

ELECTRONIC FILE GENERATION

By

P. 2293



P. Shanmugagothandapani

Reg. No. 71205621045

Of

**KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE**

**A PROJECT REPORT
Submitted to the**

**FACUTLY OF INFORMATION AND COMMUNICATION
ENGINEERING**

*In partial fulfillment of the requirements
For the award of the degree
Of*

MASTER OF COMPUTER APPLICATIONS

July 2008

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641006

DEPARTMENT OF COMPUTER APPLICATION

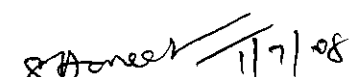
BONAFIDE CERTIFICATE


Certified that this project report titled Electronic File Generation is the bonafide work of Mr. P.Shanmugagothandapani (Registration Number: 71205621045) who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate

N. Jayakanthan
Project Guide


Head of Department

Submitted to Project and Viva Examination held on 01.07.08


Internal Examiner

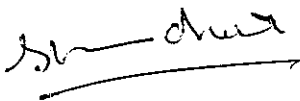

External Examiner

To Whom It May Concern

30.05.2008.
Chennai.

I certify that Mr. P.Shanmugagothandapani, final year MCA student from Kumaraguru College of Technology has completed the project assignment titled ELECTRONIC FILE GENERATION. The tenure of the project was from 28-01-2008 to 15-05-2008.

Thanking you



Shashidhar Krishnamurthy,
Associated Vice President,
Technology Center of Excellence,
Scope International.

ABSTRACT

The project entitled as “Electronic File Generation”. The main idea behind this project is design a system for bank, which provides monthly report by means of electronic. As a bank consists of huge database with different work group of a bank are interconnect and different transaction and operation are carried out. Especially in the commercial and wholesale banking data processing will be huge and continues every day. For every customer transaction the system need provide a monthly report.

In the current age of computer, effort has been made to cover every aspect of human life with faster computing power of digital computers. Especially in the commercial organizations where there is huge data to process every day and interaction between different working groups of the bank for taking crucial decisions exists, there is a desire to have comprehensive information available more quickly in order to improve internal administration, planning and processing. Therefore, systematic effort to reduce the international uncertainties becomes an important driving factor for the establishment of automatic system.

As the versatile and speed becomes the most important factor of any system, the need of any system came into the picture. Computerized system posses these properties and it is very useful, especially when huge data are to be processed and interaction between different working groups are necessary for taking crucial decisions. Therefore, the systematic effort to reduce the information uncertainty becomes important driving factor for establishment of the computerized system

ACKNOWLEDGEMENT

First and foremost I thank the Almighty for his continuous blessings showered on me in completing this project successfully.

I wish to express my profound gratitude to **Dr. Joseph V. Thanikal, Ph.D, Principal – Kumaraguru College of Technology, Coimbatore** for providing an opportunity and necessary facilities in carrying out this project work.

I am very glad to extend my deepest thanks to **Dr. M. Gururajan M.Sc., Ph.D, Head of the Department, Kumaraguru College of Technology, Coimbatore**, who has always been a source of inspiration.

I also take this opportunity to thank **Mr. A. Muthukumar M.Sc., M.C.A., Kumaraguru College of Technology, Coimbatore** for his continuous support.

My wholesome thanks to my Project mentor **Mr.S.Hameed Ibrahim M.C.A.**, for his continued encouragement and timely help during the project.

My hearty thanks to my Project mentor **Mr.N.Jayakanthan M.C.A.**, for their valuable guidance throughout the project. Also, I am grateful to my parents and friends who were the real source of my project.

I also dedicate equal and grateful acknowledgements to all the respectable members of the faculty and lab in-charges of the **Department of Computer Applications, Kumaraguru College of Technology, Coimbatore** and student friends for their motivation, encouragement and continuous support.

It's always a pleasure and privileges to be associated with a prestigious outstanding esteemed organization "**Scope International**", Chennai. I am very happy and grateful to be a part of **Scope International**.

TABLE OF CONTENTS

| CHAPTER NO. | TITLE | PAGE NO. |
|-------------|---------------------------------------|-----------|
| | ABSTRACT | iii |
| 1 | INTRODUCTION | 1 |
| | 1.1 Organisation Profile | |
| 1 | | |
| | 1.2 Problem Definition | 5 |
| 2 | SYSTEM ANALYSIS | 7 |
| | 2.1 Existing System Architecture | 5 |
| | 2.2 Proposed System Architecture | 6 |
| 3 | SYSTEM REQUIREMENTS | 10 |
| | 3.1 H/W Environment | 10 |
| | 3.2 S/W Environment | 10 |
| 4 | SYSTEM DESIGN | 23 |
| | 4.1 Data Model | 23 |
| | 4.2 Process Model | 31 |
| | 4.2.1 System Flow Diagram | 31 |
| | 4.2.2 Class Diagram | 32 |
| | 4.2.3 Use Case Diagram | 33 |
| | 4.2.4 Data Flow Diagram (DFD-Level-1) | 38 |
| | 4.2.5Data Flow Diagram (DFD-Level-2) | 39 |
| 5 | MODULES DESCRIPTION | 40 |
| 6 | SYSYTEM TESTING | 45 |

| | | |
|---|--------------------|----|
| 7 | CONCLUSION | 49 |
| 8 | APPENDIXES | 50 |
| | 8.1 Sample screens | 50 |
| 9 | REFERENCE | 66 |

List of Figures

| Figure No | Figure Name | Page No |
|------------------|---------------------------------|----------------|
| 3.2 | Java Swing class hierarchy | 16 |
| 4.2.1 | System Flow Diagram | 31 |
| 4.2.2 | Class Diagram | 32 |
| 4.2.3 | Use Case Diagram | 33 |
| 4.2.4 | Data Flow Diagram (DFD-Level-1) | 38 |
| 4.2.5 | Data Flow Diagram (DFD-Level-1) | 39 |

INTRODUCTION

CHAPTER-1

INTRODUCTION

1.1 ORGANIZATION PROFILE

Scope International, a wholly owned subsidiary of Standard Chartered Bank, UK, is an integral part of the Bank's strategy to consolidate and rationalise operations into Shared Services Centres. This is an ambitious project that the Bank undertook and today Scope International services all 56 countries of the Standard Chartered Group processing over 80 million transactions a year.

The facility in Chennai is one of two hubs that the Bank has set up, the other being in Kuala Lumpur. Scope International has today emerged as one of the top captive banking back offices in India for international companies.

The operations in Scope are very diverse and range from banking operations to supporting global HR processes and finance and accounting services, software development and maintenance, support for global treasury operations and providing IT helpdesk support. Scope International has adopted IDEAL as the Quality methodology in the organization to create a highly effective metrics driven performance culture.

Scope employs approximately 10% of the Group's global workforce. Talented professionals from various backgrounds enrich the skill mix of the company. The average age in Scope is 28 with a male to female ratio of 70% to 30%. Scope Centre has been created as a world-class custom designed facility with a built up area of about half a million square feet. The state-of-the-art Disaster Recovery site is located in Padur, 35 kilometres from Scope Centre.

The organization is motivated and totally committed to the development of its people; Scope has a flat structure; its HR policies are innovative and employee friendly. Scope embraces the Bank's values and employees are encouraged and rewarded for living the values at work. Scope differentiates itself in the community by being a trusted and proactive member.

Scope International, the Global Shared Services Centre of the Standard Chartered Group, is a pioneer in off-shoring knowledge-based services in the banking industry. The company was incorporated in 2000 and commenced actual operations in Chennai from 2001. In a short span of six years, Scope has grown from zero to over 6000 employees and services all 50 countries of the Group. Scope International is housed in world-class facilities across five campuses in Chennai. Its flagship, Scope Centre, is in the heart of the city and stretches across nearly 6 acres of prime real estate. Scope International has robust disaster recovery practices; it has an 800 seat disaster recovery site about 35 km away from its main centre.

The average age at Scope is 29 with a male female ratio of 69 to 31. Scope nurtures its employees with innovative HR practices and learning and development opportunities. It has the unique distinction of being able to offer short, long term and permanent moves into the Standard Chartered Group countries for aspiring and qualified staff.

Scope handles a wide range of value-added and complex services from Banking Operations to supporting global HR processes, Finance and Accounting services, Software Development and maintenance, and providing IT Service and Helpdesk support to the Group globally. It has also expanded to offer services such as Basel, Anti-Money Laundering, Compliance, Legal and Credit Analytics.

In line with the Corporate Responsibility (CR) initiatives of the Group, Scope focuses on improving the quality of life for the community through programmes that are long term and sustainable in the areas of health, education, the differently abled and the environment. This involves employee volunteering, staff involvement and building credibility in the community, making the company an employer of choice. Its Diversity and Inclusion initiatives focus on women empowerment and getting the best out of the broadest spectrum of people to sustain strong business performance and competitive advantage.

The company's Corporate Social Responsibility agenda is closely aligned with the Group's strategy. Scope is currently into the second stage of the Group's 'Seeing is Believing' programme, which aims to restore sight to 1 million people in three years, with support from customers, suppliers and special events. 'Living with HIV', a major Group initiative, to spread awareness on HIV/AIDS is pursued relentlessly by Scope. Their staffs are involved in championing this awareness program in the community.

The Scope culture promotes customer focus, teamwork, excellence, openness and trust. Scope International aspires to be a benchmark amongst banking BPOs.

Scope International is the Right Partner to the Bank in its strategic journey of becoming the world's best international bank leading the way in Asia, Africa and the Middle East.

1.2 Problem Definition

The Electronic File Generation is a system which allows you to record, control and analyze customer transaction in the bank. The system need to maintain the data base to record the customer details and their transaction details. The bank manager analysis customer transaction to produce monthly report and based on the criteria he has the right to select text file need for report or reject the file. And in addition the managers use this application for assessing the customer transaction. The customer will use this application for withdraw money and e-shopping by specifying the account types. It is difficult to maintain every customer transaction report in ledger book. It is also a difficult task for the bank manger to merge all transaction file in manual.

ANALYSIS OF REQUIREMENTS

CHAPTER-2

SYSYTEM ANALYSIS

The Bank operation includes customer transaction such as core Banking, Loan system and Card transaction. For a commercial banking the operation of a customer will be opening a saving account for money transfer and for a loan customer it will be a repayment of due amount and also customer credit and debit card transaction are done. In the Wholesale banking the operation of a customer will be opening a privilege account for fund transfer and customer corporate loan access directly to bank through “Straight 2 Bank” system.

Existing System:

In traditional Banking system it involves a huge data processing and transactions that need to be loaded on the database. All transaction of customer are updated in the database and written on the ledger book. It leads to bunch of paper work every day. A monthly report is generated based on the ledger book which contains all customer transaction as text documents.

Problems in existing system

- In above case, it is tedious to generate monthly report on paper for all transaction.
- This creates slow Work in progress
- The maintenance of all customer transaction leads to confusing for the bank manager.

Proposed System:

The Bank has different database for its operation such as Core banking, Loan System and Card transaction. All transaction in the database is feeds into the single database to provide monthly report. For every customer transaction text file is generated. And File Merger is a method which is used to merge all transactions files. All text file generated are merged into a single electronic document.

Advantage in Proposed system:

- Any size of text file can be merged and report is generated in very short time.
- The report contains page header, page footer and path which is useful for manager to analyzes.

SYSTEM REQUIREMENTS

CHAPTER-3

SYSTEM REQUIREMENTS

3.1 Hardware Environment:

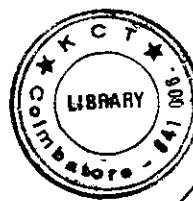
To develop the project titled “Electronic File Generation” the following hardware specification is used.

| | |
|-----------|------------------|
| Processor | INTEL Pentium IV |
| CPU Speed | 2.66GHz |
| Hard Disk | 80GB |
| Monitor | HP LCD |
| Keyboard | HP |
| RAM | 1GB |
| Mouse | HP |

3.2 Software Environment:

To develop the project titled “Electronic File Generation” the following software specification is used.

| | |
|------------------|-----------------------------------|
| Operating System | Microsoft Windows XP Professional |
| IDE | Eclipse |
| Front End | JAVA 1.5/1.6 |
| Server | IBM Websphere |
| Back End | SQL Server |



Software Description:

Windows XP Professional:

Windows XP Professional integrates the strengths of Windows 2000, such as standards-based security, manageability, and reliability, with the best features of Windows 98 and Windows Millennium Edition.

The Microsoft® Windows® XP Professional operating system includes a variety of technologies that communicate with the Internet to provide increased ease of use and functionality. Browser and e-mail technologies are obvious examples, but there are also technologies such as Automatic Updates that help users obtain the latest software and product information, including bug fixes and security patches. These technologies provide many benefits, but they also involve communication with Internet sites, which administrators might want to control.

Control of this communication can be achieved through a variety of options built into individual components, into the operating system as a whole, and into server components designed for managing configurations across your organization. For example, as an administrator, you can use Group Policy to control the way some components communicate. For some components, you can direct all communication to the organization's own internal Web site instead of to an external site on the Internet.

This white paper provides information about the communication that flows between components in Windows XP Professional with either Service Pack 1 or Service Pack 1a (both referred to as "SP1" in this white paper) and sites on the Internet, and describes steps to take to limit, control, or prevent that communication in an organization with many users. The white paper is designed to assist you, the administrator, in planning strategies for deploying and maintaining Windows XP Professional with SP1 in a way that helps to provide an appropriate level of security and privacy for your organization's networked assets.

Eclipse:

If you closely follow open source or Java programming, you may have heard some of the buzz surrounding Eclipse. Eclipse is an extensible, open source IDE (integrated development environment). The project was originally launched in November 2001, when IBM donated \$40 million worth of source code from Websphere Studio Workbench and formed the Eclipse Consortium to manage the continued development of the tool.

The stated goals of Eclipse are "to develop a robust, full-featured, commercial-quality industry platform for the development of highly integrated tools." To that end, the Eclipse Consortium has been focused on three major projects:

1. The Eclipse Project is responsible for developing the Eclipse IDE workbench (the "platform" for hosting Eclipse tools), the Java Development Tools (JDT), and the Plug-In Development Environment (PDE) used to extend the platform.
2. The Eclipse Tools Project is focused on creating best-of-breed tools for the Eclipse platform. Current subprojects include a Cobol IDE, a C/C++ IDE, and an EMF modeling tool.
3. The Eclipse Technology Project focuses on technology research, incubation, and education using the Eclipse platform.

The Eclipse platform, when combined with the JDT, offers many of the features you'd expect from a commercial-quality IDE: a syntax-highlighting editor, incremental code compilation, a thread-aware source-level debugger, a class navigator, a file/project manager, and interfaces to standard source control systems, such as CVS and Clear Case.

Eclipse also includes a number of unique features such as code refactoring, automatic code updates/installs (via the Update Manager), a task list, support for unit testing with JUnit, and integration with the Jakarta Ant build tool.

Despite the large number of standard features, Eclipse is different from traditional IDEs in a number of fundamental ways. Perhaps the most interesting feature of Eclipse is that it is completely platform- and language-neutral. In addition to the eclectic mix of languages supported by the Eclipse Consortium (Java, C/C++, Cobol), there are also projects underway to add support for languages as diverse as Python, Eiffel, PHP, Ruby.

With hundreds of plug-in development projects in progress, industry giants like IBM, HP, and Rational (just acquired by IBM) providing resources, and design heavy-weights like Erich Gamma helping to guide the process, the future indeed looks bright for Eclipse.

Java:

Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (name after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C. Sun Microsystems currently maintains and updates Java regularly.

Java, a platform independent programming language helps in building any kind of application, Java uses a compiler to convert the source code into architectural independent byte code. These are executed over a Java Virtual Machine (JVM), which is an idealized java processor chip usually implemented in software rather than hardware. Java was developed to include methods for internet data manipulation, Java applications can be written once and run on any machine having a Java Virtual machine as part of its operating system.

Features of Java:

Platform Independent

Object-Oriented language

Secure

Portable

Robust

Image processing

Web development support

Supports Multithreading

Java Swing:

Swing is a GUI toolkit for Java. It is one part of the Java Foundation Classes (JFC). Swing includes graphical user interface (GUI) widgets such as text boxes, buttons, splitpanes and tables. Swing widgets provide more sophisticated GUL components than the early Abstract Window Toolkit. Since they are written in pure Java, they run the same on all platforms, that is uniform behavior in all platform, unlike the AWT which is tied to the underlying platform's windowing system.

Swing supports pluggable look and feel – not by using the native platform's facilities, but by roughly emulating them. This means you can get any supported look and feel on any platform .It follows a single-threaded programming model.

A part of The JFC:

Swing Java consists of

Look and feel

Accessibility

Java 2D

Drag and Drop, etc

if you do not explicitly add a GUI component to a container, the GUI component will not be displayed when the container appears on the screen.

Swing, which is an extension library to the AWT, includes new and improved components that enhance the look and functionality of GUIs. Swing can be used to build Standalone swing gui Apps as well as Servlets and Applets. It employs a model/view design architecture. Swing is more portable and more flexible than AWT.

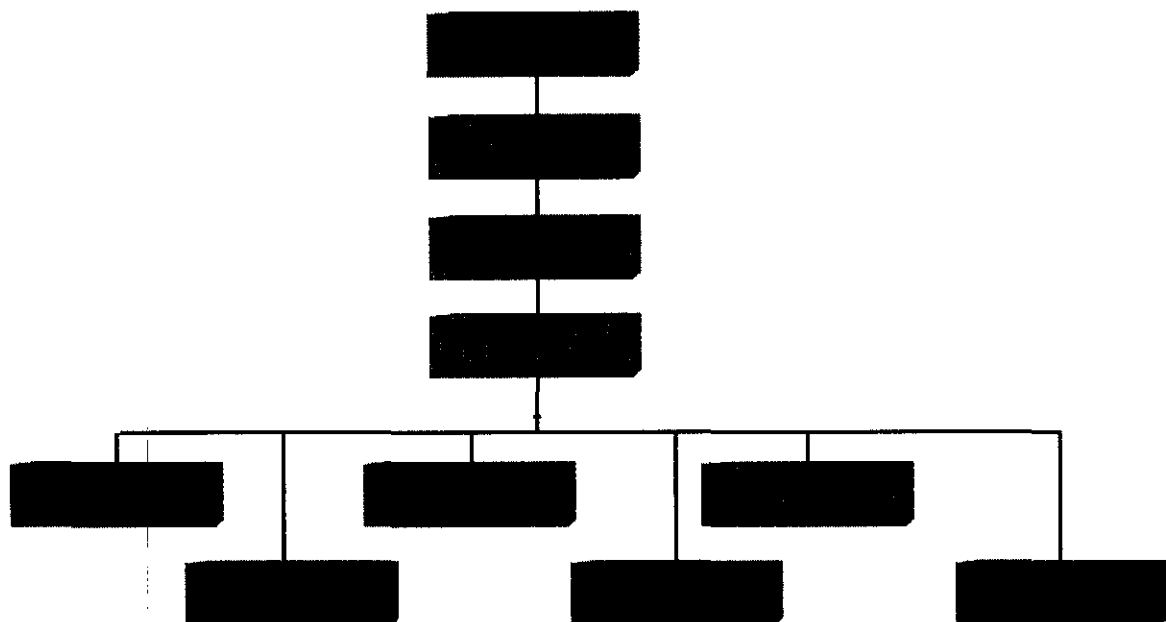
Swing Model/View Design: The "view part" of the MV design is implemented with a component object and the UI object. The "model part" of the MV design is implemented by a model object and a change listener object.

Swing is built on top of AWT and is entirely written in Java, using AWT's lightweight component support. In particular, unlike AWT, the architecture of Swing components makes it easy to customize both their appearance and behavior. Components from AWT and Swing can be mixed, allowing you to add Swing support to existing AWT-based programs. For example, swing components such as JSlider, JButton and JCheckbox could be used in the same program with standard AWT labels, textfields and scrollbars. You could subclass the existing Swing UI, model, or change listener classes without having to reinvent the entire implementation. Swing also has the ability to replace these objects on-the-fly.

In Swing, classes that represent GUI components have names beginning with the letter J. Some examples are JButton, JLabel, and JSlider. Altogether there are more than 250 new classes and 75 interfaces in Swing — twice as many as in AWT.

Java Swing class hierarchy:

The class JComponent, descended directly from Container, is the root class for most of Swing's user interface components.



Swing contains components that you'll use to build a GUI. I am listing you some of the commonly used Swing components. To learn and understand these swing programs, AWT Programming knowledge **is not** required.

JPanel is Swing's version of the AWT class Panel and uses the same default layout, FlowLayout. JPanel is descended directly from JComponent.

JFrame is Swing's version of Frame and is descended directly from that class. The components added to the frame are referred to as its contents; these are managed by the contentPane. To add a component to a JFrame, we must use its contentPane instead.

JInternalFrame is confined to a visible area of a container it is placed in. It can be iconified , maximized and layered.

JWindow is Swing's version of Window and is descended directly from that class. Like Window, it uses BorderLayout by default.

JDialog is Swing's version of Dialog and is descended directly from that class. Like Dialog, it uses BorderLayout by default. Like JFrame and JWindow, JDialog contains a rootPane hierarchy including a contentPane, and it allows layered and glass panes. All dialogs are modal, which means the current thread is blocked until user interaction with it has been completed. JDialog class is intended as the basis for creating custom dialogs; however, some of the most common dialogs are provided through static methods in the class JOptionPane.

JLabel, descended from JComponent, is used to create text labels.

The abstract class AbstractButton extends class JComponent and provides a foundation for a family of button classes, including

JBotton_

JTextField allows editing of a single line of text. New features include the ability to justify the text left, right, or center, and to set the text's font.

JPasswordField (a direct subclass of JTextField) you can suppress the display of input. Each character entered can be replaced by an echo character.

This allows confidential input for passwords, for example. By default, the echo character is the asterisk, *.

JTextArea allows editing of multiple lines of text. JTextArea can be used in conjunction with class JScrollPane to achieve scrolling. The underlying JScrollPane can be forced to always or never have either the vertical or horizontal scrollbar;

JButton is a component the user clicks to trigger a specific action.

JRadioButton is similar to JCheckbox, except for the default icon for each class. A set of radio buttons can be associated as a group in which only one button at a time can be selected.

JCheckBox is not a member of a checkbox group. A checkbox can be selected and deselected, and it also displays its current state.

JComboBox is like a drop down box. You can click a drop-down arrow and select an option from a list. For example, when the component has focus, pressing a key that corresponds to the first character in some entry's name selects that entry. A vertical scrollbar is used for longer lists.

JList provides a scrollable set of items from which one or more may be selected. JList can be populated from an Array or Vector. JList does not support scrolling directly, instead, the list must be associated with a scrollpane. The view port used by the scroll pane can also have a user-defined border. JList actions are handled using ListSelectionListener.

JTabbedPane contains a tab that can have a tool tip and a mnemonic, and it can display both text and an image.

JToolBar contains a number of components whose type is usually some kind of button which can also include separators to group related components within the toolbar.

FlowLayout when used arranges swing components from left to right until there's no more space available. Then it begins a new row below it and moves from left to right again. Each component in a FlowLayout gets as much space as it needs and no more.

BorderLayout places swing components in the North, South, East, West and center of a container. You can add horizontal and vertical gaps between the areas.

GridLayout is a layout manager that lays out a container's components in a rectangular grid. The container is divided into equal-sized rectangles, and one component is placed in each rectangle.

GridBagLayout is a layout manager that lays out a container's components in a grid of cells with each component occupying one or more cells, called its display area. The display area aligns components vertically and horizontally, without requiring that the components be of the same size.

JMenuBar can contain several JMenu's. Each of the JMenu's can contain a series of JMenuItem 's that you can select. Swing provides support for pull-down and popup menus.

Scrollable JPopupMenu is a scrollable popup menu that can be used whenever we have so many items in a popup menu that exceeds the screen visible height.

Swing Features:

Platform Independent

Extensibility

Component-Oriented

Customizable

Configurable

Lightweight User Interface

Loosely-Coupled

Look and feel

SQL Server:

SQL Server is a database. A database is a data storage feature. It can be used to store, sort, arrange, and display information. SQL Server is a functional feature on it's own. For our tutorials, we will be using PHP commands to use the functions of a SQL Server database. phpMyAdmin is a graphical interface program that allows you to use the functions of a SQL Server database. Some hosts may have this program available. SQL Server is a data storage area. In this storage area, there are small sections called tables. Very similar to a normal HTML table, the SQL Server tables consist of rows, columns, and cells.

SQL Server is currently the most popular open source database server in existence. On top of that, it is very commonly used in conjunction with PHP scripts to create powerful and dynamic server-side applications.

SQL Server has been criticized in the past for not supporting all the features of other popular and more expensive DataBase Management Systems. However, SQL Server continues to improve with each release (currently version 5), and it has become widely popular with individuals and businesses of many different sizes.

A database is a structure that comes in two flavors: a flat database and a relational database. A relational database is much more oriented to the human mind and is often preferred over the gabble-de-gook flat database that are just stored on hard drives like a text file. SQL Server is a relational database.

In a relational structured database there are tables that store data. The columns define which kinds of information will be stored in the table. An individual column must be created for each type of data you wish to store .

On the other hand, a row contains the actual values for these specified columns. Each row will have 1 value for each and every column. For example a table with columns) could have a row with the values.If all this relational database talk is too confusing, don't despair. We will talk about and show a few examples in the coming lessons.

Databases are most useful when it comes to storing information that fits into logical categories. For example, say that you wanted to store information of all the employees in a company. With a database you can group different parts of your business into separate tables to help store your information logically.

SYSTEM DESIGN

CHAPTER-4

SYSTEM DESIGN

4.1 Data Design:

The system uses MSSQL SERVER as database and it contains the following tables.

Login

Customer_Details

Customer_Transaction

Card_Details

Credit_Card_Transaction

Debit_Card_Transaction

Loan_Details

Loan_Customer

Customer_Loan_Repayments

Table Name: Login

Primary Key: User_Id

Foreign Key: nil

Table 4.1.1 Login

| Attribute | Data Type | Size | Description |
|-----------|-----------|------|-------------------------|
| User_Id | Varchar | 20 | Username |
| Password | Varchar | 10 | Password for User login |
| Category | Varchar | 15 | Category for User |

Table Name: Customer Details

Primary Key: Account _no

Foreign Key: nil

Table 4.1.Customer Details

| Attribute | Data Type | Size | Description |
|--------------------|-----------|------|------------------------------------|
| Account number | Integer | | Auto increment Primary Key |
| Name | Varchar | 20 | Name of the Customer |
| Father name | Varchar | 20 | Father name of the Customer |
| Sex | Varchar | 6 | Gender of the Customer |
| Date of Birth | Date | | Birth date of Customer |
| Profession | Varchar | 15 | Profession of the Customer |
| Qualification | Varchar | 10 | Qualification of the Customer |
| Marital status | Varchar | 10 | Marital status of the Customer |
| Income by salary | Float | | Salary of the Customer |
| Income by business | Float | | Income of the Customer by business |
| Address | Varchar | 30 | Address of the Customer |
| City | Varchar | 15 | City of the Customer |
| Pin code | Integer | | Pin code of Customer |
| State | Varchar | 15 | State of the Customer |
| Contact number | Integer | | Phone number of the Customer |
| Fax | Integer | | Fax number of the Customer |
| Email Id | Varchar | 20 | Email Id of the Customer |
| Bank branch | Varchar | 20 | Bank branch of the Customer |
| Account type | Varchar | 10 | Account type of the Customer |
| Category | Varchar | 15 | Category of the Customer |
| Balance amount | Float | | Balance amount |
| Date | Date | | Account open date |

Table Name: Customer Transaction
 Primary Key: Transaction_no
 Foreign Key: Account_no

Table 4.1.3 Customer Transaction

| Attribute | Data Type | Size | Description |
|---------------------|-----------|------|--|
| Account number | Integer | | Account number of the Customer |
| Name | Varchar | 20 | Name of the Customer |
| Bank branch | Varchar | 15 | Bank branch of the Customer |
| Account type | Varchar | 10 | Account type of the Customer |
| Category | Varchar | 15 | Category of the account |
| Type of transaction | Varchar | 10 | Transaction type i.e., Deposit or Withdraw |
| Transaction amount | Float | | Transaction amount of the Customer |
| Balance amount | Float | | Balance amount of the Customer |
| Date | Date | | Date of transaction |
| Mode of transaction | Varchar | 10 | Mode of amount transfer |

Table Name: Card Details
 Primary Key: Card_Id
 Foreign Key: Account_no

Table 4.1.4 Card Details

| Attribute | Data Type | Size | Description |
|----------------|-----------|------|--------------------------------|
| Account number | Integer | | Account number of the customer |
| Name | Varchar | 15 | Name of the customer |
| Account type | Varchar | 10 | Account type of the customer |
| Card Id | Integer | | Card id of the customer |

| | | | |
|-------------------|---------|----|--------------------------------|
| Card type | Varchar | 10 | Card type of the customer |
| Valid date | Date | | Card valid date |
| Card value | Float | | Card amount or value |
| Card charge | Float | | Card charges for transaction |
| Balance | Float | | Balance amount of the customer |
| Card renewal date | Date | | Date of card renewal |
| Category | Varchar | 15 | Category of the Customer |

Table Name: Credit_card_Transaction

Primary Key: Card_Id

Foreign Key: Account_no

Table 4.1.5 Credit Card Transaction

| Attribute | Data Type | Size | Description |
|----------------|-----------|------|---------------------------------|
| Account number | Integer | | Account number of the Customer |
| Card Id | Integer | | Card Id of the Credit card |
| Name | Varchar | 20 | Name of the Customer |
| Card type | Varchar | 10 | Card type of the Customer |
| Pin number | Integer | | Pin number of the Credit card |
| Bank branch | Varchar | 15 | Bank branch of the Customer |
| Account type | Varchar | 10 | Account type of the Customer |
| Category | Varchar | 15 | Category of the Customer |
| Date | Date | | Date of Credit card transaction |
| Amount spent | Float | | Amount spent on the purchase |
| Card Value | Float | | Value of the Card |
| Due Date | Date | | Repaying Date |
| Interest | Float | | Interest for repayable amount |

Table Name: Debit_card_Transaction

Primary Key: Card_Id

Foreign Key: Account_no

Table 4.1.6 Debit Card Transaction

| Attribute | Data Type | Size | Description |
|-----------------|-----------|------|---------------------------------|
| Account number | Integer | | Account number of the Customer |
| Card Id | Integer | | Card Id of the Debit card |
| Name | Varchar | 20 | Name of the Customer |
| Card type | Varchar | 10 | Card type of the Customer |
| Pin number | Integer | | Pin number of the Debit card |
| Bank branch | Varchar | 15 | Bank branch of the Customer |
| Account type | Varchar | 10 | Account type of the Customer |
| Category | Varchar | 15 | Category of the Customer |
| Date | Date | | Date of Credit card transaction |
| Amount withdraw | Float | | Amount withdraw from ATM |
| Balance amount | Float | | Balance amount of the Customer |
| Category | Varchar | 15 | Category of the Customer |
| Card Charge | Float | | Transaction charge |

Table Name: Bank_Loan_Details

Primary Key: Loan_Id

Foreign Key: nil

Table 4.1.7 Bank Loan Details

| Attribute | Data Type | Size | Description |
|----------------------------|-----------|------|--|
| Loan Id | Integer | | Loan number for the loan details |
| Loan type | Varchar | 15 | Type of loan |
| Finance amount | Float | | Finance amount for the Loan |
| Flat rate | Float | | Flat rate for the loan |
| Interest amount | Float | | Interest amount for the loan |
| Bank branch | Varchar | 15 | Bank branch |
| Number of loans allotted | Integer | | Number of loans allotted for the Bank branch |
| Number of loans sanctioned | Integer | | Number of loans sanctioned by Bank branch |

Table Name: Loan_Customer_Details

Primary Key: Customer_Loan_no

Foreign Key: Loan_Id

Table 4.1.8 Loan Customer Details

| Attribute | Data Type | Size | Description |
|----------------------|-----------|------|--------------------------------|
| Customer loan number | Integer | | Auto increment Primary Key |
| Name | Varchar | 20 | Name of the Customer |
| Father name | Varchar | 20 | Father name of the Customer |
| Sex | Varchar | 6 | Gender of the Customer |
| Date of Birth | Date | | Birth date of Customer |
| Profession | Varchar | 15 | Profession of the Customer |
| Qualification | Varchar | 10 | Qualification of the Customer |
| Marital status | Varchar | 10 | Marital status of the Customer |

| | | | |
|-----------------------|---------|----|---|
| Income by salary | Float | | Salary of the Customer |
| Income by business | Float | | Income of the Customer by business |
| Income by property | Float | | Income of the Customer by property |
| Own | Varchar | 20 | Details of the property owned by the Customer |
| Address | Varchar | 30 | Address of the Customer |
| City | Varchar | 15 | City of the Customer |
| Pin code | Integer | | Pin code of Customer |
| State | Varchar | 15 | State of the Customer |
| Contact number | Integer | | Phone number of the Customer |
| Fax | Integer | | Fax number of the Customer |
| Email Id | Varchar | 20 | Email Id of the Customer |
| Loan Id | Integer | | Loan Id of the Customer |
| Invoice amount | Float | | Invoice amount for the loan |
| Finance amount | Float | | Finance amount for the loan |
| Flat rate | Float | | Flat rate of the loan amount |
| Interest amount | Float | | Interest for the loan amount |
| Total amount | Float | | Total loan amount |
| Number of installment | Float | | Number of installment for the loan amount |
| Due Amount | Float | | Due amount for each month |
| Bank branch | Varchar | 15 | Bank branch of the customer |
| Date | Date | | Date of the loan taken by Customer |
| Mode of pay | Varchar | 10 | Mode of amount paid |
| Loan type | Varchar | 15 | Loan type |

Table Name: Loan_Repayment
Primary Key: Customer_Loan_Id
Foreign Key: Bank_Loan_id

Table 4.1.Loan Repayment

| Attribute | Data Type | Size | Description |
|-----------------------|-----------|------|------------------------------------|
| Customer loan number | Integer | | Loan number for the Customer |
| Name | Varchar | 20 | Name of the Customer |
| Loan Id | Integer | | Loan Id of the Customer |
| Loan type | Varchar | 15 | Loan type of the Customer |
| Due amount | Float | | Due amount need to pay by Customer |
| Installment number | Integer | | Installment number |
| Fine amount | Float | | Fine for late pay by customer |
| Mode of pay | Varchar | 10 | Mode of amount paid |
| Date | Date | | Due date for the Customer |
| Number of Installment | Integer | | Number of Installment |
| Bank branch | Varchar | | Bank branch of the Customer |

4.2 Process Model

4.2.1 System Flow Diagram:

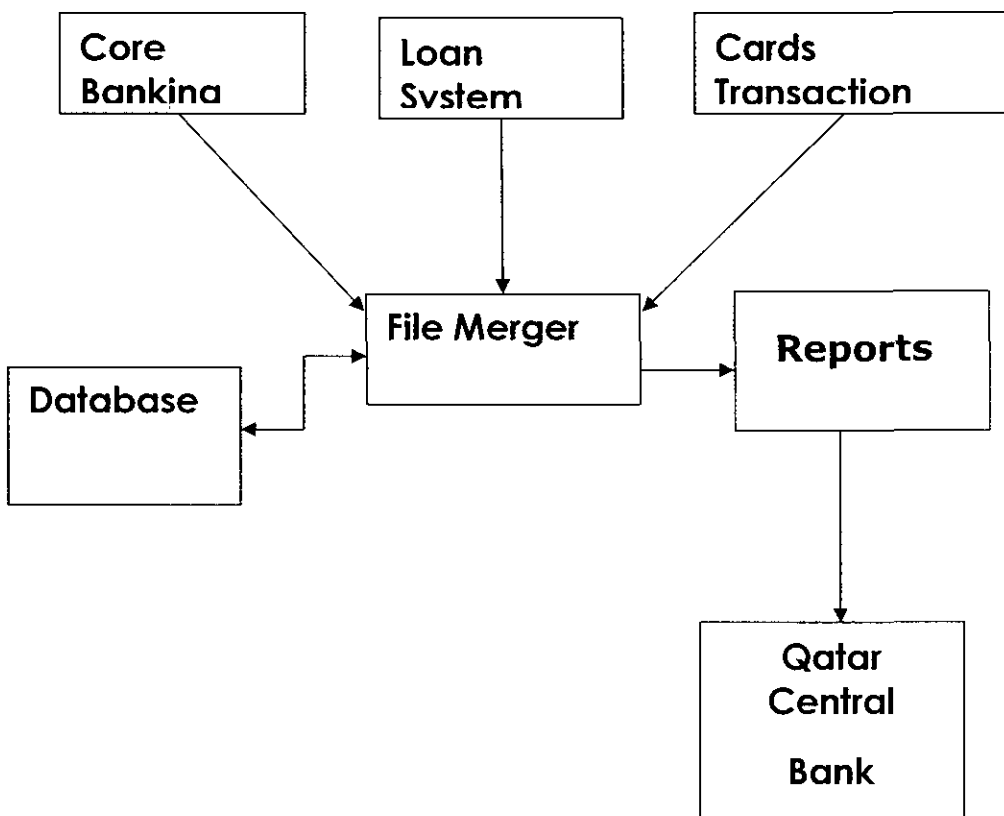


Figure 4.2.1- System Flow – Electronic File Generation

4.2.2 Class Diagram:

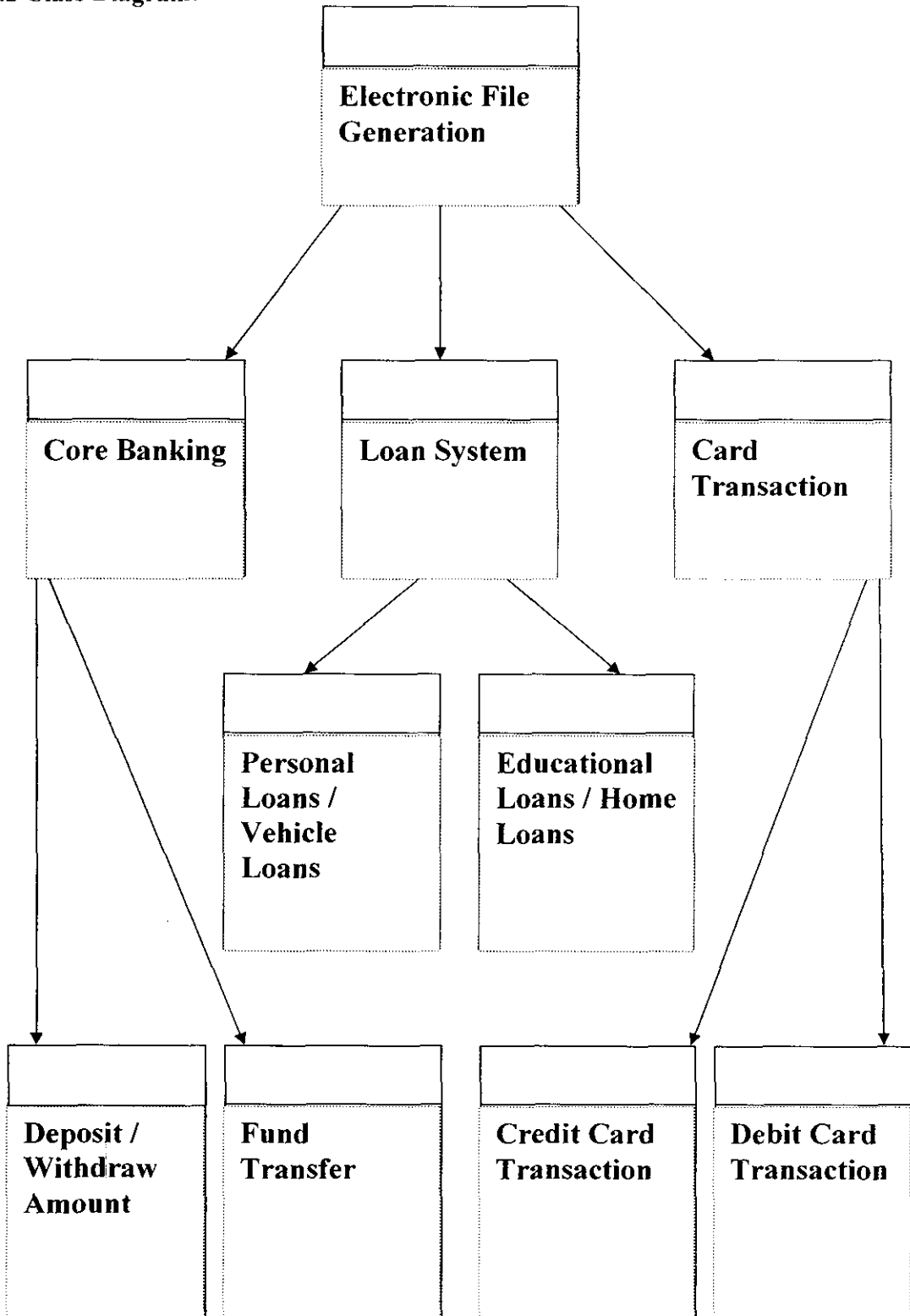
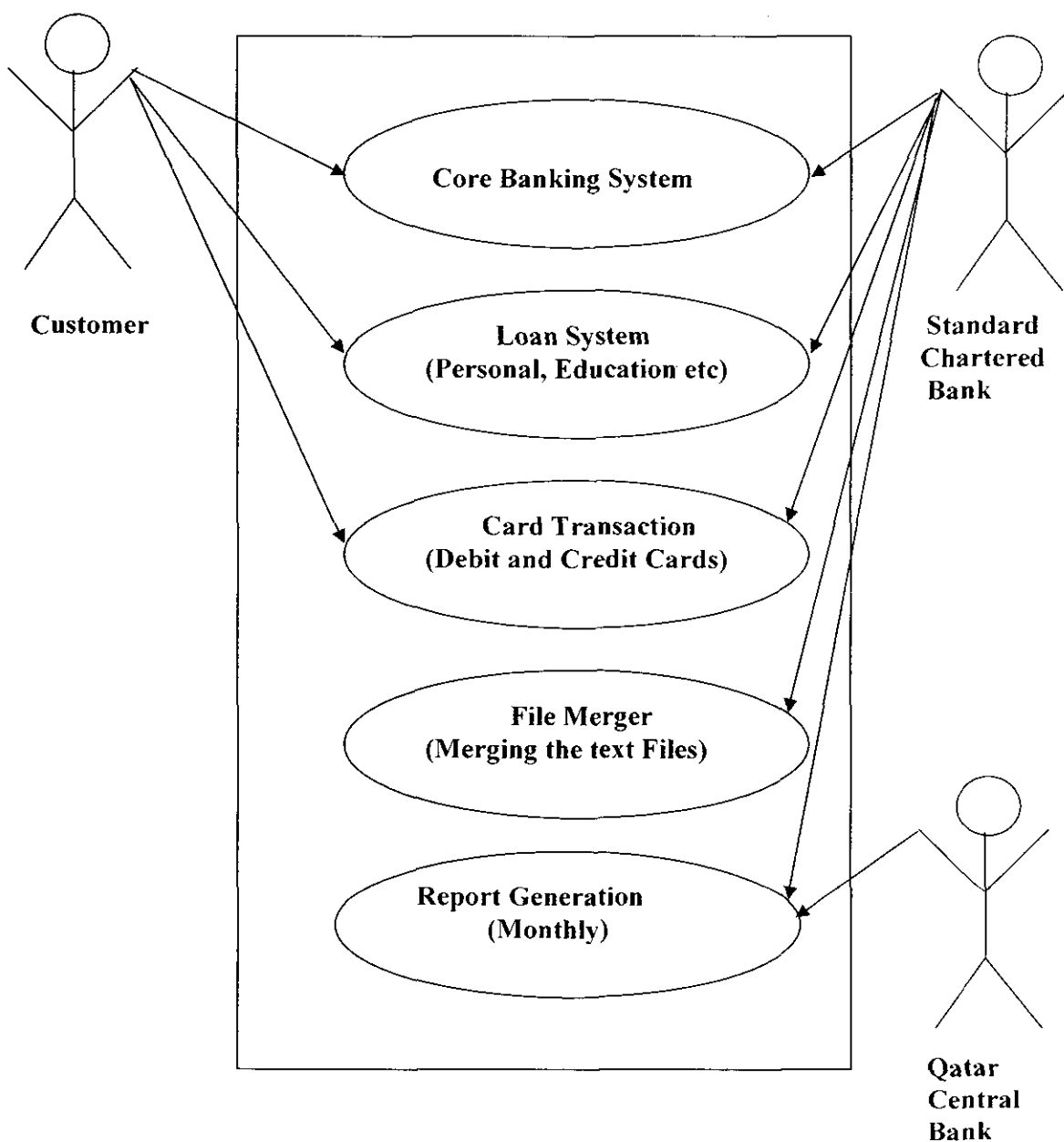


Figure 4.2..2- Class Diagram Electronic File Generation**4.2.3 Use Case Diagram:**

This section lists the use cases or scenarios from the use case model which depict significant, central functionality of the final system.

**Figure 4.2.3 Use Case – Electronic File Generation**

Use-Case Description

Core Banking

Primary Actors: Customer, Bank

Precondition : Customer must have opened a bank account

Main Flow 1:

Step 1: opening a bank account

Step 2: modifies the required details

Step 3: submit to bank manager

Alternate Flow:

Step 1: transaction through account no

Step 2: update the balance details

Step 3: clicks the ok button

Main Flow 2:

Step 1: selects the delete profile option

Step 2: confirms the operation

Alternate Flow:

Step 1: selects the delete profile option

Step 2: cancels the operation

Loan System

Primary Actors: Customer, Bank

Precondition : Customer must have opened a Loan account

Main Flow 1:

Step 1: opening a loan account

Step 2: modifies the required details

Step 3: submit to bank manger

Alternate Flow:

Step 1: Paying due amount every month

Step 2: updated on the loan status details

Step 3: clicks the ok button

Main Flow 2:

Step 1: selects the delete profile option

Step 2: confirms the operation

Alternate Flow:

Step 1: selects the delete profile option

Step 2: cancels the operation

Card Transaction

Primary Actors: Customer

Precondition : Customer must have opened a bank account

Main Flow 1:

Step 1: request the desired card option

Step 2: update customer details

Step 3: submit the changes

Alternate Flow:

Step 1: selects type of card transaction

Step 2: modifies the required details

Step 3: clicks the ok button

Main Flow 2:

Step 1: selects the delete profile option

Step 2: confirms the operation

Alternate Flow:

Step 1: selects the delete profile option

Step 2: cancels the operation

File Merger

Primary Actors: Bank

Precondition : Bank must have all monthly transaction of Customer

Main Flow 1:

Step 1: selects the text file profile option

Step 2: modifies the required details

Step 3: saves the changes

Alternate Flow:

Step 1: selects text file option that contain .ini file

Step 2: group the text file to merge

Step 3: clicks the ok button

Main Flow 2:

Step 1: selects the lock profile option

Step 2: confirms lock operation

Alternate Flow:

Step 1: selects the delete profile option

Step 2: cancels the operation

4.2.4 Data Flow Diagram

4.2.4.1 Data Flow Diagram Level - 1:

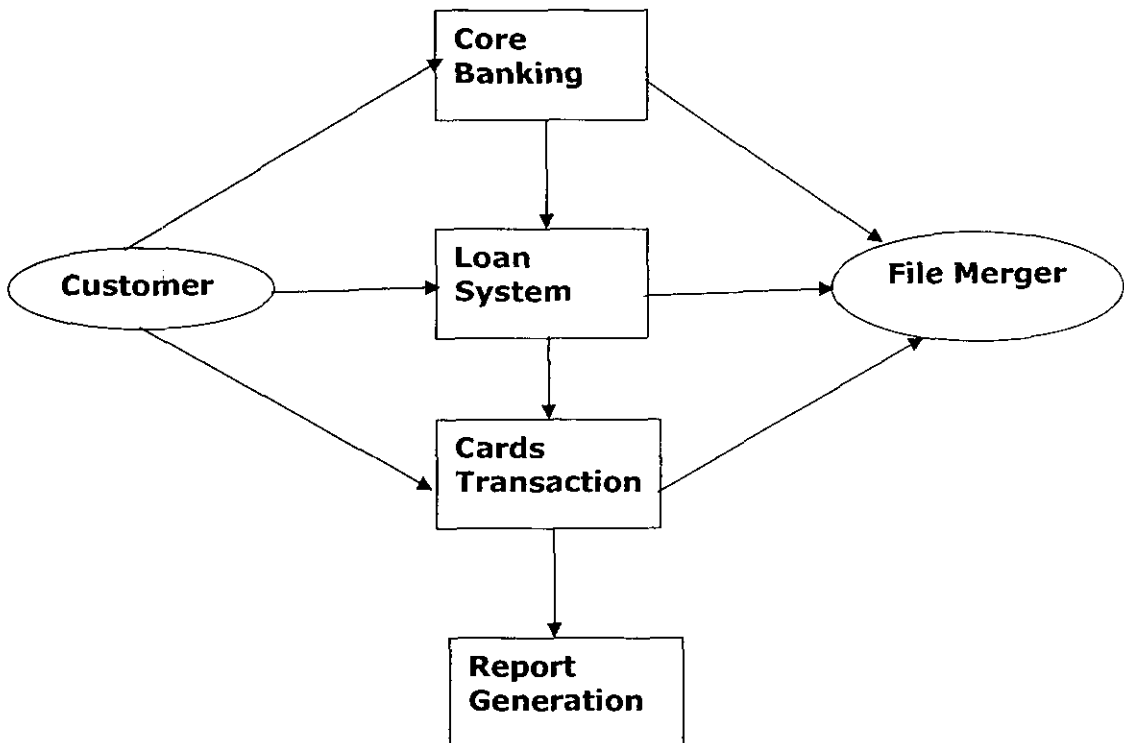


Figure 4.2.4.1 Data Flow Diagram Level – 1 - Electronic File Generation

4.2.4.2 Data Flow Diagram Level - 2:

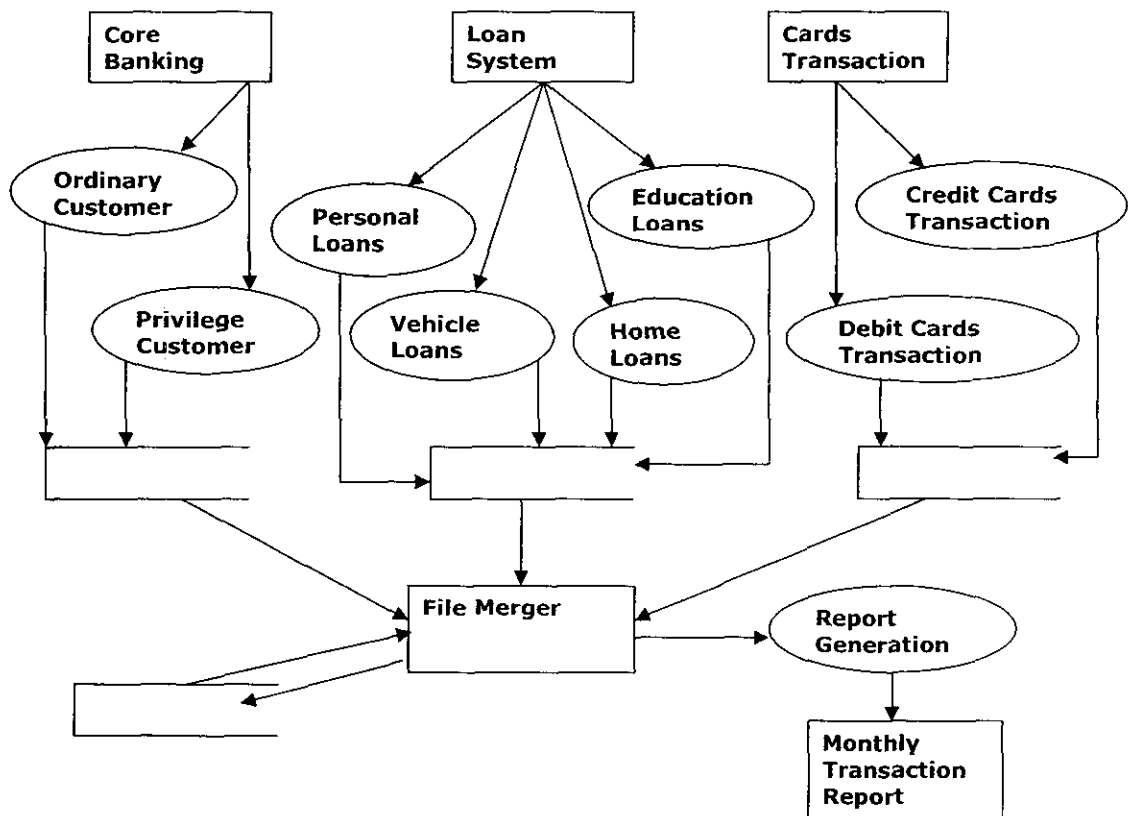


Figure 4.2.4.1 Data Flow Diagram Level – 2 - Electronic File Generation

CHAPTER-5

MODULES DESCRIPTION

The Electronic File Generation project consist of following modules

Login

Customer Details

Customer Transaction

Card Details

Loan Details

Loan Customer Details

Customer Loan Repayments

Customer Transaction Status

Loan Customer Status

File Merger

Login form:

This form is designed so that it allows user to access the project through verifying username, password and category.

Account Customer:

In this modules the A to Z details of account customer will be collected and maintained for future uses. This form is completely based upon the account number that is assigned to the customer in the customer form itself .the account customer form basically include the following information, they are as follows

Personal details

Address for Communication

Income Assessment

Bank Details

Personal details:

Here details about the full name, father name, sex, date of birth, marital status, profession, education, qualification.

Address for communication

Here the customer residential and office address is maintained along with their phone number and email id.

Income Assessment:

Here all the basic information about the customer should be maintained. The details include the income from salary from business, from property, annual income, Liabilities like loan and details about owned things are maintained.

Bank details

Here details about the bank in which the customer have their account .The details includes bank account number, account type, category, branch, balance and date.

Customer Transaction:

Here the details about customer transaction. This includes the details about the account number, customer name, account type, transaction type mode of transaction, date and balance amount are maintained.

Card details:

Here the details about bank cards i.e., credit and debit cards, this includes account number, customer name, account type ,card id, card type, card value, validate, rewable date, card charge, repayment date and bank branch are maintained.

Customer Status:

This includes the customer transaction chart for the use of bank. This includes the details about the account number, customer name ,account type, transaction type ,mode of transaction, balance amount and also include card transaction detail of card id ,card type, amount withdraw or amount spent, repayment period, and date in a tabular format.

Loan details:

Here details about bank loan. This includes the details about the loan id, loan type, invoice amount, flat rate, interest amount, number of loans allotted and number of loans sanctioned.

Loan Customer:

In this module the A to Z details of loan customer will be collected and maintained for future uses. This form is completely based upon the account number that is assigned to the customer in the customer form itself .the account customer form basically include the following information, they are as follows

Personal details

Address for Communication

Income Assessment

Bank Loan Details

Personal details:

Here details about the full name, father name, sex, date of birth, marital status, profession, education, qualification.

Address for communication

Here the customer residential and office address is maintained along with their phone number and email id

Income Assessment:

Here all the basic information about the customer should be maintained. The details include the income from salary from business, from property, annual income, Liabilities like loan and details about owned things are maintained.

Bank Loan details:

Here details about the Loan bank in which the customer have their account .The details includes bank loan number, loan type ,category, branch, invoice amount, flat rate, number installments, repayment period and date...are maintained.

Loan Repayment:

This includes the repayment schedule chart for the use of bank. This includes the details about the loan number, customer name ,invoice amount ,finance amount, flat rate, installment amount, repayment period, number of installment and also the details about the date of due and details about the number of installments paid and number of installment to be paid in a tabular format.

File Merger:

Here the details about text file need to be merged. This include mutable listing all directories, text files can be select on the panel and merged to produce report.

Loan Customer Status:

This Module describes customer loan status. This include details about installment number, installment date, due amount, paid date ,mode of pay and status of pay.

Customer Transaction Status:

This Module describes customer transaction status. This include details about transaction n number, transaction type ,date of transaction, transaction amount, mode of transaction and balance amount.

SYSTEM TESTING

CHAPTER-6

SYSTEM TESTING

Testing Methodology:

It is the stage of implementation, which ensures that system works accurately and effectively before the live operation Commences. It is a confirmation that all are correct and opportunity to show the users that the system must be tested with test data and show that the system will operate successfully and produce expected results under expected conditions.

Before implementation, the proposed system must be tested with raw data to ensure that the modules of the system work correctly and satisfactorily. The system must be tested with valid data to achieve its objective.

The purpose of system testing is to identify and correct errors in the candidate system. As important as this phase is, it is one that is frequently compromised. Typically, the project the schedule or the user is eager to go directly to conversion. Actually, testing is done to achieve the system goal. Testing is vital to the parts of the system are correct; the goal will be successfully achieved. Inadequate testing or non-testing leads to errors that may not appear until months later. This creates two problems:

- The time lag between the cause and appearance of the problem.
- The effect of system errors on files and records within the system. A small system error can conceivably exploded into much larger problem. Effectively early in the process translates directly into long term cost savings from a reduced number of errors.

Unit Testing:

A unit test is a procedure used to validate that a particular module of source code is working properly. Thus, we can say that this is a module-level testing where each of the modules are tested individually. This type of testing is mostly done by the developers and not by end-users. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. Unit testing provides a strict, written contract that the piece of code must satisfy.

Here in this project, we have tested each pair of modules before it could be integrated and packaged. Each module pair has been tested for their functionality. They were originally developed to run as CLI (Command Line Interface) and were forced to undergo Black Box testing, White Box Testing and Domain testing. At later stage in the system development, a GUI was developed to view the monitored status.

Integration Testing:

A novice in the software world might ask a seemingly legitimate question once all modules have been unit tested. If they all work individually, why do you doubt that they'll work when we put them together? The problem, of course, is "putting them together" that is interfacing. Data can be lost across when sub-functions are combined and it may not produce the desired major function individually may be magnified to unacceptable levels and global data structures can present problems. Integration Testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit-tested modules and build a program structure that has been dictated by design.

There is often a tendency to attempt Non-increment Integration; that is to construct the program using a “big Band” approach. All modules are combined in advance. The entire program is tested as a whole and chooses usually results. Sets of errors are encountered. Corrections are difficult because isolating the causes is complicated by the vast expanse of the entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

In order to avoid the chaos in above approach, we go for Incremental Testing where the modules are integrated on incremental basis rather Big Bang basis. Incremental Testing can be classified into two groups as under:

- > Top –Down Approach where modules are integrated on top-down basis.
- > Bottom –Up Approach where modules are integrated on bottom-up basis.

System Testing:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the “integrated” software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limiting type of testing; it seeks to detect defects both within the “inter-assemblages” and also within the system as a whole.

CONCLUSION

CHAPTER-7

CONCLUSION

The main idea behind this project is design a system for bank, which provides monthly report by means of electronic. As a bank consists of huge database with different work group of a bank are interconnect and different transaction and operation are carried out. Especially in the commercial and wholesale banking data processing will be huge and continues every day. For every customer transaction the system need provide a monthly report.

The Electronic File Generation is a system which allows you to record, control and analyze customer transaction in the bank. The system need to maintain the data base to record the customer details and their transaction details. The bank manager analysis customer transaction to produce monthly report and based on the criteria he has the right to select text file need for report or reject the file. And in addition the managers use this application for assessing the customer transaction. The customer will use this application for withdraw money and e-shopping by specifying the account types. For every customer transaction the system need provide a monthly report.

The project “**Electronic File Generation**” has been developed as per the requirement specification .It has been developed in Java & SQL Server, the complete system is thoroughly tested with the availability data and throughput reports which are prepared manually.

CHAPTER-8 APPENDICES

8.1 SAMPLE SCREENS



ELECTRONIC FILE GENERATION

FIG A.1.1 HOME PAGE

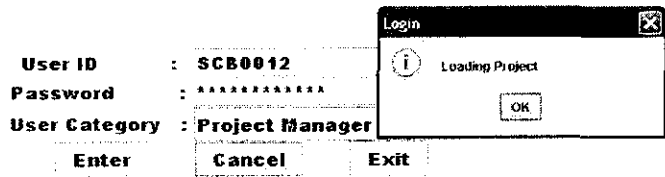


FIG A.1.2 LOGIN FORM

Customer Details

ACCOUNT CUSTOMER DETAILS

Standard Chartered

Electronic File Generator

- Customer
 - Details
 - Transaction
- Loan
 - Loan Details
 - Customer Details
 - Repayment Details
- Cards
 - Details
- Customer Status
 - Transaction
 - Loan
- Report
 - File Merge

Personal Details Address Bank Account Details

Account Number : 187690

Customer Name : Suresh

Father Name : Jaya Kumar

Gender Type : Male Female

Date of birth : 12/04/1970

Marital Status : Married Single

Qualification : M.Sc.,

Profession : Company Manager

Income by Salary : 12,500

Income Business : 10,000

Account Customer
Record Add Successfully
OK

< << >> >| CANCEL
ADD DELETE EDIT SAVE EXIT



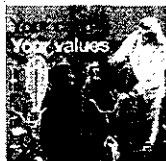




FIG A.1.3 ACCOUNT CUSTOMER DETAILS – PERSONAL DETAILS

Customer Details

ACCOUNT CUSTOMER DETAILS

Standard Chartered

Electronic File Generation

Personal Details | Address | Bank Account Details

- Customer
 - Details
 - Transaction
- Loan
 - Loan Details
 - Customer Details
 - Repayment Details
- Card
 - Details
- Customer Status
 - Transaction
 - Loan
- Report
 - File Merge

Address : 141 / 12 A, Meadows road, opposite to Sastri Bhavan, Nungambakam

City Name : Chennai

Pincode : 600097

State : Tamil Nadu

Contact Number : 044 3462901

Fax Number : 044 2545866

Email Id : suresb@gmail.com

Your wealth

Your way of life

Your values

FIG A.1.4 ACCOUNT CUSTOMER DETAILS – ADDRESS DETAILS

Customer Details

ACCOUNT CUSTOMER DETAILS

Standard Chartered

Electronic File Generation

Personal Details | Address | Bank Account Details

- Customer
 - Details
 - Transaction
- Loan
 - Loan Details
 - Customer Details
 - Repayment Details
- Cards
 - Details
- Customer Status
 - Transaction
 - Loan
- Report
 - File Merger

Account Type : Current Saving

Account Category : Individual privilege

Balance Amount : 230500.00

Bank Branch : Nungambakkam

Date : 02/01/2008





FIG A.1.5 ACCOUNT CUSTOMER DETAILS – BANK DETAILS

Customer Transaction Details

Customer Transaction Form

Standard Chartered 

Electronic File Generation

- Customer
 - Details
 - Transaction
- Loan
 - Loan Details
 - Customer Details
 - Repayment Details
- Cards
 - Details
- Customer Status
 - Transaction
 - Loan
- Report
 - File Merge

Account Number : 192350

Customer Name : Suresh

Account Type : Current Saving

Account Category : Individual Prevelige

Transaction Type : Withdraw Deposit

Transaction Amount : 13300

Mode Of Transaction : Cash




Balance Amount : 11200

Bank Branch : Nugumbakkam

Date : 08.05.2008

<<
>>
CANCEL

ADD
DELETE
EDIT
SAVE
EXIT

Customer Transaction

Transaction to continue ?

OK

FIG A.1.6 CUSTOMER TRANSACTION DETAILS

Customer Transaction Status

CUSTOMER TRANSACTION STATUS

Standard Chartered

- Electronic File Generation
 - Customer
 - Details
 - Transaction
 - Loan
 - Loan Details
 - Customer Details
 - Repayment Details
 - Cards
 - Details
 - Customer Status
 - Transaction
 - Loan
 - Report
 - File Merger


| Trans No | Date | Tran Type | Tran Amt | Mode Tran | Bal Amt | Branch |
|----------|----------|-----------|----------|-----------|---------|----------|
| 1902864 | 09/04/08 | Deposit | 20000 | Cash | 27600 | Egmore |
| 2032538 | 10/04/08 | Withdraw | 5630 | Cheque | 17960 | T. Nagar |
| 2576980 | 11/04/08 | Deposit | 15000 | Draft | 27960 | Egmore |
| 2576890 | 12/04/08 | Deposit | 5000 | Cash | 36960 | Chetpet |

Bank Loan Details


Do you Want Go Home

OK

HOME



Cashless
ATM



24 hour
Service
Assurance
SMS service to
99300 33333

FIG A.1.7 CUSTOMER TRANSACTION STATUS

Bank Loan Details

Bank Loan Details

Standard Chartered

- Electronic File Generation
 - Customer
 - Details
 - Transaction
 - Loan
 - Loan Details
 - Customer Details
 - Repayment Details
 - Cards
 - Details
 - Customer Status
 - Transaction
 - Loan
 - Report
 - File Manger

Bank Loan Id : 10243
Loan Type : Education
Finance Amount : 300000
Flat Rate : 10
Interest Amount : 30000
Loans Alloted : 30
Loan Sanctioed : 18
Bank Branch : T Nagar
Date : 09.04.2008

Bank Loan Details

Record Updated Successfully

OK

You will get
 Accounts to get them
 we'll give you
 the card to get
 you there

We'll give you
 the card to get
 you there

We'll give you
 the card to get
 you there

FIG A.1.8 BANK LOAN DETAILS

Loan Customer Details : Standard Chartered

Loan Customer Details

Electronic File Generation | **Personal Details** | Address | Income Assessment | Bank loan Details

- Customer
 - Details
 - Transaction
- Loan
 - Loan Details
 - Customer Details
 - Repayment Details
- Cards
 - Details
- Customer Status
 - Transaction
 - Loan
- Report
 - File Merger

Account Customer

Record Add Successfully

OK

Customer Loan Number : 2489089

Customer Name : Arjun

Father Name : Anu

Gender Type : Male Female

Date of Birth : 08-11-1978

Mantal Status : Single Married

Qualification : M.B.A

Profession : Business









FIG A.1.9 LOAN CUSTOMER DETAILS – PERSONAL DETAILS

Loan Customer Details : Standard Chartered 

Loan Customer Details

Electronic File Generation | **Personal Details** | **Address** | Income Assessment | Bank loan Details

- Customer
 - Details
 - Transaction
- Loan
 - Loan Details
 - Customer Details
 - Repayment Details
- Cards
 - Details
- Customer Status
 - Transaction
- Report
 - File Merger




| | | | |
|-----------------------|---|---|--|
| Address | : | 12 B/45 H, College Road, Opposite to City Center, T - Nagar |    |
| City Name | : | Chennai | |
| Pincode | : | 600032 | |
| State | : | Tamil Nadu | |
| Contact Number | : | 044 2354662 | |
| Fax Number | : | 044 2675805 | |
| Email Id | : | arjun@gmail.com | |

FIG A.1.10 LOAN CUSTOMER DETAILS – ADDRESS

Loan Customer Details

Standard Chartered

Electronic File Generation

Personal Details | Address | **Income Assessment** | Bank loan Details

- Customer
 - Details
 - Transaction
- Loan
 - Loan Details
 - Customer Details
 - Repayment Details
- Cards
 - Details
- Customer Status
 - Transaction
 - Loan
- Report
 - File Merge

| | | |
|---------------------------|---|---------------------------------|
| Income | : | 17000 |
| Income Salary | : | 20000 |
| Income by Business | : | 10000 |
| Income by Property | : | 15000 |
| Loan | : | Car Loan from SBI |
| Property Own | : | House, Car, Two wheeler. |

Your outlook

YOUR PROGRESS

YOUR PROGRESS




FIG A.1.11 LOAN ACCOUNT CUSTOMER DETAILS – INCOME ASSESSMENT

Loan Customer Details : Standard Chartered

Loan Customer Details

Personal Details | Address | Income Assessment | **Bank loan Details**

- Electronic File Generation
- Customer
 - Details
 - Transaction
- Loan
 - Loan Details
 - Customer Details
 - Repayment Details
- Cards
 - Details
- Customer Status
 - Transaction
 - Loan
- Report
 - File Merge

| | | | |
|------------------------------|---|--|--|
| Loan Id | : | 4572436 | |
| Loan Type | : | Education | |
| Invoice Amount | : | 185000 | |
| Finance Amount | : | 130000 | |
| Flat Rate | : | 7 | |
| Interest Amount | : | 15500 | |
| Total Amount | : | 165000 | |
| Due Amount | : | 1580 | |
| Number of installment | : | 18 | |
| Mode of Pay | : | <input checked="" type="radio"/> Cash <input type="radio"/> Draft <input type="radio"/> Cheque | |
| First Instalment Date | : | 02/05/2008 | |
| Bank Branch | : | Egmore | |
| Date | : | 02/04/2008 | |








FIG A.1.12 LOAN CUSTOMER DETAILS – BANK DETAILS

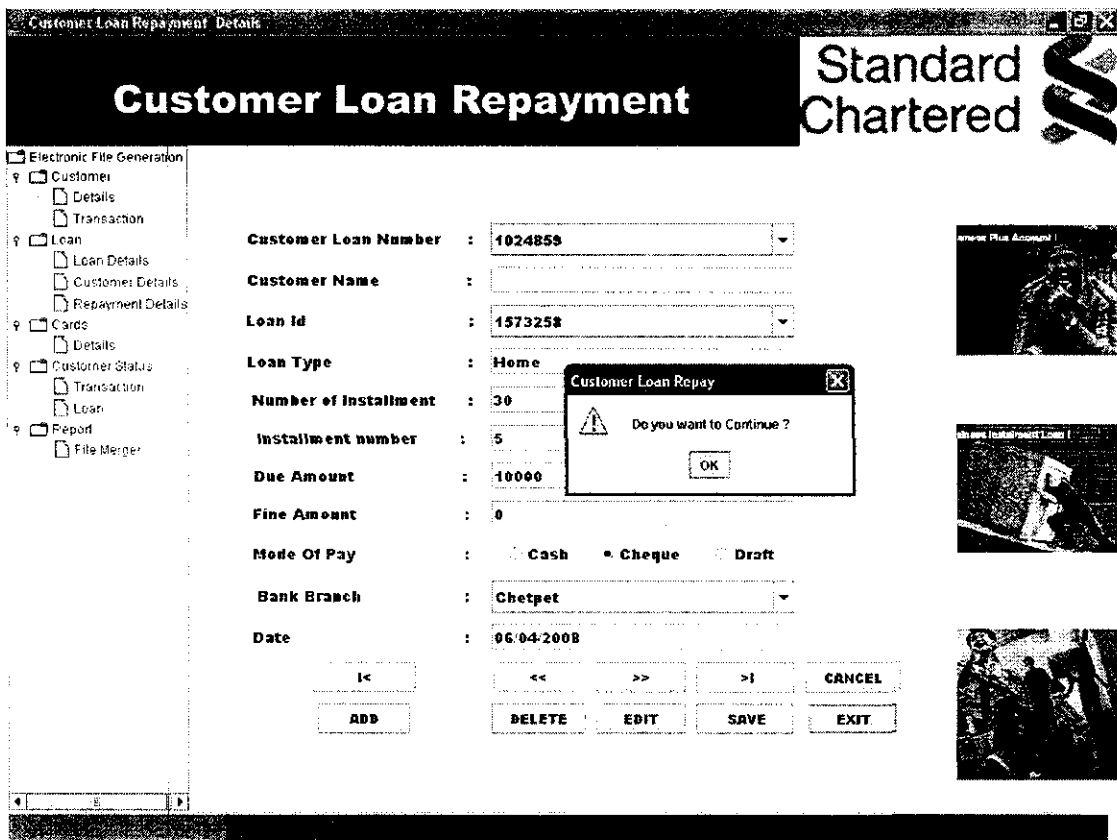


FIG A.1.13 CUSTOMER LOAN REPAYMENT DETAILS

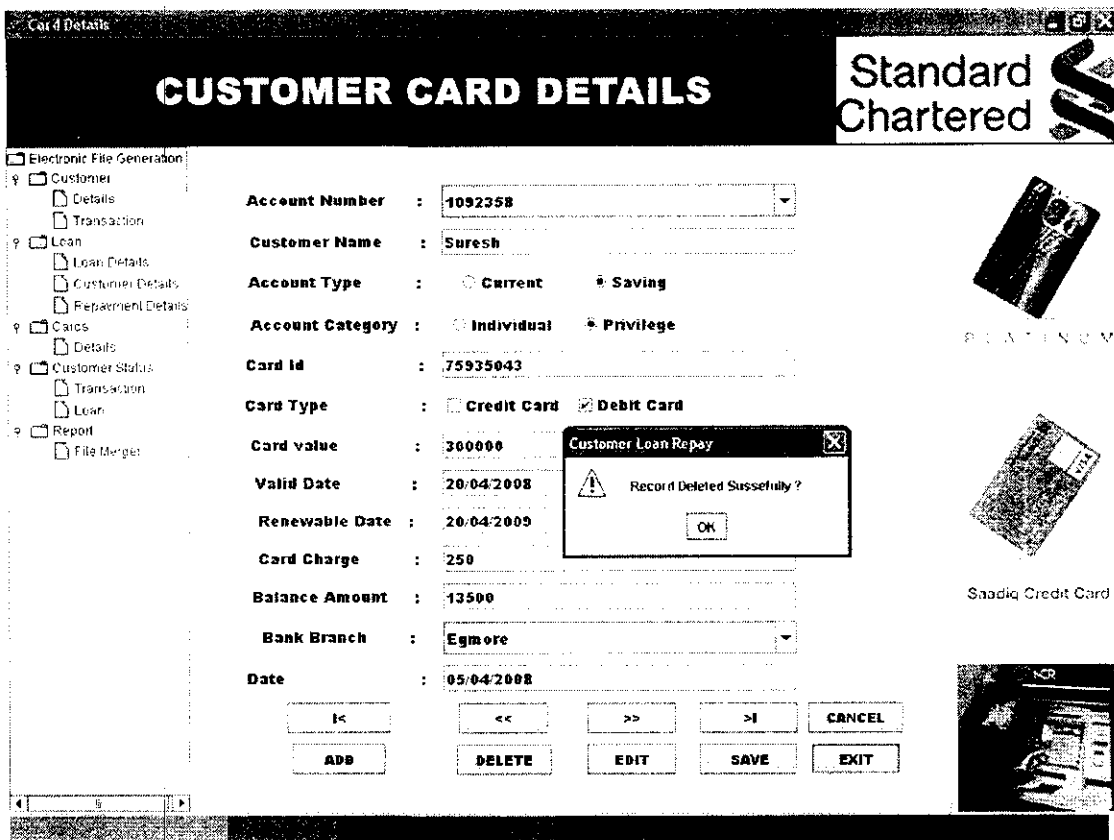



FIG A.1.14 CUSTOMER CARD DETAILS

Customer Loan Status




CUSTOMER LOAN STATUS


- Electronic File Generation
- Customer
 - Details
 - Transaction
- Loan
 - Loan Details
 - Customer Details
 - Repayment Details
- Cards
 - Details
- Customer status
 - Transaction
 - Loan
- Report
 - File Manager

| Ins.No | Ins.Amt | Due Date | Paid date | Mode of Pay | Status |
|--------|---------|------------|------------|-------------|----------|
| 01 | 2500 | 02/03/2008 | 05/03/2008 | Cash | Paid |
| 02 | 2500 | 02/04/2008 | 10/04/2008 | Cash | Paid |
| 03 | 2500 | 02/05/2008 | 07/05/2008 | Cheque | Paid |
| 04 | 2500 | 02/06/2008 | | | Not Paid |
| 05 | 2500 | 02/07/2008 | | | Not Paid |
| 06 | 2500 | 02/08/2008 | | | Not Paid |
| 07 | 2500 | 02/09/2008 | | | Not Paid |
| 08 | 2500 | 02/10/2008 | | | Not Paid |
| 09 | 2500 | 02/11/2008 | | | Not Paid |
| 10 | 2500 | 02/12/2008 | | | Not Paid |
| 11 | 2500 | 02/01/2009 | | | Not Paid |
| 12 | 2500 | 02/02/2009 | | | Not Paid |
| 13 | 2500 | 02/03/2009 | | | Not Paid |
| 14 | 2500 | 02/04/2009 | | | Not Paid |
| 15 | 2500 | 02/05/2009 | | | Not Paid |
| 16 | 2500 | 02/06/2009 | | | Not Paid |
| 17 | 2500 | 02/07/2009 | | | Not Paid |
| 18 | 2500 | 02/08/2009 | | | Not Paid |
| 19 | 2500 | 02/09/2009 | | | Not Paid |
| 20 | 2500 | 02/10/2009 | | | Not Paid |
| 21 | 2500 | 02/11/2009 | | | Not Paid |
| 22 | 2500 | 02/12/2009 | | | Not Paid |
| 23 | 2500 | 02/01/2010 | | | Not Paid |
| 24 | 2500 | 02/02/2010 | | | Not Paid |
| 25 | 2500 | 02/03/2010 | | | Not Paid |

Home



70,000 reasons why we're a different kind of bank



Standard Chartered
Platinum
VISA

PLATINUM CARD




FIG A.1.15 CUSTOMER LOAN STATUS

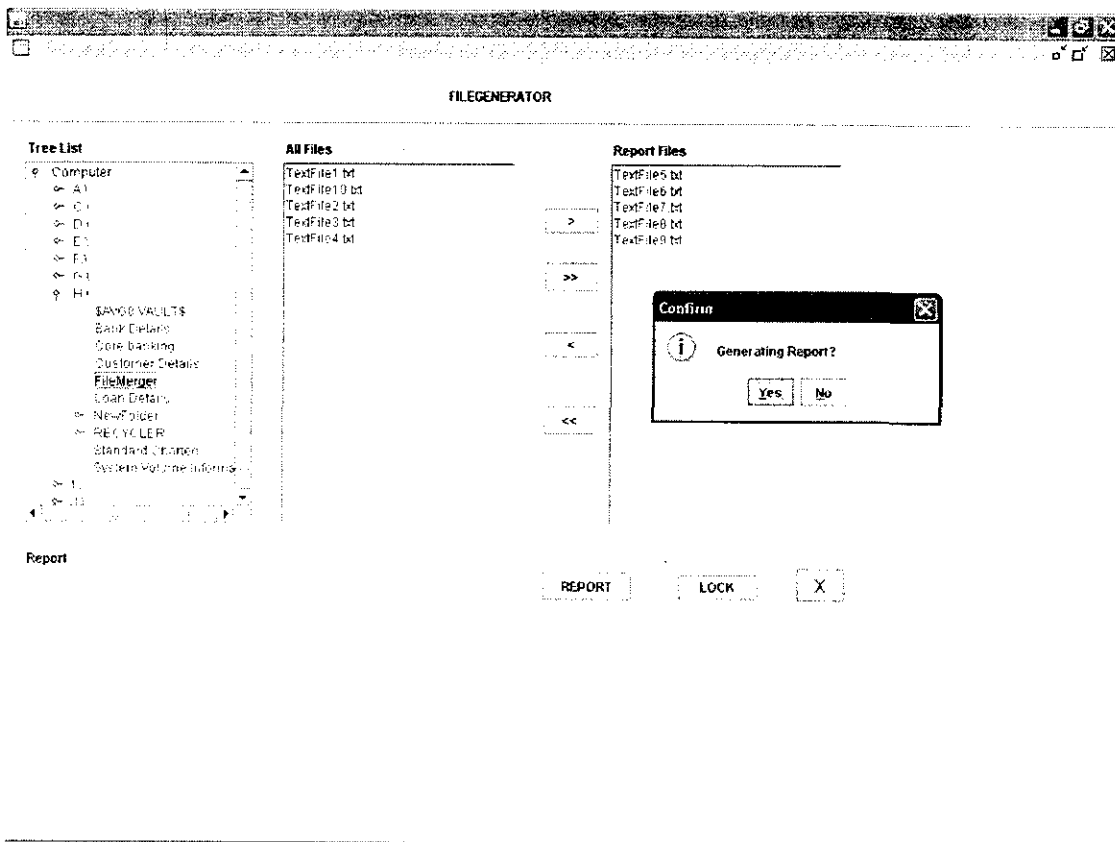


FIG A.1.16 FILE MERGER FORM

CHAPTER-9

REFERENCES

Books:

- Patrick Naughton, “**Java Hand Book** “, Osborne/mc Graw-Hill, April, 1996
- MartinBond, Dan Haywood, Debbie Law “**Professional Java Server Programming** “, WroxPress, 17 June, 2001, J2EE 1.3 Edition
- Mark Cade “**Sun Certified EnterPrise Architect for J2EE Technology**” Wrox press,11 March,2002
- Pressman R.S, “**Software Engineering**”- A practitioner’s Approach (4th Edition).
- Herbert Schildt,” **Java 2 Complete Reference**”-Tata McGraw Hill (3rd Edition).

Website:

- www.java.sun.com
- www.javalobby.org
- www.visualbuilder.com
- www.knowledgeisland.com
- www.codenotes.com
- www.coreservlets.com
- www.javaboutique.com
- www.jguru.com
- www.w3school.com