The Delivery Information System of SIAMLIFT Industries Company

by

Mr. Dejsaridh Harnratanakul

A Final Report of the Three-Credit Course
CS 6998 System Development Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer Information Systems
Assumption University

December, 2001
The Delivery Information System of SIAMLIFT Industries Company

by

Mr. Dejsaridh Harnratanakul

A Final Report of the Three-Credit Course
CS 6998 System Development Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer Information Systems
Assumption University

December 2001
Project Title: The Delivery Information System of SIAMLIFT Industries Company

Name: Mr. Dejsaridh Harnratanakul

Project Advisor: Air Marshal Dr. Chulit Meesajjee

Academic Year: December 2, 2001

The Graduate School of Assumption University has approved this final report of the three-credit course, CS 6998 System Development Project, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Information Systems.

Approval Committee:

-Air Marshal Dr. Chulit Meesajjee (Dean and Advisor)
-Prof Dr. Srisakdi Charmonman (Chairman)
-Asst. Prof Dr. Vichit Avatchanakorn (Member)
-Assoc. Prof. Somchai Thayarnyong (MUA Representative)

December 2, 2001
SIAMLIFT Industries Co., Ltd., is one of the big Lift engine manufactures in the Thai Lift's system community, and the products of SIAMLIFT is wildly use in many famous high building and important project such as BTS electronic sky train in Bangkok city.

The current existing system is based on manual operationally. All of the data are currently stored on paper, which requires a lot of administrative work and staffs to maintain the system. Facing classical problems implied by the use of the manual system is therefore obvious and those problems are basically described as error-prone and high maintenance cost, therefore, this project is to develop and effective information system to facilitate the existing process.

The new proposed Information System will be developed to replace the manual system with a two-tiered local area networking topology system (LAN). All data will be kept in a database server, Interbase, and are accesses through the client/server architecture computing. The user interfaces, moreover, are implemented through a GUI design in Borland Jbuilder Enterprise Edition 5. It will reduce the mount of administrative tasks, reduce the required staffs, solve the problem of the manual system, decrease the high maintenance cost and more importantly considerably improve the processing time concerning queries or delivery transactions.
ACKNOWLEDGEMENTS

Several people have made contributions to this project. The writer would like to acknowledge their efforts and thank them for their contributions.

He would like to thanks Air Marshal Dr. Chulit Meesajjee, his project advisor, for this valuable suggestion and advice given in to the preparation of this project.

He extends his sincere thanks to Miss Sureeporn Pratoomsuvan, the co-operator in this project for her timely assistance and information provided to him while carrying out the data collection required for this project.

Finally, the writer would like to thanks Asst.Prof.Dr. Ouen Pingern for this valuable advice and his excellence in teach System Analysis and System Design courses, part of the degree curriculum, that gave the background to this project development.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>i</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>ii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background of the Project</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Objectives of the Project</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Scope of the Project</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Deliverables</td>
<td>3</td>
</tr>
<tr>
<td>1.5 Project Plan</td>
<td>4</td>
</tr>
<tr>
<td>II. THE EXISTING SYSTEM</td>
<td>7</td>
</tr>
<tr>
<td>2.1 Background of the Organization</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Overview of Existing System</td>
<td>11</td>
</tr>
<tr>
<td>2.3 Current Problems and Recommended Area for Improvement</td>
<td>15</td>
</tr>
<tr>
<td>III. THE PROPOSED SYSTEM</td>
<td>17</td>
</tr>
<tr>
<td>3.1 System Specification</td>
<td>17</td>
</tr>
<tr>
<td>3.2 Data Modeling</td>
<td>18</td>
</tr>
<tr>
<td>3.3 Process Modeling</td>
<td>20</td>
</tr>
<tr>
<td>3.4 Physical Data Flow Design</td>
<td>45</td>
</tr>
<tr>
<td>3.5 Database Design</td>
<td>45</td>
</tr>
<tr>
<td>3.6 Input and Output Design</td>
<td>49</td>
</tr>
<tr>
<td>3.7 Software Design</td>
<td>50</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>3.8 System Architecture Design</td>
<td>60</td>
</tr>
<tr>
<td>3.9 Hardware and Software Requirement</td>
<td>63</td>
</tr>
<tr>
<td>3.10 Network Architecture</td>
<td>65</td>
</tr>
<tr>
<td>3.11 Network Configuration</td>
<td>70</td>
</tr>
<tr>
<td>3.12 System Cost Analysis</td>
<td>76</td>
</tr>
<tr>
<td>IV. PROJECT IMPLEMENTATION</td>
<td>84</td>
</tr>
<tr>
<td>4.1 Overview of Project Implementation</td>
<td>84</td>
</tr>
<tr>
<td>4.2 Programming</td>
<td>84</td>
</tr>
<tr>
<td>4.3 Testing</td>
<td>84</td>
</tr>
<tr>
<td>4.4 Conversion</td>
<td>85</td>
</tr>
<tr>
<td>4.5 Training</td>
<td>86</td>
</tr>
<tr>
<td>V. CONCLUSIONS AND RECOMMENDATIONS</td>
<td>88</td>
</tr>
<tr>
<td>5.1 Conclusions</td>
<td>88</td>
</tr>
<tr>
<td>5.2 Recommendations</td>
<td>90</td>
</tr>
<tr>
<td>APPENDIX A DATA DICTIONARY</td>
<td>92</td>
</tr>
<tr>
<td>APPENDIX B DATABASE DESIGN</td>
<td>97</td>
</tr>
<tr>
<td>APPENDIX C INPUT DESIGN</td>
<td>103</td>
</tr>
<tr>
<td>APPENDIX D OUTPUT DESIGN</td>
<td>111</td>
</tr>
<tr>
<td>APPENDIX E SQL DATABASE SCHEMA CODE</td>
<td>112</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>133</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>1.1</td>
<td>Project Plan of Delivery Information System (Gantt Chart)</td>
</tr>
<tr>
<td>2.1</td>
<td>Organization Chart of SIAMLIFT Industries Co., Ltd.</td>
</tr>
<tr>
<td>3.1</td>
<td>Context Data Model</td>
</tr>
<tr>
<td>3.2</td>
<td>Key-Based Data Model</td>
</tr>
<tr>
<td>3.3</td>
<td>Fully Attribute Data Model</td>
</tr>
<tr>
<td>3.4</td>
<td>Context Diagram of Delivery Information System</td>
</tr>
<tr>
<td>3.5</td>
<td>Function Decomposition Diagram of Delivery Information System</td>
</tr>
<tr>
<td>3.6</td>
<td>Function Decomposition Diagram of Package Subsystem</td>
</tr>
<tr>
<td>3.7</td>
<td>Function Decomposition Diagram of Delivery Subsystem</td>
</tr>
<tr>
<td>3.8</td>
<td>First Level of DFD</td>
</tr>
<tr>
<td>3.9</td>
<td>Second DFD Level Explosion of Process 1</td>
</tr>
<tr>
<td>3.10</td>
<td>Second DFD Level Explosion of Process 2</td>
</tr>
<tr>
<td>3.11</td>
<td>Second DFD Level Explosion of Process 3</td>
</tr>
<tr>
<td>3.12</td>
<td>Second DFD Level Explosion of Process 4</td>
</tr>
<tr>
<td>3.13</td>
<td>Third DFD Level Explosion of Process 1</td>
</tr>
<tr>
<td>3.14</td>
<td>Second Third DFD Level Explosion of Process 1</td>
</tr>
<tr>
<td>3.15</td>
<td>Third DFD Level Explosion of Process 2</td>
</tr>
<tr>
<td>3.16</td>
<td>Second Third DFD Level Explosion of Process 2</td>
</tr>
<tr>
<td>3.17</td>
<td>Third DFD Level Explosion of Process 3</td>
</tr>
<tr>
<td>3.18</td>
<td>Second Third DFD Level Explosion of Process 3</td>
</tr>
<tr>
<td>3.19</td>
<td>Third Third DFD Level Explosion of Process 3</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------</td>
<td>------</td>
</tr>
<tr>
<td>3.20 Third DFD Level Explosion of Process 4</td>
<td>41</td>
</tr>
<tr>
<td>3.21 Second Third DFD Level Explosion of Process 4</td>
<td>42</td>
</tr>
<tr>
<td>3.22 Third Third DFD Level Explosion of Process 4</td>
<td>43</td>
</tr>
<tr>
<td>3.23 Forth Third DFD Level Explosion of Process 4</td>
<td>44</td>
</tr>
<tr>
<td>3.24 Proposed System's Physical Database Schema</td>
<td>48</td>
</tr>
<tr>
<td>3.25 Revise Third DFD Level of Process 1</td>
<td>52</td>
</tr>
<tr>
<td>3.26 Revise Third DFD Level of Process 2</td>
<td>53</td>
</tr>
<tr>
<td>3.27 Revise Third DFD Level of Process 3</td>
<td>54</td>
</tr>
<tr>
<td>3.28 Revise Third DFD Level of Process 4</td>
<td>55</td>
</tr>
<tr>
<td>3.29 Structure Chart of Process 1</td>
<td>56</td>
</tr>
<tr>
<td>3.30 Structure Chart of Process 2</td>
<td>57</td>
</tr>
<tr>
<td>3.31 Structure Chart of Process 3</td>
<td>58</td>
</tr>
<tr>
<td>3.32 Structure Chart of Process 4</td>
<td>59</td>
</tr>
<tr>
<td>3.33 Network Configuration</td>
<td>72</td>
</tr>
<tr>
<td>3.34 Diagram Overview of Network Connection of Delivery Information System</td>
<td>73</td>
</tr>
<tr>
<td>3.35 Virtual Private Network Configuration of Delivery Information System</td>
<td>74</td>
</tr>
<tr>
<td>3.36 Security Schema of VPN for Delivery Information System</td>
<td>75</td>
</tr>
<tr>
<td>3.37 Cost Comparison between Manual and Computerized System</td>
<td>82</td>
</tr>
<tr>
<td>3.38 Payback Analysis of Logistic Management System</td>
<td>83</td>
</tr>
<tr>
<td>C.1 Bus Data Editor Input Screen of Delivery Information System</td>
<td>103</td>
</tr>
<tr>
<td>C.2 Destination Data Editor Input Screen of Delivery Information System</td>
<td>104</td>
</tr>
<tr>
<td>C.3 Package Data Editor Input Screen of Delivery Information System</td>
<td>105</td>
</tr>
<tr>
<td>C.4 Delivery Destination Data Input Screen of Delivery Information System</td>
<td>106</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>C.5</td>
<td>Destination Data Selective Screen of Delivery Information System</td>
</tr>
<tr>
<td>C.6</td>
<td>Destination Data Search Screen of Delivery Information System</td>
</tr>
<tr>
<td>C.7</td>
<td>Main Delivery Input Screen of Delivery Information System</td>
</tr>
<tr>
<td>C.8</td>
<td>Delivery History Report Screen of Delivery Information System</td>
</tr>
<tr>
<td>D.1</td>
<td>Receipt Report Output of Delivery Information System</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Manual Working Procedure</td>
<td>14</td>
</tr>
<tr>
<td>3.1</td>
<td>List of Entity</td>
<td>20</td>
</tr>
<tr>
<td>3.2</td>
<td>Database Size Evaluation</td>
<td>47</td>
</tr>
<tr>
<td>3.3</td>
<td>Estimated Projected Cost</td>
<td>76</td>
</tr>
<tr>
<td>3.4</td>
<td>Manual System Cost Analysis</td>
<td>77</td>
</tr>
<tr>
<td>3.5</td>
<td>Five Years Accumulated Manual System Cost</td>
<td>77</td>
</tr>
<tr>
<td>3.6</td>
<td>Computerized System Cost Analysis</td>
<td>78</td>
</tr>
<tr>
<td>3.7</td>
<td>Five Years Accumulated Computerized Cost</td>
<td>79</td>
</tr>
<tr>
<td>3.8</td>
<td>The Comparison of the System Costs</td>
<td>79</td>
</tr>
<tr>
<td>3.9</td>
<td>Payback Analysis of Delivery Information System</td>
<td>80</td>
</tr>
<tr>
<td>3.10</td>
<td>Net Present Value and ROI of Delivery Information System</td>
<td>81</td>
</tr>
<tr>
<td>5.1</td>
<td>The Degree of Achievement of the Proposed System</td>
<td>89</td>
</tr>
<tr>
<td>A.1</td>
<td>Structure of Bus_Data Table</td>
<td>92</td>
</tr>
<tr>
<td>A.2</td>
<td>Structure of Package_Data Table</td>
<td>93</td>
</tr>
<tr>
<td>A.3</td>
<td>Structure of Part_Data Table</td>
<td>94</td>
</tr>
<tr>
<td>A.4</td>
<td>Structure of Package_Part Table</td>
<td>94</td>
</tr>
<tr>
<td>A.5</td>
<td>Structure of Bus_Schedule Table</td>
<td>95</td>
</tr>
<tr>
<td>A.6</td>
<td>Structure of Bus_on_site Table</td>
<td>95</td>
</tr>
<tr>
<td>A.7</td>
<td>Structure of Delivery_Package Table</td>
<td>96</td>
</tr>
<tr>
<td>A.8</td>
<td>Structure of Destination_Data Table</td>
<td>96</td>
</tr>
<tr>
<td>A.9</td>
<td>Structure of Site Table</td>
<td>96</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

1.1 Background of the Project

SIAMLIFT Industries Co., Ltd., first Thai Elevator, was established with the initial registered capital of 2 million Baht in April 1977. The production from this factory was under the trademark “SIAMAN” which contained locally produced parts instead of the imported ones.

The first factory had the total production area of only 600 square meters, located near the Bangsue waterway (Prachanukul). During that period, the factory had the capacity to produce only 50 – 60 elevators annually. Even so, the first Thai factory continued to produce elevators with such high quality that was well accepted in both domestic and international markets.

In 1984, only 7 years after the establishment of the company, the new plant in Rangsit District had the production area of up to 2,400 square meters to support the quantities of machinery and work force. The production capacity had increased to 300 elevators a year. At the same time, the company established its own research and development unit for developing elevator components.

In 1992, the company applied for registration as a public company limited and was approved to be a registered company in the Stock Exchange of Thailand in 1994. The registered capital was increased to 300 million Baht.

KONE Elevator, the Finnish biggest elevator manufacturer and distributor, and also one of the worlds, top three elevator manufacturers and distributors, made a joint venture with the company. Under the joint venture, KONE granted SIAMLIFT Industries Co., Ltd. the right to delivery, offer maintenance services and manufacture elevators under KONE trademark.
In addition, the company has expanded the business by acquiring the role of distributor for other products with related technologies as follows:

WOHR: a car parking system of a German company OTTO WOHR.

Stannah: an English produced stairlift.

1.2 Objectives of the Project

The objectives expected from the development of this new information system are as follows:

1. To change the existing manual system to the new computer system.
2. Better management control obtained by the possibility of timely and consistent reports establishing the exact stock and delivery state.
3. Better Sharing of data and information between departments resulting in a more efficient work.
4. Increase job satisfaction for employee by elimination tedious tasks and reducing processing work.
5. Better services to customers (faster services) by nearly instant responses to queries.
6. Easier billing process system enabling less mistakes from the staff and more reliable hard copy backup.
7. Time saving in operation processing and cost management.
8. Change the way of information flow by documents to the electronic information for speeding up work process and easy to prepare the material in stock before the new project arrive the factory.
9. Easy to track the materials that we deliver to the other places.
10. Reduce cost and maximize the profit by computerize schedule control.
11. Forecasting and improvement of management planning activities through a
better Management Information System (MIS).

Note: The above objectives can be subjected to change during the analysis part of the project development depending on the available resources (budget, time) or modifications in scope.

1.3 Scope of the Project

This project will emphasize the development of a new system (computerized) to replace the legacy (existing) manual system.

The scope will define the boundaries of the project and will concern it as follows:

(1) Manage efficiency and fast delivery.

(2) Implementation of the new computerized system to reduce paper work and paper use.

(3) Creating the Database for an exact delivery.

(4) Providing a better decision making supporting tool.

(5) Generate report templates for delivery, request forms, Deliver form.

(6) Providing an Internet connection to make supplier contact, ordering or payment possible through data communication (if supported by suppliers).

(7) Implementation of mail system to communicate between each site and department.

1.4 Deliverables

After performing the system analysis and design of the new system, the deliverable will be the final “blueprint” of the newly designed system and thus readiness for implementation and practical use.

The deliverables of this project are as followings:

Analysis Design

(1) System Design diagram.
1.5 Project Plan

1.5.1 Cost Estimate

The time estimation to implement the proposed system described in previous section depends mostly on three factors that can be identified as the hardware setup, the Software Utility and Integrated Development Environment setup, and finally, the Software development and the training of the staff and system testing.

The time features of purchasing and procurement should be roughly around three months, and the time for configuration and setup should be around one month.

1.5.2 Time Estimate

Costs involved in the implementation of the new system can be evaluated on the following factors:

Total cost = Hardware cost + Software cost + Setup + System development cost

= 701,500 + 25,000 + 245,000 + 131,000 = 1,102,500 Baht

The time schedule for completion of the entire project from survey phase to delivery phase is roughly approximated quantitatively to four months as detailed below:

(1) Survey phase: 1 week.

(2) Study phase: 2 weeks.

(3) Definition phase: 6 weeks.

(4) Configuration phase: 1 week.
(5) Design phase: 6 weeks.
(6) Procurement phase: 6 weeks.
(7) Construction phase: 6 weeks.
(8) Delivery phase: 2 weeks.

This schedule is likely to be subjected to further changes during (at the end of) each of some phases of the project since we are only in the early stages of the development system. The corresponding Gantt Chart is shown in Figure 1.1.
<table>
<thead>
<tr>
<th>No.</th>
<th>Task Name</th>
<th>May</th>
<th>June</th>
<th>July</th>
<th>August</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 2 3 4</td>
<td>1 2 3 4</td>
<td>1 2 3 4</td>
<td>1 2 3 4</td>
</tr>
<tr>
<td>I.</td>
<td>Analysis of the Existing System</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Define the Objective and Scope</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Study the Existing System</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Identify the Existing Problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Study the Existing Computer System</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Develop Context Diagram</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Develop Data Flow Diagram</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Cost and Benefit Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>II.</td>
<td>Analysis and Design of the Proposed System</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Web Interface Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Report Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Database Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Network Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Program Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>III.</td>
<td>Implementation of the Proposed System</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Develop the software</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Hardware Installation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Software Installation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Conversion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1.1. Project Plan of Delivery Information System.
II. THE EXISTING SYSTEM

2.1 Overview and Background of the Organization

SIAMLIFT Industries Co., Ltd., first Thai Elevator, was established with the initially registered capital of 2 million Baht in April 1977. The production from this factory was under the trademark “SIAMAN” which contained locally produced parts instead of the imported ones.

In the midst of a growing Thai economy in the past two decades, together with an increasing number of population and overcrowding residential areas, a piece of land has therefore been, much needed and become more and more expensive. Consequent upon that, the behavioral pattern of land usage had to be altered. The building construction shifted from horizontal to vertical direction in order to fully utilize each piece of land.

That is the beginning of skyscrapers, which have been developed into residences, offices and various stores. The roles of skyscrapers have created modern image and new perspective of elevator standards of the people.

As such, skyscrapers can be regarded as a reflection of modern lifestyle. Such elevator style provided more opportunity for people to use elevators as means of transport inside the buildings. It can be seen that the more skyscrapers there are, the greater the elevators are in demand.

The first factory had the total production area of only 600 square meters, located near the Bangsue waterway (Klong Prapa). During that period, the factory had the capacity to produce only 50 – 60 elevators annually. Even so, the first Thai factory continued to produce elevators with such high quality that it was well accepted in both domestic and international markets.
Before having gained such success, the first stage of market penetration required much physical and mental efforts from every party concerned. The engineering team had to work so intensively that finally it was awarded with customer acceptance and trust, particularly in the area of the quality and after sale services. As a result, the factory was not able to produce sufficient elevators to meet the higher demand of the markets.

In 1984, only 7 years after the establishment of the company, the new plant in Rangsit District had the production area of up to 2,400 square meters to support the quantities of machinery and work force. The production capacity had increased to 300 elevators a year. At the same time, the company has established its own research and development unit for developing elevator components.

SIAMLIFT Industries Co., Ltd., was to overcome a lot of obstacles and at the same time had to build up the image. Endless physical, mental and intellectual efforts had to be invested along the same line with new technologies, modern production process, strict systems of inspection and quality control, as well as prompt attention in service to ensure customers full satisfaction. Such investment had won trust and popularity of the company among the customers.

Demand for high-speed elevators, both in quality and quantity, has increased with the continuous expansion of skyscrapers. SIAMLIFT Industries Co., Ltd., which aimed for medium-rise markets, has had to adapt itself to be able to compete for a more market share.

In 1992, the company applied for registration as a public company limited and was approved to be registered a company in the Stock Exchange of Thailand in 1994. The registered capital was increased to 300 million Baht.

KONE Elevator, the biggest Finnish elevator manufacturer and distributor, and
also one of the world's top three elevator manufacturers and distributors, had a joint venture with the company. Under the joint venture, KONE granted SIAMLIFT Industries Co., Ltd. the right to delivery, offer maintenance services and manufacture elevators under the KONE trademark. The right granted includes assistance in technical, marketing and workforce training of the company which strengthened SIAMLIFT Industries Co., Ltd., with having a higher potential in the production.

In addition, the company has expanded its business by acquiring the role of distributor for other products with related technologies as follows:

- **WOHR**: a car parking system of a German company OTTO WOHR.
- **Stannah**: an English produced stairlift.

**Location**: SIAMLIFT Industries Co., Ltd., is at 239 Rimklongprapa Road, Bangsue District, Bangkok. The office houses various departments which are ready to serve as a coordinating center in order to provide convenience, effective operation and prompt services to the customers.

The Organization Chart corresponding to the above described business is represented in Figure 2.1.
Figure 2.1. Organization Chart of SIAMLIFT Industries Co., Ltd.
2.2 Overview of the Existing System

The existing Delivery System at SIAM LIFT Industries Co., Ltd., is a manual system that the employees of Distributed Center must try to make everything manually. So, the Distributed Center has many problems that could not be solved by business process reengineering process. The following is the work process of the manual Distributed Center.

The process of Delivery Department’ distributed center started when the other internal departments need to send parts or materials to the other internal department or the internal department need to send elevator components to the customer’s site for assemble and install new elevators in the shaft. The example of internal to internal movement is when the warehouse department sends the parts or raw materials to the factory for manufacturing the elevator’s component. After the production process is completed the factory has to send the elevator’s component to the warehouse to be kept as spare parts or ready to install elevator’s components.

All the working process of above have many details in each task and is related to many people that may cause the error in progress with the result that estimate the costs could not be. The following paragraphs explain the working process of the manual Delivery Department’s distributed center system by using a base case that the warehouse department needs to send parts or materials to the factory.

The warehouse needs to create the RFD (Request for delivery) to the distributeing center that is located in every site. The RFD contains the date and times that the warehouse needs the DC to get the parts from warehouse and the date times that the warehouse need the DC to deliver to the destination, such as the factory in this case.

After receiveing the RFD, DC will send the copy of RFD to the Delivery Department at the head office to know the delivery task and wait for the approved
document with the signature of managing director of delivery control back from the Delivery Department. Then the DC sends the staff to the warehouse department on the date and times as specified in RFD.

When the DC’s staff reached the warehouse department, they got the parts and list of parts to send documents. Then the DC’s staff will estimate the possibility of sending those parts or materials in case the parts doesn’t match the DC’s delivery bus or that parts and materials may cause harm to the delivery procedure and in the worse case, the parts may be useless when it reached the destination. Which the DC must be responsible for this mistake and error of delivery control.

If the staffs accept to deliver all the parts and materials, they will issue the received confirmation document to the warehouse and take the parts to store temporary at the DC’s storage. Then it’s time to estimate the size of parts and materials for putting it down into the container such as wood boxes, and palettes with the approved document of DC manager at that site.

After completing containing procedure, DC must reserve the bus schedule plan to go to the destination. If the bus isn’t available to send the parts to the destination on time as specified by RFD, the DC must issue the delivery delay document to the destination with one copy to the Delivery Department at head office and the other copy to the source, which is the warehouse department in this case.

When the DC’s bus reached the destination address, the DC’s staff at the source site will give the parts and materials to the DC’s staff at the destination site. And the destination site DC’s staff will check the amount and type of parts to make sure that all parts match to the delivery part document. Then DC’s staff at the destination site will issue the received confirmation document to the staff at the source site.

Finally, the DC’s staff at the destination site will extract the parts from the box of
container and to the destination department as specified in the RFD document. Then the destination department will verify that all of parts are correct and issue the complete received confirmation document to the DC’s staff for confirming the end of delivery procedure. After that, the destination site DC’s manager will send the destination confirmation document to the Delivery Department at head office to guarantee the complete deliver finished and all parts are correct.

However, all delivery tasks above doesn’t include the cancel or reject procedure that made the delivery process halt and interrupt the other delivery bus plan or procedure which may cause of effect that made the other delivery tasks to become unsuccessful. The example of these situations come from many interrupted procedures such as, after the DC packaged the parts into the package box, the warehouse may ask cancelation to deliver parts and required to send all parts back to the warehouse again while the bus schedule is already fixed to go to the destination address. In the worse case, the fixed schedule bus may contain the other delivery package that joins the same bus to the same destination but different department.

The other problem is error in delivery control because the manual system has to pass the parts list document to the other one about three steps, which may cause lost parts and unable to trace the point of “out of control” procedure. The last main problem is the staff doesn’t know the parts that the source needs to send before going to receive the parts for delivery, because the RFD isn’t specific about the parts that needed to send on that delivery request. This is the big problem for the DC manager at that site to manage the bus schedule and prepare the package that matches the parts to send in packaging procedure, which may cause the delay of delivery if the package is suitable to contain materials out of stock.

The following table list the manual working procedure step by step for easy to
understanding of the overview of distributed center procedure.

Table 2.1. Manual Working Procedure.

<table>
<thead>
<tr>
<th>Source and Destination</th>
<th>DC’s staff</th>
<th>Delivery Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warehouse request to sends parts by create RFD</td>
<td>Get parts. And issue the received confirmation document.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create list of parts and send to the head office for approved to delivery.</td>
<td>Create approved document to delivery and send back.</td>
</tr>
<tr>
<td></td>
<td>Create confirm document for both source and destination of delivery.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Packaging all the parts procedure and booking the bus schedule.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Destination DC’ staff get the parts and check for all parts are completely deliver.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create received confirmation document for the DC’s staff at the source of delivery.</td>
<td></td>
</tr>
</tbody>
</table>
Table 2.1. Manual Working Procedure (Continued).

<table>
<thead>
<tr>
<th>Source and Destination</th>
<th>DC's staff</th>
<th>Delivery Department</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Extract the package and send all parts to destination department.</td>
<td></td>
</tr>
<tr>
<td>Issue received parts confirmation to DC's staff after completely verify all parts are correct.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Send the confirmation of finished delivery procedure to the head office.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Keep record of finished delivery task for reference later at the end of year.</td>
<td></td>
</tr>
</tbody>
</table>

2.3 Current Problems and Recommended Areas for Improvement

The current problems of the existing system will be viewed and stated from the management's point of view as well as from the user’s point of view and also from the analyst’s point of view.

Problems from the management and users will be analyzed providing a forward determination; some opportunities and directives, by the system analyst through the Performance, Information, Economics, Control, Efficiency and Service. This is condensed and named the “PIECES” framework. Each of the previous points will also be described in term of problems, opportunities and directives.
2.3.1 Causal Analysis

This method is in the line of the previous method (already discussed in the parts above) consisting the determination of problems using the FAST methodology through the PIECES framework of the Representative Problems Statement Table. We will discuss now the Cause–Effect relationship analysis of the existing system problems through another perspective involving “Problems, Objectives and Constraints Matrix.”

2.3.2 Recommendations

(1) Implementation of an Information System based on Database (relational) approach, so that all transactions can be managed by the computerized system.

(2) The Database management system (DBMS) and application programs will be chosen and designed (respectively) to ensure and enhance accurate update of records (delivery).

(3) Network architecture should be implemented with the proposed system to improve data sharing.

(4) All information will be centralized to a central Database.

(5) The data of the central Database can be shared by different departments.

(6) Consistent reports can be produced to provide a better decision support system.

(7) The proposed system should also support Word-Wide-Web applications as new technology permits online transaction.

(8) Maintenance and security features of the system should also be considered seriously to ensure the correctness and protection of the Database.
III. THE PROPOSED SYSTEM

3.1 System Specification

The methodology used to develop this project is based on a well established framework known as the FAST framework methodology for system analysis and design development.

The whole picture that will be described in the project will reflect the work performed at different stages, namely the Survey phase, Study phase, definition phase, Configuration phase (if necessary) for the Analysis tasks and the Design phase, Procurement phase, Construction phase and Delivery phase for the Design tasks.

In order to solve problems occurring from the existing manual system, SIAMLIFT Industries Co., Ltd. Needs an effective Delivery Information System which can effectively support distribution operations and increase customer satisfaction.

The new proposed information system should be as follows:

(1) the system is able to store all necessary information of distribution, delivery and checking the parts delivery process.

(2) The system should minimize the data handling.

(3) The system should reduce required number of documents.

(4) The system should be easy to retrieve and modify the information such as delivery information and cancel part information.

(5) The system should be the main information for delivery procedure in every step to maintain the procedure and qualify the working procedure traceable to the staffs that are responsible in that working process.

(6) The system should allow staff more time to concentrate on delivery procedure with quality and care.
(7) The system should be accurate the delivery records.
(8) The system should provide better security of staff responsible for data.
(9) The system should improve communications among operating departments.
(10) The system should generate timely and comprehensive reports.
(11) The system should provide information for planning and forecasting decisions.
(12) Information could be shared among several users at the same times.

The proposed system is “Delivery Information System” related many aspect of technology infrastructure for building up the system. All of technology defined here based on the technical required to implement the system and delivery working procedure.

The following computing technique required was categorized on the aspect of each computer technology.

3.2 Data Modeling

3.2.1 Context Data Modeling

The first data model will consist of the simplest; called the context model and will represent only the entities and their relationships as shown in Figure 3.1. The entities chosen for the system are the following entities:

Table 3.1. List of Entity.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination_data</td>
<td>Contain all master destination record.</td>
</tr>
<tr>
<td>Package_data</td>
<td>Record all package type for contain the parts and material.</td>
</tr>
</tbody>
</table>
Table 3.1. List of Entity (Continued).

<table>
<thead>
<tr>
<th>Entity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part_data</td>
<td>Collect all part that sender request to deliver.</td>
</tr>
<tr>
<td>Package_Part</td>
<td>Specific on which part contain on each package.</td>
</tr>
<tr>
<td>Deliver_Package</td>
<td>Completely define the package on the bus schedule.</td>
</tr>
<tr>
<td>Site</td>
<td>Mainly to use as the site identifier for delivery data.</td>
</tr>
<tr>
<td>Bus_data</td>
<td>Keep record on all bus type that use in company.</td>
</tr>
<tr>
<td>Bus_schedule</td>
<td>Contain all bus’s schedule to deliver the package.</td>
</tr>
<tr>
<td>Bus_on_Site</td>
<td>Record the bus on each site.</td>
</tr>
</tbody>
</table>

3.2.2 Key-Based Data Modeling

The next data modeling is the elaboration of the Key-Based entity relationship diagram by:

1. Identifying keys (primary keys, foreign keys and etc.) of each entity type.
2. Resolving the two non-specific relationships that appear in the context ERD by introducing Associative Entities.
3. Identifying any generalization hierarchies that might occur. Here, the part, package, and bus can be of different categories, which determine and obvious generalization hierarchy.

This work is shown in Figure 3.2.

3.2.3 Fully Attributed Data Modeling

The next and last step is the elaboration of the fully attributed data model by:

1. Identifying the remaining data attributes.
2. Updating the key-based ERD with those new attributes.
This work is shown in Figure 3.3.

The repository (Data dictionary) is also attached (in the appendix) for better reference and information about data types, entity types and relationships.

3.3 Process Modeling

The objective of the proposed system is to solve the problem in the existing manual Delivery department's delivery center at each site. Then the proposed system have three main objectives as follows:

(1) To solve the communication problem between the Delivery department and the other departments in case of interrupted process or delivery delay and the communication problem within the Delivery department.

(2) To reengineer the existing closed manual system to become the open system that the sender and receiver or customer could track the Delivery departments working procedure.

(3) To reduce the mistaken process in delivery procedure and make the system controllable in keeping all working procedure in step by step for reference in performance evaluation and business process redesign in the future plan.

The section below described the new computerized system that will help the Delivery department's working process to be more manageable, controllable, and completely loop back the checking on each process entire the system.

As SIAMLIFT Industries Co., Ltd., has many departments that co-operating all work together, setting up the new system for one department has to be concerned with the communication linked between the system and other departments as the main criteria. The new systems have to be created in the interface for external input and output for the other departments or the customer to access system and their transactions.
The context diagram of the proposed system as shown in Figure 3.4. describe the external system or persons that have interaction with the Delivery Information System to perform their responsible tasks with the system.

From the Figure 3.4, the system has the following entities:

(1) Customer : existing or potential customer.

(2) Supplier : the person who supply the units.

(3) Warehouse : the store that manage the stock.

(4) Factory : the main production.

The system has four main subsystems and each subsystem has their own task to perform related operations together. After analyzing the process of all interactions within the system and external entities, it can be divided into four subsystems as shown in Figure 3.5.

The first is “Destination subsystem” which performs all tasks related to the RFD (Request for Deliver), confirmation, postpone process, and update delivery schedule in case the sender change the delivery request.

The second is “Part subsystem” which performs all parts procedures that included getting part process, approve part to send process, and reject the delivery part.

The third is “Package subsystem” that includes all process of packaging the parts or materials for delivery such as determine package size and generate list of package in the bus or generate list of parts in the package, and cancel package process plus verifying the package process.

The final subsystem is “Delivery subsystem” which provides the main function of all Delivery department. This subsystem could be categorized in to four main processes composed with transaction process, on-site response process, allocate package process to bus, and manage bus process.
Figure 3.1. Context Data Model.
Figure 3.2. Key-Based Data Model.
Figure 3.3. Fully Attributed Data Model.
Figure 3.4. Context Diagram of Delivery Information System.
Figure 3.5. Function Decomposition Diagram of Delivery Information System.
Figure 3.6. Function Decomposition Diagram of Package Subsystem.
Figure 3.7. Function Decomposition Diagram of Delivery Subsystem.
Figure 3.8. First Level DFD.
Figure 3.9. Second DFD Level Explosion of Process 1.
Figure 3.10. Second DFD Level Explosion of Process 2.
Figure 3.11. Second DFD Level Explosion of Process 3.
Figure 3.12. Second DFD Level Explosion of Process 4.
Figure 3.13. Third DFD Level Explosion of Process 1.
Figure 3.14. Second Third DFD Level Explosion of Process 1.
Figure 3.15. Third DFD Level Explosion of Process 2
Figure 3.16. Second Third DFD Level Explosion of Process 2.
3.1. I Package and Part Process

3.1.1 Determine the Package to use

3.1.2 Process Calculate Amount of Part

3.1.3 Process Create Group of Package

3.1.4 Process Generate Part in Package Report

D9 Package Data

Package Type

Approve Part

Reject Part

Approve Send Part

Approve/Reject Part

Package and Part

Package Create Group of Package

Package and Part Mapping Data

Package for Delivery

Part in Package

Figure 3.17. Third DFD Level Explosion of Process 3.
Figure 3.18. Second Third DFD Level Explosion of Process 3.
Figure 3.19. Third DFD Level Explosion of Process 3.
Figure 3.20. Third DFD Level Explosion of Process 4.
Figure 3.21. Second Third DFD Level Explosion of Process 4.
Figure 3.22. Third DFD Level Explosion of Process 4.
4.4.1 Process Determine and Verify Package to Use

4.4.2 Calculate Amount of Package on Bus Type

4.4.3 Store Package to Bus

4.4.4 Process Generate Package Group for Bus Report

Package Data

Figure 3.23. Fourth Third DFD Level Explosion of Process 4.
3.4 Physical Data Flow Diagram

Physical data Flow diagrams model the information system’s application architectures and processes, because they serve as a general system design or blueprint for subsequent detailed design, prototyping and construction. It is shown in the attached appendix. Physical data flow diagrams are constructed from logical data model, the logical process model and the logical network model and consist of:

1. The network topology DFD that allocates processors and devices to the network and establishes the connectivity between the clients and servers and where users will interact with the processors.

2. The data distribution DFD shows how data will be stored and how the logical DFDs physically stored and how they will be implemented.

3. The process distribution DFD shows the logical event diagrams must be assigned to processors and partitioned accordingly so that each physical DFD corresponds to a design unit for a given business event.

3.5 Database Design

Since data storage is a critical component of most Information Systems, the Database design becomes also a critical success factor to the Information System.

The Database schema (or physical model) will represent the technical implementation of the logical data model. The goal of the database design should ensure a good distribution and a good data replication.

The Database (physical) design will take its source from the logical analysis done during the CS6401 course and more precisely from the logical data modeling in the form of the Entity Relationship Diagram (ERD).

This ERD will therefore serve as the basic input for the Database design and the technique that organizes data attributes such that they are grouped to form stable
flexible and adaptive entities should be enforced. That technique is called “Normalization.”

In this paper, we will then construct the Database of the proposed system up to the third normal form (3NF) by normalizing at the ERD level, and at the same time, the size of the Database will be evaluated (as an indication of the desirable storage required).

(1) The first normal form (1NF) will ensure that no attributes can have repeating groups (no more than one value for single instance of an entity).

(2) The second normal form (2NF) will ensure that 1NF is already performed and all non primary key attributes are dependent on the full primary key and not just part of it.

(2) The third Normal form (3NF) will ensure that 2NF is performed and non primary key attributes are not dependent on any other non primary attributes.

The Database model chosen for the target system is a Relational Database so its structure is defined in terms of Tables, keys, indexes and integrity rules.

The rules for transforming a logical data model into a physical database schema are as follows:

(1) Each entity becomes a Table.

(2) Each attribute becomes a field.

(3) Each primary and secondary key becomes an index into the Table.

(4) Each foreign key implements a possible relationship between instances of the Table.

The ERD (fully attributed data model) has already been depicted in Figure 3.3. and the corresponding database schema as show in the Figure 3.24.

The size needed for the Database storage can be evaluated. The calculations and
size evaluation are summarized in Table 3.2.

Adding a slack capacity buffer of 15% and a 20% margin for further growth of data volume, we can estimate the Database size to approximately 12,150,000 Bytes, that is about 12.5 MB.

### Table 3.2. Database Size Evaluation (for a Period of Three Years).

<table>
<thead>
<tr>
<th>Table</th>
<th>Size (in byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination_data</td>
<td>104 x 5,000 = 520,000</td>
</tr>
<tr>
<td>Part_data</td>
<td>112 x 300,000 = 33,600,000</td>
</tr>
<tr>
<td>Package_data</td>
<td>128 x 500 = 62,000</td>
</tr>
<tr>
<td>Package_Part</td>
<td>70 x 200,000 = 14,000,000</td>
</tr>
<tr>
<td>Deliver_package</td>
<td>20 x 100,000 = 2,000,000</td>
</tr>
<tr>
<td>Site</td>
<td>54 x 3 = 162</td>
</tr>
<tr>
<td>Bus_On_Site</td>
<td>17 x 20 = 340</td>
</tr>
<tr>
<td>Bus_schedule</td>
<td>70 x 7,000 = 490,000</td>
</tr>
<tr>
<td>Bus_data</td>
<td>94 x 50 = 4,700</td>
</tr>
</tbody>
</table>

Database approximate size is 

\[
\sum (\text{Table size}) = 53,677,202 \text{ bytes} + \text{further growth } 20\% \approx 65 \text{ Megabytes}
\]
Figure 3.24. Proposed System's Physical Database Schema.
3.6 Input and Output Design

Since most new applications developed to day rely on a Graphical User Interface (GUI), it is highly desirable to take a close look at this aspect of the system design.

At the same time, outputs produced by the information system present information to the users, managers, stakeholder, system auditor, etc and should be designed as such a visible component of the system.

3.6.1 Input Design

Business Transactions create data.

The key points regarding Input design include the following:

(1) Data capture is the identification of new data to be input.

(2) A Source document is a paper form used to record data that will eventually be input to a computer.

(3) Data entry is the process of translating the source document into a machine readable format.

(4) Data input is the actual entry of data in a machine-readable format into the computer.

Most new applications developed uses Graphical User Interface (GUI). Inputs should be designed as simple as possible and reduce the possibility of faulty inputs.

Input screens for some strategic processes management have been chosen for the purpose system to develop using the Java Swing API.

3.6.2 Output Design

Outputs consist of external and internal outputs. The outputs have to be used by many departments that interact with the Logistic department. The following general principles are important for output design:

(1) Computer outputs must be simple to use.
(2) The timing of computer outputs is important.

(3) The distribution of computer outputs must be sufficient to assist in all relevant system users.

(4) The computer outputs must be acceptable to the system users who will receive them.

3.7 Software Design

In designing the software for the information system to be implemented, two following steps are taken:

(1) Modular design to perform the decomposition of a program into modules. A module is a group of executable instructions with a point of entry and a single point of exit.

(2) Packaging which emphasize on the assembly of data, process, interface and geography design specification for each module.

Structured design was a popular strategy in the 1980ies for determining an optimal modular design for programs.

Realizing a Structure chart is the primary tool used in structured design and software design.

Structure charts are used to graphically depict a modular design of a program, and are constructed on the basis of Data Flow Diagrams according to the following rules:

(1) Transform analysis is an examination of the DFD to divide the processes into those that perform input and editing, those that do processing of data transformation, and those that do output.

During the transformation analysis, the logical DFD will be the source of the operation although it may have to be manipulated or expanded to identify clearly the multifunction process of the DFD.
In such a case, additional processes should be implemented obviously along with additional data access.

The emphasis of the transform analysis will finally show clearly the three different groups of processes:

(a) **Afferent processes** are processes performing input.

(b) **Central transforms** are the process doing the actual processing or transformation of data.

(c) **Efferent processes** refer to processes doing output.

(2) **Transaction analysis** is the examination of the DFD to identify processes that represent transaction centers that are processes that do not perform actual transformation on the incoming data.

Once again, it may arise that the DFD has to be revised to introduce the transaction center along with the corresponding types of process transactions.

Once both transform analysis and transaction analysis have been done, the structure chart can be sketched in the form of a “tree” architecture (top-down and left-right) using the standard Gane & Sarson shapes.

The two measures of quality of structure charts are:

(1) **Coupling** to refer to the level of dependency existing between modules, the lowest the dependency the better (loose coupling).

(2) **Cohesion** is inversely proportion to coupling and thus refers to the degree to which a module’s instruction are functionally related. If loose coupling is achieved, modules will be highly cohesive and therefore the program resulting from the software design will tend to be easier to understand and also to maintain.
Figure 3.25. Revise Third DFD Level of Process 1.
Figure 3.26. Revise Third DFD Level of Process 2.
Figure 3.27. Revise Third DFD Level of Process 3.
Figure 3.28. Revise Third DFD Level of Process 4.
Figure 3.29. Structure Chart of Process 1.
Figure 3.30. Structure Chart of Process 2.
Figure 3.31. Structure Chart of Process 3.
Figure 3.32. Structure Chart of Process 4.
3.8 System Architecture Design

3.8.1 Networking

(1) Cause & requirements

The Delivery Department controls all the three “Distributed Center” subsystem on each site that must communicate with each other all the time for the other site to know the up-coming delivery plan and easier to update the schedule to be the same set of data that help the Delivery department of SIAMLIFT Industries Co., Ltd., to be more manageable at the Head office.

The system must allow the customer or other department that use the Delivery department’s service tracking the delivery process over the local area network (LAN), wide area network (WAN) and dial-up connection from the customer or the management level of the company.

(2) Analyze

The requirement of the Delivery department’s working procedure need a permanent networking connection all the time to communicate and update data of all three sites.

The system must have the connectionless system for the normal phone line dial in customer to check the process of delivery and the Internet connection for the customer at the other regions.

(3) Solution

The most suitable technology that could support all requirements above is the VPN (Virtual Private Network) system. VPN allows the system and other departments on each site to communicate together and the customer could track their product deliver procedure anywhere in the world.
The system needs to have the dial-up connection server for the customer to track the process and manager to verify and approve all the process instantly.

By the plan, the proposed system has three workstations computer at each site. The workstation that pass the test process burn-in 24 hrs. have the following specifications.

3.8.2 Primary Development System Technology

(1) Cause & requirements

The Delivery Department plan to use extensive system that can support the whole different platform and workstation system in the future investment in case some system reduce the price to expand the volume of sales. The proposed system must expand to any of them without the cost of re-writing the new code from ground up.

(2) Analyze

The specific requirement of this system needs to develop the customized application that could run on any different workstation system.

(3) Solution

In 1995, Sun Microsystems invented the new pure object oriented programming language structure that could run on any platform by using Java virtual machine software. Sun opens the computing community to proposed idea and joint develop this programming language to open to the third-party vendor to modify the use of language to match their criteria and problem with no charge like the proprietary technology from the specific vendor.
3.8.3 Primary Database Server

(1) Cause & requirements

The Delivery Department controls all the three “distributed center” subsystem on each site that must communicate with each other all the time and the main criteria is the data that must match with each other all the times.

The system must keep track of all the delivery transaction procedure whether it is complete or not for the estimation process at the end of the year to evaluate the performance and is able to trace for the problem and reason that found in each process to tune up the procedure and reduce unnecessary or redundant tasks in the future.

(2) Analyze

This requirement needs to set up the database server to keeping every transaction and record the data storage to recall to tracking, working and analyze anytime.

(3) Solution

The solution for the Database server that is suitable for the systems which plan to develop the crossable platform language such as Java is the system that scalability and wildly use as standard, which is “Interbase” database server. It is the open-source database server with high performance in the market and not required license.

3.8.4 Mail and Web Technology

(1) Cause & requirements

The Delivery Department controls all the three “Distributed Center” subsystem on each site that must communicate all the time and the main
criteria is the data that must match all the times.

The system must have the easiest way to communicate within the internal organization whether employee or manager stays on any customer's site and to communicate with the customer schedule plan and progress.

(2) Analyze

This requirement needs the standard system that everyone could contact and get information together anywhere including the customer, manager, and employee at the work site.

(3) Solution

Using the Apache web server is required for this project because it's the most stable web server in the world with the open-source that the IBM used as base code for their IBM Websphere web server. For the mail server, the system chooses the SendMail server which is widely use on the internet. Both of them are free license.

3.9 Hardware and Software Requirement

3.9.1 Hardware Requirement

(1) Database server

(a) 450-MHz Pentium III

(b) 512 MB maximum, 4 DIMM slots, 50ns 168-pin EDO

(c) Three 32-bit PCI slots, two full size, one short, 33 MHz, 5 volt

(d) Ethernet/Fast Ethernet, twisted pair standard (10-BaseT and 100-BaseT)

(e) 9.1 GB, 3.5-in. Enhanced IDE HDD, 7200 rpm

(f) 32X-speed CD-ROM
(g) One 3.5-in., 1.44-MB floppy
(h) Solaris 7 operating environment
(i) Network link: NFS™, TCP/IP, IPX/SPX, Solaris™ PC NetLink
(j) One front-access bay for third-party options
(k) UPS 1,000 watt. 220 volt 1000 VA.

(2) PC Workstation
   (a) CPU AMD Duron 800 MHz
   (b) Main board ASUS K7 with VGA on board
   (c) SD-RAM 64 MB
   (d) Hard disk 1000 MB
   (e) 14” Phillips Monitor
   (f) Keyboard / mouse PS/2
   (g) 300 Watt ATX Chassis with cooler fan

(3) Network Adapter
   (a) 3Com Ethernet 10/100 LAN card

(4) Network Peripherals
   (a) Cisco Router 5200 series
   (b) 3Com switching hub

(5) Canon Inkjet Printer

(6) UTP Cable 100 Mbps CAT5

3.9.2 Software Requirement

(1) Borland JBuilder Enterprise edition
(2) Interbase Database server
(3) Apache web server with Tomcat
(4) Send Mail server
3.10 Network Architecture

The Internet has forever changed the way in doing business. An outgrowth of Internet technology and thinking, Virtual Private Networks are transforming the daily method of doing business faster than any other technology. A Virtual Private Network, or VPN, typically uses the Internet as the transport backbone to establish secure links with business partners, extend communications to regional and isolated offices, and significantly decrease the cost of communications for an increasingly mobile workforce.

Basically, VPNs are private network overlays on a public IP network infrastructure such as the Internet. A VPN provides secured connections between several points, sort of an express lane on the information superhighway, where communications travel along the Internet but are encrypted for security.

The effects a VPN can have on an organization are dramatic: sales can be increased, product development can be accelerated, and strategic partnerships can be strengthened in a way never before possible. Prior to the advent of VPNs, the only other option for creating this type of communication was prohibitively expensive and consisted of leasing dedicated lines from the phone company. Internet access is generally local and much less expensive than dedicated Remote Access Server connections.

There are as many types of VPN implementations as there are companies taking advantage of the concept’s benefits, each with its own specific set of technology requirements. However, VPN configurations can be grouped into three primary categories:

(1) Intranet VPNs between internal corporate departments and branch offices.

(2) Remote Access VPNs between a corporation and remote or mobile employees.
(3) Extranet VPNs between a corporation and its strategic partners, customers, and suppliers.

In Intranet VPNs that facilitate secure communications between a company’s internal departments and its branch offices. The primary technology requirements are fast, strong encryption to accommodate the high-speed links present in internal LANs, reliability to ensure the prioritization of mission-critical applications, such as financial systems, sales and customer database management, and document exchange; and ease administration to accommodate the changing requirements of new users, new offices and new applications.

There are a dizzying number of methods for employing VPNs in today's computing environment. The VPN market is populated with point products and incomplete solutions, focusing only on one type of VPN application. Most VPN vendors are offering products that only provide authentication and encryption, leading customers to believe that these two components alone comprise a VPN. However, encryption and authentication alone are inadequate to implement the various types of mission-critical VPNs demanded by companies today.

Standalone VPN products do not usually provide basic access control. In fact, most vendors sell their VPN products as yet one more networking device which customers must manage separately and somehow integrate into their overall security policy. In addition, most leave the Quality of Service or reliability aspect to the service providers, rather than giving users the tools they need to put performance predictability in their own hands. This piecemeal approach leads to possible security threats, because the VPNs are not integrated into an organization’s enterprise security policy, and they provide limited manageability, scalability and interoperability.
Virtual Private Networks provide a relatively easy way to connect remote users and the mobile workforce without losing the ability to connect to common applications and protocols. VPNs offer ways to simplify and consolidate the administration of remote access users. Access is available worldwide via fixed and cellular networks as well as over satellite links. This global accessibility allows employees within an organization the ability to access corporate information anywhere in the world where their service provider has a presence, all for the cost of a local call. An Intranet VPN is able to provide connections between internal corporate departments and branch offices. While an Extranet VPN can provide connections between a corporation and its strategic partners, customers, and suppliers.

The cost reductions of using a Virtual Private Network are tremendous when compared to the standard wide area network (WAN). VPNs eliminate the need for using dedicated or leased lines, which have a high monthly cost especially across large distances. Instead VPNs use a shared public IP network, which eliminates multiple remote access hardware devices. Also, there is savings on telecommunications costs by using the Internet to carry traffic, rather than paying long distance phone charges. Long distance phone charges are reduced with a VPN because a user typically dials a local call to an ISP rather than placing a long distance call directly to the company. Another way to save telecommunications costs by reducing the number of access lines into a corporate site. Finally, VPNs also provide the opportunity to save operational costs by outsourcing the management of remote access equipment to a service provider. This allows a company can get rid of its modem pools and remote access servers. The operational cost savings come from not having to manage those devices.

Security is a very critical part of Virtual Private Networks as it will maintain the privacy of important corporate data. There are several factors that contribute to the
security of the VPNs such as authentication of the sender and receiver, integrity of the data, encryption of the data, and reliability of delivery. To provide security the VPN creates a tunnel through the public Internet from your user to the company network resources. This tunnel is invisible to outsiders and all data passing through it is encrypted. The data is verified at each end of the tunnel to ensure that it is from a legitimate source and has not been tampered with while in transit. Each user is verified against a database of authorized users and access is granted based on their network privileges. Access can be controlled by two-factor security employing both something.

The user has, in the form of a security token, and something the user knows, in the form of a password.

One of the most important factors is authentication, which allows only those who are authorized to get onto the network. This will restrict access to all unauthorized users who’s identity is not verified. This is done by the client PC presenting the user’s credentials to the remote server when connecting to the VPN. Next each user is verified against a database of authorized users and access is granted based on their network privileges. This protects against replay attacks and remote client impersonations. A replay attack is when a third party copies data packets and then plays them back to the remote clients response. A remote client impersonation happens when a third party takes over the actual authenticated connection.

Data Encryption is another very important factor in the security of VPNs. The data that is traveling on the network must be completely unreadable to unauthorized users on the network. This can be done by either symmetric or asymmetric encryption. Symmetric encryption is known as the conventional encryption and uses a private key. This secret key is shared only between the two authorized communicating parties. The sender uses the key to encipher the information by a mathematical operation. The
receiver than uses the same key to decipher the same information. Only the user with
the key is able to change the message back into plain text.

An alternative method to symmetric encryption is using asymmetric or public key
encryption. This uses two different keys for each user, one is a private key know only to
one user and the other is the corresponding public key, that anyone can access. The two
keys are mathematically related to each other by an encryption algorithm so they will
recognize each other. Asymmetric encryption also allows for digital signatures to be
sent on the messages. This is another way to verify the sender’s identity.

Virtual Private Networks also provide many different performance advantages.
First of all there are NSPs in nearly every city in developed countries, which creates a
worldwide presence for VPNs. Second by dialing in though local access throughput is
improved by minimized line noise. Third, mesh redundancy and fault tolerance ensure
continuous reliability. And finally user familiarity to the Internet interface simplifies
training needs.

In addition to these performance advantages an organization using VPNs can add
and delete connections instantaneously, Provide permanent, periodic or temporary
connectivity as needed and integrate third-party users, such as customers and suppliers,
almost effortlessly.
3.11 Network Configuration

Because of the proposed system use the VPN as main link between each Site, So the configuration is based on the ISP (Internet Service Provider) that provide the setup and configuration at the customer system’s location.

The main configuration of VPN is based on the server and router. The most reliable operating system that provide more efficient for VPN is “Linux”. Because it’s built-in the algorithm that specifically support VPN in their kernel. And the most popular router that provide more configuration in routing packet is “Cisco”. Both of the best suitable operating system and router for VPN is listed as main hardware for the proposed system.

There’re two type of VPN protocol, reliable and unreliable protocol. The proposed system will use the reliable protocol such as TCP and GRE that developed by CISCO and supported natively by Linux kernel. To setup the “Network to Network” the router configuration examples for the server look like this:

```bash
up {
    ifconfig "%% 192.168.1.254 pointopoint 192.168.2.254 mtu 1450";
    route "add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.2.254";
    firewall "-I forward -s 192.168.1.0/24 -d 192.168.2.0/24 -j ACCEPT -b";
    program "/sbin/arp -sD 192.168.1.254 eth1 pub";
}

down {
    firewall "-D forward -s 192.168.1.0/24 -d 192.168.2.0/24 -j ACCEPT -b";
    program "/sbin/arp -d 192.168.1.254 -i eth1";
}
```
The configuration above assume that the “eth1” (Ethernet port number 1) is internal network interface card and the “eth0” (Ethernet port number 2) is the external network interface card. The next configuration is the client configuration will look like this:

```
up {
    ifconfig "%%% 192.168.2.254 pointopoint 192.168.1.254 mtu 1450";
    route "add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.254";
    firewall "-I forward -s 192.168.1.0/24 -d 192.168.2.0/24 -j ACCEPT -b";
    program "/sbin/arp -sD 192.168.2.254 eth1 pub";
}

down {
    firewall "-D forward -s 192.168.1.0/24 -d 192.168.2.0/24 -j ACCEPT -b";
    program "/sbin/arp -d 192.168.2.254 -i eth1";
}
```

Because of the VPN categorized the setting in to 2 group, which are “Remote-to-Network” and “Network-to-Network”, then the configuration for the proposed system become the client server configuration. These mean that one site should be config as server and the other site should be config as client to create the virtual tunnel over the Internet.
Figure 3.33. Network Configuration of a Site for Delivery Information System.
Figure 3.34. Diagram Overview of Network Connection of Delivery Information System.
Figure 3.35. Virtual Private Network Configuration of Delivery Information System.
3.12 System Cost Analysis

(1) Cost of Estimated Project

Table 3.3. Estimated Projected Cost, Baht.

<table>
<thead>
<tr>
<th>Cost Items</th>
<th>Description</th>
<th>Amount</th>
<th>Unit Price</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>1. Development Cost:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>1.1 New Hardware:</strong></td>
<td>Sun Fire Server</td>
<td>3</td>
<td>200,000.00</td>
<td>600,000.00</td>
</tr>
<tr>
<td></td>
<td>PC Workstation</td>
<td>6</td>
<td>35,000.00</td>
<td>210,000.00</td>
</tr>
<tr>
<td></td>
<td>Cisco Router 5200 series</td>
<td>3</td>
<td>10,000.00</td>
<td>30,000.00</td>
</tr>
<tr>
<td></td>
<td>3Com Switching hub</td>
<td>3</td>
<td>7,500.00</td>
<td>22,500.00</td>
</tr>
<tr>
<td></td>
<td>100 Mbps CAT5 Cable</td>
<td>3</td>
<td>5,000.00</td>
<td>15,000.00</td>
</tr>
<tr>
<td></td>
<td>Printer</td>
<td>3</td>
<td>20,000.00</td>
<td>60,000.00</td>
</tr>
<tr>
<td></td>
<td>UPS 1000 VA</td>
<td>3</td>
<td>6,000.00</td>
<td>18,000.00</td>
</tr>
<tr>
<td></td>
<td><strong>Subtotal 1:</strong></td>
<td></td>
<td></td>
<td>955,500.00</td>
</tr>
<tr>
<td><strong>1.2 New Software:</strong></td>
<td>Borland Jbuilder 5</td>
<td>1</td>
<td>100,000.00</td>
<td>100,000.00</td>
</tr>
<tr>
<td></td>
<td>Interbase database server</td>
<td>3</td>
<td>10,000.00</td>
<td>30,000.00</td>
</tr>
<tr>
<td></td>
<td>Apache web server</td>
<td>1</td>
<td>10,000.00</td>
<td>10,000.00</td>
</tr>
<tr>
<td></td>
<td>SendMail mail server</td>
<td>1</td>
<td>10,000.00</td>
<td>10,000.00</td>
</tr>
<tr>
<td></td>
<td><strong>Subtotal 2:</strong></td>
<td></td>
<td></td>
<td>150,000.00</td>
</tr>
<tr>
<td><strong>1.3 Implementation Cost:</strong></td>
<td>Set up Cost</td>
<td></td>
<td>200,000.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Personnel Cost</td>
<td></td>
<td>200,000.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Subtotal 3:</strong></td>
<td></td>
<td></td>
<td>400,000.00</td>
</tr>
<tr>
<td></td>
<td><strong>Total Development Cost</strong></td>
<td></td>
<td></td>
<td>1,505,500.00</td>
</tr>
<tr>
<td><strong>2. Operating Cost:</strong></td>
<td><strong>Subtotal 1:</strong></td>
<td></td>
<td></td>
<td>45,500.00</td>
</tr>
<tr>
<td>**2.1 Supplies &amp; Miscellaneous</td>
<td>Stationary</td>
<td></td>
<td>1,000.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Paper</td>
<td></td>
<td>1,500.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Utility</td>
<td></td>
<td>10,000.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Internet and leased-line</td>
<td></td>
<td>30,000.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Miscellaneous</td>
<td></td>
<td>3,000.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Subtotal 2:</strong></td>
<td></td>
<td></td>
<td>75,000.00</td>
</tr>
<tr>
<td></td>
<td><strong>Total Operating Cost</strong></td>
<td></td>
<td></td>
<td>120,500.00</td>
</tr>
<tr>
<td></td>
<td><strong>Total Projected Annual Cost</strong></td>
<td></td>
<td></td>
<td>1,626,000.00</td>
</tr>
</tbody>
</table>
(2) Cost of Manual System

Table 3.4. Manual System Cost Analysis, Baht.

<table>
<thead>
<tr>
<th>Cost items</th>
<th>Years</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Type writer</td>
<td>1,800.00</td>
</tr>
<tr>
<td>Computer</td>
<td>18,000.00</td>
</tr>
<tr>
<td>Total Fix cost</td>
<td>19,800.00</td>
</tr>
<tr>
<td>Salary cost</td>
<td></td>
</tr>
<tr>
<td>manager 3 persons @ 30,000</td>
<td>1,080,000.00</td>
</tr>
<tr>
<td>Staff</td>
<td></td>
</tr>
<tr>
<td>Head office 7 persons @ 7500</td>
<td>630,000.00</td>
</tr>
<tr>
<td>Service 7 persons @ 7500</td>
<td>630,000.00</td>
</tr>
<tr>
<td>Factory 7 persons @ 7500</td>
<td>630,000.00</td>
</tr>
<tr>
<td>Total salary cost</td>
<td>2,970,000.00</td>
</tr>
<tr>
<td>Office Miscellaneous Cost:</td>
<td></td>
</tr>
<tr>
<td>Stationary</td>
<td>18,000.00</td>
</tr>
<tr>
<td>Paper</td>
<td>24,000.00</td>
</tr>
<tr>
<td>Utility</td>
<td>60,000.00</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>10,200.00</td>
</tr>
<tr>
<td>Total annual misc cost</td>
<td>112,200.00</td>
</tr>
<tr>
<td>Total operating cost</td>
<td>3,082,200.00</td>
</tr>
<tr>
<td>Total Manual cost</td>
<td>3,102,000.00</td>
</tr>
</tbody>
</table>

Table 3.5. Five Years Accumulated Manual System Cost, Baht.

<table>
<thead>
<tr>
<th>Year</th>
<th>Total Manual Cost</th>
<th>Accumulated Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3,102,000.00</td>
<td>3,102,000.00</td>
</tr>
<tr>
<td>2</td>
<td>3,410,220.00</td>
<td>6,512,220.00</td>
</tr>
<tr>
<td>3</td>
<td>3,749,262.00</td>
<td>10,261,482.00</td>
</tr>
<tr>
<td>4</td>
<td>4,122,208.20</td>
<td>14,383,690.20</td>
</tr>
<tr>
<td>5</td>
<td>4,532,449.02</td>
<td>18,916,139.22</td>
</tr>
<tr>
<td>Total</td>
<td>18,916,139.22</td>
<td>-</td>
</tr>
</tbody>
</table>
## Table 3.6  Computerized System Cost Analysis, Baht.

<table>
<thead>
<tr>
<th>Cost items</th>
<th>Years</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Development Cost</td>
<td></td>
</tr>
<tr>
<td>Hardware Cost:</td>
<td></td>
</tr>
<tr>
<td>Computer Server Cost</td>
<td>120,000.00</td>
</tr>
<tr>
<td>Workstation Cost</td>
<td>42,000.00</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>29,100.00</td>
</tr>
<tr>
<td>Total Hardware Cost</td>
<td>191,100.00</td>
</tr>
<tr>
<td>Software Cost:</td>
<td></td>
</tr>
<tr>
<td>Software Cost</td>
<td>30,000.00</td>
</tr>
<tr>
<td>Total Software Cost</td>
<td>30,000.00</td>
</tr>
<tr>
<td>Implementation Cost:</td>
<td></td>
</tr>
<tr>
<td>Setup Cost</td>
<td>200,000.00</td>
</tr>
<tr>
<td>Total Implementation Cost</td>
<td>200,000.00</td>
</tr>
<tr>
<td>Personnel Cost:</td>
<td></td>
</tr>
<tr>
<td>Personnel Cost</td>
<td>200,000.00</td>
</tr>
<tr>
<td>Total Personnel Cost</td>
<td>200,000.00</td>
</tr>
<tr>
<td>Maintenance Cost:</td>
<td></td>
</tr>
<tr>
<td>Hardware Maintenance</td>
<td>300,000.00</td>
</tr>
<tr>
<td>Software Maintenance</td>
<td>600,000.00</td>
</tr>
<tr>
<td>Total Development Cost</td>
<td>1,521,100.00</td>
</tr>
</tbody>
</table>

### Operating Cost

| People-Ware Cost                  |             |             |             |             |
| Manager                           | 480,000.00  | 528,000.00  | 580,800.00  | 638,880.00  | 702,768.00  |
| Staff:                            |             |             |             |             |
| Head Office                       | 288,000.00  | 316,800.00  | 348,480.00  | 383,328.00  | 421,660.80  |
| Service                           | 240,000.00  | 264,000.00  | 290,400.00  | 319,440.00  | 351,384.00  |
| Factory                           | 360,000.00  | 369,000.00  | 435,600.00  | 479,160.00  | 527,076.00  |
| Total Monthly Salary Cost         | 1,268,000.00| 1,594,800.00| 1,865,280.00| 2,082,888.00| 2,202,888.00|

### Office Supplies & Miscellaneous Cost

| Stationary                        | Per Annual  |             |             |             |
| Paper                             | Per Annual  |             |             |             |
| Utility                           | Per Annual  |             |             |             |
| Internet Cost                     | Per Annual  |             |             |             |
| Miscellaneous                     | Per Annual  |             |             |             |
| Annual Office Supplies & Miscellaneous | Per Annual |             |             |             |
|                                   | 12,000.00   | 13,200.00   | 14,520.00   | 15,972.00   | 17,569.20   |
|                                   | 18,000.00   | 19,800.00   | 21,780.00   | 23,958.00   | 26,353.80   |
|                                   | 120,000.00  | 132,000.00  | 145,200.00  | 159,720.00  | 175,692.00  |
|                                   | 360,000.00  | 396,000.00  | 435,600.00  | 479,160.00  | 527,076.00  |
|                                   | 36,000.00   | 39,600.00   | 43,560.00   | 47,916.00   | 52,708.00   |
|                                   | 546,000.00  | 600,600.00  | 660,660.00  | 726,726.00  | 799,398.60  |
| Total Operating Cost              | 1,914,000.00| 2,105,400.00| 2,315,940.00| 2,547,534.00| 2,802,287.40|

### Total Computerized System Cost

|                                   |             |             |             |             |
|                                   | 3,435,100.00| 3,226,500.00| 3,437,040.00| 3,668,634.00| 3,923,387.40|
Table 3.7. Five Years Accumulated Computerized Cost, Baht.

<table>
<thead>
<tr>
<th>Year</th>
<th>Total Computerized Cost</th>
<th>Accumulated Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3,435,100.00</td>
<td>3,435,100.00</td>
</tr>
<tr>
<td>2</td>
<td>3,226,500.00</td>
<td>6,661,600.00</td>
</tr>
<tr>
<td>3</td>
<td>3,437,040.00</td>
<td>10,098,640.00</td>
</tr>
<tr>
<td>4</td>
<td>3,668,634.00</td>
<td>13,767,274.00</td>
</tr>
<tr>
<td>5</td>
<td>3,923,387.40</td>
<td>17,690,661.40</td>
</tr>
<tr>
<td>Total</td>
<td>17,690,661.40</td>
<td></td>
</tr>
</tbody>
</table>

(4) The Comparison of the System Costs between Computerized System and Manual System

Table 3.8. The Comparison of the System Costs, Baht.

<table>
<thead>
<tr>
<th>Year</th>
<th>Accumulated Manual Cost</th>
<th>Accumulated Computerized Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3,102,000.00</td>
<td>3,435,100.00</td>
</tr>
<tr>
<td>2</td>
<td>6,512,220.00</td>
<td>6,661,600.00</td>
</tr>
<tr>
<td>3</td>
<td>10,261,482.00</td>
<td>10,098,640.00</td>
</tr>
<tr>
<td>4</td>
<td>14,383,690.20</td>
<td>13,767,274.00</td>
</tr>
<tr>
<td>5</td>
<td>18,916,139.22</td>
<td>17,690,661.40</td>
</tr>
</tbody>
</table>
Table 3.9. Payback Analysis of Logistic System, Baht.

<table>
<thead>
<tr>
<th>Cost Items</th>
<th>Years</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>System and Development cost</td>
<td>1,505,500.00</td>
</tr>
<tr>
<td>Operation &amp; Maintenance cost</td>
<td>0</td>
</tr>
<tr>
<td>Discount factors for 12%</td>
<td>1.00</td>
</tr>
<tr>
<td>Time-adjusted costs (adjusted to present value)</td>
<td>1,505,500.00</td>
</tr>
<tr>
<td>Cumulative time-adjusted costs over lifetime</td>
<td>1,505,500.00</td>
</tr>
<tr>
<td>Benefits derived from operation of new system</td>
<td>0</td>
</tr>
<tr>
<td>Discount factors for 12%</td>
<td>1.00</td>
</tr>
<tr>
<td>Time-adjusted costs (adjusted to present value)</td>
<td>0</td>
</tr>
<tr>
<td>Cumulative time-adjusted benefits over lifetime</td>
<td>0</td>
</tr>
<tr>
<td>Cumulative lifetime time-adjusted cost + benefit</td>
<td>-1,505,500.00</td>
</tr>
</tbody>
</table>
Table 3.10. Net Present Value and ROI of Logistic System, Baht.

<table>
<thead>
<tr>
<th>Cost Items</th>
<th>Year 0</th>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
<th>Year 4</th>
<th>Year 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development cost</td>
<td>1,505,500.00</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Operation &amp; Maintenance cost</td>
<td>0</td>
<td>120,500.00</td>
<td>132,550.00</td>
<td>145,805.00</td>
<td>160,385.50</td>
<td>176,424.05</td>
</tr>
<tr>
<td>Discount factors for 12%</td>
<td>1.00</td>
<td>0.89</td>
<td>0.80</td>
<td>0.71</td>
<td>0.64</td>
<td>0.57</td>
</tr>
<tr>
<td>Present Value of annual costs:</td>
<td>1,505,500.00</td>
<td>107,606.50</td>
<td>105,642.35</td>
<td>103,813.16</td>
<td>102,005.18</td>
<td>100,032.44</td>
</tr>
<tr>
<td>Total present value of lifetime costs:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2,024,599.62</td>
</tr>
<tr>
<td>Benefits derived from operation of new system</td>
<td>0</td>
<td>1,168,200.00</td>
<td>1,285,020.00</td>
<td>1,413,522.00</td>
<td>1,554,874.20</td>
<td>1,710,361.62</td>
</tr>
<tr>
<td>Discount factors for 12%</td>
<td>1.00</td>
<td>0.89</td>
<td>0.80</td>
<td>0.71</td>
<td>0.64</td>
<td>0.57</td>
</tr>
<tr>
<td>Present Value of annual costs:</td>
<td>0</td>
<td>1,043,202.60</td>
<td>1,024,160.94</td>
<td>1,006,427.66</td>
<td>988,899.99</td>
<td>969,775.04</td>
</tr>
<tr>
<td>Total present value of lifetime costs:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5,032,466.23</td>
</tr>
</tbody>
</table>

Net Present Value of New System                        |   |   |   |   |   | 3,007,866.61 |

Return on Investment (ROI) of new system                  |   |   |   |   |   | 148.57
Figure 3.37. Cost Comparison between Existing System and Proposed System.
Figure 3.38. Payback Analysis of Delivery Information System.
IV. PROJECT IMPLEMENTATION

4.1 Overview of Project Implementation

System Implementation is the Construction of the new system and the delivery of that system into day-to-day operation. To smoothly convert from the old system to the new system, the following objectives must be accomplished:

(1) Conduct a system test to ensure that the new system works properly.

(2) Prepare a conversion plan to provide a smooth transition to the new system.

(3) Install databases to be used by the new system.

(4) Provide training and documentation for individuals who will be using the new system.

(5) Convert from the old system to the new system and evaluate the project and final system.

4.2 Programming

The database application program, Java programming language, is used to manage the database of SIAMLIFT Industries Co., Ltd. The concept of Normalization is applied to prepare a data model for implementation as a simple, nonredundant, flexible, and adaptable database.

4.3 Testing

The goals of testing are:

(1) To force a program to work incorrectly.

(2) To discover the causes of these errors.

(3) To revise the program code to eliminate errors.
Testing means to turn up heretofore unknown problems, not to demonstrate the perfection of programs, manuals or equipment. Testing can be done on the following ways:

1. System performance. Is the throughput and response time for processing adequate to meet a normal processing workload? If not, some programs may have to be rewritten to improve efficiency or processing hardware may have to be replaced or upgraded to handle the additional workload.

2. Peak workload processing performance. Can the system handle the workload during peak processing periods? If not, we may have to improve hardware or software to increase efficiency.

3. Human engineering test. Is the system as easy to learn and use as anticipated? If not, is it adequate?

4. Methods and procedures test. During conversion, the methods and procedures for the new system will be put to their first real test. Methods and procedures may have to be modified if they prove to be awkward and inefficient from the end-users’ standpoint.

5. Backup and recovery testing. To test for recovering data, whenever there is a problem about data such as loss of data.

4.4 Conversion

Conversion is a process of converting from the existing to the new production information system. Some commonly used strategies include:

1. Abrupt cut-over. On a specific date, the old system is terminated and the new system is placed into operation.
(2) Parallel conversion. Under this approach, both the old and new systems are operated for sometime. This is done to ensure that all major problems in the new system have been solved before the old system is discarded.

(3) Location conversion. When the same system will be used in numerous geographical locations, it is usually converted at one location. As soon as that site has approved the system, it can be farmed to the other sites.

(4) Staged conversion is based on the version concept. Each successive version of the new system is converted as it is developed.

Conversion is more than just starting to use the new system. It includes the creation of all required master and transaction files, establishing backup copies of master file and database, and converting tested programs to operating status.

Normally, the existing system is manual, so abrupt cut-over conversion system is not appropriate. Because clerks are not familiar with computerized system and abrupt cut-over is a high-risk approach, if the system is in failure, all work will stop.

The appropriate strategy for SIAMLIFT Industries Co., Ltd., is parallel conversion. The front office information system is operated side by side with the old one to ensure that data will not be lost if problem arises.

4.5 Training

Systems can succeed or fail because of the way they are operated and used. Therefore, the quality of training received by the personnel involved with the system in various capacities helps or hinders, and may even prevent, the successful implementation of an information system. The users must be trained to use equipment and to follow the procedures required of the new system.

The objectives of training depend on requirements of user's jobs. For SIAMLIFT Industries Co., Ltd., the manager needs a broad view of how the new system will help
him more productively in order to fulfill his responsibilities, so his training focuses heavily on the information that the system produces. While the front office staffs and the service staffs need to know how to run the program and backup the system, they must know in detail how to enter data, respond to error messages, and call up routines that will print reports excepting statistical report.
V. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The Delivery Information System is the system that uses SIAMLIFT Industries Co., Ltd., as a case study. Deliver System of SIAMLIFT Industries Co., Ltd., uses a manual operation which has many weaknesses. Therefore, the new Delivery Information System is proposed to replace the old one. A system, which automated the delivery and get parts processes offers users, a great deal of benefit, compared to the manual system. Although the initial cost is high, it will reach the break-even point in 2 months operation. Then the cumulative cost of the proposed system will be less than the cumulative cost of the existing manual system. The computerized system also provides a lot of intangible benefits, in term of data accuracy, efficiency and control, to the organization.

Analysis of the existing system and defining the current problems and areas for improvement is the first step. Then interviewing the customer for the customer requirements, developing system design that includes database design, input design and output design. Java programming language technology is selected for Distributed Delivery Information system application program because it is efficient to manage the database and has a very user-friendly customizable interface. The project provided cost and benefit analysis to compare costs of the existing system and the new proposed system. Before the proposed system is used, it has been tested to ensure that it is free from errors. The users of the new computerized system have also been trained.

Table 5.1. shows the time spent on each process of the proposed system compared with the existing system. It shows that each process of the proposed system spends less time than each process of the existing system which has to pass many manual work
steps. This can be explained as the proposed system is more effective and efficient than the existing system.

<table>
<thead>
<tr>
<th>Process</th>
<th>Existing System</th>
<th>Proposed System</th>
<th>Faster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query destination</td>
<td>3 mins.</td>
<td>0.52 mins.</td>
<td>2.48 mins.</td>
</tr>
<tr>
<td>Check parts</td>
<td>12 mins.</td>
<td>0.78 mins.</td>
<td>11.2 mins.</td>
</tr>
<tr>
<td>Decision on delivery</td>
<td>30 mins.</td>
<td>0.85 mins.</td>
<td>29.1 mins.</td>
</tr>
<tr>
<td>Manage package</td>
<td>20 mins.</td>
<td>0.92 mins.</td>
<td>19.0 mins.</td>
</tr>
<tr>
<td>Manage bus</td>
<td>25 mins.</td>
<td>0.88 mins.</td>
<td>24.1 mins.</td>
</tr>
<tr>
<td>Allocate package to bus</td>
<td>10 mins.</td>
<td>0.73 mins.</td>
<td>9.2 mins.</td>
</tr>
<tr>
<td>Enquiry Process</td>
<td>5 mins.</td>
<td>0.91 mins.</td>
<td>4.0 mins.</td>
</tr>
</tbody>
</table>

(1) Query destination process

The computerized proposed system is able to store all necessary information of destination. It is also easy to retrieve and modify the send parts destination information, so it fastens query procedure.

(2) Check parts process

The proposed system reduces the volume of paperwork, so carrying out the parts and assigning to acceptable send parts list are more efficient and more comfortable.

(3) Decision of delivery process

The proposed system is able to retrieve the decision on send parts easily and produce receipts quickly. Therefore, check out process will be
done much faster.

(4) Manage package process

The proposed system is able to manage package with more precision because the process is more quick to retrieve data and size of package and estimate the amount of parts in the package.

(5) Manage bus process

The proposed system could retrieve the bus schedule in real-time. This made the managed bus time more accurate in booking, canceling and instantly booking and canceling at anytime.

(6) Allocate package to bus

The proposed system has the package management functions that allow the staff to manage the package to the bus more efficiently and reduce the bus delivery times.

(7) Enquiry Process

The proposed system has well integrated information and easy commands so the users can inquire the information faster than the existing manual system.

5.2 Recommendations

This project represents the development of the computerized Delivery Information System for SIAMLIFT Industries Co., Ltd. The company has never used a computerized system before. Most of its clerks lack computer experience so they are not familiar with this kind of change. The company is recommended to encourage the clerks to realize the importance of information system. In any step of system development, the clerks should have participation. The reasons are that they will gradually be familiar with the new system and have a chance to know
misunderstandings and miscommunications between developers and users. So both sides can tune or correct any mistakes before it will be too late.

Another important thing is the security. Passwords must be assigned and updated in time for new users and immediately deleted for resigned users. A more sophisticated control and security procedure should be redesigned when the organization expanded. In addition, the system developed in this project still needs the ongoing development, revision and modification similar to any other software system to meet the future user requirements.

Further enhancement of this system should be done step by step and start from identifying the necessity and the problems that occur in another system such as Inventory management information system. The Delivery information system should be developed further in the next phase especially on-line delivery system.
APPENDIX A
DATA DICTIONARY
Table A.1. Structure of Bus_Data Table.

<table>
<thead>
<tr>
<th>No.</th>
<th>Field Name</th>
<th>Field Type</th>
<th>Index</th>
<th>Unique</th>
<th>Nullable</th>
<th>Foreign Key to Table</th>
<th>Check</th>
<th>Key Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bus_type_id</td>
<td>Int(5)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td>Primary Key</td>
</tr>
<tr>
<td>2</td>
<td>Bus_type_name</td>
<td>Varchar (50)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>3</td>
<td>Bus_type_weight</td>
<td>Float(5)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>4</td>
<td>Bus_type_width</td>
<td>Float(5)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>5</td>
<td>Bus_type_height</td>
<td>Float(5)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>6</td>
<td>Bus_type_depth</td>
<td>Float(5)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>7</td>
<td>Bus_type_space_width</td>
<td>Float(5)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>8</td>
<td>Bus_type_space_height</td>
<td>Float(5)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>9</td>
<td>Bus_type_space_depth</td>
<td>Float(5)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>10</td>
<td>Bus_type_max_load</td>
<td>Float(5)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>11</td>
<td>Bus_type_min_speed</td>
<td>Float(5)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>12</td>
<td>Bus_type_max_speed</td>
<td>Float(5)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
</tbody>
</table>
Table A.2. Structure of Package_Data Table.

<table>
<thead>
<tr>
<th>No.</th>
<th>Field Name</th>
<th>Field Type</th>
<th>Index</th>
<th>Unique</th>
<th>Nullable</th>
<th>Foreign Key to Table</th>
<th>Check</th>
<th>Key Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Package_type_id</td>
<td>Int(5)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td>Primary Key</td>
</tr>
<tr>
<td>2</td>
<td>Package_type_name</td>
<td>Varchar (50)</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>3</td>
<td>Package_type_weight</td>
<td>Float(5)</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>4</td>
<td>Package_type_width</td>
<td>Float(5)</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>5</td>
<td>Package_type_height</td>
<td>Float(5)</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>6</td>
<td>Package_type_depth</td>
<td>Float(5)</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>7</td>
<td>Package_type_space_width</td>
<td>Float(5)</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>8</td>
<td>Package_type_space_height</td>
<td>Float(5)</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>9</td>
<td>Package_type_space_depth</td>
<td>Float(5)</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>10</td>
<td>Package_type_comment</td>
<td>Varchar (50)</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
</tbody>
</table>
### Table A.3. Structure of Part_Data Table.

<table>
<thead>
<tr>
<th>No.</th>
<th>Field Name</th>
<th>Field Type</th>
<th>Index</th>
<th>Unique</th>
<th>Nullable</th>
<th>Foreign Key to Table</th>
<th>Check</th>
<th>Key Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Part_id</td>
<td>Int(5)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Site</td>
<td></td>
<td>Primary Key</td>
</tr>
<tr>
<td>2</td>
<td>Site_id</td>
<td>Int(5)</td>
<td></td>
<td>N</td>
<td></td>
<td>Site</td>
<td></td>
<td>Foreign Key</td>
</tr>
<tr>
<td>3</td>
<td>Destination_id</td>
<td>Int(5)</td>
<td></td>
<td>N</td>
<td></td>
<td>Destination_data</td>
<td></td>
<td>Foreign Key</td>
</tr>
<tr>
<td>4</td>
<td>Part_name</td>
<td>Varchar (50)</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>5</td>
<td>Part_comment</td>
<td>Varchar (50)</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
</tbody>
</table>

### Table A.4. Structure of Package_Part Table.

<table>
<thead>
<tr>
<th>No.</th>
<th>Field Name</th>
<th>Field Type</th>
<th>Index</th>
<th>Unique</th>
<th>Nullable</th>
<th>Foreign Key to Table</th>
<th>Check</th>
<th>Key Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Package_no</td>
<td>Int(5)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Package_data</td>
<td></td>
<td>Primary Key</td>
</tr>
<tr>
<td>2</td>
<td>Package_type_id</td>
<td>Int(5)</td>
<td></td>
<td>N</td>
<td></td>
<td>Package_data</td>
<td></td>
<td>Foreign Key</td>
</tr>
<tr>
<td>3</td>
<td>Part_id</td>
<td>Int(5)</td>
<td></td>
<td>N</td>
<td></td>
<td>Part_data</td>
<td></td>
<td>Foreign Key</td>
</tr>
<tr>
<td>4</td>
<td>Packaging_date</td>
<td>DateTime</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>5</td>
<td>Package_comment</td>
<td>Varchar (50)</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
</tbody>
</table>
Table A.5. Structure of Bus_schedule Table.

<table>
<thead>
<tr>
<th>No.</th>
<th>Field Name</th>
<th>Field Type</th>
<th>Index</th>
<th>Unique</th>
<th>Nullable</th>
<th>Foreign Key to Table</th>
<th>Check</th>
<th>Key Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bus_schedule_id</td>
<td>Int(5)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td>Primary Key</td>
</tr>
<tr>
<td>2</td>
<td>Bus_on_site_id</td>
<td>Int(5)</td>
<td></td>
<td></td>
<td>N</td>
<td>Bus_on_site</td>
<td></td>
<td>Foreign Key</td>
</tr>
<tr>
<td>3</td>
<td>Deliver_destination</td>
<td>Varchar (50)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>4</td>
<td>Deliver_date</td>
<td>DateTime</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
</tbody>
</table>

Table A.6. Structure of Bus_on_site Table.

<table>
<thead>
<tr>
<th>No.</th>
<th>Field Name</th>
<th>Field Type</th>
<th>Index</th>
<th>Unique</th>
<th>Nullable</th>
<th>Foreign Key to Table</th>
<th>Check</th>
<th>Key Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bus_on_site_id</td>
<td>Int(5)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td>Primary Key</td>
</tr>
<tr>
<td>2</td>
<td>Site_id</td>
<td>Int(5)</td>
<td></td>
<td></td>
<td>N</td>
<td>Site</td>
<td></td>
<td>Foreign Key</td>
</tr>
<tr>
<td>3</td>
<td>Bus_type_id</td>
<td>Int(5)</td>
<td></td>
<td></td>
<td>N</td>
<td>Bus_data</td>
<td></td>
<td>Foreign Key</td>
</tr>
<tr>
<td>4</td>
<td>Bus_maintenance_times</td>
<td>Int(5)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>5</td>
<td>Bus_ready_to_use</td>
<td>Bit</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
</tbody>
</table>
Table A.7. Structure of Deliver_Package Table.

<table>
<thead>
<tr>
<th>No.</th>
<th>Field Name</th>
<th>Field Type</th>
<th>Index</th>
<th>Unique</th>
<th>Nullable</th>
<th>Foreign Key to Table</th>
<th>Check</th>
<th>Key Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Deliver_package_no</td>
<td>Int(5)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td>Primary Key</td>
</tr>
<tr>
<td>2</td>
<td>Bus_schedule_id</td>
<td>Int(5)</td>
<td></td>
<td></td>
<td>N</td>
<td>Bus_schedule</td>
<td></td>
<td>Foreign Key</td>
</tr>
<tr>
<td>3</td>
<td>Package_no</td>
<td>Int(5)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>4</td>
<td>Install_on_bus_date</td>
<td>DateTime</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
</tbody>
</table>

Table A.8. Structure of Destination_data Table.

<table>
<thead>
<tr>
<th>No.</th>
<th>Field Name</th>
<th>Field Type</th>
<th>Index</th>
<th>Unique</th>
<th>Nullable</th>
<th>Foreign Key to Table</th>
<th>Check</th>
<th>Key Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Destination_id</td>
<td>Int(5)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td>Primary Key</td>
</tr>
<tr>
<td>2</td>
<td>Destination_address</td>
<td>Varchar (50)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td>3</td>
<td>Destination_comment</td>
<td>Varchar (50)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
</tbody>
</table>

Table A.9. Structure of Site Table.

<table>
<thead>
<tr>
<th>No.</th>
<th>Field Name</th>
<th>Field Type</th>
<th>Index</th>
<th>Unique</th>
<th>Nullable</th>
<th>Foreign Key to Table</th>
<th>Check</th>
<th>Key Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Site_id</td>
<td>Int(5)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td>Primary Key</td>
</tr>
<tr>
<td>2</td>
<td>Site_name</td>
<td>Varchar (50)</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td>Attribute</td>
</tr>
</tbody>
</table>
APPENDIX B
DATABASE DESIGN
**Bus_data**

Composition:
- Bus_type_id: Integer 4
- Bus_type_name: Unicode VarChar
- Bus_type_weight: Float
- Bus_type_width: Float
- Bus_type_height: Float
- Bus_type_depth: Float
- Bus_type_space_width: Float
- Bus_type_space_height: Float
- Bus_type_space_depth: Float
- Bus_type_max_load: Float
- Bus_type_min_speed: Float
- Bus_type_max_speed: Float

Primary Key:
- Index Name: Generated by VAW
- Column(s): Bus_type_id [ ASC ]

Location:
- ER Diagram

Attached relationships on ER Diagram:
- stand by MIN: 1 MAX: many
  - Bus_on_Site

Date Last Altered: 24/10/2001  Date Created: 1/10/2001

**Bus_on_Site**

Composition:
- Site_id: Integer 4
- Bus_type_id: Integer 4
- Bus_maintainance_times: Integer 4
- Bus_ready_to_use: Bit

Primary Key:
- Index Name: Generated by VAW
- Column(s): Site_id [ ASC ]
- Bus_type_id [ ASC ]

Foreign Key(s):
- Bus_data 'stand by' Bus_on_Site
  - Bus_type_id -> Bus_type_id
- On Delete Restrict
- On Update Restrict
- On Insert of Child Row Restrict
- Site 'has' Bus_on_Site
  - Site_id -> Site_id
- On Delete Restrict
- On Update Restrict
- On Insert of Child Row Restrict

Location:
- ER Diagram

Attached relationships on ER Diagram:
- [ has ] MIN: 1 MAX: 1
Site
  [ stand by ] MIN: 1 MAX: 1
Bus_data
  set on MIN: 1 MAX: many
Bus_schedule

Date Last Altered: 24/10/2001 Date Created: 1/10/2001

Bus_schedule

Entity

Composition:
  Bus_schedule_id : Integer 4
  Site_id : Integer 4
  Bus_type_id : Integer 4
  Deliver_destination : Unicode VarChar
  Deliver_date : DateTime

Primary Key:
  Index Name: Generated by VAW
  Column(s): Bus_schedule_id [ ASC ]

Foreign Key(s):
  Bus_on_Site 'set on' Bus_schedule
    Site_id -> Site_id
    Bus_type_id -> Bus_type_id
  On Delete Restrict
  On Update Restrict
  On Insert of Child Row Restrict

Location:
  ER Diagram

Attached relationships on ER Diagram:
  [ set on ] MIN: 1 MAX: 1
  Bus_on_Site deliver MIN: 1 MAX: many
  Deliver_package

Date Last Altered: 24/10/2001 Date Created: 1/10/2001

Deliver_package

Entity

Composition:
  Deliver_package_no : Integer 4
  Bus_schedule_id : Integer 4
  Package_no : Integer 4
  Install_on_bus_date : DateTime

Primary Key:
  Index Name: Generated by VAW
  Column(s): Deliver_package_no [ ASC ]

Foreign Key(s):
  Package_Part 'install on' Deliver_package
    Package_no -> Package_no
  On Delete Restrict
  On Update Restrict
  On Insert of Child Row Restrict
  Bus_schedule 'deliver' Deliver_package
Bus schedule -> Bus_schedule_id
On Delete Restrict
On Update Restrict
On Insert of Child Row Restrict

Location:
ER Diagram
Attached relationships on ER Diagram:
[ install on ] MIN: 1 MAX: 1
[ deliver ] MIN: 1 MAX: 1

---

Destination data

Entity
Composition:
Destination_id : Integer 4
Destination_address : Unicode VarChar
Destination_comment : Unicode VarChar

Primary Key:
Index Name: Generated by VAW
Column(s): Destination_id [ ASC ]

Location:
ER Diagram
Attached relationships on ER Diagram:
has MIN: 1 MAX: many

Date Last Altered: 23/10/2001 Date Created: 23/10/2001
---

Package data

Entity
Composition:
Package_type_id : Integer 4
Package_type_name : Unicode VarChar
Package_type_width : Float
Package_type_height : Float
Package_type_depth : Float
Package_type_space_width : Float
Package_type_space_height : Float
Package_type_space_depth : Float
Package_type_comment : Unicode VarChar

Primary Key:
Index Name: Generated by VAW
Column(s): Package_type_id [ ASC ]

Location:
ER Diagram
Attached relationships on ER Diagram:
contain MIN: 1 MAX: many

Date Last Altered: 24/10/2001 Date Created: 1/10/2001
---
Package_Part

Composition:
- Package_no : Integer 4
- Package_type_id : Integer 4
- Part_id : Integer 4
- Packaging_date : DateTime
- Package_comment : Unicode VarChar

Primary Key:
- Index Name: Generated by VAW
- Column(s): Package_no [ ASC ]

Foreign Key(s):
- Part_data 'stored in' Package_Part
  - Part_id -> Part_id
  - On Delete Restrict
  - On Update Restrict
  - On Insert of Child Row Restrict
- Package_data 'contain' Package_Part
  - Package_type_id -> Package_type_id
  - On Delete Restrict
  - On Update Restrict
  - On Insert of Child Row Restrict

Location:
- ER Diagram
  - Attached relationships on ER Diagram:
    - [ contain ] MIN: 1 MAX: 1
      - Package_data
        - [ stored in ] MIN: 1 MAX: 1
        - Part_data
          - install on MIN: 1 MAX: 1
          - Deliver_package

Part_data

Composition:
- Part_id : Integer 4
- Site_id : Integer 4
- Destination_id : Integer 4
- Part_name : Unicode VarChar
- Part_comment : Unicode VarChar

Primary Key:
- Index Name: Generated by VAW
- Column(s): Part_id [ ASC ]

Foreign Key(s):
- Site 'has' Part_data
  - Site_id -> Site_id
  - On Delete Restrict
  - On Update Restrict
  - On Insert of Child Row Restrict
- Destination_data 'has' Part_data
Destination id -> Destination id
On Delete Restrict
On Update Restrict
On Insert of Child Row Restrict

Location:
ER Diagram
Attached relationships on ER Diagram:
[ has ] MIN: 1 MAX: 1
Site
[ has ] MIN: 1 MAX: 1
Destination_data
stored in MIN: 1 MAX: 1
Package_Part

Date Last Altered: 24/10/2001 Date Created: 23/10/2001

Site
Composition:
  Site_id : Integer 4
  Site_name : Unicode VarChar
Primary Key:
  Index Name: Generated by VAW
  Column(s): Site_id [ ASC ]

Location:
ER Diagram
Attached relationships on ER Diagram:
  has MIN: 1 MAX: many
  Part_data
  has MIN: 1 MAX: many
  Bus_on_Site

Date Last Altered: 24/10/2001 Date Created: 1/10/2001

Bus_on_Site
Composition:
  Site_id : Integer 4
  Bus_type_id : Integer 4
  Bus_maintainance_times : Integer 4
  Bus_ready_to_use : Bit
Primary Key:
  Index Name: Generated by VAW
  Column(s): Site_id [ ASC ]
  Bus_type_id [ ASC ]

Foreign Key(s):
  Bus_data 'stand by' Bus_on_Site
    Bus_type_id -> Bus_type_id
  On Delete Restrict
  On Update Restrict
  On Insert of Child Row Restrict
  Site 'has' Bus_on_Site
    Site_id -> Site_id
On Delete Restrict
On Update Restrict
On Insert of Child Row Restrict
Location:

ER Diagram
Attached relationships on ER Diagram:

- [ has ]  MIN: 1  MAX: 1
  Site
- [ stand by ]  MIN: 1  MAX: 1
  Bus_data
  set on  MIN: 1  MAX: many
  Bus_schedule

Date Last Altered: 24/10/2001  Date Created: 1/10/2001
Figure C.1. Bus Data Editor input screen of Delivery Information System.
Figure C.2. Destination Data Editor input screen of Delivery Information System.
Figure C.3. Package Data Editor input screen of Delivery Information System.
Figure C.4. Delivery Destination Data input screen of Delivery Information System.
Figure C.5. Destination Data Selective screen of Delivery Information System.
Figure C.6. Destination Data Search screen of Delivery Information System.
Figure C.7. Main Delivery input screen of Delivery Information System.
Figure C.8. Delivery History Report screen of Delivery Information System.
figure D.1. Receipt report output of Delivery Information System.
APPENDIX E

SQL DATABASE SCHEMA CODE
CREATE TABLE BUS_DATA
(
    BUS_TYPE_ID INTEGER NOT NULL,
    BUS_TYPE_NAME VARCHAR(100) NOT NULL,
    BUS_TYPE_WEIGHT FLOAT(5) NOT NULL,
    BUS_TYPE_WIDTH FLOAT(5) NOT NULL,
    BUS_TYPE_HEIGHT FLOAT(5) NOT NULL,
    BUS_TYPE_DEPTH FLOAT(5) NOT NULL,
    BUS_TYPE_SPACE_WIDTH FLOAT(5) NOT NULL,
    BUS_TYPE_SPACE_HEIGHT FLOAT(5) NOT NULL,
    BUS_TYPE_SPACEDEPTH FLOAT(5) NOT NULL,
    BUS_TYPE_MAX_LOAD FLOAT(5) NOT NULL,
    BUS_TYPE_MIN_SPEED FLOAT(5) NOT NULL,
    BUS_TYPE_MAX_SPEED FLOAT(5) NOT NULL,
    PRIMARY KEY (BUS_TYPE_ID)
)

CREATE TABLE BUS_ON_SITE
(
    BUS_ON_SITE_ID INTEGER NOT NULL,
    SITE_ID INTEGER NOT NULL,
    BUS_TYPE_ID INTEGER NOT NULL,
    BUS_MAINTAINANCE_TIMES INTEGER NOT NULL,
    BUS_READY_TO_USE RAW(1) NOT NULL,
    PRIMARY KEY (BUS_ON_SITE_ID)
)

CREATE TABLE BUS_SCHEDULE
(
    BUS_SCHEDULE_ID INTEGER NOT NULL,
    BUS_ON_SITE_ID INTEGER NOT NULL,
    DELIVER_DESTINATION VARCHAR(100) NOT NULL,
    DELIVER_DATE DATE NOT NULL,
    PRIMARY KEY (BUS_SCHEDULE_ID)
)

CREATE TABLE DELIVER_PACKAGE
(
    DELIVER_PACKAGE_NO INTEGER NOT NULL,
    BUS_SCHEDULE_ID INTEGER NOT NULL,
    PACKAGE_NO INTEGER NOT NULL,
    INSTALL_ON_BUS_DATE DATE NOT NULL,
    PRIMARY KEY (DELIVER_PACKAGE_NO)
)

CREATE TABLE DESTINATION_DATA
(
CREATE TABLE PACKAGE_DATA
(  PACKAGE_TYPE_ID INTEGER NOT NULL,
  PACKAGE_TYPE_NAME VARCHAR(100) NOT NULL,
  PACKAGE_TYPE_WIDTH FLOAT(5) NOT NULL,
  PACKAGE_TYPE_HEIGHT FLOAT(5) NOT NULL,
  PACKAGE_TYPE_DEPTH FLOAT(5) NOT NULL,
  PACKAGE_TYPE_SPACE_WIDTH FLOAT(5) NOT NULL,
  PACKAGE_TYPE_SPACE_HEIGHT FLOAT(5) NOT NULL,
  PACKAGE_TYPE_SPACE_DEPTH FLOAT(5) NOT NULL,
  PACKAGE_TYPE_COMMENT VARCHAR(100),
  PRIMARY KEY (PACKAGE_TYPE_ID)
)

CREATE TABLE PACKAGE_PART
(  PACKAGE_NO INTEGER NOT NULL,
  PACKAGE_TYPE_ID INTEGER NOT NULL,
  PART_ID INTEGER NOT NULL,
  PACKAGING_DATE DATE NOT NULL,
  PACKAGE_COMMENT VARCHAR(100) NOT NULL,
  PRIMARY KEY (PACKAGE_NO)
)

CREATE TABLE PART_DATA
(  PART_ID INTEGER NOT NULL,
  SITE_ID INTEGER NOT NULL,
  DESTINATION_ID INTEGER NOT NULL,
  PART_NAME VARCHAR(100) NOT NULL,
  PART_COMMENT VARCHAR(100) NOT NULL,
  PRIMARY KEY (PART_ID)
)

CREATE TABLE SITE
(  SITE_ID INTEGER NOT NULL,
  SITE_NAME VARCHAR(100) NOT NULL,
  PRIMARY KEY (SITE_ID)
)

CREATE UNIQUE INDEX PKBUS_DATA ON BUS_DATA (BUS_TYPE_ID ASC)
CREATE UNIQUE INDEX PKBUS_ON_SITE ON BUS_ON_SITE (BUS_ON_SITE_ID ASC)
CREATE UNIQUE INDEX PKBUS_SCHEDULE ON BUS_SCHEDULE (BUS_SCHEDULE_ID ASC)
CREATE UNIQUE INDEX PKDELIVER_PACKAGE ON DELIVER_PACKAGE (DELIVER_PACKAGE_NO ASC)
CREATE UNIQUE INDEX PKDESTINATION_DATA ON DESTINATION_DATA (DESTINATION_ID ASC)
CREATE UNIQUE INDEX PKPACKAGE_DATA ON PACKAGE_DATA (PACKAGE_TYPE_ID ASC)
CREATE UNIQUE INDEX PKPACKAGE_PART ON PACKAGE_PART (PACKAGE_NO ASC)
CREATE UNIQUE INDEX PKPART_DATA ON PART_DATA (PART_ID ASC)
CREATE UNIQUE INDEX PKSITE ON SITE (SITE_ID ASC)
CREATE TRIGGER BUS_DATA_TD AFTER DELETE ON BUS_DATA FOR EACH ROW
BEGIN
--VA_Extension InsertTrigger
--VA_Extension begin
/*VA_Extension
EXEC SQL
ALTER TABLE BUS_ON_SITE ADD CONSTRAINT stand by FOREIGN KEY (BUS_TYPE_ID) REFERENCES BUS_DATA (BUS_TYPE_ID) ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD ROW RESTRICT
END-EXEC
*/
delete
from BUS_ON_SITE
where BUS_ON_SITE.BUS_TYPE_ID = :old.BUS_TYPE_ID;
--VA_Extension end
END;
CREATE TRIGGER BUS_DATA_TU AFTER UPDATE ON BUS_DATA FOR EACH ROW
BEGIN
--VA_Extension InsertTrigger
--VA_Extension begin
/*VA_Extension

*/

114
EXEC SQL
    ALTER TABLE BUS_ON_SITE ADD CONSTRAINT stand by
    FOREIGN KEY (BUS>Type_ID)
    REFERENCES BUS_DATA (BUS_TYPE_ID)
    ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
IF :old.BUS_TYPE_ID != :new.BUS_TYPE_ID
THEN
    Update BUS_ON_SITE
    set BUS_ON_SITE.BUS_TYPE_ID = :new.BUS_TYPE_ID
    where BUS_ON_SITE.BUS_TYPE_ID = :old.BUS_TYPE_ID;
END IF;
--VA_Extension end
END;
/
CREATE TRIGGER BUS_ON_SITE_TD AFTER DELETE ON BUS_ON_SITE FOR EACH ROW
BEGIN
    --VA_Extension InsertTrigger
    --VA_Extension begin
    /*VA_Extension
    EXEC SQL
        ALTER TABLE BUS_SCHEDULE ADD CONSTRAINT set on
        FOREIGN KEY (BUS_ON_SITE_ID)
        REFERENCES BUS_ON_SITE (BUS_ON_SITE_ID)
        ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
delete
    from BUS_SCHEDULE
    where BUS_SCHEDULE.BUS_ON_SITE_ID = :old.BUS_ON_SITE_ID;
    --VA_Extension end
END;
/
CREATE TRIGGER BUS_ON_SITE_TI AFTER INSERT ON BUS_ON_SITE FOR EACH ROW
BEGIN
    --VA_Extension InsertTrigger
    --VA_Extension begin
    /*VA_Extension
    EXEC SQL
        ALTER TABLE BUS_ON_SITE ADD CONSTRAINT stand by
        FOREIGN KEY (BUS_TYPE_ID)
        REFERENCES BUS_DATA (BUS_TYPE_ID)
        ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
DECLARE
  ParentRowCnt Integer;
BEGIN
  select count(*) into ParentRowCnt
  from BUS_DATA
  where :new.BUS_TYPE_ID = BUS_DATA.BUS_TYPE_ID;

  IF ParentRowCnt <> 1
  THEN
    raise_application_error(-20010,
    'Insert of BUS_ON_SITE prevented because referential key does not exist in
    BUS_DATA');
  END IF;
END;
--VA_Extension end
--VA_Extension begin
/*VA_Extension
EXEC SQL
ALTER TABLE BUS_ON_SITE ADD CONSTRAINT has
FOREIGN KEY (SITE_ID)
REFERENCES SITE (SITE_ID)
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
DECLARE
  ParentRowCnt Integer;
BEGIN
  select count(*) into ParentRowCnt
  from SITE
  where :new.SITE_ID = SITE.SITE_ID;

  IF ParentRowCnt <> 1
  THEN
    raise_application_error(-20010,
    'Insert of BUS_ON_SITE prevented because referential key does not exist in
    SITE');
  END IF;
END;
--VA_Extension end
END;
/
CREATE TRIGGER BUS_ON_SITE TU AFTER UPDATE ON BUS_ON_SITE
FOR EACH ROW
BEGIN
EXEC SQL
   ALTER TABLE BUS_ON_SITE ADD CONSTRAINT stand by
   FOREIGN KEY (BUS_TYPE_ID)
   REFERENCES BUS_DATA (BUS_TYPE_ID)
   ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/

IF :old.BUS_TYPE_ID != :new.BUS_TYPE_ID
THEN
   DECLARE
      ParentRowCnt Integer;
   BEGIN
      select count(*) into ParentRowCnt
      from BUS_DATA
      where :new.BUS_TYPE_ID = BUS_DATA.BUS_TYPE_ID;

      IF ParentRowCnt <> 1
      THEN
         raise_application_error(-20010,
            'Update of BUS_ON_SITE prevented because referential key does not exist
in BUS_DATA');
      END IF;
   END;
END IF;
-- VA_Extension end
-- VA_Extension begin

EXEC SQL
   ALTER TABLE BUS_ON_SITE ADD CONSTRAINT has
   FOREIGN KEY (SITE_ID)
   REFERENCES SITE (SITE_ID)
   ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/

IF :old.SITE_ID != :new.SITE_ID
THEN
   DECLARE
      ParentRowCnt Integer;
   BEGIN
      select count(*) into ParentRowCnt
      from SITE
      where :new.SITE_ID = SITE.SITE_ID;

IF ParentRowCnt <> 1
THEN
    raise_application_error(-20010,
    'Update of BUS_ON_SITE prevented because referential key does not exist
in SITE');

    END IF;
END;
END IF;
-- VA_Extension end
-- VA_Extension begin
/*VA_Extension
EXEC SQL
ALTER TABLE BUS_SCHEDULE ADD CONSTRAINT set on
FOREIGN KEY (BUS_ON_SITE_ID)
REFERENCES BUS_ON_SITE (BUS_ON_SITE_ID)
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC */
IF :old.BUS_ON_SITE_ID != :new.BUS_ON_SITE_ID
THEN
    Update BUS_SCHEDULE
    set BUS_SCHEDULE.BUS_ON_SITE_ID = :new.BUS_ON_SITE_ID
    where BUS_SCHEDULE.BUS_ON_SITE_ID = :old.BUS_ON_SITE_ID;
END IF;
-- VA_Extension end
END;

CREATE TRIGGER BUS_SCHEDULE_TD AFTER DELETE ON
BUS_SCHEDULE FOR EACH ROW
BEGIN
-- VA_Extension Insert Trigger
-- VA_Extension begin
/*VA_Extension
EXEC SQL
ALTER TABLE DELIVER_PACKAGE ADD CONSTRAINT deliver
FOREIGN KEY (BUS_SCHEDULE_ID)
REFERENCES BUS_SCHEDULE (BUS_SCHEDULE_ID)
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC */
delete
from DELIVER_PACKAGE
where DELIVER_PACKAGE.BUS_SCHEDULE_ID = :old.BUS_SCHEDULE_ID;
-- VA_Extension end
END;
/
CREATE TRIGGER BUS_SCHEDULE_TI AFTER INSERT ON BUS_SCHEDULE
FOR EACH ROW
BEGIN
--VA_Extension InsertTrigger
--VA_Extension begin
/*VA_Extension
EXEC SQL
   ALTER TABLE BUS_SCHEDULE ADD CONSTRAINT set on
   FOREIGN KEY ( BUS_ON_SITE_ID )
   REFERENCES BUS_ON_SITE ( BUS_ON_SITE_ID )
   ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
DECLARE ParentRowCnt Integer;
BEGIN
   select count(*) into ParentRowCnt
   from BUS_ON_SITE
   where :new.BUS_ON_SITE_ID = BUS_ON_SITE.BUS_ON_SITE_ID;
   IF ParentRowCnt <> 1 THEN
      raise_application_error( -20010, 'Insert of BUS_SCHEDULE prevented because referential key does not exist in BUS_ON_SITE');
   END IF;
END;
--VA_Extension end
END;

CREATE TRIGGER BUS_SCHEDULE_TU AFTER UPDATE ON
BUS_SCHEDULE FOR EACH ROW
BEGIN
--VA_Extension InsertTrigger
--VA_Extension begin
/*VA_Extension
EXEC SQL
   ALTER TABLE BUS_SCHEDULE ADD CONSTRAINT set on
   FOREIGN KEY ( BUS_ON_SITE_ID )
   REFERENCES BUS_ON_SITE ( BUS_ON_SITE_ID )
   ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
IF :old.BUS_ON_SITE_ID ! = :new.BUS_ON_SITE_ID THEN
   DECLARE
      119
ParentRowCnt Integer;
BEGIN
    select count(*) into ParentRowCnt
    from BUS_ON_SITE
    where :new.BUS_ON_SITE_ID = BUS_ON_SITE.BUS_ON_SITE_ID;
IF ParentRowCnt <> 1
    THEN
        raise_application_error(-20010,
        'Update of BUS_SCHEDULE prevented because referential key does not exist in BUS_ON_SITE');
    END IF;
END IF;
/*VA Extension
EXEC SQL
ALTER TABLE DELIVER_PACKAGE ADD CONSTRAINT deliver
FOREIGN KEY ( BUS_SCHEDULE_ID )
REFERENCES BUS_SCHEDULE ( BUS_SCHEDULE_ID )
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD ROW RESTRICT
END-EXEC
*/
IF :old.BUS_SCHEDULE_ID != :new.BUS_SCHEDULE_ID
    THEN
        Update DELIVER_PACKAGE
        set DELIVER_PACKAGE.BUS_SCHEDULE_ID = :new.BUS_SCHEDULE_ID
        where DELIVER_PACKAGE.BUS_SCHEDULE_ID = :old.BUS_SCHEDULE_ID;
    END IF;
END IF;
/*VA_Extension InsertTrigger
EXEC SQL
ALTER TABLE DELIVER_PACKAGE ADD CONSTRAINT deliver
FOREIGN KEY ( BUS_SCHEDULE_ID )
REFERENCES BUS_SCHEDULE ( BUS_SCHEDULE_ID )
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD ROW RESTRICT
*/
CREATE TRIGGER DELIVER_PACKAGE_TI AFTER INSERT ON DELIVER_PACKAGE FOR EACH ROW
BEGIN
    --VA_Extension InsertTrigger
    --VA_Extension begin
    /*VA Extension
EXEC SQL
ALTER TABLE DELIVER_PACKAGE ADD CONSTRAINT deliver
FOREIGN KEY ( BUS_SCHEDULE_ID )
REFERENCES BUS_SCHEDULE ( BUS_SCHEDULE_ID )
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD ROW RESTRICT
*/
DECLARE
  ParentRowCnt    Integer;
BEGIN
  select count(*) into ParentRowCnt
  from BUS_SCHEDULE
  where :new.BUS_SCHEDULE_ID = BUS_SCHEDULE.BUS_SCHEDULE_ID;
  IF ParentRowCnt <> 1 THEN
    raise_application_error(-20010,
    'Insert of DELIVER_PACKAGE prevented because referential key does not exist in BUS_SCHEDULE');
  END IF;
END;
-- VA_Extension end
-- VA_Extension begin
/*VA_Extension
EXEC SQL
  ALTER TABLE DELIVER_PACKAGE ADD CONSTRAINT install on
FOREIGN KEY (PACKAGE_NO )
REFERENCES PACKAGE_PART (PACKAGE_NO )
on UPDATE CASCADE on DELETE CASCADE on insert of CHILD
ROW RESTRICT
END-EXEC
*/
DECLARE
  ParentRowCnt    Integer;
BEGIN
  select count(*) into ParentRowCnt
  from PACKAGE_PART
  where :new.PACKAGE_NO = PACKAGE_PART.PACKAGE_NO;
  IF ParentRowCnt <> 1 THEN
    raise_application_error(-20010,
    'Insert of DELIVER_PACKAGE prevented because referential key does not exist in PACKAGE_PART');
  END IF;
END;
-- VA_Extension end
/CREATE TRIGGER DELIVER_PACKAGE_TU AFTER UPDATE ON DELIVER_PACKAGE FOR EACH ROW
BEGIN
--VA_Extension InsertTriger
--VA_Extension begin
/*VA_Extension
EXEC SQL
ALTER TABLE DELIVER_PACKAGE ADD CONSTRAINT deliver
FOREIGN KEY ( BUS_SCHEDULE_ID )
REFERENCES BUS_SCHEDULE ( BUS_SCHEDULE_ID )
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
IF :old.BUS_SCHEDULE_ID != :new.BUS_SCHEDULE_ID
THEN
DECLARE
  ParentRowCnt Integer;
BEGIN
  select count(*) into ParentRowCnt
  from BUS_SCHEDULE
  where :new.BUS_SCHEDULE_ID = BUS_SCHEDULE.BUS_SCHEDULE_ID;
  IF ParentRowCnt <> 1
  THEN
    raise_application_error(-20010,
    'Update of DELIVER_PACKAGE prevented because referential key does not exist in BUS_SCHEDULE');
  END IF;
END IF;
END;
END IF;
--VA_Extension end
--VA_Extension begin
/*VA_Extension
EXEC SQL
ALTER TABLE DELIVER_PACKAGE ADD CONSTRAINT install on
FOREIGN KEY ( PACKAGE_NO )
REFERENCES PACKAGE_PART ( PACKAGE_NO )
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
IF :old PACKAGE_NO != :new PACKAGE_NO
THEN
DECLARE
  ParentRowCnt Integer;
BEGIN
  select count(*) into ParentRowCnt
  from PACKAGE_PART
  where :new PACKAGE_NO = PACKAGE_PART.PACKAGE_NO;
IF ParentRowCnt <> 1
THEN
    raise_application_error(-20010,
       'Update of DELIVER_PACKAGE prevented because referential key does not exist in PACKAGE_PART');

    END IF;
END;
END IF;
--VA_Extension end
END;
/

CREATE TRIGGER DESTINATION_DATA AFTER DELETE ON
DESTINATION_DATA FOR EACH ROW
BEGIN
    --VA_Extension InsertTrigger
    --VA_Extension begin
    /*VA_Extension
    EXEC SQL
    ALTER TABLE PART_DATA ADD CONSTRAINT has
    FOREIGN KEY ( DESTINATION_ID )
    REFERENCES DESTINATION_DATA ( DESTINATION_ID )
    ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
    ROW RESTRICT
    END-EXEC
    */
    delete
    from PART_DATA
    where PART_DATA.DESTINATION_ID = :old.DESTINATION_ID;
    --VA_Extension end
    END;
/

CREATE TRIGGER DESTINATION_DATA_1 AFTER UPDATE ON
DESTINATION_DATA FOR EACH ROW
BEGIN
    --VA_Extension InsertTrigger
    --VA_Extension begin
    /*VA_Extension
    EXEC SQL
    ALTER TABLE PART_DATA ADD CONSTRAINT has
    FOREIGN KEY ( DESTINATION_ID )
    REFERENCES DESTINATION_DATA ( DESTINATION_ID )
    ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
    ROW RESTRICT
    END-EXEC
    */
    IF :old.DESTINATION_ID != :new.DESTINATION_ID
    THEN
UPDATE PART_DATA
    set PART_DATA.DESTINATION_ID = :new.DESTINATION_ID
    where PART_DATA.DESTINATION_ID = :old.DESTINATION_ID;
END IF;
    --VA_Extension end
END;

CREATE TRIGGER PACKAGE_DATA_TD AFTER DELETE ON
PACKAGE_DATA FOR EACH ROW
    BEGIN
    --VA_Extension InsertTrigger
    --VA_Extension begin
    /*VA_Extension
    EXEC SQL
     ALTER TABLE PACKAGE_PART ADD CONSTRAINT contain
     FOREIGN KEY ( PACKAGE_TYPE_ID )
     REFERENCES PACKAGE_DATA ( PACKAGE_TYPE_ID )
     ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
    ROW RESTRICT
    END-EXEC
    */
    delete
    from PACKAGE_PART
    where PACKAGE_PART.PACKAGE_TYPE_ID = :old.PACKAGE_TYPE_ID;
    --VA_Extension end
END;

CREATE TRIGGER PACKAGE_DATA_TU AFTER UPDATE ON
PACKAGE_DATA FOR EACH ROW
    BEGIN
    --VA_Extension InsertTrigger
    --VA_Extension begin
    /*VA_Extension
    EXEC SQL
     ALTER TABLE PACKAGE_PART ADD CONSTRAINT contain
     FOREIGN KEY ( PACKAGE_TYPE_ID )
     REFERENCES PACKAGE_DATA ( PACKAGE_TYPE_ID )
     ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
    ROW RESTRICT
    END-EXEC
    */
    IF :old.PACKAGE_TYPE_ID != :new.PACKAGE_TYPE_ID
    THEN
        Update PACKAGE_PART
        set PACKAGE_PART.PACKAGE_TYPE_ID = :new.PACKAGE_TYPE_ID
        where PACKAGE_PART.PACKAGE_TYPE_ID = :old.PACKAGE_TYPE_ID;
    END IF;
    --VA_Extension end
END;
CREATE TRIGGER PACKAGE_PART_TD AFTER DELETE ON PACKAGE_PART FOR EACH ROW
BEGIN
--VA_Extension InsertTrigger
--VA_Extension begin
/*VA_Extension
EXEC SQL
   ALTER TABLE DELIVER_PACKAGE ADD CONSTRAINT install on
      FOREIGN KEY ( PACKAGE_NO )
      REFERENCES PACKAGE_PART ( PACKAGE_NO )
      ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
delete
   from DELIVER_PACKAGE
where DELIVER_PACKAGE.PACKAGE_NO = :old.PACKAGE_NO;
--VA_Extension end
END;

CREATE TRIGGER PACKAGE_PART_TI AFTER INSERT ON PACKAGE_PART FOR EACH ROW
BEGIN
--VA_Extension InsertTrigger
--VA_Extension begin
/*VA_Extension
EXEC SQL
   ALTER TABLE PACKAGE_PART ADD CONSTRAINT stored in
      FOREIGN KEY ( PART_ID )
      REFERENCES PART_DATA ( PART_ID )
      ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
DECLARE
   ParentRowCnt Integer;
BEGIN
   select count(*) into ParentRowCnt
      from PART_DATA
         where :new.PART_ID = PART_DATA.PART_ID;
   IF ParentRowCnt <> 1
      THEN
         raise_application_error( -20010,
            'Insert of PACKAGE_PART prevented because referential key does not exist in PART_DATA');
   END IF;
125
EXEC SQL
   ALTER TABLE PACKAGE_PART ADD CONSTRAINT contain
   FOREIGN KEY ( PACKAGE_TYPE_ID )
   REFERENCES PACKAGE_DATA ( PACKAGE_TYPE_ID )
   ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC

DECLARE
   ParentRowCnt Integer;
BEGIN
   select count(*) into ParentRowCnt
   from PACKAGE_DATA
   where :new.PACKAGE_TYPE_ID = PACKAGE_DATA.PACKAGE_TYPE_ID;

   IF ParentRowCnt <> 1 THEN
      raise_application_error(-20010,
      'Insert of PACKAGE_PART prevented because referential key does not exist
      in PACKAGE_DATA');
   END IF;
END;

CREATE TRIGGER PACKAGE_PART_TU AFTER UPDATE ON
PACKAGE_PART FOR EACH ROW
BEGIN
   --VA_Extension InsertTrigger
   --VA_Extension begin
   /*VA_Extension
   EXEC SQL
      ALTER TABLE PACKAGE_PART ADD CONSTRAINT stored in
      FOREIGN KEY ( PART_ID )
      REFERENCES PART_DATA ( PART_ID )
      ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
IF :old.PART_ID != :new.PART_ID THEN
   DECLARE
      ParentRowCnt Integer;

BEGIN
    select count(*) into ParentRowCnt
    from PART_DATA
    where :new.PART_ID = PART_DATA.PART_ID;

    IF ParentRowCnt <> 1
    THEN
        raise_application_error(-20010,
                   'Update of PACKAGE_PART prevented because referential key does not
exist in PART_DATA');
    END IF;
END;
END IF;
--VA_Extension end
--VA_Extension begin
/*VA_Extension
EXEC SQL
ALTER TABLE PACKAGE PART ADD CONSTRAINT contain
FOREIGN KEY ( PACKAGE_TYPE_ID )
REFERENCES PACKAGE_DATA ( PACKAGE_TYPE_ID )
on UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
IF :old.PACKAGE_TYPE_ID != :new.PACKAGE_TYPE_ID
THEN
    DECLARE
        ParentRowCnt Integer;
    BEGIN
        select count(*) into ParentRowCnt
        from PACKAGE_DATA
        where :new.PACKAGE_TYPE_ID = PACKAGE_DATA.PACKAGE_TYPE_ID;

        IF ParentRowCnt <> 1
        THEN
            raise_application_error(-20010,
                       'Update of PACKAGE_PART prevented because referential key does not
exist in PACKAGE_DATA');
        END IF;
    END;
END IF;
--VA_Extension end
--VA_Extension begin
/*VA_Extension
EXEC SQL
ALTER TABLE DELIVER PACKAGE ADD CONSTRAINT install on
FOREIGN KEY ( PACKAGE_NO )
REFERENCES PACKAGE_PART ( PACKAGE_NO )
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/

IF :old.PACKAGE_NO != :new.PACKAGE_NO
THEN
  Update DELIVER_PACKAGE
  set DELIVER_PACKAGE.PACKAGE_NO = :new.PACKAGE_NO
  where DELIVER_PACKAGE.PACKAGE_NO = :old.PACKAGE_NO;
END IF;
--VA_Extension end
END;

CREATE TRIGGER PART_DATA_TD AFTER DELETE ON PART_DATA FOR EACH ROW
BEGIN
--VA_Extension InsertTrigger
--VA_Extension begin
/*VA_Extension
EXEC SQL
  ALTER TABLE PACKAGE_PART ADD CONSTRAINT stored in
  FOREIGN KEY ( PART_ID )
  REFERENCES PART_DATA ( PART_ID )
  ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
delete
  from PACKAGE_PART
  where PACKAGE_PART.PART_ID = :old.PART_ID;
--VA_Extension end
END;

CREATE TRIGGER PART_DATA_TI AFTER INSERT ON PART_DATA FOR EACH ROW
BEGIN
--VA_Extension InsertTrigger
--VA_Extension begin
/*VA_Extension
EXEC SQL
  ALTER TABLE PART_DATA ADD CONSTRAINT has
  FOREIGN KEY ( SITE_ID )
  REFERENCES SITE ( SITE_ID )
  ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
DECLARE
  ParentRowCnt Integer;
BEGIN
  select count(*) into ParentRowCnt
  from SITE
  where :new.SITE_ID = SITE.SITE_ID;

  IF ParentRowCnt <> 1
  THEN
    raise_application_error(-20010,
      'Insert of PART_DATA prevented because referential key does not exist in SITE');
  END IF;
END;

CREATE TRIGGER PART_DATA_TU AFTER UPDATE ON PART_DATA FOR EACH ROW
BEGIN
  --VA_Extension InsertTrigger
  --VA_Extension begin
  /*VA_Extension
  EXEC SQL
  ALTER TABLE PART_DATA ADD CONSTRAINT has FOREIGN KEY (DESTINATION_ID)
  REFERENCES DESTINATION_DATA (DESTINATION_ID)
  ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD ROW RESTRICT
  END-EXEC
  */
  DECLARE
    ParentRowCnt Integer;
  BEGIN
    select count(*) into ParentRowCnt
    from DESTINATION_DATA
    where :new.DESTINATION_ID = DESTINATION_DATA.DESTINATION_ID;

    IF ParentRowCnt <> 1
    THEN
      raise_application_error(-20010,
        'Insert of PART_DATA prevented because referential key does not exist in DESTINATION_DATA');
    END IF;
  END;
  --VA_Extension end
  END;
  /
/*VA Extension
EXEC SQL
ALTER TABLE PART_DATA ADD CONSTRAINT has
FOREIGN KEY (SITE_ID)
REFERENCES SITE (SITE_ID)
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
IF :old.SITE_ID != :new.SITE_ID
THEN
  DECLARE
    ParentRowCnt Integer;
BEGIN
  select count(*) into ParentRowCnt
  from SITE
  where :new.SITE_ID = SITE.SITE_ID;

  IF ParentRowCnt <> 1
  THEN
    raise_application_error(-20010,
      'Update of PART_DATA prevented because referential key does not exist
in SITE');
  END IF;
END;
END IF;
-- VA Extension end
-- VA Extension begin
/*VA Extension
EXEC SQL
ALTER TABLE PART_DATA ADD CONSTRAINT has
FOREIGN KEY (DESTINATION_ID)
REFERENCES DESTINATION_DATA (DESTINATION_ID)
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
IF :old.DESTINATION_ID != :new.DESTINATION_ID
THEN
  DECLARE
    ParentRowCnt Integer;
BEGIN
  select count(*) into ParentRowCnt
  from DESTINATION_DATA
  where :new.DESTINATION_ID = DESTINATION_DATA.DESTINATION_ID;

  IF ParentRowCnt <> 1
  THEN
    raise_application_error(-20010,
Update of PART_DATA prevented because referential key does not exist in DESTINATION_DATA;

END IF;
END;
END IF;
--VA_Extension end
--VA_Extension begin
/*VA_Extension
EXEC SQL
ALTER TABLE PACKAGE_PART ADD CONSTRAINT stored in
FOREIGN KEY (PART_ID)
REFERENCES PART_DATA (PART_ID)
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/

IF :old.PART_ID != :new.PART_ID
THEN
Update PACKAGE_PART
set PACKAGE_PART.PART_ID = :new.PART_ID
where PACKAGE_PART.PART_ID = :old.PART_ID;
END IF;
--VA_Extension end
END;

CREATE TRIGGER SITE_TD AFTER DELETE ON SITE FOR EACH ROW
BEGIN
--VA_Extension InsertTrigger
--VA_Extension begin
/*VA_Extension
EXEC SQL
ALTER TABLE BUS_ON_SITE ADD CONSTRAINT has
FOREIGN KEY (SITE_ID)
REFERENCES SITE (SITE_ID)
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
delete
from BUS_ON_SITE
where BUS_ON_SITE.SITE_ID = :old.SITE_ID;
--VA_Extension end
--VA_Extension begin
/*VA_Extension
EXEC SQL
ALTER TABLE PART_DATA ADD CONSTRAINT has
FOREIGN KEY (SITE_ID)
REFERENCES SITE (SITE_ID)
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/

delete
from PART_DATA
where PART_DATA.SITE_ID = :old.SITE_ID;
-- VA_Extension end
END;
/
CREATETRIGGER SITE_TU AFTER UPDATE ON SITE FOR EACH ROW
BEGIN
-- VA_Extension InsertTrigger
-- VA_Extension begin
/* VA_Extension
EXEC SQL
ALTER TABLE BUS_ON_SITE ADD CONSTRAINT has
FOREIGN KEY (SITE_ID)
REFERENCES SITE (SITE_ID)
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
IF :old.SITE_ID != :new.SITE_ID
THEN
Update BUS_ON_SITE
set BUS_ON_SITE.SITE_ID = :new.SITE_ID
where BUS_ON_SITE.SITE_ID = :old.SITE_ID;
END IF;
-- VA_Extension end
-- VA_Extension begin
/* VA_Extension
EXEC SQL
ALTER TABLE PART_DATA ADD CONSTRAINT has
FOREIGN KEY (SITE_ID)
REFERENCES SITE (SITE_ID)
ON UPDATE CASCADE ON DELETE CASCADE ON INSERT OF CHILD
ROW RESTRICT
END-EXEC
*/
IF :old.SITE_ID != :new.SITE_ID
THEN
Update PART_DATA
set PART_DATA.SITE_ID = :new.SITE_ID
where PART_DATA.SITE_ID = :old.SITE_ID;
END IF;
-- VA_Extension end
END;
/
BIBLIOGRAPHY


