Customer relationship management for banking system

Pingyu Hou
CUSTOMER RELATIONSHIP MANAGEMENT
FOR BANKING SYSTEM

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Ping-Yu Hou

June 2004
CUSTOMER RELATIONSHIP MANAGEMENT
FOR BANKING SYSTEM

A Project
Presented to the
Faculty of
California State University,
San Bernardino

by
Ping-Yu Hou
June 2004

Approved by:

Dr. Kerstin Voigt, Chair, Computer Science
Dr. Ernesto Gomez
Dr. David Turner

Date: 05/13/04
ABSTRACT

Customer Relationship Management for Banking System
(CRM BANKING) is a web application used by banks to study
customer information. It interacts with banking data
warehouse to retrieve and analyze customer’s personal
information, account profile, and transaction history. CRM
BANKING applies statistics and data mining techniques to
understand the correlation between banking data and its
customers through pattern recognition. This information is
crucial for customer relationship management and
improvement.
ACKNOWLEDGMENTS

I thank the faculty of Computer Science department for giving me an opportunity to pursue my Master Science in Computer Science at California State University, San Bernardino. I gratefully acknowledge the invaluable guidance of my advisor, Dr. Kerstin Voigt, who directed me through this entire effort. I would also like to express my sincere appreciation to the other committee members, Dr. Ernesto Gomez and Dr. David Turner for their valuable suggestions.

Last but not least, I would like to thank my family for their patience and encouragement during the course of my Master Science program.
TABLE OF CONTENTS

ABSTRACT ......................................................... iii
ACKNOWLEDGMENTS ........................................ iv
LIST OF TABLES .............................................. viii
LIST OF FIGURES ............................................ ix

CHAPTER ONE: INTRODUCTION
  1.1 Purpose of this Project ................................. 1
  1.2 Project Products .......................................... 2

CHAPTER TWO: CUSTOMER RELATIONSHIP MANAGEMENT FOR
  BANKING SYSTEM ARCHITECTURE ............................ 3
  2.1 Software Interfaces ....................................... 5

CHAPTER THREE: DATABASE DESIGN
  3.1 Data Analysis ........................................... 6
  3.2 Database Schema Conceptual Model – ER
          Diagram ................................................. 6
  3.3 Database Schema Logical Model –
          Relational Schema .................................. 9
  3.4 Data Type and Details ................................. 11

CHAPTER FOUR: PROJECT IMPLEMENTATION .................. 17
  4.1 Customer Relationship Management For
          Banking System Graphical User Interface
          Design ................................................... 18
    4.1.1 Customer Relationship Management
            For Banking System Login ....................... 18
  4.1.2 Inquire User Information Function ............. 19
  4.1.3 Add User Function .................................. 20
  4.1.4 Edit User Information Function ................ 21
4.1.5 Delete User Function ....................... 22
4.1.6 Customer Inquiry Function ............. 23
4.1.7 Channel Affinity Function ............... 24
4.1.8 Percentile Analysis Function .......... 26
4.1.9 Segmentation Function .................... 31
4.1.10 Customer Behavior Analysis Function ........................................... 33

CHAPTER FIVE: DATA MINING ALGORITHMS

5.1 K-Means Algorithm ............................. 38
5.2 Decision Tree Algorithm ....................... 39

CHAPTER SIX: SYSTEM VALIDATION

6.1 Unit Test ........................................ 41
6.2 Subsystem Testing ............................... 45
6.3 System Testing ....................................... 46

CHAPTER SEVEN: MAINTENANCE MANUAL

7.1 Software Installation ......................... 47
7.1.1 RedHat Installation ......................... 47
7.1.2 MySQL Installation ............................. 48
7.1.3 JAVA 2 Platform, Standard Edition (J2SE) ............................................ 49
7.1.4 Tomcat ........................................ 50
7.1.5 JAVA Database Connectivity (JDBC) ..... 51
7.2 Variables Modification ......................... 52
7.2.1 System Variables ............................... 52
7.2.2 Copying Files ................................ 52
7.3 Customer Relationship Management For Banking System Installation/Migration ....... 53
7.4 Backup .................................................. 53
   7.4.1 System Backup ................................. 53
   7.4.2 Database Backup ............................... 54

CHAPTER EIGHT: CONCLUSION AND FUTURE DIRECTIONS

8.1 Conclusion ............................................. 55
8.2 Future Directions ................................. 55

REFERENCES .................................................. 57
LIST OF TABLES

Table 1. Structure of Table User .................... 11
Table 2. Structure of Table Role .................... 11
Table 3. Structure of Table Cif .................... 12
Table 4. Structure of Table Pbacno ................ 12
Table 5. Structure of Table Ckacno ................ 13
Table 6. Structure of Table Ctaclno ............... 13
Table 7. Structure of Table Arff ................... 14
Table 8. Structure of Table Criteria ............... 15
Table 9. Structure of Table Pbtxn .................. 15
Table 10. Structure of Table Cktxn ................ 16
Table 11. Unit Test Results ......................... 41
Table 12. Subsystem Test Results ................... 46
Table 13. System Test Results ....................... 46
LIST OF FIGURES

Figure 1. CRM BANKING System Architecture ............ 3
Figure 2. Banking Data Warehouse E-R Diagram .......... 7
Figure 3. Application System E-R Diagram ............... 8
Figure 4. CRM BANKING System Database
Relational Schema ...................................... 9
Figure 5. Use Case Diagram .................................. 17
Figure 6. Login Page ........................................ 19
Figure 7. Inquire User Information Function ............. 20
Figure 8. Add User Function ................................ 21
Figure 9. Edit User Function .............................. 22
Figure 10. Delete User Function ............................ 23
Figure 11. Customer Inquiry Function ..................... 24
Figure 12. Selection Page of Channel Affinity
Function .................................................. 25
Figure 13. Applied Channels Statistic of Channel
Affinity .................................................. 26
Figure 14. Group Selection Page of Percentile
Analysis .................................................. 27
Figure 15. Criterion Selection Page of
Percentile Analysis ..................................... 28
Figure 16. Percentile Analysis Function by
Customer Groups ....................................... 29
Figure 17. Percentile Function by Account
Groups ................................................... 30
Figure 18. Percentile Function by Transaction
Groups .................................................... 31
Figure 19. Selection Page of Segmentation
Function .................................................. 32
Figure 20. A List of Customers by Segmentation
Function .................................................. 33
Figure 21. Selection Page of Customer Behavior Function ......................................... 35

Figure 22. Customer Behavior Analysis by K-Means Algorithm ................................. 36

Figure 23. Decision Tree of Customer Behavior Analysis ........................................... 37
CHAPTER ONE
INTRODUCTION

CRM BANKING is an online application that caters to the strengthening and stabilizing customer relationships in a bank. With millions of customers, varied channels and products, lots of agents and representatives, banks face a series of complex data and information challenges. The CRM BANKING application is going to discover and manage the relationships of abundant data and information of customers in a bank. It provides nine functions, Add User, Edit User, Delete User, Inquire User, Segmentation, Percentile Analysis, Customer Behavior Analysis, Channel Affinity Analysis, and Customer Inquiry Functions, for four different roles of a bank which are administrator, manager, agent and clerk. In addition, it applies some statistics and data mining techniques in the above functions.

1.1 Purpose of this Project

The purpose of this project is to design, build, and implement a Customer Relationship Management system for a bank. All the information will be stored in MySQL database and retrieved by JSP and JDBC. The main purpose of the project is on-line analysis and on-line process of banking...
data, which can let the users understand the customer contributions and manage the relationships with customers. Also, this system provides some functionality for the users to analyze customers’ dealing behaviors and measure banking channel’s affinities.

1.2 Project Products

This project would lead to the following products:

- **Implementation of CRM BANKING**: a working web site with JSP programs, Java programs and MySQL database, which would achieve the specific needs of customer relationship management in a bank.
- **Users manual**: an implementation manual will be available for the user.
- **Systems Manual**: a project report (this report) will be available with design details and specifications.
CHAPTER TWO

CUSTOMER RELATIONSHIP MANAGEMENT FOR BANKING

SYSTEM ARCHITECTURE

This project, Customer Relationship Management For Banking System (CRM BANKING), implements a web system to provide an environment for the users to discover and manage all relationships of customers in a bank. Thus, the components needed to implement CRM BANKING are a database server, a web server, graphical user interface components, and a database interface Application Programming Interface (API) to programmatically access the database. The following figure describes the interaction among the components used in CRM BANKING.

![Diagram of CRM BANKING System Architecture](image)

Figure 1. CRM BANKING System Architecture

The components used to build CRM BANKING were chosen with the following criteria: (i) the components should be shareware, i.e., available freely for non-commercial purposes, (ii) be part of a standard, i.e., the do not
depend on a specific operating system and hence are easily portable across systems with ease, (iii) database server independent, so that new and different versions of the server can be plugged in easily.

The user interface components are built by using HTML 6.0 forms, frames, and JavaScript. The applications are launched using JavaServer Pages (JSP) and Java Servlet. JSP and Java Servlet were used because they are easily portable and can be reused. Also, it is easy to process whole user input from the HTML forms. Moreover, Java provides the convenient function, Java Database Connector (JDBC), to connect database.

The database choice available to CRM BANKING is MySQL. MySQL is a real multi-user database and it is free. The availability of the JDBC driver for MySQL and the rapid performance of the access data are the most important reasons to choose it. Moreover, the same code could be used to link with another version of MySQL database by changing the proper JDBC driver, thus, making it database independent.
2.1 Software Interfaces

- Internet browser: Netscape or Internet Explorer.
- Operating system: Windows 98/Me/2000/XP or Unix/Linux.
- Database: MySQL.
- Compiler: JDK 1.4.2
- Language: HTML / JAVA / JavaScript / JSP.
- Database connector: JDBC.
- JSP Container/Web server: Jakarta Tomcat.
CHAPTER THREE
DATABASE DESIGN

3.1 Data Analysis

There are two parts of data for designing and implementing the schema of the database. The first part of data is live data from a local commercial bank. In view of privacy, names, addresses, e-mail addresses, telephone numbers and other sensitive information are removed and also identification numbers and account numbers are changed by some random parameters. The second part of data needed by the system is user information, user privilege, category of banking data, and prepared data for further analysis.

3.2 Database Schema Conceptual Model - ER Diagram

In designing the schema for the CRM BANKING database, two distinct parts have been identified. The first part is a banking data warehouse which includes customer personal information, banking account information, customer transaction history. All the entities and attributes are described in Figure 2.
Figure 2. Banking Data Warehouse E-R Diagram
The second part is the system part which includes user information, user privilege, categories of banking data, and the information of segmentation. All the entities and attributes are detailed in Figure 3.

![Application System E-R Diagram](image)

Figure 3. Application System E-R Diagram
3.3 Database Schema Logical Model - Relational Schema

The conceptual model ER diagrams map into the following relational table design. In the following tables, Underlined fields indicate the primary key.

<table>
<thead>
<tr>
<th>User</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>user_id</td>
<td>role_id</td>
<td>password</td>
<td>employee_id</td>
<td>fname</td>
<td>lname</td>
</tr>
<tr>
<td>email_addr</td>
<td>pswd_count</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Role</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>role_id</td>
<td>role_name</td>
<td>privilege</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cif</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>cif_id</td>
<td>teller</td>
<td>create_date</td>
<td>birthdate</td>
<td>mod_mark</td>
<td>mod_date</td>
</tr>
<tr>
<td>atm_code</td>
<td>i-mark</td>
<td>n_txn</td>
<td>a_txn</td>
<td>i_txn</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pbacno</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>pbac_acno</td>
<td>char_code</td>
<td>int_code</td>
<td>id</td>
<td>open_date</td>
<td>last_date</td>
</tr>
<tr>
<td>bal</td>
<td>i-mark</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 4. Database Relational Schema (continued)
3.4 Data Type and Details

The logical model translates into the following detailed design in MySQL database. The following tables describe data type, length, primary key or not, null or non-null key and default value.

Table 1. Structure of Table User

<table>
<thead>
<tr>
<th>field</th>
<th>type</th>
<th>null</th>
<th>key</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_id</td>
<td>varchar(10)</td>
<td></td>
<td>PRI</td>
<td></td>
</tr>
<tr>
<td>role_id</td>
<td>int(1)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>password</td>
<td>varchar(32)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>employee_id</td>
<td>varchar(10)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>user_fname</td>
<td>varchar(15)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>user_lname</td>
<td>varchar(15)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>email_addr</td>
<td>varchar(40)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>pswd_count</td>
<td>int(2)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2. Structure of Table Role

<table>
<thead>
<tr>
<th>field</th>
<th>type</th>
<th>null</th>
<th>key</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>role_id</td>
<td>int(1)</td>
<td></td>
<td>PRI</td>
<td></td>
</tr>
<tr>
<td>role_name</td>
<td>varchar(10)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>privilege</td>
<td>varchar(32)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
</tbody>
</table>
Table 3. Structure of Table Cif

<table>
<thead>
<tr>
<th>field</th>
<th>type</th>
<th>null</th>
<th>key</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>cif_id</td>
<td>varchar(10)</td>
<td></td>
<td>PRI</td>
<td></td>
</tr>
<tr>
<td>cif_teller</td>
<td>varchar(5)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>cif_create_date</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>cif_birthdate</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>cif_mod_mark</td>
<td>varchar(1)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>cif_mod_date</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>cif_atm_code</td>
<td>varchar(1)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>cif_i_mark</td>
<td>varchar(1)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>cif_n_txn</td>
<td>varchar(1)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>cif_a_txn</td>
<td>varchar(1)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>cif_i_txn</td>
<td>varchar(1)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
</tbody>
</table>

Table 4. Structure of Table Pbacno

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>null</th>
<th>key</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>pbac_acno</td>
<td>varchar(11)</td>
<td>PRI</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>pbac_char_code</td>
<td>varchar(2)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>pbac_int_code</td>
<td>varchar(1)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>Pbac_id</td>
<td>varchar(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pbac_open_date</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>pbac_last_date</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>pbac_pb_bal</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>pbac_bal</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>pbac_i_mark</td>
<td>varchar(1)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
</tbody>
</table>
### Table 5. Structure of Table Ckacno

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>null</th>
<th>key</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ckac_acno</td>
<td>varchar(11)</td>
<td></td>
<td>PRI</td>
<td>NULL</td>
</tr>
<tr>
<td>ckac_char_code</td>
<td>varchar(2)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>ckac_int_code</td>
<td>varchar(1)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>ckac_id</td>
<td>varchar(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ckac_open_date</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ckac_last_date</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ckac_bal</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ckac_withld_bal</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ckac_i_mark</td>
<td>varchar(1)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
</tbody>
</table>

### Table 6. Structure of Table Ctacno

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>null</th>
<th>key</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ctac_acno</td>
<td>varchar(11)</td>
<td></td>
<td>PRI</td>
<td>NULL</td>
</tr>
<tr>
<td>ctac_char_code</td>
<td>varchar(2)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>ctac_int_code</td>
<td>varchar(1)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>ctac_id</td>
<td>varchar(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ctac_open_date</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ctac_due_date</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ctac_bill_bal</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ctac_bal</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ctac_i_mark</td>
<td>varchar(1)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
</tr>
</tbody>
</table>
Table 7. Structure of Table Arff

<table>
<thead>
<tr>
<th>field</th>
<th>type</th>
<th>null</th>
<th>key</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>arff_userid</td>
<td>varchar(1)</td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arff_date</td>
<td>varchar(8)</td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arff_time</td>
<td>varchar(6)</td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arff_count</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>arff_table1</td>
<td>varchar(10)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_attribute1</td>
<td>varchar(30)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_begin1</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_end1</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_table2</td>
<td>varchar(10)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_attribute2</td>
<td>varchar(30)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_begin2</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_end2</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_table3</td>
<td>varchar(10)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_attribute3</td>
<td>varchar(30)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_begin3</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_end3</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_table4</td>
<td>varchar(10)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_attribute4</td>
<td>varchar(30)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_begin4</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_end4</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_table5</td>
<td>varchar(10)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_attribute5</td>
<td>varchar(30)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_begin5</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_end5</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_table6</td>
<td>varchar(10)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_attribute6</td>
<td>varchar(30)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_begin6</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>arff_end6</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>
Table 8. Structure of Table Criteria

<table>
<thead>
<tr>
<th>field</th>
<th>type</th>
<th>null</th>
<th>key</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>criteria_table</td>
<td>varchar(10)</td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>criteria_attribute</td>
<td>varchar(30)</td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>criteria_item</td>
<td>varchar(1)</td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>criteria_begin</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>criteria_end</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>criteria_field</td>
<td>varchar(40)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>criteria_content</td>
<td>varchar(10)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Table 9. Structure of Table Pbtxn

<table>
<thead>
<tr>
<th>field</th>
<th>type</th>
<th>null</th>
<th>key</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>pbtxn_acno</td>
<td>varchar(11)</td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pbtxn_txn_date</td>
<td>int(8)</td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pbtxn_txn_time</td>
<td>int(6)</td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pbtxn_termid</td>
<td>varchar(4)</td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pbtxn_txn_status</td>
<td>int(1)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>pbtxn_txn_id</td>
<td>varchar(4)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>pbtxn_dc_code</td>
<td>varchar(1)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>pbtxn_txn_amt</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>pbtxn_pre_bal</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
### Table 10. Structure of Table Cktxn

<table>
<thead>
<tr>
<th>field</th>
<th>type</th>
<th>null</th>
<th>key</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>cktxn_acno</td>
<td>varchar(11)</td>
<td></td>
<td>PRI</td>
<td></td>
</tr>
<tr>
<td>cktxn_txn_date</td>
<td>int(8)</td>
<td></td>
<td>PRI</td>
<td></td>
</tr>
<tr>
<td>cktxn_txn_time</td>
<td>int(6)</td>
<td></td>
<td>PRI</td>
<td></td>
</tr>
<tr>
<td>cktxn_termid</td>
<td>varchar(4)</td>
<td></td>
<td>PRI</td>
<td></td>
</tr>
<tr>
<td>cktxn_txn_status</td>
<td>int(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cktxn_txn_id</td>
<td>varchar(4)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>cktxn_dc_code</td>
<td>varchar(1)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>cktxn_txn_amt</td>
<td>int(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cktxn_pre_bal</td>
<td>int(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CRM BANKING is designed to perform 9 different functions for 4 different users. The following Figure 5 is the Use Case Diagram of this project.

Figure 5. Use Case Diagram
4.1 Customer Relationship Management
For Banking System Graphical User Interface Design

CRM BANKING GUI is easy to use. The GUI is written using Hyper Text Markup Language (HTML) forms and frames. Also, it also uses JavaScript to check the user input’s accuracy. Hence, the CRM BANKING GUI is executable with browsers that support JavaScript. The following sub sections explain the GUI work and details.

4.1.1 Customer Relationship Management For Banking System Login

The user logs in by providing a user id and a password. After verifying the user id and password, the JSP program will send the page to main menu. Also, the program will record the user id and privilege into session for later use. The menu program will display different user menu depending on the authority value. If the user id or password is in error, the program will show the error message and the user can re-login.
4.1.2 Inquire User Information Function

The administrator is able to inquire the individual information of a user by user id or employee id or browse the information of all users which is sorted by users' last name with this function.
Figure 7. Inquire User Information Function

4.1.3 Add User Function

In this function, the administrator is able to add a user and assign a privilege to a user.
4.1.4 Edit User Information Function

The administrator is able to modify one user personal information and the privilege by this function. Before editing user information, the administrator needs to inquire the user who is going to be modified.
4.1.5 Delete User Function

Once a user does not need to access this system, the administrator can delete the user with this function. Before deleting a user, the administrator needs to inquire the user information by user id or employee id to verify.
4.1.6 Customer Inquiry Function

This function provides manager, agent and clerk to inquire individual customer information. Users not only can view summary personal information and account information, but also can obtain a statistic of transaction history.
4.1.7 Channel Affinity Function

This function allows users to rank each channel such as ATM, ONLINE Banking, Phone Banking, and Branch Counter by access rate, transaction amount, and one type of transaction in a particular time interval. Moreover, this function also allows users to rank each channel applied by customers within a range of age, a type of gender, or all of the customers.
**Customer Relationship Management System**

<table>
<thead>
<tr>
<th>Customer Inquiry</th>
<th>Percentile Analysis</th>
<th>Channel Affinity</th>
<th>Segmentation</th>
<th>Customer Behavior Analysis</th>
</tr>
</thead>
</table>

**Accessed Channels Statistic By Transactions:**
- Access Times Period From: yyyy/mm/dd To: yyyy/mm/dd
- Transaction Amounts Period From: yyyy/mm/dd To: yyyy/mm/dd
- Deposit Transactions Period From: yyyy/mm/dd To: yyyy/mm/dd
- Withdraw Transactions Period From: yyyy/mm/dd To: yyyy/mm/dd

**Applied Channels Statistic By Customers:**
- One Range of Age Age Range
- Gender Gender
- All Customers

SUBMIT  |  RESET

---

Figure 12. Selection Page of Channel Affinity Function
Figure 13. Applied Channels Statistic of Channel Affinity

4.1.8 Percentile Analysis Function

This function provides users with the ability to calculate percentage by customer groups, account groups, and transaction groups. The groups defined for analysis are split into some ranges according to an attribute or a time interval.
### Customer Relationship Management System

<table>
<thead>
<tr>
<th>Customer Inquiry</th>
<th>Percentile Analysis</th>
<th>Channel Affinity</th>
<th>Segmentation</th>
<th>Customer Behavior Analysis</th>
</tr>
</thead>
</table>

Please select one group in the following:

**Customer Attribute Groups:**
- Age
- Gender
- Account Type

**Account Attribute Groups:**
- Saving Active Duration
- Checking Active Duration
- CD Deposit Term
- Saving Balance
- Checking Balance
- CD Balance

**Transaction Attribute Groups:**
- Transaction Amount
- Transaction Type
- Transaction Term
- 1 Month

![Figure 14. Group Selection Page of Percentile Analysis](image-url)

---

27
Figure 15. Criterion Selection Page of Percentile Analysis
Figure 16. Percentile Analysis Function by Customer Groups
Figure 17. Percentile Function by Account Groups
4.1.9 Segmentation Function

This function provides users with some selections for creating customer segments or account segments based on some customer attributes and account attributes. It also will create a file which is for further analysis with Customer Behavior Analysis function.
**Customer Relationship Management System**

Customer Inquiry ▶ Percentile Analysis ▶ Channel Affinity ▶ Segmentation ▶ Customer Behavior Analysis

Please select the criteria in the following:

**Customer Attributes:**
- **Age:** under 18 ▶
- **Gender:**
- **Account:** Saving ▶
- **Applied Channel:**

Please select one of account types:

**Account Attributes:**
- **Saving Balance:** ▶
- **Checking Balance:** ▶
- **CD Balance:** ▶

- **Active Duration:** ▶
- **Deposit Term:** ▶

[Submit] [Reset]

**Figure 19. Selection Page of Segmentation Function**

32
4.1.10 Customer Behavior Analysis Function

In this function, users select one segment which is created previously from Segmentation Function and to analyze customer behavior. First of all, this function shows all created segments and users decide the number of cluster for clustering. Then, it shows the result of clustering by applying the K-Means algorithm [2]. Users can continue to analyze the patterns of customer behaviors by applying a Decision Tree Learning algorithm [2]. Following will describe the result of clustering and decision tree.
1. After applying the K-Means algorithm for clustering, a result shows the mean values and the standard deviations of numerical attributes and the modes of nominal attributes for each cluster in Figure 22. User can realize what kind of clusters they obtain from clustering. In addition, the number and percentile of each cluster is also shown in the result.

2. After applying the Decision Tree Learning algorithm, a result shows a decision tree learner in Figure 23. The decision tree tells user that what patterns this segment of customers has. For example, in Figure 23, user can see the customers are divided into two branches by applied and not applied the ATM channel in the level of root. Secondly, the customers in the first branch are those who have applied ATM channel. Secondly, those customers are divided again by their ages and later divided by applied and not applied the online channel. The last level of this branch is shown which class of divided branch is.
### Customer Relationship Management System

<table>
<thead>
<tr>
<th>File Name</th>
<th>Count</th>
<th>Criterion</th>
<th>From</th>
<th>To</th>
<th>Criterion</th>
<th>From</th>
<th>To</th>
<th>criterion</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>20040417_20108.arff</td>
<td>578</td>
<td>age</td>
<td>31</td>
<td>50</td>
<td>saving</td>
<td>null</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20040417_20434.arff</td>
<td>578</td>
<td>age</td>
<td>31</td>
<td>50</td>
<td>saving</td>
<td>null</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20040420_224317.arff</td>
<td>15</td>
<td>age</td>
<td>0</td>
<td>18</td>
<td>saving</td>
<td>null</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20040420_224351.arff</td>
<td>15</td>
<td>age</td>
<td>0</td>
<td>18</td>
<td>saving</td>
<td>null</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20040420_233717.arff</td>
<td>47</td>
<td>account</td>
<td>checking</td>
<td>null</td>
<td>duration</td>
<td>0</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20040420_233728.arff</td>
<td>47</td>
<td>account</td>
<td>checking</td>
<td>null</td>
<td>duration</td>
<td>0</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20040420_234040.arff</td>
<td>71</td>
<td>account</td>
<td>saving</td>
<td>null</td>
<td>duration</td>
<td>0</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Input the number of clusters: [ ]

---

**Figure 21.** Selection Page of Customer Behavior Function
### Customer Relationship Management System

**Customer Inquiry** | **Percentile Analysis** | **Channel Affinity** | **Segmentation** | **Customer Behavior Analysis** | **K-Means**
--- | --- | --- | --- | --- | ---

Number of iterations: 3

Cluster centroids:

**Cluster 0**
- Gender: female
- Age: 27.5625
- ATM: yes
- ONLINE: yes
- PHONE: yes

**Cluster 1**
- Gender: female
- Age: 29
- ATM: no
- ONLINE: no
- PHONE: yes

---

**Clustering stats for training data**

Clustered Instances:
- 0-16 (80%)
- 1-4 (20%)

---

Figure 22. Customer Behavior Analysis by K-Means Algorithm
Figure 23. Decision Tree of Customer Behavior Analysis
CHAPTER FIVE

DATA MINING ALGORITHMS

In this project, two data mining algorithms, K-Means algorithm and Decision Tree Learning algorithm, are applied to analyze the complicated banking data and discover some patterns of customers' behaviors with the Customer Behavior Function.

5.1 K-Means Algorithm

The K-Means algorithm [2] generates a specific number of disjoint and non-hierarchical clusters. The process of this algorithm is shown below. The dataset is partitioned into K clusters and the data points are randomly assigned to the clusters resulting in clusters that have roughly the same number of data points. For each data point, calculate the distance from the data point to each cluster. If the data point is closest to its own cluster, leave it where it is. If the data point is not closest to its own cluster, move it into the closest cluster. Repeating the above step until a complete pass through all the data points result in no data point moving from one cluster to another. At this point the clusters are stable and the clustering process ends.
5.2 Decision Tree Algorithm

Decision Tree Learner [2] is one of popular classification models. Decision Trees are constructed recursively, applying divide-and-conquer to the training instances. The process of this algorithm is shown below.

- Start out with the original set of training instances.
- Pick the most informative attribute A from the list of attributes describing the training instances.
- For each value V for attribute A, create a branch, labeled V, and send down this branch the subset of the training instances with A=V.
- Examine each subset of training instances from the previous step: If the subset contains all instances of the same class, then create a leaf node labeled with this class. If the subset contains instances from multiple classes, repeat one to find an attribute to split the subset of training instances further.
- The Decision Tree Learning process terminates when no further splits training data subsets are possible. Ideally, at this point, all branches
of the Decision Tree end up in decision making leaf nodes.
CHAPTER SIX
SYSTEM VALIDATION

Testing of system validation is the testing process to ensure that the system meets user requirements, functions as designed, and is free of defects. The purpose of system validation is to have assurance about the software quality and functionalities. It also guarantees system performance and reliability.

6.1 Unit Test

Unit test is the basic level of testing. It verifies the functionality and performance of individual software modules and ensures that they work correctly. The unit testing result of CRM BANKING are shown in Table 11.

Table 11. Unit Test Results

<table>
<thead>
<tr>
<th>Function</th>
<th>Tests Performed</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login page</td>
<td>• Verify handling valid data input.</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>• Check all the buttons work properly.</td>
<td></td>
</tr>
<tr>
<td>Administrator menu</td>
<td>• Check all the links work as expected.</td>
<td>Pass</td>
</tr>
<tr>
<td>Manager menu</td>
<td>• Check all the links work as expected.</td>
<td>Pass</td>
</tr>
<tr>
<td>Agent menu</td>
<td>• Check all the links work as expected.</td>
<td>Pass</td>
</tr>
<tr>
<td>Clerk menu</td>
<td>• Check all the links work as expected.</td>
<td>Pass</td>
</tr>
<tr>
<td>Function</td>
<td>Tests Performed</td>
<td>Results</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| Inquire user                          | • Verify all required input validated.  
• Make sure all the required information appeared and same content as database.                                                                                                                              | Pass    |
| Add user                              | • Verify all required input validated.  
• Make sure all input data stored in database.                                                                                                                                                        | Pass    |
| Edit user                             | • Verify all required input validated.  
• Make sure all modified data stored in database.                                                                                                                                                     | Pass    |
| Delete user                           | • Verify all required input validated.  
• Make sure deleted data removed from database.                                                                                                                                                     | Pass    |
| Customer Inquiry                      | • Verify all required input validated.  
• Make sure all the required information appeared and same content as database.  
• Verify all data transferring from pervious input is correct.  
• Make sure all calculations are correct.  
• Check all the displayed data transfer to string correctly.                                                                                                                                             | Pass    |
| Channel Affinity measuring by transactions | • Verify all input validated.  
• Verify all data transferring from pervious input is correct.  
• Make sure all the required information appeared.  
• Make sure all displayed data is from a selected time interval.  
• Make sure all calculations are correct.  
• Check all the displayed data transfer to string correctly.                                                                                                                                         | Pass    |
<table>
<thead>
<tr>
<th>Function</th>
<th>Tests Performed</th>
<th>Results</th>
</tr>
</thead>
</table>
| Channel Affinity measuring by customers applying | - Verify all input validated.  
                                            - Verify all data transferring from previous input is correct.  
                                            - Make sure all the required information appeared.  
                                            - Make sure all displayed data is from a selected range of age or a selected type of gender.  
                                            - Make sure all calculations are correct.  
                                            - Check all the displayed data transfer to string correctly. | Pass    |
| Percentile Analysis by customer personal information groups | - Verify all input validated.  
                                            - Verify all data transferring from previous input is correct.  
                                            - Make sure all the required information appeared.  
                                            - Make sure all displayed data is from a selected range of age, a selected type of gender, or a selected type of account.  
                                            - Make sure all calculations are correct.  
                                            - Check all the displayed data transfer to string correctly. | Pass    |
| Percentile Analysis by account groups   | - Verify all input validated.  
                                            - Verify all data transferring from previous input is correct.  
                                            - Make sure all the required information appeared.  
                                            - Make sure all displayed data is from a selected range of account balance, or a selected duration/term.  
                                            - Make sure all calculations are correct.  
                                            - Check all the displayed data transfer to string correctly. | Pass    |
<table>
<thead>
<tr>
<th>Function</th>
<th>Tests Performed</th>
<th>Results</th>
</tr>
</thead>
</table>
| Percentile Analysis by transaction groups | • Verify all input validated.  
• Verify all data transferring from previous input is correct.  
• Make sure all the required information appeared.  
• Make sure all displayed data is from a selected type of transaction, a selected time interval, or a selected range of transaction amount.  
• Make sure all calculations are correct.  
• Check all the displayed data transfer to string correctly. | Pass    |
| Segmentation by customer attributes | • Verify all input validated.  
• Verify all data transferring from previous input is correct.  
• Make sure all the required information appeared.  
• Make sure all displayed data is from a selected range of age, a selected type of gender, a selected type of channel, and a selected type of account.  
• Check all the displayed data transfer to string correctly.  
• Make sure the downloaded file is correct. | Pass    |
| Segmentation by account attributes | • Verify all input validated.  
• Verify all data transferring from previous input is correct.  
• Make sure all the required information appeared.  
• Make sure all displayed data is from a selected range of account balance and a selected active duration/deposit term.  
• Check all the displayed data transfer to string correctly.  
• Make sure the downloaded file is correct. | Pass    |
<table>
<thead>
<tr>
<th>Function</th>
<th>Tests Performed</th>
<th>Results</th>
</tr>
</thead>
</table>
| Customer Behavior Analysis | • Verify all input validated.  
                          | • Verify all data transferring from previous input is correct.  
                          | • Make sure all the required information appeared and same as database.  
                          | • Make sure all calculations are correct.  
                          | • Make sure output a file clustered by K-Means Algorithm based on the selected file.  
                          | • Check all the displayed data transfer to string correctly. | Pass    |

6.2 Subsystem Testing

Subsystem testing is the second step in the testing process. It verifies that the software functions properly when individual software modules are linked together to form sub-system. Thus, the subsystem test process is useful for detecting interface errors and specific functions. Table 12 show subsystem test results in detail.
Table 12. Subsystem Test Results

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Tests Performed</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-Line Application Subsystem</td>
<td>• Test all the required input.</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>• Check all the input in session variable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Make sure all the information correctly passes to the next process.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Make sure all the information correctly shows up in application process.</td>
<td></td>
</tr>
<tr>
<td>Database Maintenance Subsystem</td>
<td>• Test insert / inquire / edit / delete function of each table.</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>• Check all the information in database.</td>
<td></td>
</tr>
</tbody>
</table>

6.3 System Testing

System testing is the testing process that uses real data, which the system is intended to manipulate, to test the system. First of all, all subsystems will be integrated into one system. Then test the system by using a variety of data to see the overall result.

System testing of CRM BANKING system begins with the following steps (Table 13):

Table 13. System Test Results

<table>
<thead>
<tr>
<th>System Testing</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Install CRM BANKING system into server.</td>
<td>Pass</td>
</tr>
<tr>
<td>2. Start up all services such as JSP engine, MySQL database engine.</td>
<td>Pass</td>
</tr>
</tbody>
</table>
CHAPTER SEVEN
MAINTENANCE MANUAL

System maintenance is an important step to ensure that the system runs smoothly and meets the user’s expectation. In CRM BANKING system, there are 3 major issues: Software Installation, Variables Modification, and CRM BANKING Installation/Migration.

7.1 Software Installation

In CRM BANKING, it requires RedHat, MySQL, JSDK, TOMCAT, and JDBC to run the programs. Following will detail the installations of software.

7.1.1 RedHat Installation

RedHat is a Linux base operating system which is offered freely and be downloaded from internet. Following are the steps to install RedHat onto your machine.


2. Install the operating system by inserting CD 1 into the CD-ROM and start up the machine which is going to install the operating system.
3. The machine will startup via CD-ROM and start to install RedHat.

4. Follow the install wizard and sets up the required information such as network setting and the hardware environment.

5. After all the necessary files are copied into the computer and install it, the machine will restart and RedHat is installed.

7.1.2 MySQL Installation

MySQL is the database system we use in CRM BANKING. Because it provides JDBC to easily connect with JAVA program and the access performance is also good, it's a good choice for designing this project. To install MySQL, follow the following 2 steps:

1. Because MySQL may install on to RedHat when the operating system is installed, the first thing we have to do is to check if the MySQL is already in the operating system. Using the command to check if MySQL exists in the operating system:

   ```bash
   rpm -q mysql
   ```

   If MySQL is not installed in the operating system, then use rpm to install it.
2. In order to create a database user, "crm," and database, "crmdb" for CRM BANKING use, have the following commands executed.

   su mysql
   initdb -D /var/lib/mysql/data
   createuser crm
   createdb crmdb

At the first command, the "root" is the default user for MySQL. Starting the database by using the command "initdb" with the directory "/var/lib/mysql/data/" which is the default database directory will start up the database system. Login root as the supervisor and create a database user, "crm," and the database "crmdb."

7.1.3 JAVA 2 Platform, Standard Edition (J2SE)

J2SE is the compiler program for JSP and JAVA Servlet programs and it's required in TOMCAT JAVA Container.

1. Download SDK Linux at
   http://java.sun.com/j2se/1.4.2/download.html to the directory /usr/java, then execute the following commands:

   chmod +x j2sdk-1_4_2_02-linux-i586-rpm.bin
   ./j2sdk-1_4_2_02-linux-i586-rpm.bin
   rpm -ivh j2sdk-1_4_2_02-linux-i586.rpm
2. Set the environment variables in the file /etc/profile.d/*.sh:

   JAVA_HOME=/usr/java/j2sdk1.4.2_02
   PATH=${PATH}:${JAVA_HOME}/bin

   Export JAVA_HOME

7.1.4 Tomcat

   TOMCAT is one of the Apache Jakarta projects, which is a web container to process JSP and JAVA Servlet programs, and to serve static web pages.

   1. Go to the tomcat’s official download ftp server at
      http://ftp.epix.net/apache/jakarta/tomcat-4/v4.1.29/bin/ to download the file of tomcat server for Linux jakarta-tomcat-4.1.29.tar.gz to /usr/java/ and extract it to the hard drive.
      
      tar -xzvf jakarta-tomcat-4.1.29.tar.gz

   2. Set the environment variable by adding the following lines in the file /etc/profile.d/*.sh:

      CATALINA_HOME=/usr/java/jarkata-tomcat-4.1.29
      PATH=${PATH}:${JAVA_HOME}/bin:${CATALINA_HOME}/bin

      export CATALINA_HOME
3. Add the following lines in the file
/etc/rc.local to have the tomcat run when the
system boots:

```bash
export JAVA_HOME=/usr/java/j2sdk1.4.2_02
export
CATALINA_HOME=/usr/java/jarkata-tomcat-4.1.29
${CATALINA_HOME}/bin/startup.sh
```

### 7.1.5 JAVA Database Connectivity (JDBC)

The API used to execute SQL statement is different
for each database engine. Java programmers, however, are
lucky and are freed from such database portability issues.
They have a single API, the Java Database Connectivity API
(JDBC), that’s portable between database engines. The JDBC
library provides an interface for executing SQL
statements. It provides the basic functionality for data
access. First of all, we can download the file
mysql0jdbc.tar.gz at
http://dev.mysql.com/downloads/connector/j/3.0.html and
copy the file mysql-connector-java-3.0.9-stable-bin.jar to
/usr/java/jakarta-tomcat-4.1.29/common/lib/.
7.2 Variables Modification

In CRM BANKING, we have to change some environment variables in Tomcat server.

7.2.1 System Variables

1. Modify the file “server.xml” in the directory “/usr/java/jakarta-tomcat-4.1.29/conf.”

2. The variable “path” in Context indicates the context path of the web application. The default value would be “/crm.”

3. The variable “docBase” in Context is the files directory for the web application. The default value would be “/crm.”

7.2.2 Copying Files

1. In order to run JAVA Servlet on the server, we need to copy a file servlet.jar to /usr/java/jakarta-tomcat-4.1.29/common/lib/.

2. In order to present graphic on web page, we need to copy two files jfreechart-0.9.4.jar and jcommon-0.7.1.jar to /usr/java/jakarta-tomcat-4.1.29/common/lib/.
7.3 Customer Relationship Management For Banking System Installation/Migration

1. All the JSP programs and HTML programs are stored in
   $CATALINA_HOME\webapps\crm$

2. All the classes are stored in
   $CATALINA_HOME\webapps\crm\WEB-INF\classes$

3. All the database files are stored in
   VAR\LIB\MYSQL\data\crmdb$

7.4 Backup

Backup is a very important action for an administrator. Because we can not ensure the system will work and not be stuck forever, backup can recover the system to its original status. In CRM BANKING, we need to backup two components, system files and database.

7.4.1 System Backup

All the CRM BANKING system files are located in the directory "webapps.crm" and all its subdirectory. Thus, in order to backup the system files, all we need to do is to backup the files in the directory. The method here I suggest is to compress the directory of "/webapps/crm" including its subdirectory to compress files for future use by the compress program "tar." Using the following command to backup the system files:

   tar -cf CRM.tar /webapps/crm.
7.4.2 Database Backup

To backup the database system, we use `mysql_dump` command to backup the database used by CRM BANKING. The following command is used to backup the database:

```
mysql_dump crmdb | gzip > crmdb.zip
```

After executing the backup command above, the file `crmdb.zip` would be the backup file of the database.
CHAPTER EIGHT
CONCLUSION AND FUTURE DIRECTIONS

8.1 Conclusion

CRM BANKING is an integrated and powerful customer relationship management system for the banking industry. It allows you to manage relationships based on a common understanding of all customer interactions, detect events to provide for timely and relevant customer communications, and apply rules to analyze customer transactions across banking channels. The application is based on an integrated suite of functionalities that help you to understand what is important for individual customers and customer segments and communicate based on that understanding.

8.2 Future Directions

CRM BANKING provides a start point to manage customer relationships for a bank with a general web application. Due to abundant data in a bank, all of banking data is able to be analyzed by many different ways. In the future, CRM BANKING could offer more functions, so that users can discover more valuable information for customer communications. For example, a function allows users to set up some promotion policies. Later users can apply them
to generate some information for customers in order to increase customer contributions. In addition, another function lets users download a list of customers’ contact information such as email address, phone number, or mailing address for communication with these particular customers. Moreover, CRM BANKING could apply more data mining algorithms for an in-depth analysis of those customer data to help a bank take critical business decisions in the competitive financial world.
REFERENCES


