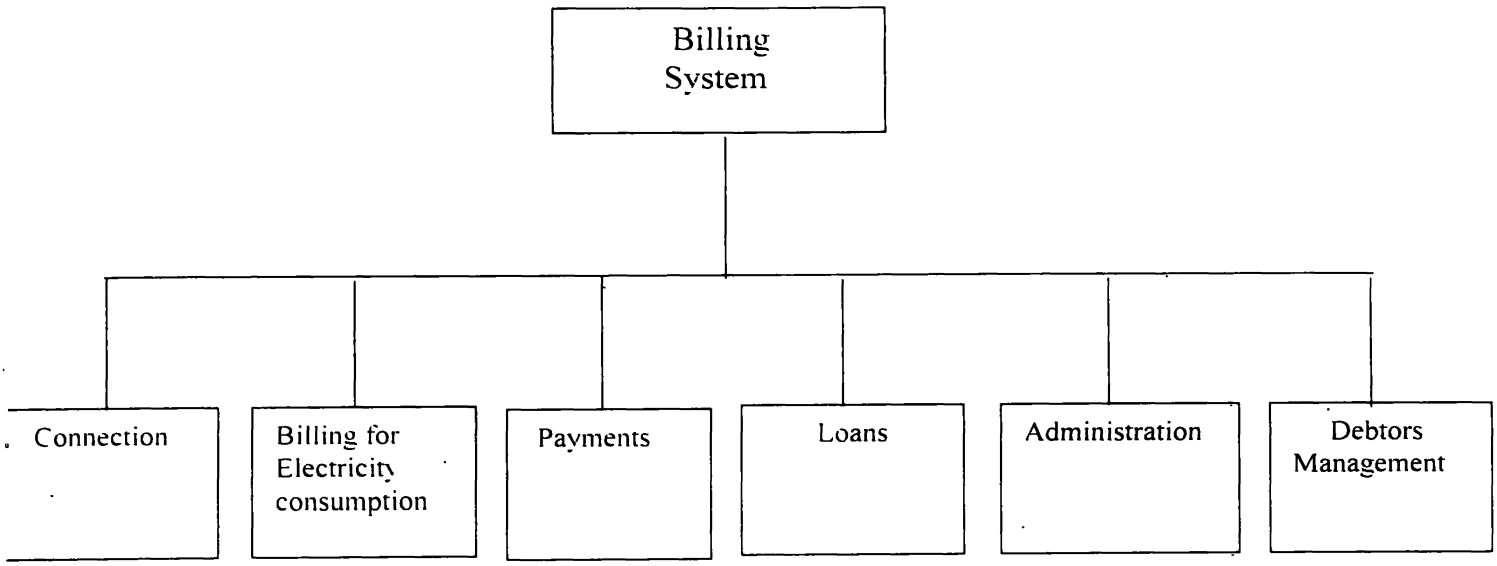


Figure 1: Decomposition of the system in to main sub systems.

- **Connection** – This function will capture all the information related to a connection.
- **Billing for Electricity consumption** – Capture the total number of units consumed. calculate the total bill and generate the billing statement.
- **Payments** – This function will capture payments related to consumer accounts.
- **Loans** – Capture loan details. calculate installments. and recover monthly installment through the bill or invoice.
- **Administration** – Capture information such as tariff. taxation and walk path to calculate and generate the billing statement.
- **Debtor's management** – Capture all receivable. receipts. payments. adjustment transactions and generate all debtors statements.

The Billing system was decomposed in to 6 sub systems as shown in Figure-1. The external entities and their interaction internally and externally with the system were identified. Then the entities. their relation ship and carnality. which were in the problem domain were identified. Next an ER diagram was drawn as shown in figure-2.

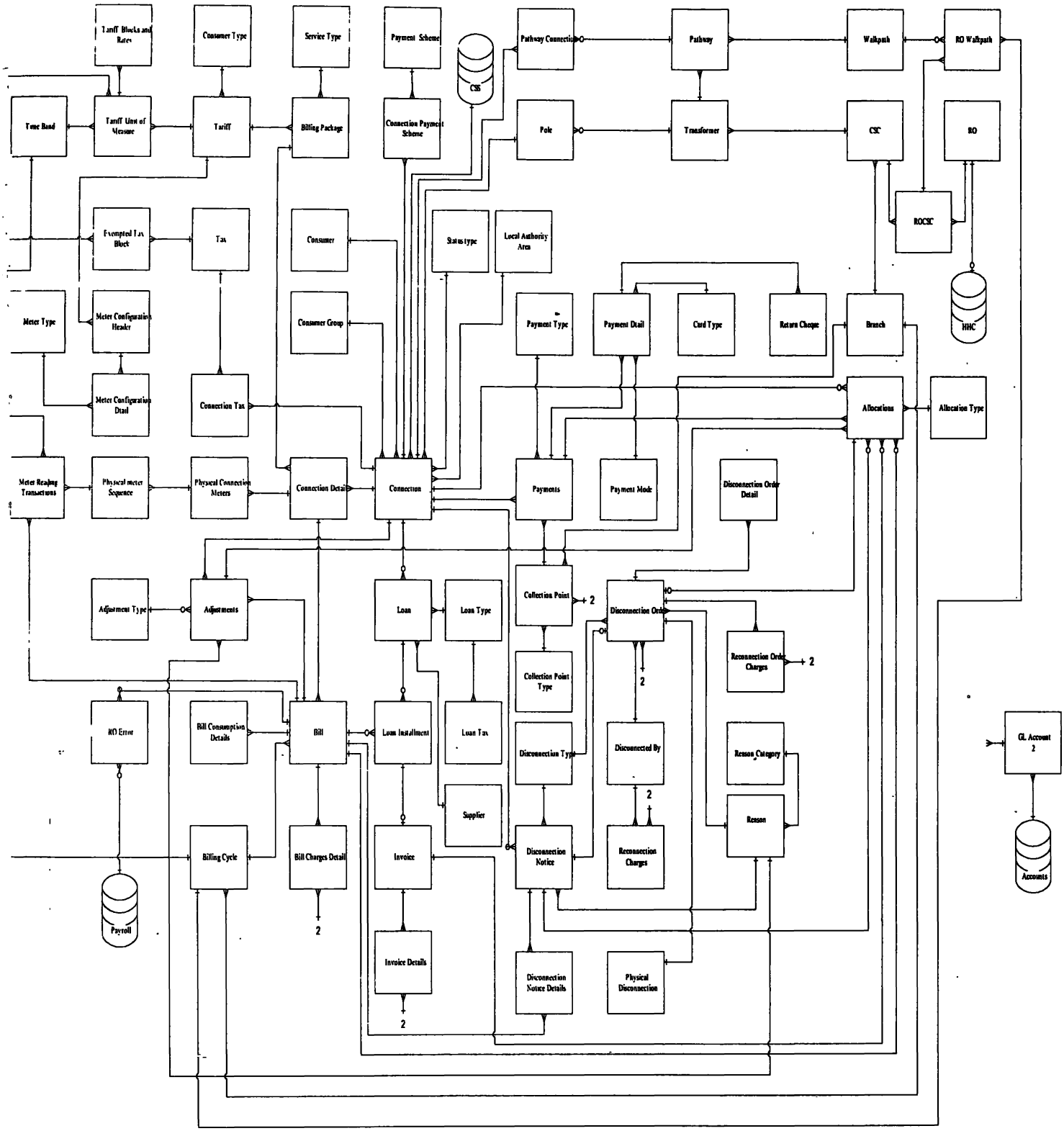


Figure 2 Entity Relationship Diagram of LECO Billing system

3.1 Database Implementation in oracle

Oracle

Oracle is a kind of a database that can be used as a database of a system. It has high data processing ability and faster. Oracle supports much kind of platforms. Like *Windows 98, WindowsNT, Unix*. The LECO has the Unix server. The oracle was used to implement the database after considering the LECO requirements. Oracle has the ability of work with Unix server and high data processing. Client computers of the LECO are Windows 98. this Oracle support Unix platform and Windows98 platform also.

Under this I was participated to some modifications of the script

The script is the document. which used to create database. this document is reusable. In the script there should be a sequence of tables that we use in the Entity Relationship Diagram. This sequence may be according to primary key reference of Entity Relationship Diagram or with using synonyms. Once run this script on the Oracle. it create database according to script.

3.1.1 Development procedure of Master tables

<u>What are master Tables</u> :

With the entity relation ship diagram (ERD) we define all the entities that are involved in the system. Data of some entities are not changing regularly or once we enter data. it keeps that data long time without changing. Those tables are master tables if the system. They give their permanent data to Transaction tables to process or to do other work. But data associated with master tables not changing due to that. The ER diagram was converted into highest normalized tables.

Example : some master tables in the system

REASON	LOCAL_AUTHORITY_AREA
CONSUMER_TYPE	CARD_TYPE
STATUS_TYPE	READING_TYPE
CONSUMER	STATUS_TYPE
SERVICE_TYPE	METER_TYPE
SUPPLIER	TIME_BAND
REASON_CATEGORY	COLLECTION_POINT_TYPE
PAYMENT_MODE	RECONNECTION_CHARGES
PAYMENT_TYPE	ALLOCATION_PRIORITY

hose tables were created using a script in *Oracle.* In the script synonym was used to refer a able instead of table name.

xample: - For consumer type table

01= consumer type.

he following sample code was used to create consumer type table and service type table

xample :-

```
-    CONSUMER_TYPE      M001
ROP TABLE APPLECO.M001 CASCADE CONSTRAINTS ;

REATE TABLE APPLECO.M001 (
     CONSUMER_TYPE_CODE          CHAR (2)       NOT NULL,
     CONSUMER_TYPE_DESCRIPTION   CHAR (40),
     PRIMARY KEY ( CONSUMER_TYPE_CODE ) )  ;

-    SERVICE_TYPE       M002
ROP TABLE APPLECO.M002 CASCADE CONSTRAINTS ;

     CREATE TABLE APPLECO.M002 (
     SERVICE_TYPE_CODE          CHAR (2)       NOT NULL,
     SERVICE_TYPE_DESCRIPTION   CHAR (40),
     PRIMARY KEY ( SERVICE_TYPE_CODE ) ) ;
```

he following transaction tables derived from ER diagram. The data related with ansaction tables are not permanent.

xample : some transaction tables used in the system

```
ISCONNECTION_ORDER
ISCONNECTION_ORDER_DETAIL
HYSICAL_DISCONNECTION
ECONNECTION_ORDER_CHARGES
REDIT_BALANCE
EPOSIT
EFUND
LLOCATIONS
ONNECTION_HISTORY
ONTHLY_CONSUMPTION
```

3.2 Graphical User Interface for master Tables

Graphical User Interfaces were developed using *java* and connected with Master tables. Some *java* codes were generated more easily using *Jbuilder* visual tool.

is a visual tool that is use java developers kit (JDK) to generate java codes more easily.

3.3 Program specification document

For each and every master table a separate program specification document was created. This document consisted the variables. methods. interfaces specific text fields and buttons. decomposition of graphical user interface etc.

Example: Program specification document of Reconnection Charges master table

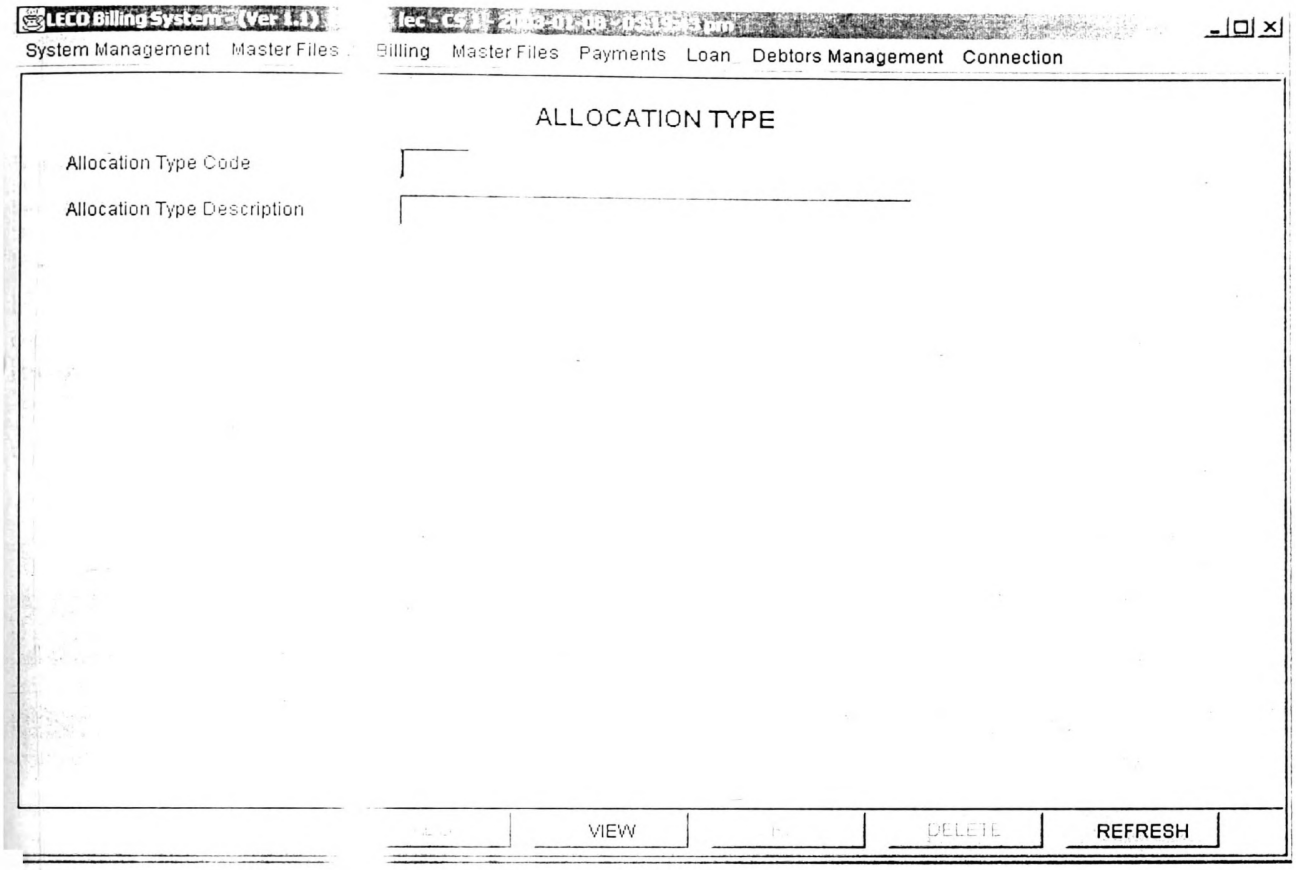
Program Specification – Billing System

Prepared by: Anura Kumarasiri	Date: 10-Dec-02
Authorized by:	Date:
Process Reference Number:	
Process Name:	Reconnection Charges
Description:	Add, Update, Delete, View, Print, Find and Refresh Reconnection Charges

Process Type:	Elementary	X
	Common Logic Module	
Class Diagram Reference Number:		
Sequence Diagram Reference Number:		

Entity Actions:	Created	Retrieved	Updated	Deleted
Reconnection Charges	Y	Y	Y	Y

GUI Class Layouts :



Report Layouts :

Class Name	:	MAF_ Reconnection_Charges_GU

Class description	:	GUI class for Reconnection Charges Table
Type of class	:	GU
Super-class	:	Jframe
Interface	:	N/A

Attributes	:	*Description*
PaneFields	:	Panel to hold the textboxes & the labels
PaneMenu	:	Panel to hold the menu bar
PaneButton	:	Panel to hold the buttons
lblReconnection_Charges	:	Label to display the name of the master file
lblDisconnectedBy_Code	:	To display the "Disconnected By Code" label
txtDisconnectedBy_Code	:	Text box to accept or display Disconnected By Code
lblCharge_Code		To display the "Charge_Code" label
txtCharge_Code		Text box to accept or display Charge_Code
LblChargeDescription		To display the "ChargeDescription" label
TxtChargeDescription		Text box to accept or display ChargeDescription
LblRate		To display the "Rate" label
TxtRate		Text box to accept or display Rate
LblAmount		To display the "Amount" label
TxtAmount		Text box to accept or display Amount
lblGlAccount_Code		To display the "GL Account Code" label
txtGlAccount_Code		Text box to accept or display GL Account Code
CmdAddModify	:	Button used to invoke the method to add the information about a new Reconnection Charges. Button used to invoke the method to modify the information about a Reconnection Charges
CmdDelete	:	Button used to invoke the method to delete the information
CmdRefresh	:	Button used to invoke the method to clear the window
CmdView	:	Button used to display the Reconnection Charges table
CmdPrint	:	Button used to print the Reconnection Charges table.
CmdFind		Two Buttons used to display Disconnectrd By Code and GL Account Code coming as forign keys to the Reconnection Charges table
IntByCodeMaxLen	:	Holds the Disconnected_By_Code MAX length "1"
IntChargeCodeMaxLen	:	Holds the ChargeCode MAX length "2"
IntChargeDescMaxLen		Holds the Charge Description MAX length "40"
IntRateMaxLen		Holds the Rate MAX length "14"
IntAmountMaxLen		Holds the Amount MAX length "14"

IntGLCodeMaxLen		Holds the GL Account Code MAX length "14"
StrByCode	:	String to assign the captured Disconnected By Code from txtBy_Code
strCharg_Code		String to assign the captured Charge Code from txtCharge_Code
strCharg_Desc	:	String to assign the captured Charge Description from txtCharge_Desc
StrRate		String to assign the captured Rate from txtRate
StrAmount		String to assign the captured Amount from txtAmount
strGL_Code		String to assign the captured GL_Account_Code from txtGL_Code
Viewer	:	ViewTableRecords object to display the data in Reconnection_Charges Table
BL_Mfile	:	MAF_Reconnection_ChargeBL type object to perform the business logic actions
	:	

Methods	:	*Description*
Constructor for MAF_Reconnection_ChargeGU	:	Constructor for MAF_Reconnection_ChargeGU class. Calls jbInit method.
JbInit	:	Method to initialise the values in the MAF_Reconnection_ChargeGU object
Listener for cmdAddModify	:	invoke method in MAF_Reconnection_ChargeBL with "ADD" to add information about a new Reconnection Charge invoke method in MAF_Reconnection_ChargeBL with "MODIFY" mode to modify information about an existing Reconnection Charge
Listener for cmdDelete	:	Invoke method in MAF_Reconnection_ChargeBL with "DELETE" mode
Listener for cmdRefresh boxes	:	Invoke local method to clear the text boxes and the form initialisation
Listener for cmdView	:	Invoke method in MAF_Reconnection_ChargeBL with "VIEW" mode to retrieve information all the existing records in Reconnection Charge Table display in a new Frame
Listener for cmdPrint	:	Invoke method in MAF_Reconnection_ChargeBL with "PRINT" to print all the existing records in Reconnection Charge Table
Listener for cmdFind	:	Invoke method in MAF_Reconnection_ChargeBL with "FIND" to find all the records in Disconnected By Code and GL Account Code as forign keys to Reconnection Charge Table
Listener for txtBy_Code	:	Validate if the content in the textbox is acceptable and Invoke

		method in MAF_Reconnection_ChargeBL with "READ" to retrieve information about the entered Disconnected By Code
Listener for txtCarge_Code	:	Validate if the content in the textbox is acceptable and Invoke method in MAF_Reconnection_ChargeBL with "READ" to retrieve information about the entered Charge Code
Listener for txtCharge_Desc		Validate if the content in the textbox is acceptable
Listener for txtRate		Validate if the content in the textbox is acceptable
Listener for txtAmount		Validate if the content in the textbox is acceptable
Listener for txtGL_Code		Validate if the content in the textbox is acceptable
ReFreshTable( )	:	Method to display the content if the table is not empty

Class Name	:	viewTableRecord
Class description	:	GUI class(inner) for displaying the Reconnection Charges master file data
Type of class	:	GU
Super-class	:	Jframe
Interface	:	N/A

Attributes	:	*Description*
ViewTable	:	Panel to hold the table object
Table	:	Table to display the column names & the data values
VecRow	:	Vector which holds the data of the table
VecCol	:	Vector which holds the column names of the table
	:	

Methods	:	*Description*
Constructor for viewTableRecord	:	Constructor for viewTableRecord class.
SetData(data, col)	:	This creates a table from the vectors that are passed by MAF_ Reconnection Charges _GU's "VIEW" button's action listener.
GetData	:	Return the currently selected row's values in a Vector
Listener for table	:	Track mouse click event to get the selected row
ATModel	:	AbstractTableModel implementation to set the properties of the table

Class Name	:	MAF_ Reconnection ChargesBL

Class description	:	Business logic class for Reconnection Charges Table
Type of class	:	BL
Super-class	:	N/A
Interface	:	N/A

Attributes	:	*Description*
UT_MFile	:	MAF_ Reconnection ChargesUT type object to perform the actions on the database
StrByCode	:	String which will be holding the Disconnected By Code
StrChargeCode		String which will be holding the Charge Code
StrDesc	:	String which will be holding the Charge Description
StrRate		String which will be holding the Rate
StrAmount		String which will be holding the Amount
StrGLCode		String which will be holding the GL Account Code
StrMode	:	Holds the Mode of action to be performed on the UT_MFile object
IntNumOfRec	:	Holds the number of records successfully Added, Updated or Deleted from the Reconnection Charges Table
VecData	:	Holds the records retrieved from the Reconnection Charges Table
	:	

Methods	:	*Description*
Constructor for MAF_ Reconnection_Charges BL	:	Constructor for MAF_ Reconnection_Charges BL class .
SetMode(mode, bycode,chargecode,desc, rate,amount,glcode)	:	Set the mode of action to be performed. Take in the Disconnected By Code , Charge Code , Charge Description, Rate, Amount & the GL Account Code depending on the mode
GetRecord( )	:	Method to return the vecData after performing the action on the database by the "processCommonData" method
GetStatus( )	:	Method to return the intNumOfRec after performing the action on the database by the "processCommonData" method
ProcessCommonData( )	:	{ if strMode = ADD{ invoke add(strByCode,strChargeCode,strDesc,strRate,strAmount, strGLCode) method in MAF_ Reconnection_ChargesUT class }

```
if strMode = MODIFY{
        invoke

update(strByCode,strChargeCode,strDesc,strRate,strAmount,
strGLCode)
        method in MAF_ Reconnection_ChargesUT class
}
if strMode = DELETE{
        invoke
        delete(strByCode,strChargeCode)
        method in MAF_ Reconnection_ChargesUT class
}
if strMode = READ{
        invoke
        read(strByCode,strChargeCode)
        method in MAF_ Reconnection_ChargesUT class
}
if strMode = VIEW{
        invoke
        read( )
        method in MAF_ Reconnection_ChargesUT class
}
if strMode = PRINT{
        invoke
        print( )
        method in MAF_ Reconnection_ChargesUT class
}
```

:

Class Name	:	MAF_Reconnection_ChargesUT
Class description	:	Utility class for accessing the data on Reconnection Charges Table
Type of class	:	UT
Super-class	:	N/A
Interface	:	N/A

Attributes	:	*Description*
StrTableName	:	Stores the table name of the database "Reconnection Charges"
StrByCodeCol		Stores the ByCode column name of the table

		" Disconnected_By_Code"
StrChargeCodeCol		Stores the ChargeCodeCol column name of the table " Cahrge_Code"
StrDescCol		Stores the DescCol column name of the table " Charge_Description"
StrRateCol		Stores the RateCol column name of the table " Rate"
StrAmountCol		Stores the AmountCol column name of the table " Amount"
StrGLCodeCol		Stores the GLCodeCol column name of the table " GL_Account_code"
ConString	:	Used to set the connection to the database
Statement	:	Used to set the statement from conString
RsetTable	:	Used to store the values when executing a query
	:	

Methods	:	*Description*
Constructor for MAF_ Reconnection_ChargesUT	:	Constructor for MAF_ Reconnection_ChargesUT class

add	:	int add(strByCode,strChargeCode,strDesc,strRate,strAmount, strGLCode){
		Insert into Reconnection Charges
		(Disconnected_By_Code,Charge_Code,Charge_Description, Rate,Amount,GL_Account_Code)
		Value
		('strByCode','strChargeCode','strDesc','strRate','strAmount ',' strGLCode')
		return the number of records added successfully
		}

Update	:	int
		update(strByCode,strChargeCode,strDesc,strRate,strAmount, strGLCode){
		Update Reconnection Charges
		Set Charge_Description =' strDesc ' Rate = 'strRate'
		GL_Account_Code =' strGLCode' Amount = 'strAmount'
		Where Disconnected_By_Code =' strByCode'
		Charge_Code = 'strChargeCode'
		return the number of records updated successfully
		}

Delete	:	int delete(strByCode,strChargeCode){ Delete * From Reconnection Charges Where Disconnected_By_Code =' strByCode' Charge_Code = 'strChargeCode' return the number of records deleted successfully

		}

Read	:	Vector read(strByCode,strChargeCode){
		Select
		Disconnected_By_Code,Charge_Code,Charge_Description,Rate.Amount,GL_Account_Code
		From Reconnection Charges
		Where
		Disconnected_By_Code =' strByCode'
		Charge_Code = 'strChargeCode'
		return the return record that was read
		}

View	:	Vector view() { Select Disconnected_By_Code,Charge_Code,Charge_Description,Rate,Amount,GL_Account_Code From Reconnection Charges

Print	:	Vector print()
		{
		return view()
		}

Finalize( )	:	Overriding the finalise method of the "Object" to explicitly clear resources held by the MAF_ Reconnection ChargesUT object No need to call this in the program. After clearing give control back to finalise method of "Object"

SCREEN LAYOUT 1	
Screen ID	
Screen Name	

SCREEN LAYOUT 2	
Screen ID	
Screen Name	

3.4 Unit test plan

For each unit (table) a separate test plan was created. The unit test plan consisted the action performed. expected result. actual result given by the program.

Example : Unit test plan for Reconnection Charges master table.

Test Plans

TEST SPECIFICATION – Reconnection Charges Header

Customer:	
Project : LECO	
Prepared by : L.M.Anura kumarasiri	**Date :17/12/2002**
Authorized by:	**Date:**

Type of Test:

Unit tests		Functional Tests	
Program Test	☐	System Test	☐
Link Test	☐	Integration Test	☐
Change Test	☐	Performance Test	☐
Error Test	☐	Assurance Test	☐

Test Scope: To testing functionality of text boxes and buttons

TEST SPECIFICATION OUTLINE

Test Numbers	Test Outline
1	Validate the text boxes with valid and invalid data
2	Validate the ADD/MODIFY,VIEW,PRINT,DELETE,REFRESH ,FIND buttons functionality

TEST SPECIFICATION TEST DESCRIPTION

Test Number : 1

Description (*What the test is aiming to improve*)

Validate the text boxes with valid ad invalid data.

Instructions:

1.1 Check initial conditions and Disconnected By Code text box
1.2 Check the Charge Code text box
1.3 Check the Charge Description text box
1.4 Check the Rate text box
1.5 Check the Amount text box
1.6 Check the GL Account text box

Instructions	Expected Results	Actual Results
1.1(a)Check focus of initially	Initially Disconnected By Code text box Find button should focus. User must select one of them. User does not allow to enter values by the key board. Null values are not allow	
1.2(a) Enter valid new Charge	If the user has not selected one of	

Code and press enter key	Disconnected By Codes error message should display Disconnected By Code Find button should focus Else Charge description textbox should request focus. ADD Button is activate VIEW and REFRESH Buttons (always) enabled	
1.2(b) Enter valid existing Charge Code and select Disconnected By Code and press enter key	If both mach and has records to display, Charge Description, Rate, Amount, GL Account Code should display. Cursor should blink at the starting of the Charge Description text box Charge Code and Disconnected By Code should be inactive ADD Button is changed as MODIFY and it is enabled DELETE Button is enabled VIEW and REFRESH Buttons (always) enabled Users allows to select one of the other text fields find button by mouse clicking	
1.2(c) Enter invalid Charge Code	Generate an error message	
1.3(a) Enter valid Charge Description	ADD Button should keep active VIEW and REFRESH Buttons (always) enabled	
1.3(b) Enter invalid Charge Description	Generate an error message	

1.4(a) Enter valid Rate	ADD Button should keep active VIEW and REFRESH Buttons (always) enabled	
1.4(b)Enter invalid Rate	Generate an error message	
1.5(a)Enter valid Amount press enter key	ADD Button should keep active VIEW and REFRESH Buttons (always) enabled GL Account Codes FIND button should focus	
1.5(b)Enter invalid Amount	Generate an error message	
1.6(a) Select valid GL Account Code and press enter key	Add button should get focus Selected GL Account Code should display in the text box. Users not allow to type values. Users allow to keep null values	

Test Number : 2
Description (*What the test is aiming to improve*) Validate the functionality of ADD/MODIFY,VIEW,PRINT,DELETE,REFRESH Buttons
Instructions: 2.1 Check initial conditions of buttons 2.2 Check ADD Button functionality 2.3 Check MODIFY Button functionality 2.4 Check DELETE Button functionality 2.5 Check VIEW Button functionality 2.6 Check REFRESH Button functionality 2.7 Check PRINT Button functionality 2.8 Check FIND Buttons functionality

Instructions	Expected Results	Actual Results
2.1(a) Initial condition of Buttons	VIEW , FIND and REFRESH Buttons should be activated .Other should be disable	
2.2(a) Enter valid data press ADD button	Data should be added and massage should display Data should be able to retrieve	
2.3(a) Change the existing data and press MODIFY Button	Modified data should be added to the data base Message should display Data should be able to retrieve	
2.4(a) Enter valid Disconnected By Code and Charge Code press DELETE Button	Record should be deleted Massage should display record deleted If the record is used by another table proper massage should display	
2.5(a) Press VIEW Button	Existing Disconnected By Code, Charge Code, Charge Description, Rate, Amount, GL Account Code	

	should display on separate form Existing should be display according to an order of Disconnected By Code If existing is null proper massage should display	
2.6(a) Press REFRESH Button	Form should be initiate	
2.7(a) Press PRINT Button	Existing Disconnected By Code, Charge Code, Charge Description, Rate, Amount, GL Account Code should print	
2.8(a) Press Disconnected By Codes FIND Button	Press Disconnected By Code, Description should display	
2.8(b) Press GL Account Codes FIND Button	GL Account Codes, Description should display	

TEST SPECIFICATION
SCREEN HANDLING

Test Number : 1

Screen Level	
Screen Headings	
To Specification	
To Standard	
Conditioned headings	
Buttons	
Order of Buttons	
Size of buttons	
Alignment	
Screen specific buttons	
Grids	
To specification	
To standards	
Headings	
Correct Sequence	
Correct Selection	

Field Level	
Empty field	
Spaces	
Zeroes	
Minus sign values	
More than two decimals	
Field value more than specified range	
Field format	
Field type	
Field length	
Correct total (in total fields)	
Correct value (computed fields)	
Drop down field	
Correct selection	
Correct sequence	

TEST SPECIFICATION
REPORT HANDLING

Test Number : 1

Report Level	
Page Headings	
To Specification	
To standard	
Conditioned headings	
On over flow	
On level break	
Column headings	
Column Line up	
Conditioning	
Records	
Correct selection	
Correct sequence	
Totals	
Conditional Totals	
Permanent totals	
Headings	
Value correct	
General	
No records on table	
No records selected	

Field Level	
Line up	
Conditioning	
Correct data	
Editing	
Negatives	
Blanks/zeros	
Max Size	
Over flow truncation	

TEST DATA

	Valid	Invalid
Disconnected By Code	M	Null values
Charge Code	01.05	Null values String more than 2 character
Description	SURCHARGE .Null	**")< More than 40 characters
Rate	0.1 .Null values	Characters More than 14 Digits
Amount	1400.00. Null values	Characters More than 14 Digits
GL Account Code	26000. Null	Can not enter values

3.5 Cording master Tables

Development of master table is not a standalone program. It consists of three layers to reduce the complexity of the program. To reduce the complexity of the programming each master table was decomposes in to following three layers.

1.Presentation Layer.

2.Business Logic Layer

3.Utility Layer

Presentation Layer: This layer implements all the classes related to user interfaces. Those will be instantiated by either main control logic or any one of the business logic objects. This name as graphical user interface class. In this layer actual graphical user interface is coded called GU class. Adding panels. labels. adding text fields. combo box. radio buttons. calling another panels. text field validation. initialization. are done here. With the field validation it control the data that should be enter by cording. Some of fields do not allow entering numbers. symbols. more than one decimal places. wrong dates. months. The BL (Business Logic) class is called by this class. Calling to the message table and displaying them with the error or any other task. This GU class has action perform. event listener. mouse listener. action listener. mouse event etc. There are several methods that perform above tasks. Some methods are called by several other methods. Each and every field has individual methods to validate them separately. For ADD. MODIFY. VIEW. DELETE. REFRESH methods written separately. But REFRESH method used several times to refresh GUI.

Business Logic Layer: The complete business logic of the billing system will be implemented via a set of classes. These classes will be instantiated by the main control logic depending on the selected menu option or calling of the graphical user interface class. Business Logic object or class will Interact with the utility layer or utility class as well as graphical user interface class in order to carry out the process. The Utility (UT) class is called by this BL class and BL has the logic of calling function of UT class.

Utility Layer: The utility layer will responsible for the data base data retrieve . add data to the database. modification existing data with SQL and perform calculation with data. error handling .rollover. exceptions handling. This UT class write to error message table about database error and database access errors. SQL error. field selection errors. field-naming errors etc.

Master tables developed by me in the system

1 Consumer Type
2 Status Type
3 Payment Type
4 Allocation Type
5 Payment Scheme
6 Supplier
7 Reason
8 Reason Category
9 Reconnection Charges

3.6 Testing

The LECO project. predefined procedure was used for testing. The tests results were documented and errors reported at this stage. If any error found sent them back to the development step to fixed them.

The following testing procedure was used for LECO system.

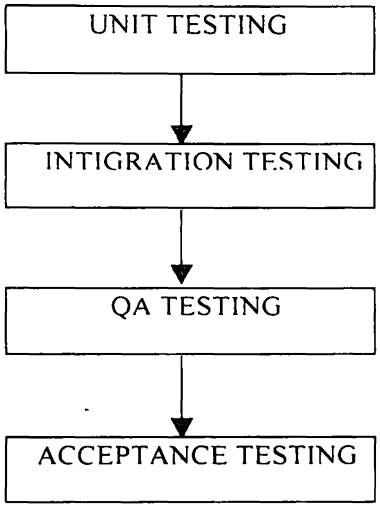


Figure 3 Testing Procedure

I did only unit and integration testing.

3.6.1 Unit Testing

The unit testing was done for each unit in the system using the test plan. According to the test plan enter test data to the unit and got the result. The result was then compared with the expected result. If any error found send it to the programmer again to fix error.

3.6.2 Integration Testing

This integration test was done after integrating all the units. Testing was conducted using a test plan designed prior to testing and synthetic data were used. All the errors found were fixed by the programmers. Then a special case test was carried out to only for the errors. which had been fixed. This process was conducted for all the integration errors. Then the system was sent to Quality Assurance testing.

3.6.3 Quality Assurance Testing

The Quality Assurance test was carried out to confirm that the system functions according to the requirements and error free. This also very similar to the integration testing but this was carried out by a special team that specially trained for Quality Assurance testing.

3.6.4 Acceptance Testing

The system was finally under went an acceptance test to accept the system by the LECO. a team specially allocated for this purpose.

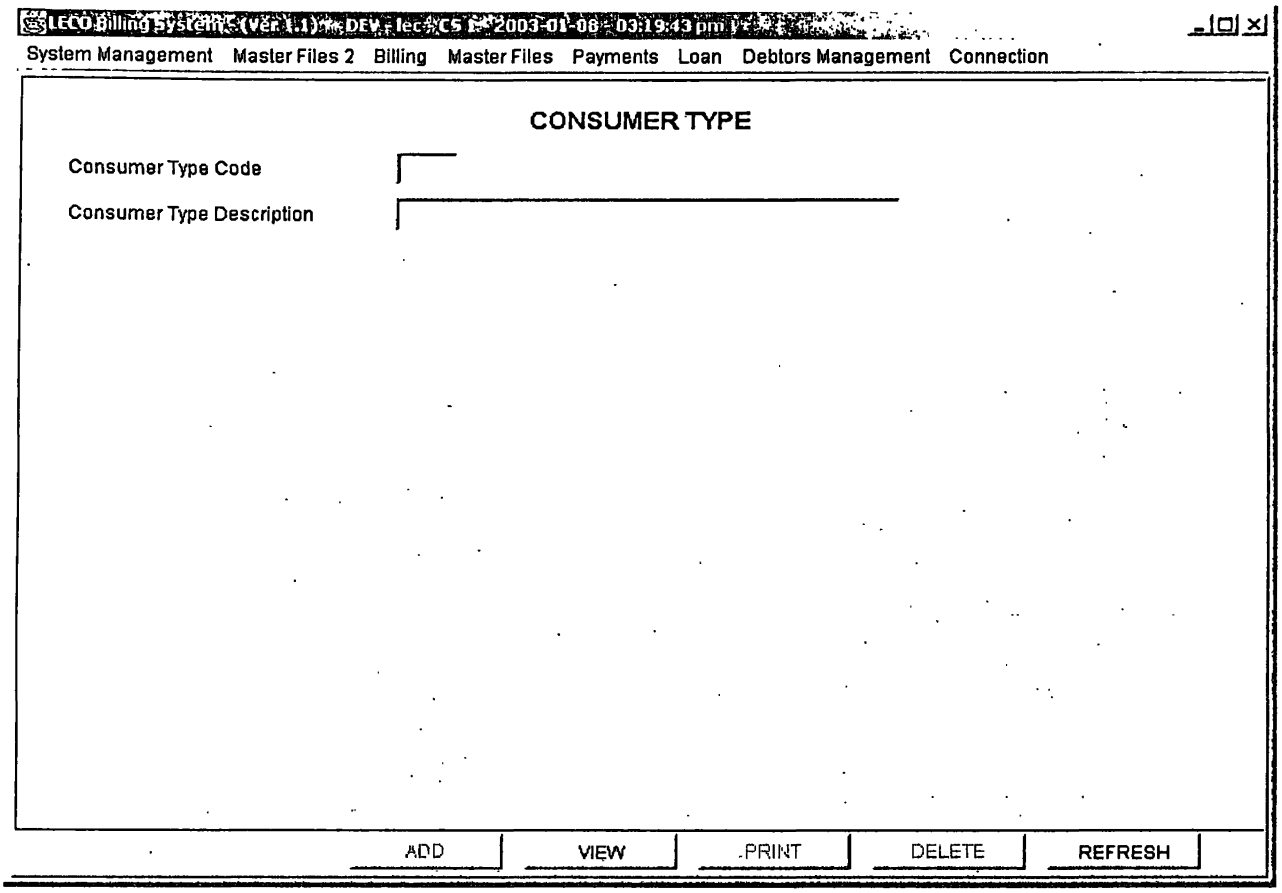
4. DISCUSSION

The goal of this project was of this project was a user friendly Billing System for LECO Pvt (Ltd). The project was initially carried out by the East West Information System company using three-tire architecture. But it failed in the acceptance level test due to poor design. We illuminated this problem using proper software development methodologies. and end up with successful software project. which meet. all the requirement of LECO. Following the correct software development methodologies can solve most of the real world IT problem.

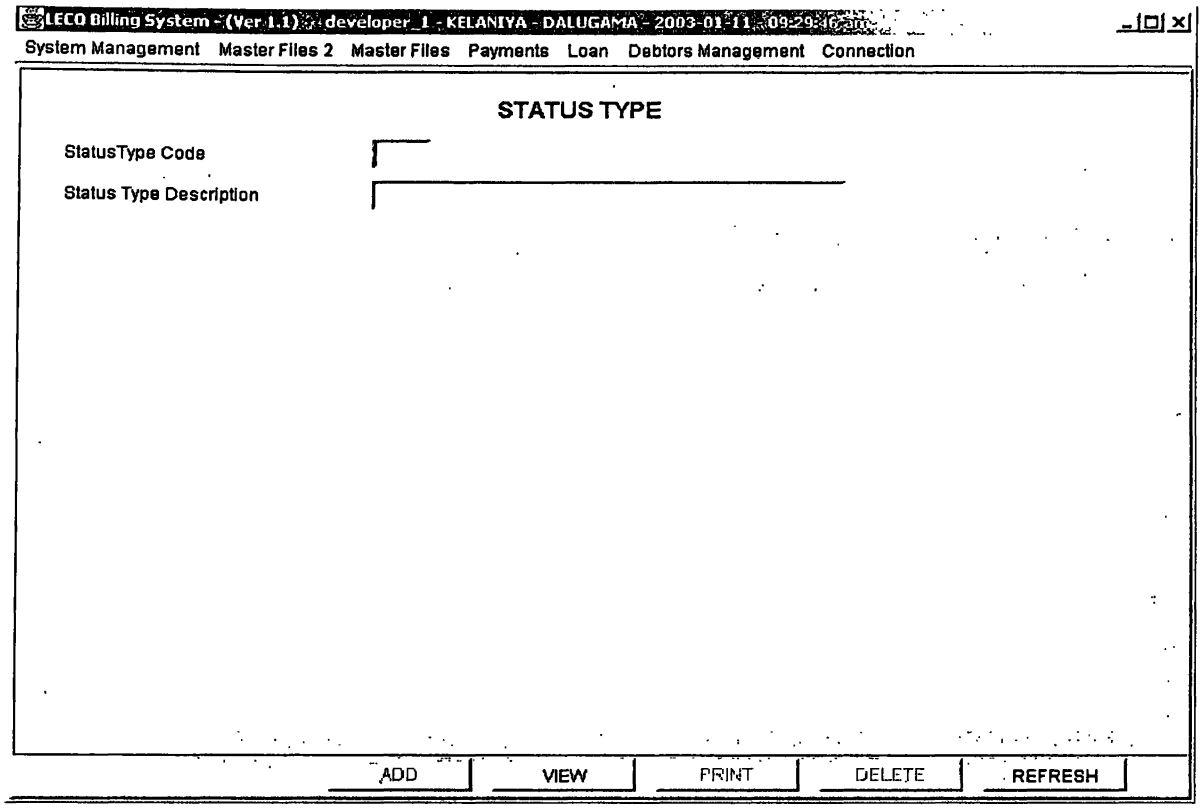
5. REFERENCE

- *Java* Application Programming interface (API) documentation of *Java.sun.com* website (Http: \\ www. Sun .java.com)
- Peter Norton's Guide To *Java Programming* Techmedia .New Delhi(1996).
- Edward Whalen's. Teach Yourself *Oracle 8T.M in 21 Days.* Techmedia .New Delhi(1998).

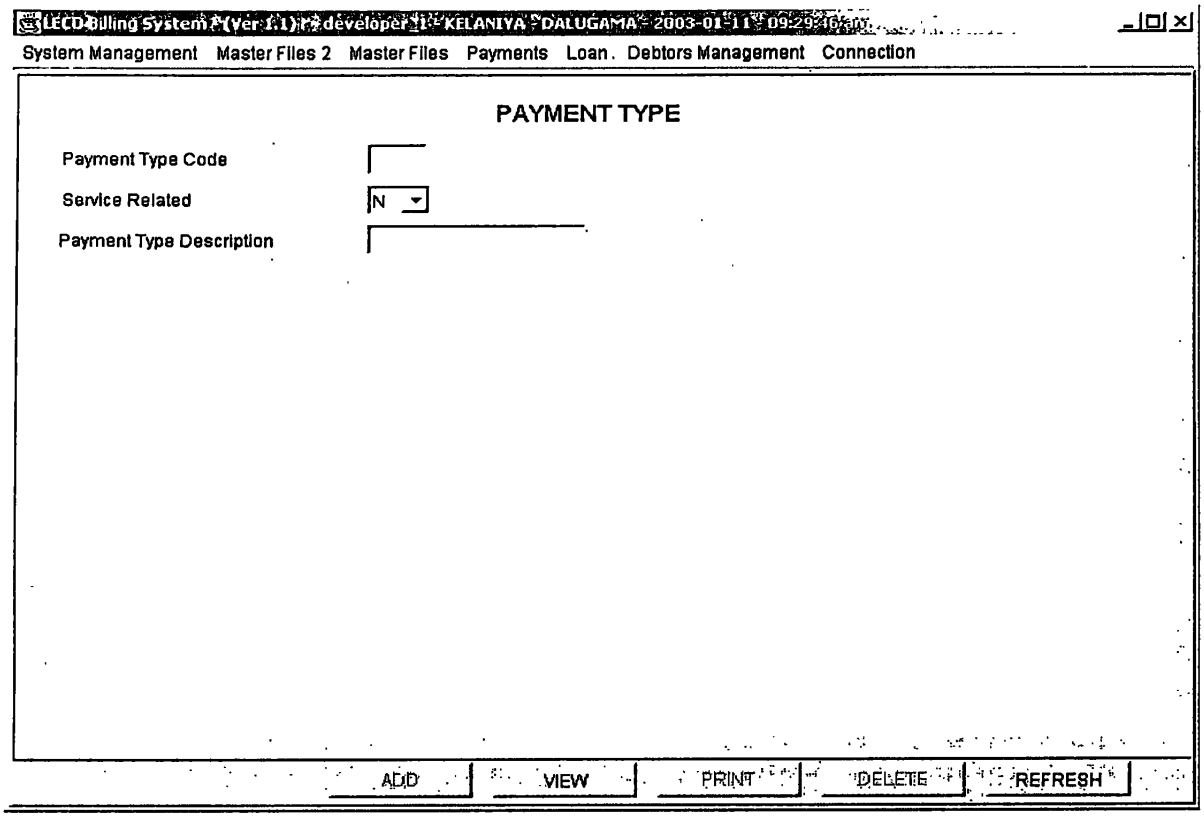
Graphical User Interface of Consumer Type master table



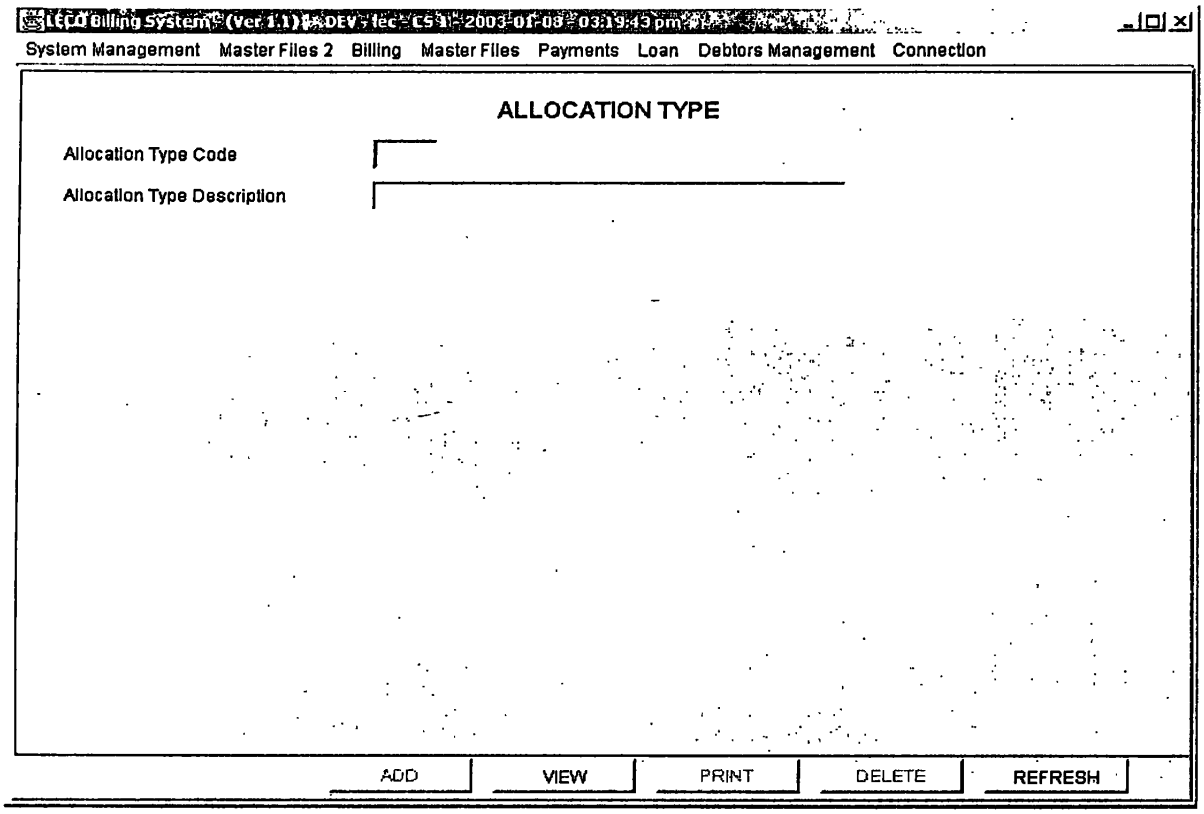
Graphical User Interface of Status Type master table



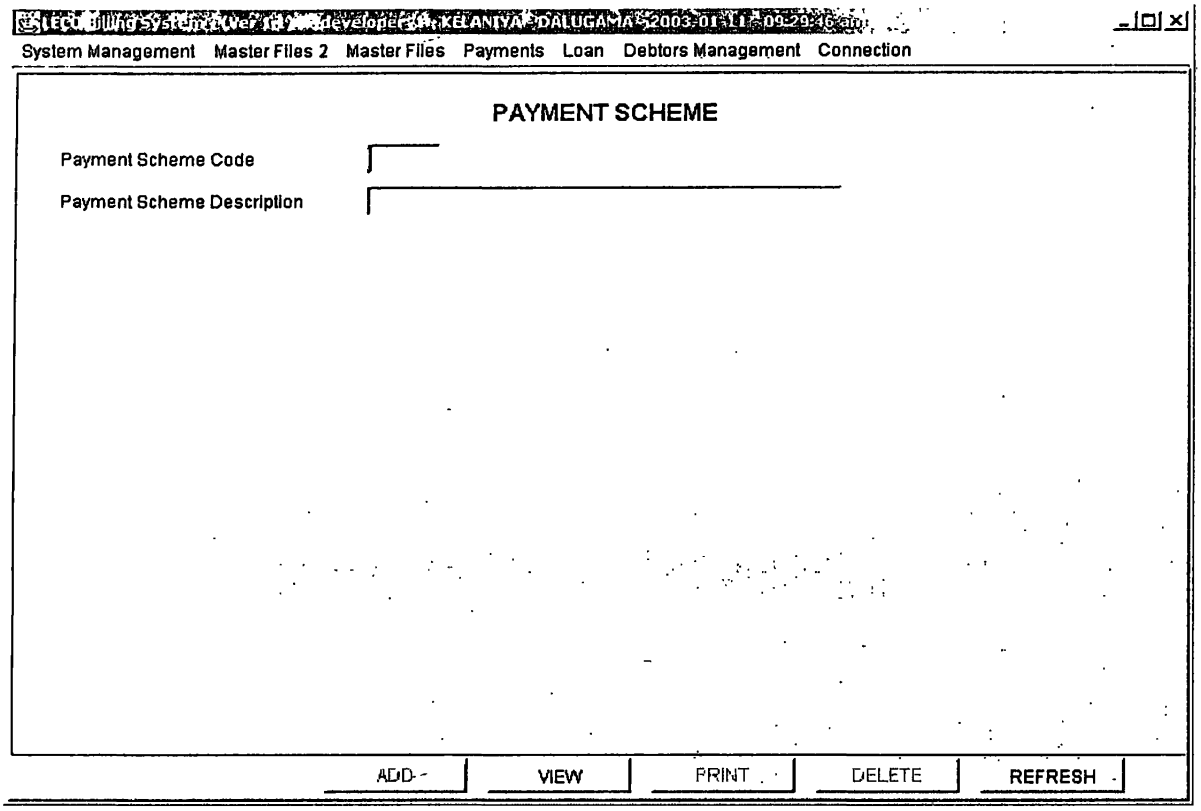
Graphical User Interface of Payment Type master table



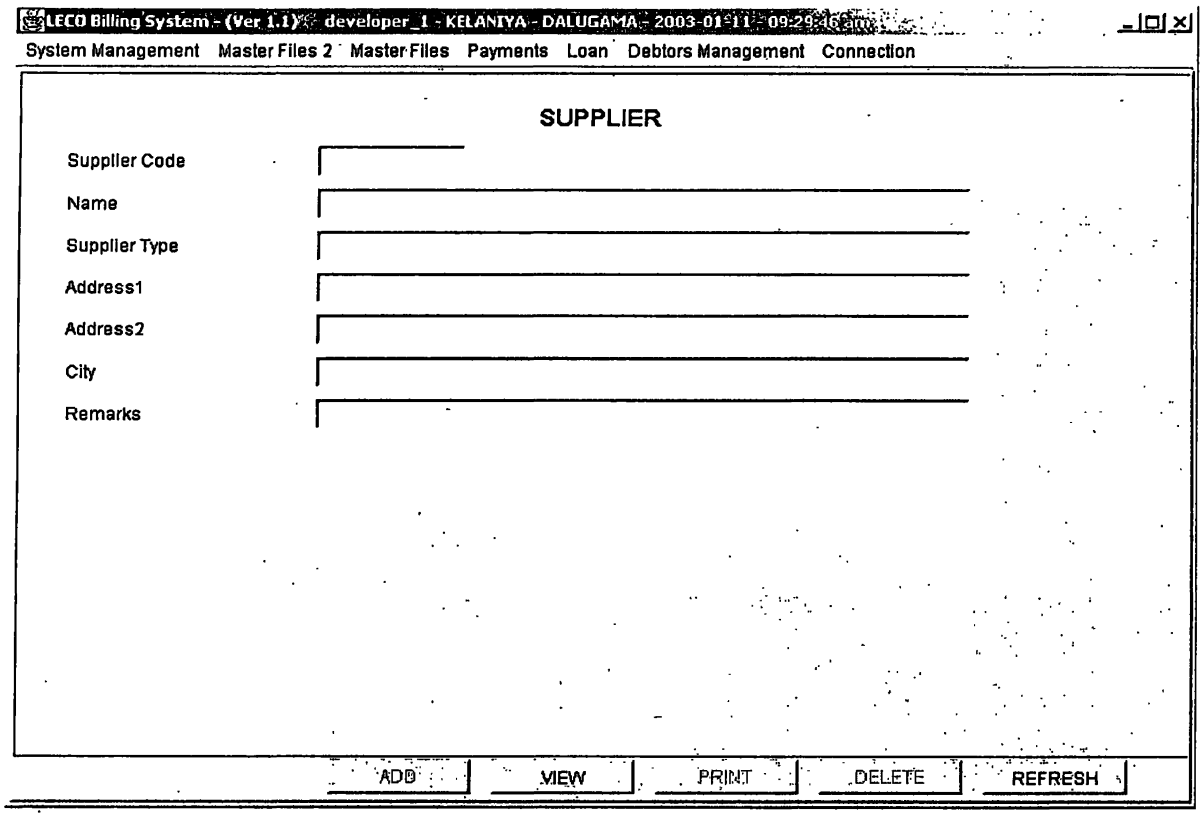
Graphical User Interface of Allocation Type master table



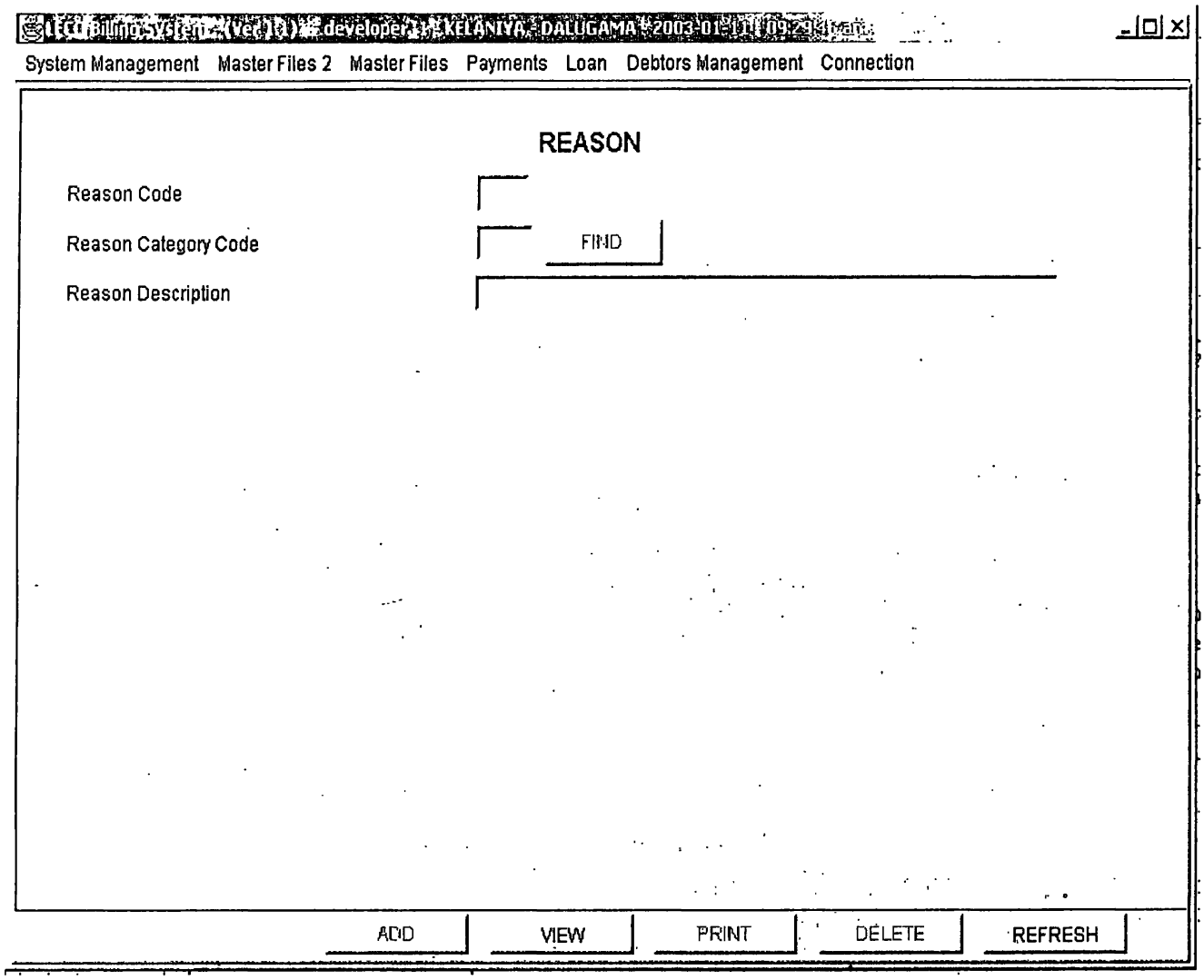
Graphical User Interface of Payment Scheme master table



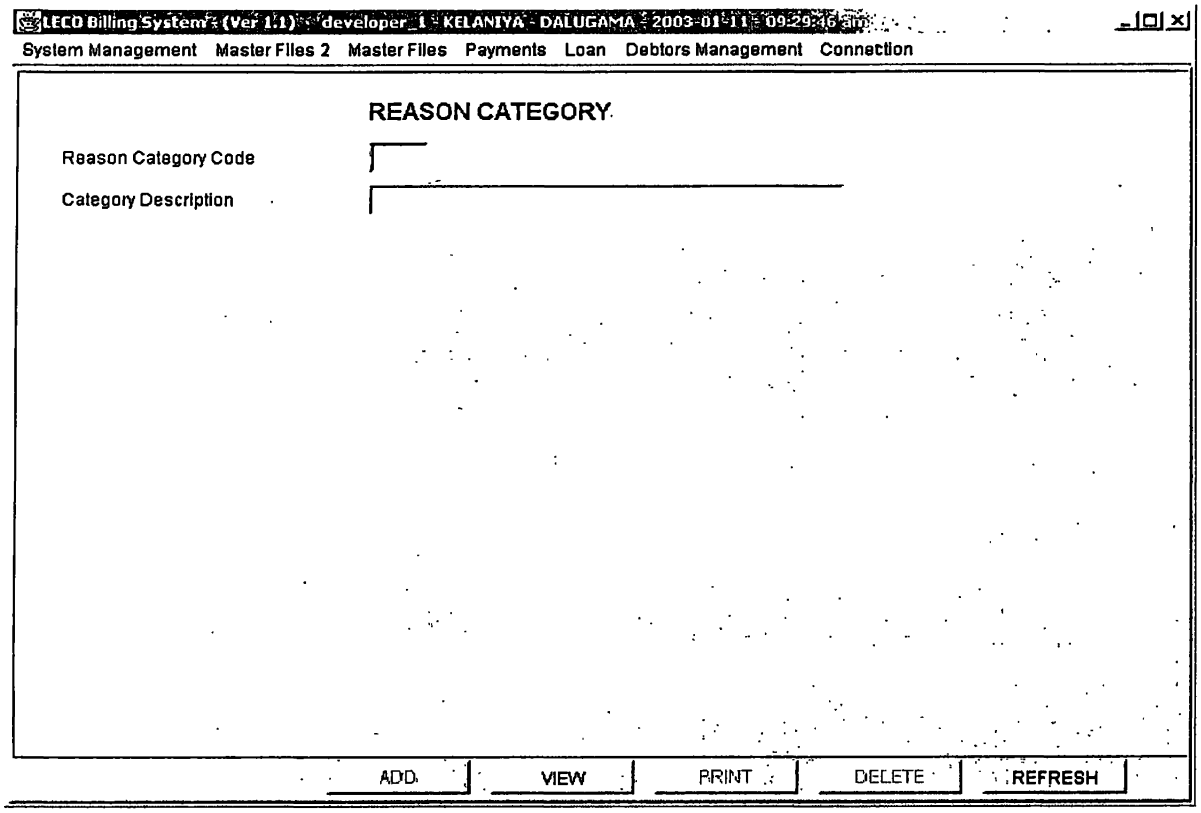
Graphical User Interface of Supplier master table



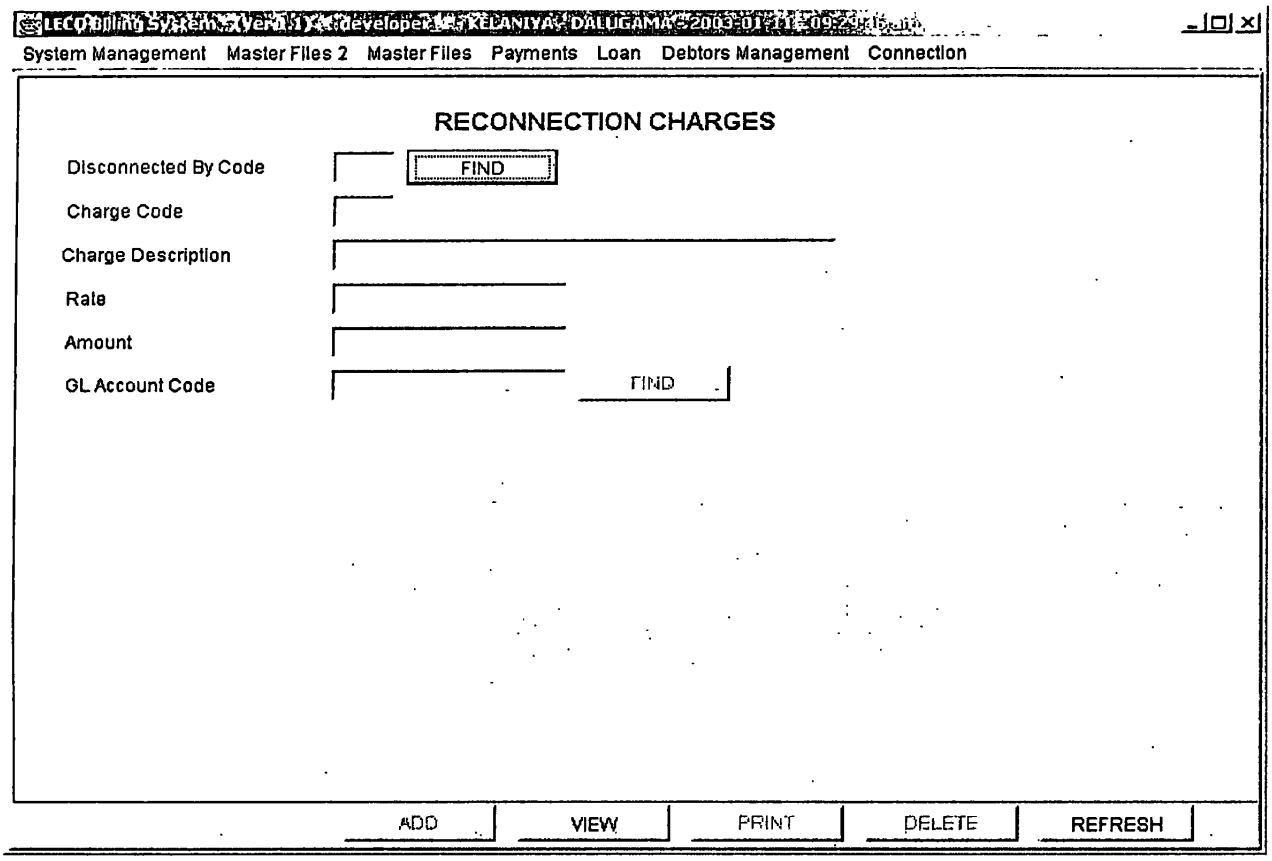
Graphical User Interface of Reason master table



Graphical User Interface of Reason Category master table



Graphical User Interface of Reconnection Charges master table



National Digitization Project

National Science Foundation

Institute : Sabaragamuwa University of Sri Lanka

1. Place of Scanning : Sabaragamuwa University of Sri Lanka, Belihuloya
2. Date Scanned : 2017-09-25
3. Name of Digitizing Company : Sanje (Private) Ltd, No 435/16, Kottawa Rd, Hokandara North, Arangala, Hokandara
4. Scanning Officer

Name : B.A.C. Gadaruwan

Signature : .....

Certification of Scanning

I hereby certify that the scanning of this document was carried out under my supervision, according to the norms and standards of digital scanning accurately, also keeping with the originality of the original document to be accepted in a court of law.

Certifying Officer

Designation : Librarian

Name : T. N. Neighsoorei

Signature : .....

Mrs.T.N.NEIGHSOOREI
(MSSc.PGD.ASLA,BA)
Librarian
Sabaragamuwa University of Sri Lanka
P.O.Box 02 Belihuloya,Sri Lanka
Tele.0094 45 2280045
Fax:0094 45 2280045

Date : 2017-09-25

"This document/publication was digitized under National Digitization Project of the National Science Foundation, Sri Lanka"