E-TAX INVOICE SYSTEM
USING
WEB SERVICES TECHNOLOGY

SUKANYA SUWISUTHIKASEM

A RESEARCH PROJECT SUBMITTED IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
(COMPUTER SCIENCE)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2007

COPYRIGHT OF MAHIDOL UNIVERSITY

Copyright by Mahidol University

Research Project
Entitled

E-TAX INVOICE SYSTEM USING WEB SERVICES TECHNOLOGY

.....
Miss Sukanya Suwisuthikasem
Candidate

.....
Lect. Songsri Tangsripairoj, Ph.D.
Major-Advisor

.....
Asst.Prof. Thanwadee Sunetnanta, Ph.D.
Co-Advisor

.....
Prof. Banchong Mahaisavariya, M.D.
Dean
Faculty of Graduate Studies

.....
Assoc.Prof. Supachai Tangwongsan, Ph.D.
Chair
Master of Science Programme in
Computer Science
Faculty of Science

Copyright by Mahidol University

Research Project
Entitled

E-TAX INVOICE SYSTEM USING WEB SERVICES TECHNOLOGY

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Science (Computer Science)
on
November 2, 2007

สุกัญญา สุวิสุทธิกเสม
.....
Miss Sukanya Suwisuthikasem
Candidate

Supanat
.....
Lect. Supanat Kitcharoensakkul, Ph.D.
Chair

Songsri Tangsripairoj
.....
Lect. Songsri Tangsripairoj, Ph.D.
Member

Thanwadee Sunetnanta
.....
Asst.Prof. Thanwadee Sunetnanta, Ph.D.
Member

B. Mahaisavariya
.....
Prof. Banchong Mahaisavariya, M.D.
Dean
Faculty of Graduate Studies
Mahidol University

Amaret Bhumiratana
.....
Prof. Amaret Bhumiratana, Ph.D.
Dean
Faculty of Science
Mahidol University

Copyright by Mahidol University

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and deep appreciations to my advisor, Dr. Songsri Tangsripairoj, for her deep interest, inspiring guidance and continued encouragement which provided the impetus to complete this research project successfully.

I am also graceful to Dr Thanwadee Sunetnanta and Dr Supanat Kitcharoensakkul who are my research project committee and provide me their insightful comments, technical review and suggestions on this research project.

I would like to thank all the lecturers of the Computer Science Department, Faculty of Science for their knowledge and education. I wish to express my sincere thanks to all my graduate friends, class 2004 for their warm support and friendship.

My special thanks is extended to Mrs. Tasana Chutivisut, for her helpful guidance, encouragement and moral support throughout the period of research project implementation and writing.

Finally, I wish to specifically thank my parent, and my brother, for their understanding and inspiration throughout my life. Their continued love and encouragement throughout this research project is the reason behind its success.

Sukanya Suwisuthikasem

Copyright by Mahidol University

E-TAX INVOICE SYSTEM USING WEB SERVICES TECHNOLOGY

SUKANYA SUWISUTHIKASEM 4737980 SCCS/M

M.Sc. (COMPUTER SCIENCE)

RESEARCH PROJECT ADVISORS: SONGSRI TANGSRIPAIROJ, Ph.D, THANWADEE SUNETNANTA, Ph.D.

ABSTRACT

This project proposed an "e-Tax Invoice System" using Web Services (WS) technology for the Revenue Department of Thailand./ This system uses four WS technologies -XML, WSDL, UDDI and SOAP- that enable the system to communicate to several applications on heterogeneous platforms. In addition, the e-Tax Invoice System applies two security technologies: SSL as a secure channel for data exchange, and Hash Function for transforming plain passwords into unreadable passwords. The new system is expected to collect tax invoice data from several businesses. Businesses can send tax invoice data to the system immediately once the payment occurs. From this, the system has up-to-date data for generating reports for RD Officers, so that reports generated from the system can support work processes especially tax payment auditing processes. Moreover the system allows customers and businesses to search tax invoice data online.

Usability testing and comparative analysis are performed to evaluate the quality of the new system. The experimental result of usability testing shows that the user satisfaction level of most users for using the system is in Good level. This implies that the new system is easy to use and the system can achieve user requirements. Moreover, comparative analysis shows that the new system can solve problems in the existing system and enhance work processes concerning tax invoice systems.

KEY WORDS : TAX INVOICE / WEB SERVICES / SOAP / XML / XSLT / XPATH

164 pp.

Copyright by Mahidol University

ระบบใบกำกับภาษีอิเล็กทรอนิกส์โดยการประยุกต์ใช้เทคโนโลยีของเว็บเซอร์วิส
(E-TAX INVOICE SYSTEM USING WEB SERVICES TECHNOLOGY)

สุกัญญา สุวิสุทธิเกษม 4737980 SCCS/M

วท.ม. (วิทยาการคอมพิวเตอร์)

คณะกรรมการควบคุมสารนิพนธ์ : ทรงศรี ตั้งศรีไพโรจน์, Ph.D, ธันวดี สุเนตนันท์, Ph.D

บทคัดย่อ

สารนิพนธ์ฉบับนี้นำเสนอ "ระบบใบกำกับภาษีอิเล็กทรอนิกส์" ซึ่งพัฒนาโดยใช้เทคโนโลยีเว็บเซอร์วิสคือ XML WSDL UDDI SOAP สำหรับนำมาใช้งานในกรมสรรพากร เพื่อให้บริการแก่ผู้เสียภาษี ผู้ประกอบการ และเจ้าหน้าที่ของกรมสรรพากร โดยเทคโนโลยีเว็บเซอร์วิสช่วยให้ระบบใบกำกับภาษีอิเล็กทรอนิกส์สามารถเชื่อมต่อเพื่อแลกเปลี่ยนข้อมูลกับระบบอื่นๆที่ถูกพัฒนาและติดตั้งบนระบบปฏิบัติการที่ต่างกันได้ นอกจากนี้ระบบใบกำกับภาษีอิเล็กทรอนิกส์นี้ยังมีการใช้เทคโนโลยีด้านความปลอดภัย คือ SSL เพื่อใช้เป็นช่องทางที่ปลอดภัยในการส่งผ่านข้อมูล และ Hash Function เพื่อใช้แปลงรหัสผ่านที่ใช้ในระบบให้อยู่ในรูปแบบที่ไม่สามารถอ่านเข้าใจได้ โดยระบบใบกำกับภาษีอิเล็กทรอนิกส์จะถูกนำมาใช้ในการจัดเก็บข้อมูลใบกำกับภาษีจากผู้ประกอบการขนาดใหญ่ โดยผู้ประกอบการสามารถส่งข้อมูลใบกำกับภาษีเข้าสู่ระบบได้ทันทีที่เกิดรายการขาย ซึ่งทำให้ข้อมูลในระบบใบกำกับภาษีอิเล็กทรอนิกส์มีความเป็นปัจจุบันเพื่อนำมาใช้ในการสร้างรายงานเพื่อ ให้เจ้าหน้าที่กรมสรรพากรใช้ช่วยในการตัดสินใจในการตรวจสอบการจ่ายภาษีของผู้ประกอบการได้มีประสิทธิภาพดียิ่งขึ้น นอกจากนี้ผู้เสียภาษียังสามารถค้นหาข้อมูลใบกำกับภาษีจากระบบใบกำกับภาษีอิเล็กทรอนิกส์ได้ เช่นกัน

การทดสอบการใช้งานและการวิเคราะห์เชิงเปรียบเทียบถูกใช้ในการประเมินผลระบบ ซึ่งจากผลการทดสอบการใช้งานพบว่าผู้ใช้ส่วนมากมีความพอใจในระดับที่ดีในการใช้งานระบบ ซึ่งหมายความว่าระบบง่ายต่อการใช้งานและสามารถตอบสนองความต้องการของผู้ใช้ได้ นอกจากนี้การวิเคราะห์เชิงเปรียบเทียบยังแสดงให้เห็นว่าระบบสามารถแก้ไขปัญหาที่มีอยู่ในระบบเดิมและยังสนับสนุนกระบวนการทำงานในระบบใบกำกับภาษีให้มีประสิทธิภาพยิ่งขึ้น

164 หน้า

Copyright by Mahidol University

CONTENTS

Page

ACKNOWLEDGEMENTS iii

ABSTRACT iv

LIST OF TABLES ix

LIST OF FIGURES xi

CHAPTER

I INTRODUCTION

1.1 Introduction 1

1.2 Objectives 2

1.3 Scope of Work 3

1.4 Contribution of Work 4

II THE EXISTING TAX INVOICE SYSTEM

2.1 System Overview 5

2.2 Use Case Diagram 8

2.3 Activity Diagram 14

2.4 Tax Invoice Data 16

2.4.1 Types of Tax Invoice Slip 17

2.4.2 Examples of Tax Invoice Slip 18

2.4.3 Reports 19

2.5 Tax Invoice File Structure 21

2.6 Problems of The Existing Tax Invoice System 22

III LITERATURE REVIEW

3.1 Web Services 23

3.1.1 Web Services Definition 23

3.1.2 Web Services Architecture 24

Copyright by Mahidol University

CONTENTS (Cont.)

		Page
	3.1.3 Web Services Core Technologies	25
	3.2 Security Technology	44
	3.2.1 Secure Socket Layer (SSL)	44
	3.2.2 Hash Function	45
	3.3 Web Services Technology in Revenue Departments of Other Countries	48
	3.3.1 Canada Revenue Agency (CRA)	48
	3.3.2 HM Revenue & Customs (HMRC)	48
	3.3.3 Inland Revenue Authority of Singapore (IRAS)	49
	3.3.4 Internal Revenue Service (IRS) of the United States	49
	3.3.5 DG Tax & Customs (DG)	50
	3.3.6 The Revenue Department of Thailand (RD)	50
IV	SYSTEM ANALYSIS AND DESIGN	
	4.1 e-Tax Invoice System Architecture	52
	4.2 Structure Chart	54
	4.2.1 User Authenticator (UA)	54
	4.2.2 Data Collector (DC)	54
	4.2.3 Data Provider (DP)	56
	4.2.4 Login Manager (LM)	56
	4.2.5 Report Generator (RG)	57
	4.2.6 Log Report Generator (LRG)	57
	4.2.7 User Manager (UM)	57
	4.3 User Case Diagram	58
	4.4 Activity Diagram	65
	4.5 Class Diagram	73
	4.6 Business Process Diagram	76

Copyright by Mahidol University

CONTENTS (Cont.)

Page

4.6.1 Web Services 77
4.6.2 Processes 77
4.7 Sequence Diagram 88
V IMPLEMENTATION
5.1 Hardware and Software Requirements 96
5.1.1 Database Layer 96
5.1.2 Application Layer 98
5.1.3 Client Layer 98
5.2 Prototype 99
5.2.1 User Interface 99
5.2.2 Web Services 114
VI SYSTEM EVALUATION AND RESULT
6.1 Usability Testing 145
6.1.1 Evaluation Objective 146
6.1.2 Evaluation Methodology 147
6.1.3 Testing Environment 147
6.1.4 Questionnaire 149
6.1.5 Usability Testing Result 152
6.3 Comparative Analysis 154
VII CONCLUSION AND FUTURE WORKS
7.1 Conclusion 158
7.2 Discussion 159
7.3 Future Works 159
REFERENCES 161
BIOGRAPHY 164

Copyright by Mahidol University

LIST OF TABLES

Page

Table 2.1: Buy Products/Services Use Case 9
Table 2.2: Pay for Products/Services Use Case 9
Table 2.3: Generate Tax Invoice Slips Use Case 10
Table 2.4: Distribute Tax Invoice Slips Use Case 10
Table 2.5: Send Tax Invoice Slip to Customer Use Case 10
Table 2.6: Keep Tax Invoice Slip at Business's Shop Use Case 11
Table 2.7: Generate Reports and Summarize Data Use Case 11
Table 2.8: Send Data to RD Use Case 12
Table 2.9: Request for Summary Data, Reports and Tax Invoice Slips Use Case 12
Table 2.10: Call to Business's Shop and Request Data Use Case 13
Table 2.11: Go to Business's Shop and Request Data Use Case 13
Table 2.12: Wait for Requested Data Use Case 13
Table 2.13: Inspect Data from Businesses Use Case 14
Table 2.14: Example of Large Businesses Transaction 16
Table 2.15: Full Tax Invoice File 21
Table 2.16: Short Tax Invoice File 22
Table 4.1: Authenticate Client Use Case 59
Table 4.2: Logging User Use Case 59
Table 4.3: View & Print Tax Invoice Slip Use Case 60
Table 4.4: Collect Tax Invoice Data Use Case 61
Table 4.5: Login Use Case 61
Table 4.6: View Reports Use Case 62
Table 4.7: View Log Report Use Case 63
Table 4.8: Add User Account Use Case 63

Copyright by Mahidol University

LIST OF TABLES (Cont.)

Page

Table 4.9: Edit User Account Use Case 64
Table 4.10: Delete User Account Use Case 64
Table 4.11: List of Data Tables 72
Table 4.12: Invoice 73
Table 4.13: InvoiceProduct 74
Table 4.14: BusinessProfile 75
Table 4.15: UserAuthentication 75
Table 4.16: UserAccess 76
Table 4.17: WService 76
Table 4.18: WSLog 76
Table 4.19: Business Process Diagram Symbols 77
Table 4.20: Sequence Diagram Symbols 89
Table 6.1: System Evaluation Criteria 146
Table 6.2: Questionnaire for Customer 149
Table 6.3: Questionnaire for RD Officer 150
Table 6.4: Customer Usability Test Result 152
Table 6.5: RD Officer Usability Test Result 152
Table 6.6: Comparison of activities in Existing System and Proposed System 155

Copyright by Mahidol University

LIST OF FIGURES

Page

Figure 2.1: System Overview of the Existing Tax Invoice 6
Figure 2.2: Use Case Diagram of The Existing Tax Invoice System Use Case 8
Figure 2.3: Generate Tax Invoice Slips Activity Diagram 14
Figure 2.4: Inspect Tax Invoice Slips Activity Diagram 15
Figure 2.5: Full Tax Invoice Slip Example : AIS 18
Figure 2.6: Short Tax Invoice Slip Example : Black Canyon 19
Figure 2.7: Sale Report Example 20
Figure 2.8: Purchase Report Example 20
Figure 3.1: Basic Web Service Architecture 24
Figure 3.2: Three Parts of The SOAP Message 25
Figure 3.3: An example of SOAP Message 26
Figure 3.4: An Example of WSDL Document 27
Figure 3.5: An Example of Definition Element 29
Figure 3.6: An Example of Type Element 29
Figure 3.7: An Example of Message Element 30
Figure 3.8: An Example of PortType Element 31
Figure 3.9: An Example of Binding Element 31
Figure 3.10: An Example of Code to specify Port Element 32
Figure 3.11: An Example of Service Element 32
Figure 3.12: XML Transformation using XSLT 38
Figure 3.13: XSLT Style Sheet Structure 39
Figure 3.14: Seven Node Types in XPath 41
Figure 3.15: The UDDI repository used to discover a Web Service 43
Figure 3.16: SSL between application protocols and TCP/IP 44
Figure 4.1: e-Tax Invoice System Architecture 53

Copyright by Mahidol University

LIST OF FIGURES (Cont.)

Page

Figure 4.2: e-Tax Invoice System Structure Chart 55
Figure 4.3: e-Tax Invoice System Use Case Diagram 58
Figure 4.4: Authenticate Client Activity Diagram 65
Figure 4.5: Logging User Activity Diagram 65
Figure 4.6: View & Print Tax Invoice Activity Diagram 66
Figure 4.7: Collect Tax Invoice Data Activity Diagram 67
Figure 4.8: View Reports Activity Diagram 68
Figure 4.9: View Log Report Activity Diagram 69
Figure 4.10: Add User Account Activity Diagram 70
Figure 4.11: Edit User Account Activity Diagram 70
Figure 4.12: Delete User Account Activity Diagram 71
Figure 4.13: Class Diagram of e-Tax Invoice System 73
Figure 4.14: Business Process Diagram of e-tax Invoice System 79
Figure 4.15: Business Process Diagram of User Authenticator 80
Figure 4.16: Business Process Diagram of Data Provider 81
Figure 4.17: Business Process Diagram of Data Collector 82
Figure 4.18: Business Process Diagram of Report Generator 83
Figure 4.19: Business Process Diagram of Log Report Generator 84
Figure 4.20: Business Process Diagram of Logging User Usage Process 85
Figure 4.21: Business Process Diagram of Generate Report Process 86
Figure 4.22: Business Process Diagram of Generate Log Report Process 87
Figure 4.23: Business Process Diagram of Manage User Account Process 88
Figure 4.24: User Authenticator Sequence Diagram 90
Figure 4.25: Data Provider Sequence Diagram 91
Figure 4.26: Data Collector Sequence Diagram 91
Figure 4.27: Login Manager Sequence Diagram 92

Copyright by Mahidol University

LIST OF FIGURES (Cont.)

Page

Figure 4.28: Report Generator Sequence Diagram 93
Figure 4.29: Log Report Generator Sequence Diagram 94
Figure 4.30: Log User Manager Sequence Diagram 95
Figure 5.1: e-Tax Invoice System Environment 96
Figure 5.2: View Tax Invoice Page 100
Figure 5.3: An Example of XSLT in e-Tax Invoice System 100
Figure 5.4: An Example of Tax Invoice XML Document 103
Figure 5.5: An Example of HTML Document Result 104
Figure 5.6: Send Tax Invoice Data Page 106
Figure 5.7: Login Page 107
Figure 5.8: Main Menu Page 107
Figure 5.9: User Account Main Page 108
Figure 5.10: Add User Account Page 109
Figure 5.11: Edit User Account Page 109
Figure 5.12: Confirm User Removal Message Dialog 110
Figure 5.13: Summary Monthly Log Report Page 110
Figure 5.14: Log Detail Page 111
Figure 5.15: Report Main Page 112
Figure 5.16: Purchases/Sales Summary Report Page 112
Figure 5.17: Total Sales Summary Report Page 113
Figure 5.18: Total Sales Summary Report Graph Page 114
Figure 5.19: WSDL of UAService 115
Figure 5.20: Input/Output Schema of UAService: Design View 117
Figure 5.21: Input/Output Schema of UAService: Code View 117
Figure 5.22: WSDL of DPService 118

Copyright by Mahidol University

LIST OF FIGURES (Cont.)

Page

Figure 5.23: Input/Output Schema of DPService: Design View 120
Figure 5.24: Invoice Schema: Design View 121
Figure 5.25: Product Schema: Design View 122
Figure 5.26: CFullname Schema: Design View 122
Figure 5.27: Caddr Schema: Design View 123
Figure 5.28: Input/Output Schema of DPService: Code View 123
Figure 5.29: WSDL of DCService 127
Figure 5.30: Input/Output Schema of DCService: Design View 129
Figure 5.31: Input/Output Schema of DCService: Code View 129
Figure 5.32: WSDL of RGService 133
Figure 5.33: Input/Output Schema of RGService: Design View 135
Figure 5.34: ReportCond Schema: Design View 135
Figure 5.35: Sum_Report Schema: Design View 136
Figure 5.36: Tot_Sum_Report Schema: Design View 137
Figure 5.37: Input/Output Schema of RGService: Code View 137
Figure 5.38: WSDL of LRGService 141
Figure 5.39: Input/Output Schema of LRGService: Design View 143
Figure 5.40: Input/Output Schema of LRGService: Code View 143
Figure 6.1: Qualitative Assessment of Usability 145
Figure 6.2: Questionnaire for Customer in Web Application 151
Figure 6.3: Questionnaire for RD Officer in Web Application 151

Copyright by Mahidol University

CHAPTER I
INTRODUCTION

1.1 Introduction

The Revenue Department (RD) is one of the three tax collection agencies in Thailand with the responsibility to collect taxes which are approximately 2/3 of the national tax collection. Being an agency that is responsible for raising the majority of revenue and an agency that works closely with the people, the RD always tries to continually improve the efficiency and effectiveness of its operations that will stabilize the economic and social growth. For this reason, the core success factor is the ability to adopt information technology in all operation levels. The RD realizes the significance of the information technology and constantly develops information system in all tax types to facilitate tax collection, specifically Personal Income Tax (PIT), Corporate Income Tax (CIT) and Value Added Tax (VAT). From these tax types, VAT collection amount is approximately 40% of all tax collection.

Tax invoice data is a part of VAT information system that presently the RD cannot collect. For this reason, there are vulnerabilities that fraud business can use tax invoice slip dishonestly to avoid tax payment or to get tax refund from fraud tax invoice. This will make unfairness in tax collection. However, presently most of business organizations that deal with VAT system are most likely to use computer for collecting and generating tax invoice and to keep tax invoices data in electronic form. Each business uses software developed by different programming languages on heterogeneous platforms. This is the limitation for the RD to develop application for communicating with business's applications.

Copyright by Mahidol University

Web Services (WS) is the service-oriented architecture (SOA), a loosely coupled mechanism for software providers to make modularized functionality accessible over a computer network. WS provides a new and excellent solution to the data integration among heterogeneous systems. It presents an innovative software technology which provides application-to-application interaction. Web Services are based on communication protocols (SOAP), service description (WSDL), and service discovery (UDDI) and are built on top of existing Web protocols and based on open XML standards. This will support the new system to interact with others business organizations easily. Moreover, this project applies security technologies – Secure Socket Layer (SSL) and Hash Function - to provide three capabilities: credential exchange, message integrity, and message confidentiality.

This project uses the Web Services (WS) technology to develop e-tax invoice system. The new system is expected to collect tax invoice data from allied business and provide benefits for customer, allied business and the RD officer (Supervisor). For customer and allied business, they can online view and print tax invoice. For RD officer in operation level, the new system can provide sales report and purchase report of each business that the RD officer can use them to audit business's tax payment. This is the significant benefit of the system for the RD officer. For decision making level, it can generate summary information and report using for tax collection plan. Moreover, when the new system is deployed, business no need to keep tax invoice slips. As a result, the system can save cost in purchasing paper and keeping tax invoice slips. This is the major benefit that business especially large business will receive. Beside these, the system also provides convenience in usage to satisfy the RD officer, customer and business.

1.2 Objectives

This project aims to accomplish the following tasks.

1. To study and analyze the problems of the existing tax invoice system.
2. To study methodologies and benefits of WS technologies presently available.

Copyright by Mahidol University

3. To create a new system called “e-Tax Invoice System” by using WS technology to collect and analyze tax invoice data from business and provide useful information to RD officer for inspecting business’s tax payment and planning VAT collection.

4. To construct the e-Tax Invoice system by applying security technology: SSL and Hash Function to encapsulate sensitive data from unauthorized.

5. To build a prototype of the e-Tax Invoice System for customer and allied business to view and print tax invoice online.

6. To implement and evaluate our proposed system.

1.3 Scope of Work

The scopes of this research project are as follows:

1. The e-Tax Invoice system with Web Services technology can provide services to customer, allied business, and RD officer.

2. Create the tool that can validate and analyze collected data. And provide useful information for RD officer to audit business’s tax payments.

3. Apply existing security technology in the e-Tax Invoice System Secure Socket layer: SSL and Hash Function to encapsulate sensitive data from unauthorized user.

4. The e-Tax Invoice system can provide information to other tax information system e.g. VAT system.

5. The e-Tax Invoice system can interact with existing RD’s Web Services e.g. TIN/PINService: Checking for validation and existence of Tax Identification Number (TIN) and Personal Identification Number (PIN), VATService: Providing information of taxpayer who is the Value-Added Tax registrants.

6. Customer and business can online view and print tax invoice.

Copyright by Mahidol University

1.4 Contribution of Work

This research project has contributions described as follows:

1. Control and anti-fraud in tax payment and tax return.
2. Save cost and time in operations both in business and the RD.
3. Provide faster and easier services.
4. Enhance efficiency of work process in RD that will increase efficiency and transparency in the tax administration.
5. Modernize the RD in line with e-Government policy and support e-Service among RD officer, allied business and other official organizations.

Copyright by Mahidol University

CHAPTER II
THE EXISTING TAX INVOICE SYSTEM

The Tax Invoice System is the system dealing with tax invoice data. This includes generating, collecting and inspecting tax invoice data. Tax invoice is issued by VAT registered business operators when customer consumes goods/services. There are two types of customer: corporation and individual. In case of corporation, tax invoice slips are recorded in business's account for generating reports. And for individuals, tax invoice slips are used to get money return from customer's business organization.

Moreover, the RD officer uses tax invoice slip as the tool to inspect business's tax (Value Added Tax: VAT) payment. If businesses generate tax invoice slip in every sale transactions, business must use all transactions to generate sales report, purchase report etc. that the RD officer can use these reports to audit business's tax payment. As a result the RD can collect VAT tax completely. The rest of this chapter will describe the existing tax invoice system in detail and problem of existing tax invoice system.

2.1 System Overview

Figure 2.1 illustrates the system overview of the existing tax invoice system. There are three parties involving with this system: customer, business, and the RD officer. The significant data is tax invoice data that was generated when sale transactions occur. The system composes of eight main processes: Select Products/Services, Compute Amount of Product's Price and VAT, Make a Payment, Generate Tax Invoice Slips, Summarize Tax Invoice Data and Generate Report, Send

Copyright by Mahidol University

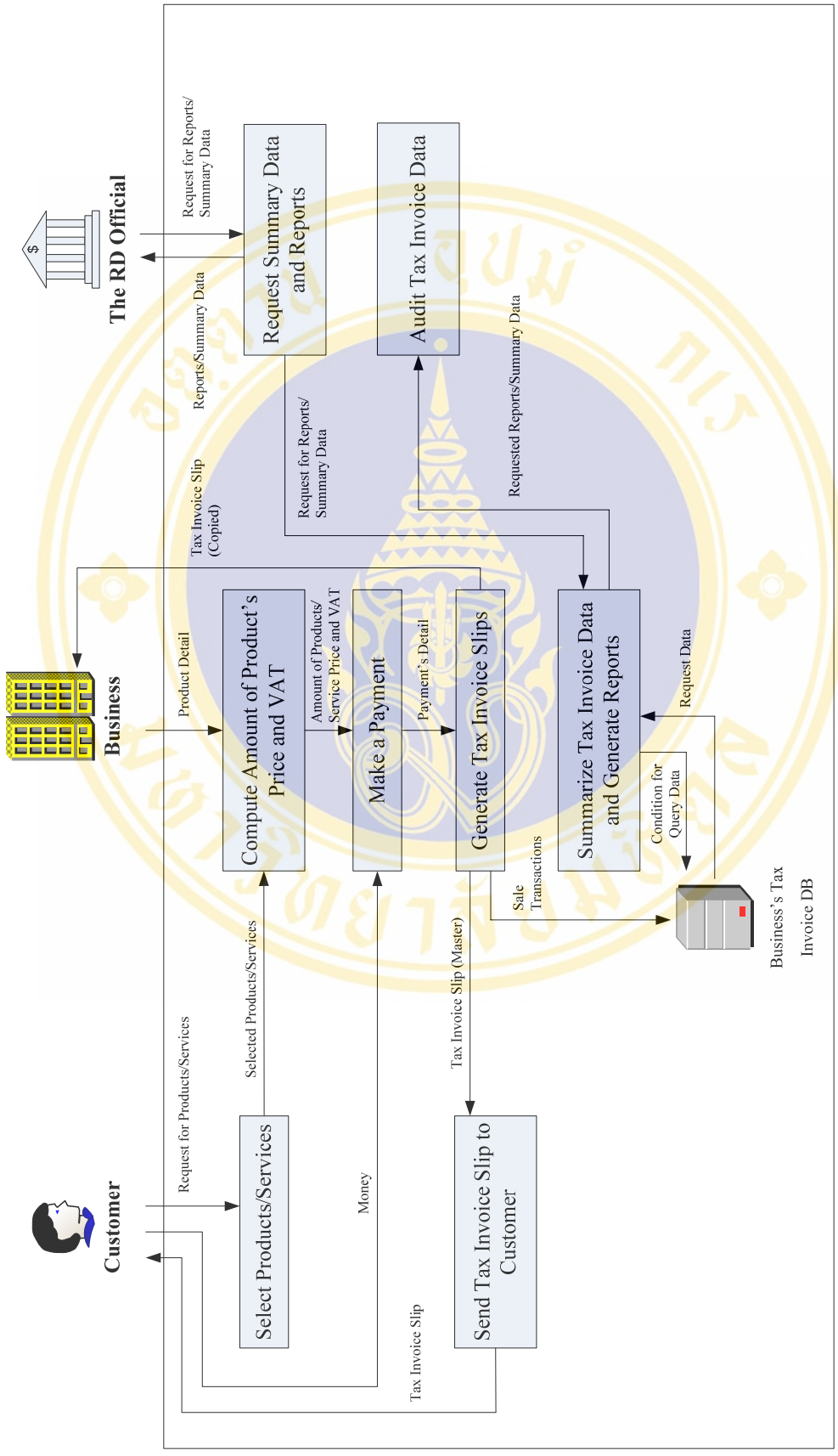


Figure 2.1 System Overview of The Existing Tax Invoice

Copyright by Mahidol University

Tax Invoice Slip to Customer, Request Summary Data and Reports, and Audit Tax Invoice Data.

First, customer goes to business's shop to purchase products/services. When customer receives desired products/services, customer goes to cashier. Business's shop computes price of products/services and informs total price (including VAT) to customer. Afterward, customer gives money to business's shop for making a payment. When business receives money from customer, business generates two copies of tax invoice slips and gives one tax slip (master) to customer. Business's shop keep tax invoice slip (copied) at the business place.

When the RD officer wants to audit business's VAT payment, they request for summary tax invoice data and reports from the business. Afterward, the RD officer waits for requested data and reports. This may takes at least one day. Business generates reports and sends to the RD officer. When the RD officer receives requested reports/summary data, they will review reports/summary to inspect the business' VAT payment.

The inspecting processes performed by the RD Officer are described in detail as follows:

1. When RD Officer finds that the business has the unusual VAT payment. They will prepare themselves for inspecting suspected business. Supervisors review business's profile such as business type, product, size of business, and initial investment. After supervisors known about suspected business, they will go to business place.

2. When RD Officer goes to business's place, they meets business's contact person. They will observe business's place to know about amount of customers, business's size, and business's product.

3. Afterward, RD Officer asks for summary of tax invoice data and reports. When RD Officer receives data and documents, they will examine those data and document with the information from the preparation and observation. If all data are consistent, it means that the business has the valid VAT payment. Otherwise the

Copyright by Mahidol University

business has the illegal VAT payment that the RD Officer must continue to inspect the business in detail.

2.2 Use Case Diagram

Use case diagram shows actors and use cases in the existing tax invoice system. This section provides use case diagram of the existing tax invoice system as illustrated in Figure 2.2, and use case descriptions that describe each use case.

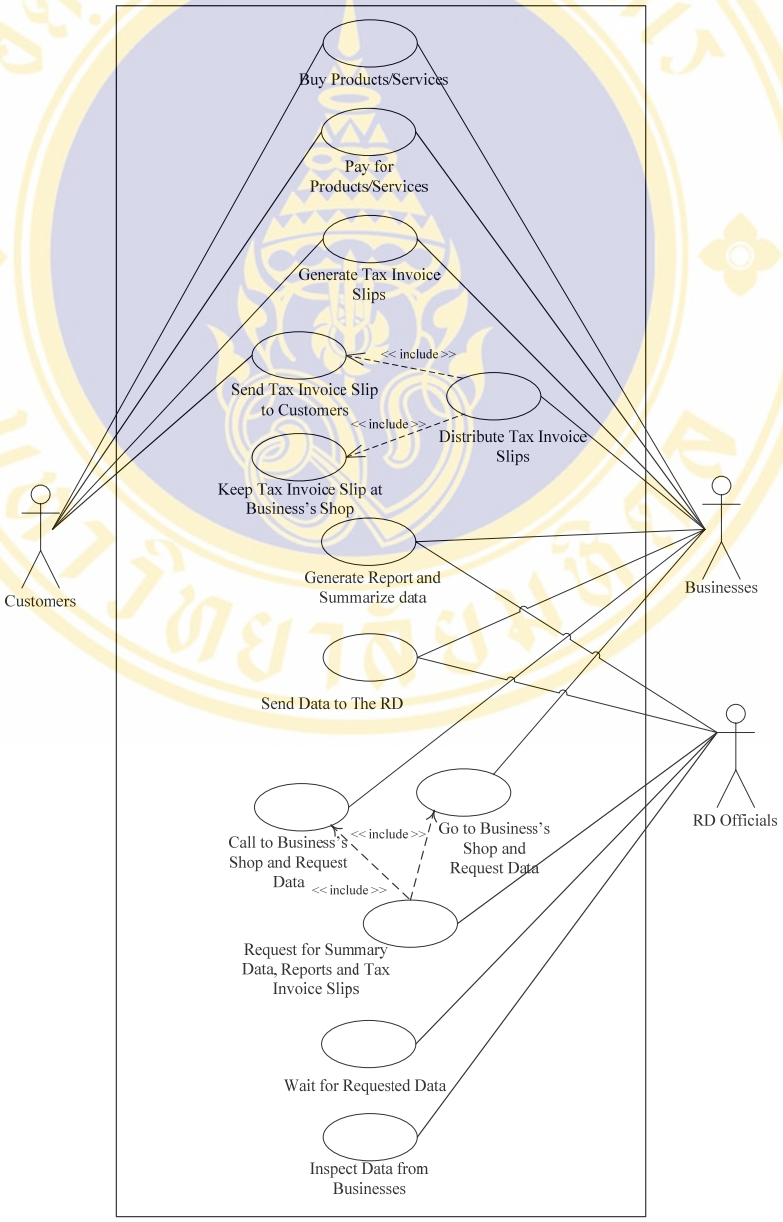


Figure 2.2 Use Case Diagram of The Existing Tax Invoice System

Copyright by Mahidol University

Table 2.1 Buy Products/Services Use Case

Use case name	**Buy Products/Services**
Participating actors	Initiated by Customer Communicates with Business
Flow of event	1. Customer goes to business's shop to purchase products/services. Customer picks desired products/services. 2. When customer chooses all desired products/services, customer will make a payment.
Entry condition	Customer goes to business's shop for purchasing products/services.
Exit Condition	Customer selects products/services and goes to cashier for payment.
Quality requirements	-

Table 2.2 Pay for Products/Services Use Case

Use case name	**Pay for Products/Services**
Participating actors	Initiated by Customer Communicates with Business
Flow of event	1. Customer goes to cashier for payment, business uses machine to compute total amount and VAT. 2. Business tells customer total price (including VAT). Customer gives money to business.
Entry condition	Customer goes to cashier for payment.
Exit Condition	Business receives money from customer.
Quality requirements	-

Copyright by Mahidol University

Table 2.3 Generate Tax Invoice Slips Use Case

Use case name	**Generate Tax Invoice Slips**
Participating actors	Initiated by Business Communicates with Customer
Flow of event	1. When business receives money from customer, business uses machine to generate two copied of tax invoice slips.
Entry condition	Customer gives money to business.
Exit Condition	Business print two copies of tax invoice slips.
Quality requirements	-

Table 2.4 Distribute Tax Invoice Slips Use Case

Use case name	**Distribute Tax Invoice Slips**
Participating actors	Initiated by Business Communicates with Customer
Flow of event	1. After business uses machine to generate and print two copies of tax invoice slip. Business gives one tax invoice slip (master) to customer and keeps the other.
Entry condition	Business uses machine to generate tax invoice slip already.
Exit Condition	Customer receives tax invoice slip from business.
Quality requirements	-

Table 2.5 Send Tax Invoice Slip to Customer Use Case

Use case name	**Send Tax Invoice Slip to Customer**
Participating actors	Initiated by Business Communicates with Customer
Flow of event	1. When business prints two copies of tax invoice, business gives master tax invoice slip to customer.
Entry condition	Business generates and prints tax invoice slip already.

Copyright by Mahidol University

Table 2.5 Send Tax Invoice Slip to Customer Use Case (Cont.)

Use case name	**Send Tax Invoice Slip to Customer**
Exit Condition	Customer receives tax invoice slip already.
Quality requirements	-

Table 2.6 Keep Tax Invoice Slip at Business's Shop Use Case

Use case name	**Keep Tax Invoice Slip at Business's Shop**
Participating actors	Initiated by RD Business Communicates with -
Flow of event	1. After business sends master of tax invoice slip to customer. Business keeps tax invoice slip at business location for RD officer.
Entry condition	Business gives tax invoice slip to customer already.
Exit Condition	Business keeps the copied of tax invoice slip already.
Quality requirements	-

Table 2.7 Generate Reports and Summarize Data Use Case

Use case name	**Generate Reports and Summarize Data**
Participating actors	Initiated by RD officer Communicates with Business
Flow of event	1. Business receives request and conditions from RD officer. 2. Business generates reports using received conditions from RD officer.
Entry condition	Business receives request from RD officer.
Exit Condition	Business generates reports/summary data already.
Quality requirements	-

Copyright by Mahidol University

Table 2.8 Send Data to RD Use Case

Use case name	**Send Data to RD**
Participating actors	Initiated by Business Communicates with RD officer
Flow of event	1. Business sends reports, summary data, and tax invoice slip to RD officer.
Entry condition	Business generates reports/summary data already.
Exit Condition	Business sends reports/summary data to RD officer already.
Quality requirements	-

Table 2.9 Request for Summary Data, Reports and Tax Invoice Slips Use Case

Use case name	**Request for Summary Data, Reports and Tax Invoice Slips**
Participating actors	Initiated by RD officer Communicates with Business
Flow of event	1. RD officer generates conditions for required reports/summary data. 2. RD officer sends request for reports, summary data, and tax invoice slip to business.
Entry condition	RD officer wants to inspect tax invoice data from suspected business.
Exit Condition	RD officer sends request to business already.
Quality requirements	-

Copyright by Mahidol University

Table 2.10 Call to Business's Shop and Request Data Use Case

Use case name	**Call to Business's Shop and Request Data**
Participating actors	Initiated by RD officer Communicates with Business
Flow of event	1. When RD officer suspects business's tax payment, RD officer calls to distrustful business and ask for report.
Entry condition	RD officer suspects business's tax payment.
Exit Condition	RD officer phones to business and asks for reports already.
Quality requirements	-

Table 2.11 Go to Business's Shop and Request Data Use Case

Use case name	**Go to Business's Shop and Request Data**
Participating actors	Initiated by RD officer Communicates with Business
Flow of event	1. When RD officer suspects business's tax payment, RD officer goes to business location and asks for report.
Entry condition	RD officer suspects business's tax payment.
Exit Condition	RD officer goes to Business location and ask for report already.
Quality requirements	-

Table 2.12 Wait for Requested Data Use Case

Use case name	**Wait for Requested Data**
Participating actors	Initiated by RD officer Communicates with -
Flow of event	1. After RD officer sends request to business. RD officer waits for the data.
Entry condition	RD officer sends request to business already.

Copyright by Mahidol University

Table 2.12 Wait for Requested Data Use Case (Cont.)

Use case name	Inspect Data from Businesses
Exit Condition	RD officer receives data already.
Quality requirements	-

Table 2.13 Inspect Data from Businesses Use Case

Use case name	Inspect Data from Businesses
Participating actors	Initiated by RD officer Communicates with -
Flow of event	1. After receives data from business. RD officer audits all data for inspecting business's tax payment.
Entry condition	RD officer receives data.
Exit Condition	RD officer audits all data already.
Quality requirements	-

2.3 Activity Diagram

Activities in the existing tax invoice system are separated to two parts: 1) Generate Tax Invoice Slips Activity Diagram and 2) Inspect Tax Invoice Slips Activity Diagram. They are illustrated in Figure 2.3 and Figure 2.4:

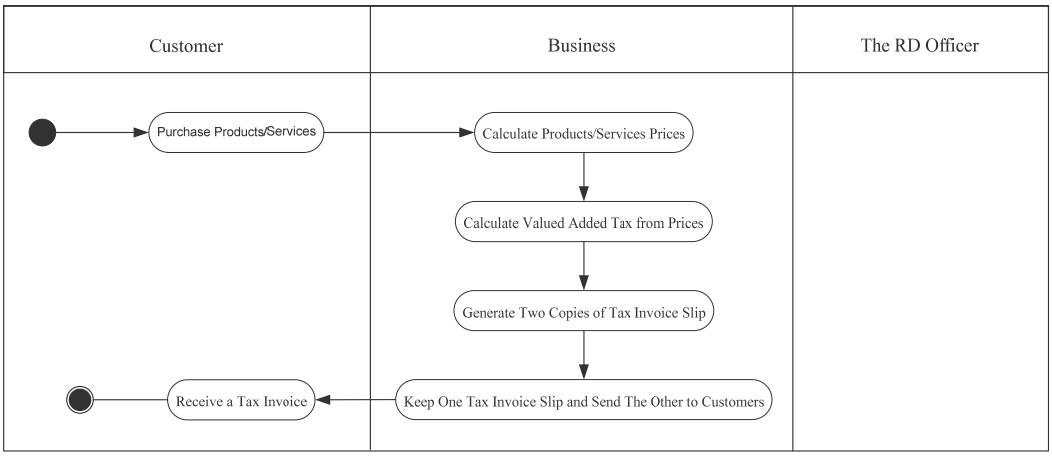


Figure 2.3 Generate Tax Invoice Slips Activity Diagram

Copyright by Mahidol University

Customers go to shop to purchase products/services. When they receive products/services already, business owner computes prices of products/services and Value Added Tax. Then, customers make a payment for products/services. After that, business owner generates and prints two copies of tax invoice slips - one for business owner and the other for customer – and gives tax invoice slip to customer. Business owner keeps tax invoice slip for RD officer to inspect tax payment.

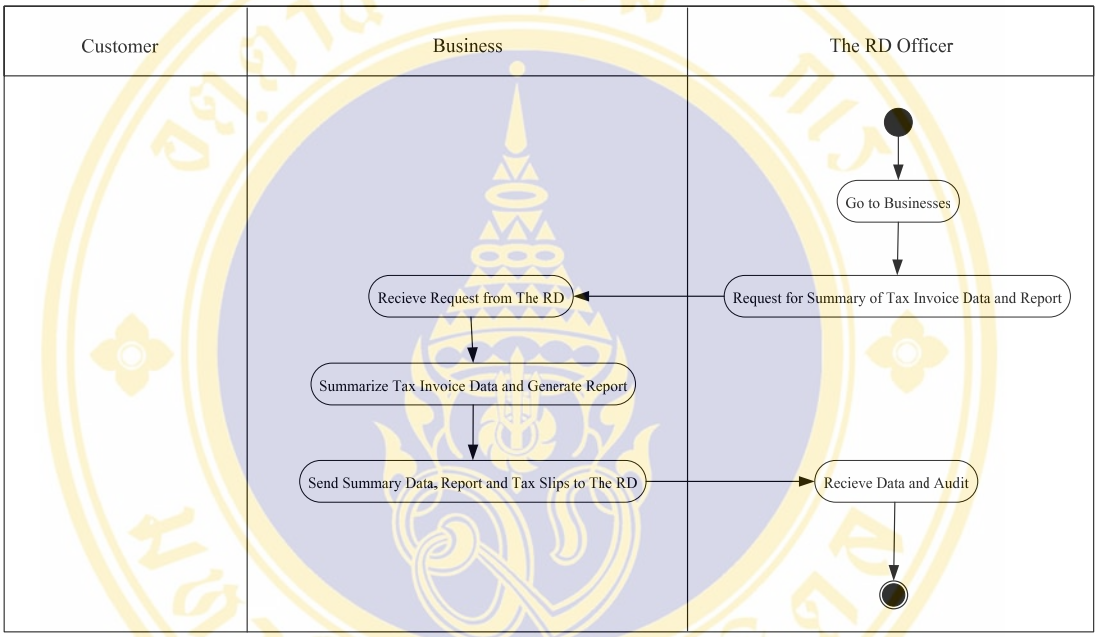


Figure 2.4 Inspect Tax Invoice Slips Activity Diagram

When the RD officer wants to inspect business tax payment, the RD officer goes to the suspected business organization and asks for summary tax invoice data, reports and tax invoice slips. RD officer takes at least one hour for waiting those documents. Otherwise, RD officer calls to the suspected business and ask for data and documents. When the business receives request from the RD officer, the business summarizes data and generates reports. Afterward, business saves these data on CD and sends it to the RD. As a result, supervisors take at least one day for waiting data from businesses and audit them. Moreover, business spends large amount of cost for keeping tax invoice slip (in paper form), especially in large business that have the large numbers of tax invoice slips.

Copyright by Mahidol University

2.4 Tax Invoice Data

Tax invoice data are data contained in tax invoice slip. These data are used to generate reports especially sales report and purchase report. Large businesses keep large amount of tax invoice slips. Currently, there are 338,642 businesses in VAT system [1]. From all businesses in VAT system, there are 2,409 large businesses [2]. For large businesses, they expense large amount of cost to buy paper for printing tax invoice slips and to keep tax invoice slips at business place. For example, 1) Advanced Info Service (AIS) has approximately 1,965,200 customers. Therefore AIS has approximately 1,965,200 transactions per month. Therefore **AIS has approximately** 1,965,200 x 12 = **23,582,400 transactions per year**. Since business must print two copies of tax invoice slips – one for customer and the other for keeping at business- for each transaction, AIS prints tax invoice slips 23,582,400 x 2 = 47,164,800 slips per year and keeps tax invoice slips 23,582,400 slips. 2) 7-Eleven has 2,700 branches [3]. In each day, each branch has tax invoice transaction approximately 1,100 transactions. Therefore, each day 7-Eleven has 1,100 x 2,700 = 2,970,000 transactions. In each month 7-Eleven has 2,970,000 x 30 = 89,100,000 transactions. Thus **7-Eleven has approximately** 89,100,000 x 12 = **1,069,200,000 transactions per year**. In each year, 7-Eleven must print tax invoice slips 1,069,200,000 x 2 = 2,138,400,000 slips. 3) Metropolitan Waterworks Authority: MWA. MWA has approximately 1,607,921 customers [4]. Therefore in each month, MWA has approximately 1,607,921 transactions. Thus **MWA has approximately** 1,607,921 x 12 = **19,295,052 transactions per year**. MWA must print tax invoice slips 19,295,052 x 2 = 38,590,104 slips. These are the example of large businesses that have the same problem. These figures are given in Table 2.14.

Table 2.14 Example of Large Businesses Transaction [3], [4]

Company	Transactions per Year	Generated Tax Invoice Slips	Stored Tax Invoice Slips
AIS	23,582,400	47,164,800	23,582,400
7-Eleven	1,069,200,000	2,138,400,000	1,069,200,000
MWA	19,295,052	38,590,104	19,295,052

Copyright by Mahidol University

2.4.1 Types of Tax Invoice Slip

In tax invoice system, there are two types of tax invoice slips:

2.4.1.1 Full Tax Invoice Slip

Generally, full tax invoice slip is generated when customer requests it. Customer can use full tax invoice slip to get money return from customer's company. Moreover, in case of customers is corporation that must record tax invoice data in company's account.

According to Section 86/4 of the Revenue Code [5], full tax invoice slip must contain data as follows:

1. Word "ใบกำกับภาษี" at the prominent position.
2. Business Detail : name, surname, address, and TIN
3. Customer Detail : name, surname, and address
4. Tax invoice slip's serial number
5. Product Detail : name, type, quantity, and price
6. VAT amount (must completely separate from product amount)
7. Generated date

2.4.1.2 Short Tax Invoice Slip

Commonly, short tax invoice slip is generated by retail business such as 7-Eleven, Tops. Customer can only use short tax invoice slip to review amount price of product and VAT. Customer cannot use short tax invoice slip to get money return from customer's company. In other words, short tax invoice slip is only the evidence that customers receive when they purchase products/services.

According to Section 86/6 of the Revenue Code [6], retail business can generate short tax invoice slip instead of full tax invoice slip. Short tax invoice slip must contain data as follows:

1. Word "ใบกำกับภาษีอย่างย่อ" at the prominent position
2. Business's name or short name

Copyright by Mahidol University

3. Tax invoice slip's serial number
4. VAT amount (must completely separate from product amount)
5. Generated date

Remark: Data contained in tax invoice slip must be in Thai language and use Baht for monetary unit.

2.4.2 Examples of Tax Invoice Slip

From the survey of tax invoice slips from several businesses, there are various styles of tax invoice slip in each business (both in full tax invoice slip and short tax invoice slip). Examples of tax invoice slip are shown as follows:

2.4.2.1 Full Tax Invoice Slip

Figure 2.5 illustrates full tax invoice slip data described in section 2.4.1.1. 1) word "ใบกำกับภาษี" 2) business detail 3) customer detail 4) generate data 5) product detail 6) VAT amount. This tax invoice slip satisfies Section 86/4 [5] of the Revenue Code that is the legal tax invoice slip that can be used in tax invoice system.

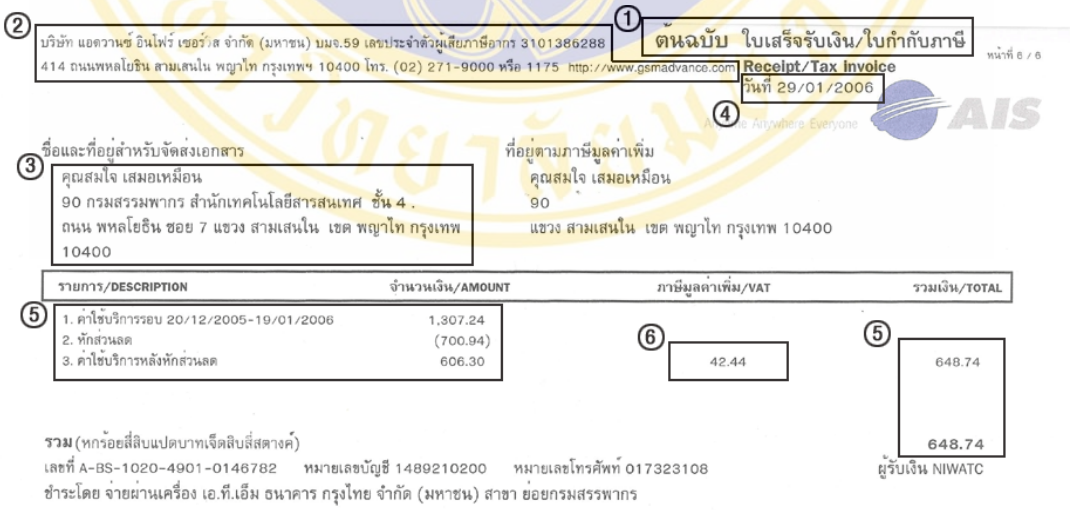


Figure 2.5 Full Tax Invoice Slip Example: AIS

Copyright by Mahidol University

2.4.2.2 Short Tax Invoice Slip

Figure 2.6 illustrates short tax invoice slip data described in section 2.4.1.2. 1) word “ใบกำกับภาษีอย่างย่อ” 2) business detail 3) generate data 4) product detail 5) VAT amount. This tax invoice slip satisfies Section 86/6 [6] of the Revenue Code that is the legal tax invoice slip that can be used in tax invoice system.

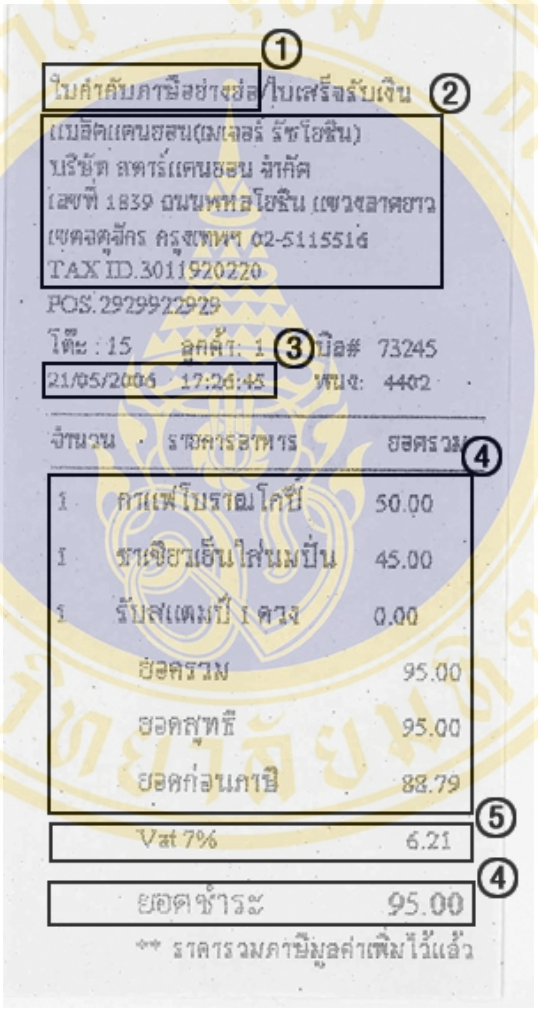


Figure 2.6 Short Tax Invoice Slip Example : Black Canyon

2.4.3 Reports

There are many reports in tax invoice system, for instance, sales report, purchase report. These reports are generated from tax invoice data. Examples of sales report and purchase report are shown in Figure 2.7 and 2.8

Copyright by Mahidol University

รายงานภาษีขาย

เดือนภาษี ....กุมภาพันธ์..... ปี .....2548....

ชื่อผู้ประกอบการ ....นาย นคร วงศาอภินันท์..... เลขประจำตัวผู้เสียภาษีอากร

2	5	4	2	7	8	5	9	2	2

ชื่อสถานประกอบการ .....มารวย เฟอร์นิเจอร์..... ..... สำนักงานใหญ่ สาขาที่

0	0	1	1

ใบกำกับภาษี		ผู้ซื้อสินค้า/ผู้รับบริการ	มูลค่าสินค้าหรือบริการ		จำนวนเงินภาษีมูลค่าเพิ่ม	
วันเดือนปี	เล่มที่/เลขที่					
12/02/2548	202	นายสินบดี เจริญดี	1250	50	87	50
12/02/2548	203	นางสาวจุฑามาศ สามวัลย์	4552	00	318	50
		รวม	5802	50	406	00

Figure 2.7 Sales Report Example

รายงานภาษีซื้อ

เดือนภาษี ....กุมภาพันธ์..... ปี .....2548....

ชื่อผู้ประกอบการ ....นาย นคร วงศาอภินันท์..... เลขประจำตัวผู้เสียภาษีอากร

2	5	4	2	7	8	5	9	2	2

ชื่อสถานประกอบการ .....มารวย เฟอร์นิเจอร์..... ..... สำนักงานใหญ่ สาขาที่

0	0	1	1

ใบกำกับภาษี		ผู้ขายสินค้า/ผู้ให้บริการ	มูลค่าสินค้าหรือบริการ		จำนวนเงินภาษีมูลค่าเพิ่ม	
วันเดือนปี	เล่มที่/เลขที่					
15/02//2548	111	หจก. ศรีสวัสดิ์	2250	50	157	50
15/02/2548	112	นางสาวจุฑามาศ สามวัลย์	5552	00	388	64
		รวม	7802	50	546	14

Figure 2.8 Purchase Report Example

Copyright by Mahidol University

2.5 Tax Invoice File Structure

Table 2.15 and 2.16 show file structure of full tax invoice and short tax invoice used in existing tax invoice system, respectively.

Table 2.15 Full Tax Invoice File

No	Field	Description	Type	Size
1.	TIN	TIN (In case of company), PIN (In case of personal)	Character	13
2.	Invoice_no	Tax Invoice Serial Number	Character	6
3.	Invoice_date	Generated Date	Character	50
4.	Machine_no	Machine number	Character	10
5.	Product_name	Product Name	Character	150
6.	Product _price	Product Price	Float	-
7.	Product_amount_no_vat	Product Amount no VAT	Float	-
8.	Discount	Discount (optional)	Float	-
9.	VAT_amount	VAT	Float	-
10.	Total_amount	Summation of all product's price included VAT	Float	-
11.	Cust_title	Customer's Title	Character	20
12.	Cust_firstname	Customer's Name	Character	100
13.	Cust_surname	Customer's Surname	Character	100
14.	Cust_building_no	Customer's Building Number	Character	3
15.	Cust_addr_no	Customer's Address number	Character	10
16.	Cust_soi	Customer's Soi Name	Character	100
17	Cust_street	Customer's Street Name	Character	100
18	Cust_tambol	Customer's Sub District	Character	100
19	Cust_amphur	Customer's District	Character	100
20	Cust_province	Customer's Province Name	Character	100
21	Cust_zipcode	Customer's Zipcode	Character	5

Copyright by Mahidol University

Table 2.16 Short Tax Invoice File

No	Field	Description	Type	Size
1.	TIN	TIN (In case of company), PIN (In case of personal)	Character	13
2.	Invoice_no	Tax Invoice Serial Number	Character	6
3.	Invoice_date	Generated Date	Character	50
4.	Product_name	Product Name	Character	150
5.	Product_price	Product price	Float	-
6.	Product_amount_no_vat	Product Amount no VAT	Float	-
7.	Discount	Discount (optional)	Float	-
8.	Vat_amount	VAT value	Float	-
9.	Total_amount	Summation of all product's price included VAT	Float	-

2.6 Problems of The Existing Tax Invoice System

There are many problems in existing tax invoice system as follows:

1. Business pays large amount of money to purchase paper for printing tax invoice slips and uses huge warehouse to keep large amount of tax invoice slips.

2. RD Officer takes at least one day to go to business place and wait for reports/summary data.

3. Business can make change/edit reports before send to supervisors. So that, the RD officer received incomplete reports. As a result, the RD cannot inspect business's tax payment effectively.

4. The RD officer doesn't have tax invoice data to support work operations more efficient.

Copyright by Mahidol University

CHAPTER III
LITERATURE REVIEW

This project focuses on applying Web Services technologies to develop e-Tax Invoice System. This chapter reviews background information that will be used in the new system especially Web Services (WS) technologies and security technologies. Moreover this chapter discusses related research works of revenue departments in other countries.

3.1 Web Services

3.1.1 Web Services Definition

The main idea in Web services is the service-oriented architecture (SOA), a loosely coupled mechanism for software providers to make modularized functionality accessible over a network. Web services are an instantiation of SOA that strictly follow XML standards, including the XML schema for data types, the Simple Object Access Protocol (SOAP) for messaging and invocation, the Web Services Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI).

According to [7], Web Services are formally defined as follows:

"Web services are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web Services supports direct interactions with other software agents using XML-based messages exchanged via Internet-Based protocols."

Because of Web Services based on XML standards, Web Services can be developed as loosely coupled application components using heterogeneous platforms, programming languages and protocols. This will facilitates business application

Copyright by Mahidol University

services as a service accessible anytime/anywhere and on different platforms.

3.1.2 Web Services Architecture

The Web Services architecture as shown in Figure 3.1 describes three roles: web service provider, web service consumer and web service broker; and three basic operations: publish, find and bind. A network component can play any or all of these roles.

- **Web Service Provider:** create a Web Services and make it available to clients who want to use it.
- **Web Service Broker:** maintain a registry of published services. This is a searchable registry of service description which provide a way for Web Service Provider and the Web Service Consumer of a Web Services to the Internet
- **Web Service Consumer:** is the client application that consumes the Web Services.

Starting with the web service provider communicates its intent to offer services to the UDDI registry. The web service provider stores a WSDL representation of the service in the registry to be distributed upon request. At some point later, a node wishing to consume a service contacts the UDDI registry to find a web service provider. If the registry finds a match, it returns the WSDL description, which the client interprets. The client then formats a request and forwards it to the web service provider, which returns a response. All the messages and data are in XML and encapsulated in SOAP messages [8], [9], [10].

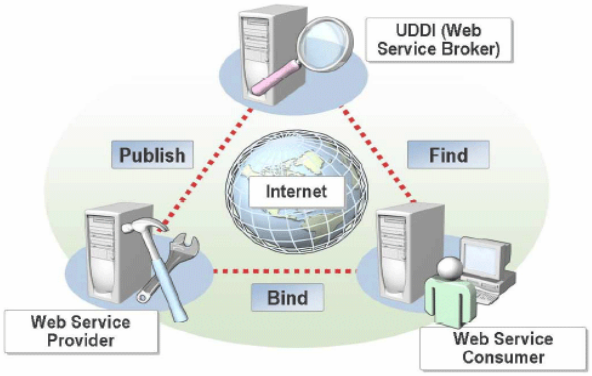


Figure 3.1 Basic Web Service Architecture [11]

Copyright by Mahidol University

3.1.3 Web Services Core Technologies

3.1.3.1 SOAP (Simple Object Access Protocol)

Soap provides a standard packaging structure for transporting XML documents over a variety of standard Internet technologies, including SMTP, HTTP, and FTP. It also defines encoding and binding standards for encoding non-XML. SOAP message consists of three major blocks: the envelope, the header and the body (see Figure 3.2) [12], [13].

1) The envelope is required and marks the start and the end of the SOAP message.

2) The header is optional, and can contain one or more header blocks carrying the attribute of the message. Headers are intended to carry contexts or any application-defined information associated with the message, such as security tokens, transaction identifiers, and message correlation mechanisms.

3) The body is required and contains one or more body blocks comprising the message itself.

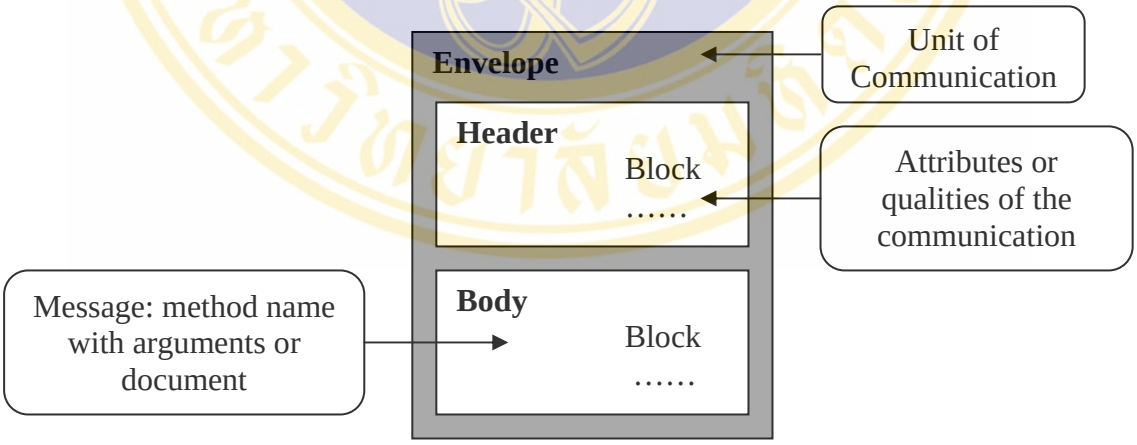


Figure 3.2 Three Parts of SOAP Message [12]

Copyright by Mahidol University

```
<env:Envelope xmlns:env="http://w3.org/2001/12/soap-envelope">
  <env:Header>
    <n:broadcastService xmlns:n="http://www.xmlbus.com/broadcastServices">
      <n:list>PDA, Cell, Email, VoiceMail, IM</n:list>
    </n:broadcastService>
  <env:Header>
  <env:Body>
    <m:Function xmlns:m="http://www.xmlbus.com/broadcastServices/send">
      <m:message>
        Eric, you are late for the concall again!!
      </m:message>
    <m:Function>
  </env:Body>
<env:Envelope>
```

Figure 3.3 An Example of SOAP Message [12]

Figure 3.3 illustrates a simple SOAP message example containing the envelope, an optional header block that defines attributes of the message, and a block that contains the message itself. In this example, the meeting reminder message will be broadcast to the device list by the *send* service located at the address defined by the URL: www.xmlbus.com/broadcastServices Separate namespaces (*broadcast Service* and *function*) are used to qualify the element and attribute for each part of the message. The envelope refers the most recent V1.2 envelope namespace, www.w3.org/2001/12/soap-envelope.

SOAP transports an XML document across the Web and, potentially across other types of networks, to reach a Web Service implementation. A SOAP messages is, in fact, an XML document create in a specific format. A conforming SOAP implementation understands how to interpret such a document and how to map the data to an underlying software implementation of the service. SOAP does not define the service itself, however, but rather just enough about the message so that a SOAP processor can recognize. SOAP defines what it means to recognize the message as a SOAP message, but the Web Service implementation has to know how to interpret the data contained in the message as a target for the underlying software implementation of the service.

Copyright by Mahidol University

3.1.3.2 WSDL (Web Services Description Language)

WSDL is an XML-based language for describing Web services and how to access them. It is a container of all the other elements. Moreover it specifies a targetNamespace attribute and default namespace. The targetNamespace is a convention of XML Schema that enables the WSDL document to refer to itself. All elements in WSDL document have the default WSDL namespace [14].

The WSDL document example is illustrated in Figure 3.4.

```
<?xml version="1.0" encoding="utf-8"?>                    <Definition> Element
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://rdws.rd.go.th/ServiceRD/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://rdws.rd.go.th/ServiceRD/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>                                            <Type> Element
    <s:schema elementFormDefault="qualified"
targetNamespace="http://rdws.rd.go.th/ServiceRD/">
      <s:element name="ReceiveInvoice">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="username" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="password" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ReceiveInvoiceResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ReceiveInvoiceResult">
              <s:complexType>
                <s:sequence>
                  <s:element ref="s:schema" />
                  <s:any />
                </s:sequence>
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
```

Figure 3.4 An Example of WSDL Document

Copyright by Mahidol University

```
<wsdl:message name="ReceiveInvoiceSoapIn">
  <wsdl:part name="parameters" element="tns:ReceiveInvoice" />
 </wsdl:message>
 <wsdl:message name="ReceiveInvoiceSoapOut">
  <wsdl:part name="parameters" element="tns:ReceiveInvoiceResponse" />
 </wsdl:message>
 <wsdl:portType name="DCServiceSoap">
  <wsdl:operation name="ReceiveInvoice">
   <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Online import
business's invocie data to database</wsdl:documentation>
   <wsdl:input message="tns:ReceiveInvoiceSoapIn" />
   <wsdl:output message="tns:ReceiveInvoiceSoapOut" />
  </wsdl:operation>
 </wsdl:portType>
 <wsdl:binding name="DCServiceSoap" type="tns:DCServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="ReceiveInvoice">
   <soap:operation soapAction="http://rdws.rd.go.th/ServiceRD/ReceiveInvoice"
style="document" />
   <wsdl:input>
    <soap:body use="literal" />
   </wsdl:input>
   <wsdl:output>
    <soap:body use="literal" />
   </wsdl:output>
  </wsdl:operation>
 </wsdl:binding>
 <wsdl:binding name="DCServiceSoap12" type="tns:DCServiceSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="ReceiveInvoice">
   <soap12:operation soapAction="http://rdws.rd.go.th/ServiceRD/ReceiveInvoice"
style="document" />
   <wsdl:input>
    <soap12:body use="literal" />
   </wsdl:input>
   <wsdl:output>
    <soap12:body use="literal" />
   </wsdl:output>
  </wsdl:operation>
 </wsdl:binding>
 <wsdl:service name="DCService">
  <wsdl:port name="DCServiceSoap" binding="tns:DCServiceSoap">
   <soap:address location="http://localhost:1045/etax_web2/DCService.asmx" />
  </wsdl:port>
  <wsdl:port name="DCServiceSoap12" binding="tns:DCServiceSoap12">
   <soap12:address location="http://localhost:1045/etax_web2/DCService.asmx" />
  </wsdl:port>
 </wsdl:service>
</wsdl:definitions>
```

<Message> Element

<PortTypee> Element

<Binding> Element

Figure 3.4 An Example of WSDL Document (Cont.)

Copyright by Mahidol University

WSDL Element

WSDL element consists of seven main elements as follows:

1) Definition: The <definition> element must be the root element of all WSDL documents. It defines the name of the web service. The example of this element is depicted in Figure 3.5.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://rdws.rd.go.th/ServiceRD/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://rdws.rd.go.th/ServiceRD/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  .....
</definitions>
```

Figure 3.5 An Example of Definition Element

2) Type: The <type> element responsible for take care of defining the data types that are used by the web service. Types are XML documents, or document parts. Data types address the problem of how to identify the data types and formats that intend to use with Web services. Type information is shared between sender and receiver. The example of this element is illustrated in Figure 3.6.

```
<wsdl:types>
 <s:schema elementFormDefault="qualified" targetNamespace="http://rdws.rd.go.th/ServiceRD/">
  <s:element name="ReceiveInvoice">
   <s:complexType>
    <s:sequence>
     <s:element minOccurs="0" maxOccurs="1" name="username" type="s:string" />
     <s:element minOccurs="0" maxOccurs="1" name="password" type="s:string" />
     <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string" />
    </s:sequence>
   </s:complexType>
  </s:element>
```

Figure 3.6 An Example of Type Element

Copyright by Mahidol University

```
    <s:element name="ReceiveInvoiceResponse">
     <s:complexType>
      <s:sequence>
       <s:element minOccurs="0" maxOccurs="1" name="ReceiveInvoiceResult">
        <s:complexType>
         <s:sequence>
          <s:element ref="s:schema" />
          <s:any />
         </s:sequence>
        </s:complexType>
       </s:element>
      </s:sequence>
     </s:complexType>
    </s:element>
   </s:schema>
  </wsdl:types>
```

Figure 3.6 An Example of Type Element (Cont.)

3) Message: The <message> element describes the data being exchanged between the Web service providers and consumers. Each Web Service has two messages: input and output. Input shows the parameters for the Web Service and output shows the return data from Web Service. Each message contains zero or more <part> parameters, one for each parameter of the Web Service's function and each <part> parameter associates with a concrete type defined in the <types> container element. The example of this element is illustrated in Figure 3.7.

```
 <wsdl:message name="ReceiveInvoiceSoapIn">
   <wsdl:part name="parameters" element="tns:ReceiveInvoice" />
 </wsdl:message>
 <wsdl:message name="ReceiveInvoiceSoapOut">
   <wsdl:part name="parameters" element="tns:ReceiveInvoiceResponse" />
 </wsdl:message>
```

Figure 3.7 An Example of Message Element

4) PortType: The <portType> element combines multiple message elements to form a complete one-way or round-trip operation. For instance <portType> can combine one request and one response message into a single request/response operation. This is most commonly used in SOAP services. A portType can define multiple operations. The example of this element is illustrated in Figure 3.8.

Copyright by Mahidol University

```
<wsdl:portType name="DCServiceSoap">
  <wsdl:operation name="ReceiveInvoice">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Online import
        business's invocie data to database</wsdl:documentation>
    <wsdl:input message="tns:ReceiveInvoiceSoapIn" />
    <wsdl:output message="tns:ReceiveInvoiceSoapOut" />
  </wsdl:operation>
</wsdl:portType>
```

Figure 3.8 An Example of PortType Element

5) Binding: The <binding> element provides specific details on how a portType operation will actually be transmitted over the wire. It can be made available via multiple transports, including HTTP GET, HTTP POST, or SOAP. The bindings provide concrete information on what protocol is being used to transfer portType operations. Further it provide information where the service is located. For SOAP protocol, the binding is <soap:binding>, and the transport is SOAP messages on top of HTTP protocol. The example of this element is illustrated in Figure 3.9.

```
<wsdl:binding name="DCServiceSoap12" type="tns:DCServiceSoap">
 <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
 <wsdl:operation name="ReceiveInvoice">
 <soap12:operation soapAction=http://rdws.rd.go.th/ServiceRD/ReceiveInvoice style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
     <soap12:body use="literal" />
  </wsdl:output>
 </wsdl:operation>
</wsdl:binding>
```

Figure 3.9 An Example of Binding Element

6) Port: A <port> element defines an individual endpoint by specifying a single address for a binding. The example code to specify port is illustrated in Figure 3.10.

Copyright by Mahidol University

```
<wsdl:definitions .... >
   <wsdl:service .... > *
   <wsdl:port name="DCServiceSoap" binding="tns:DCServiceSoap"> *
     <soap:address location="http://localhost:1045/etax_web2/DCService.asmx" />
   </wsdl:port>
   </wsdl:service>
</wsdl:definitions>
```

Figure 3.10 An Example of Port Element

7) Service: The <service> element defines the ports supported by the Web service. For each of the supported protocols, there is one port element. The service element is a collection of ports. It includes a documentation element to provide human-readable documentation. The example of this element is illustrated in Figure 3.11.

```
<wsdl:definitions .... >
  <wsdl:service name="DCService"> *
    <wsdl:port name="DCServiceSoap" binding="tns:DCServiceSoap"> *
      <soap:address location="http://localhost:1045/etax_web2/DCService.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Figure 3.11 An Example of Service Element

3.1.3.3 XML (Extensible Markup Language)

XML is a W3C-recommended general-purpose markup language for creating special-purpose markup languages. It is a simplified subset of SGML, capable of describing many different kinds of data. Its primary purpose is to facilitate the sharing of data across different systems, particularly systems connected via the Internet. Languages based on XML (for example, RDF, RSS, MathML, XHTML and SVG) are themselves described in a formal way, allowing some programs to modify and validate documents in these languages without prior knowledge of their form [12].

Copyright by Mahidol University

XML is a family of technologies: a data markup language, various content models, a linking model, a namespace model, and various transformation mechanisms. The following are significant members of the XML family used as the basis of Web Services:

- **XML schema:**

XML schema is a document that defines the data types, content, structure, and allowed elements in an associated XML document; also used to describe semantic-processing instruction associated with document elements [15].

There are three types of elements in XML schema described as follows:

1) The Schema element is the root element of every XML Schema. The example of <schema> element is shown below:

```
<?xml version="1.0"?>
 <xs:schema>
      ……..
 </xs:schema>
```

2) The Simple Types: the element in this type can contain only text. It cannot contain any other elements or attributes. Moreover this element can be added restrictions (facets) to a data type in order to limit its content, or required the data to match a specific pattern. There are two element types of simple type as follows:

a) Simple Element is an XML element that contains only text. It cannot contain any other elements or attributes. However, the "only text" restriction is quite misleading. The text can be of many different types. It can be one of the types included in the XML Schema definition (boolean, string, date, etc.), or it can be a custom type. Moreover a simple element can be added restrictions (facets) to a data type in order to limit its content, or you can require the data to match a specific pattern.

Copyright by Mahidol University

The *syntax* for defining a simple element is:

```
<xs:element name=" Ename" type=" Etype"/>
```

Where Ename is the name of the element and
Etype is the data type of the element.

b) Attribute is declared as simple types. Simple elements cannot have attributes. If an element has attributes, it is considered to be of a complex type. But the attribute itself is always declared as a simple type.

The *syntax* for defining an attribute is:

```
<xs:attribute name="Aname" type="Atype"/>
```

Where: Aname is the name of the attribute and
Atype specifies the data type of the attribute.

The example of an XML element with an attribute is shown as follows:

```
<Cfname lang="EN">Smith</ Cfname>
```

And the following is the corresponding attribute definition:

```
<xs:attribute name="lang" type="xs:string"/>
```

The most common built-in data types of XML schema are:

- **xs:string:** can contain characters, line feeds, carriage returns, and tab characters.
- **xs:decimal:** is used to specify a numeric value.

Copyright by Mahidol University

- **xs:integer:** is used to specify a numeric value without a fractional component.
- **xs:boolean:** is used to specify a true or false value.
- **xs:date:** is used to specify a date. The date is specified in the following form YYYY-MM-DD where:
    a) YYYY indicates the year
    b) MM indicates the month
    c) DD indicates the day
- **xs:time:** is used to specify a time. The time is specified in the following form "hh:mm:ss" where:
    a) hh indicates the hour
    b) mm indicates the minute
    c) ss indicates the second

3) The Complex Types: the element in this type can contain other elements and/or attributes. The following are the example of element types of complex type.

a) An "Elements-Only" contains an element that contains only other elements. For example, an XML element “Product” that can contain only other elements. The example of “Product” element in XML document is shown as follows:

```
<Product>
   <Productname>Nivea</ Productname >
  < ProductPrice>55.50</ ProductPrice >
</ Product >
```

The “Product” product can be defined in a schema like this:

```
<xs:element name="Product">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Productname" type="xs:string"/>
   <xs:element name="ProductPrice" type="xs:decimal"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

Copyright by Mahidol University

Moreover, the "Product" element can be defined by giving the complexType element a name, and let the "Product" element have a type attribute that refers to the name of the complexType (By using this method, several elements can refer to the same complex type):

```
<xs:element name=" Product " type=" Product type"/>
<xs:complexType name=" Product type">
  <xs:sequence>
    <xs:element name=" Productname " type="xs:string"/>
    <xs:element name=" ProductPrice " type="xs: decimal "/>
  </xs:sequence>
</xs:complexType>
```

b) A "Text-Only" Complex Type contains only simple content (text and attributes), therefore we add a simpleContent element around the content. When using simple content, element must be defined an extension OR a restriction within the simpleContent element, like this:

```
<xs:element name="somename">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="basetype">
        ....
        ....
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>OR

<xs:element name="somename">
  <xs:complexType>
    <xs:simpleContent>
      <xs:restriction base="basetype">
        ....
        ....
      </xs:restriction>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

The following is an example of an XML element, "Productname", that contains text-only:

```
<Productname color="red">Powder</Productname>
```

Copyright by Mahidol University

The "Productname" element shown above is defined in schema as follows: The content is defined as an integer value, and the "Productname" element also contains an attribute named "color":

```
<xs:element name="Productname">
 <xs:complexType>
  <xs:simpleContent>
   <xs:extension base="xs:string">
    <xs:attribute name="color" type="xs:string" />
   </xs:extension>
  </xs:simpleContent>
 </xs:complexType>
</xs:element>
```

Like the "Element-Only" element, the "Text-Only" element can be defined by giving the complexType element a name, and let the "Productname" element have a type attribute that refers to the name of the complexType (By using this method, several elements can refer to the same complex type)

```
<xs:element name="Productname" type="Productnametype"/>
<xs:complexType name="Productnametype">
 <xs:simpleContent>
  <xs:extension base="xs:string">
   <xs:attribute name="country" type="xs:string" />
  </xs:extension>
 </xs:simpleContent>
</xs:complexType>
```

- **Extensible Style Sheet Language Transformations (XSLT):**

XSLT is a language for transforming data extracted from an XML document into a new document. The destination format can be in simple text, HTML or XML. The most common use of XSLT is in conjunction with XML Path Language (XPath) in e-business integration, when its role is to transform XML data between the different XML structures used by each participant. XSLT is also widely used in website content management, to convert raw XML into HTML pages [16].

Copyright by Mahidol University

XSLT can create different types of output documents:

- An HTML document
- Another XML document
- A plain-text document

XSLT consists of a set of rules that determine how an XML document will be represented.

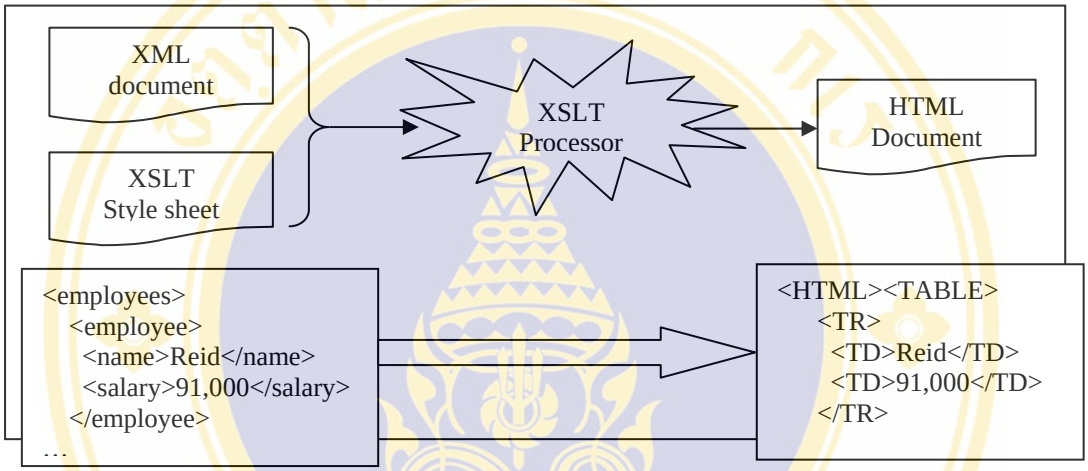


Figure 3.12 XML Transformation using XSLT [17]

XSLT Style Sheet Structure

Most XSLT stylesheet follow a typical structure and contain the following parts:

1) The <xsl:stylsheet> Element is the root element of an XSLT style sheet. The purpose of the <xsl:stylesheet> element is to identify the document as an XSLT document. Moreover, it can supply a version number attribute.

2) Top-Level Elements are a subset of all the XSLT elements defined in the W3C XSL specification. The example of XSLT defined element that is allowed as top-level element is described as follows:

<xsl:template> Element: is used to identify nodes in the source document. In Figure 3.13, <xsl:template> Element is shown. The element specifies Invoice node in XML document.

Copyright by Mahidol University

3) Instruction Elements is any XSLT element that is used in the body of a template rule. Commonly, instruction element consists of either control constructs, such as conditional statements that evaluate parts of the source document, or value placeholders. The example of Instruction Element is described as follows:

<xsl:value-of> Element: is the simplest method to produce output. This element has a *select* attribute that is used to specify the value as output. The select attribute uses any valid XPath expression as an argument.

XSLT Style Sheet Structure is illustrated in Figure 3.13.

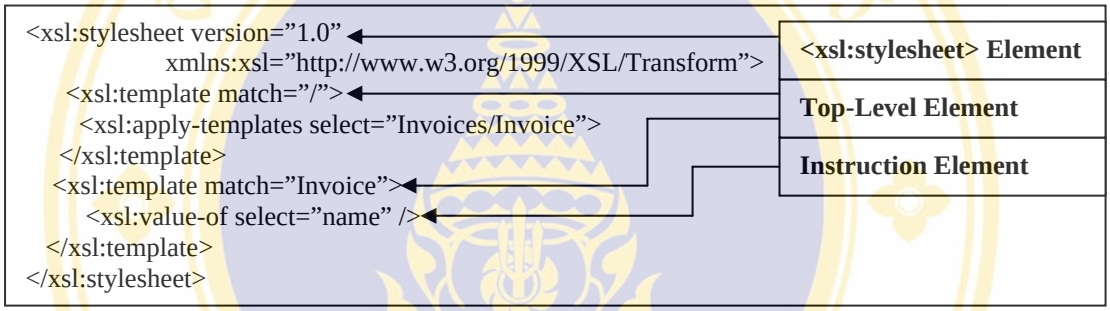


Figure 3.13 XSLT Style Sheet Structure [17]

In additions, XSLT can perform repetition processing by using repetition instruction for instance: **<xsl:for-each**>. This instruction is used to perform the same processing on a series of nodes. This instruction is implemented by using the <xsl:for-each> within a template as syntax below:

```
<xsl:for-each select="expression">
   Element-body
</xsl:for-each>
```

The following template rule matches the <Invoices> element. The <xsl:for-each> instruction loops through the <Invoice> child elements, and outputs the TIN for each invoice.

```
<xsl:template match="Invoices">
   <xsl:for-each select="Invoice">
      <xsl:value-of select="TIN"/><br/>
   </xsl:for-each>
</xsl:template>
```

Copyright by Mahidol University

Furthermore, XSLT defines a function to define the output in one of three possible formats: XML, HTML or Text. The example of XSLT function is described as follows:

Format-number Function: XSLT defines this function to convert a number into a string. This function can be used to control the precise format of a number as it appears in the result document. The syntax for using this function is demonstrated as below:

Format-number(*number, format-string***)**

This function can be used in Instruction Element to define output format of appearance as shown below:

<**xsl:value-of** select=" **format-number(**Product_price,'#.00'**)**" />.

- **XPath:**:

XPath is a standard language for locating and selecting nodes and data in an XML document. XPath is used in conjunction with other key XML technologies whenever there is a need to identify, select, or manipulate nodes. XPath treats an XML document as a tree of nodes. The ordering of nodes corresponds to the ordering of content in the XML document. There are seven node types in XPath:

1) Root Node: lies at the top of the XPath tree. Every XPath tree has a single root node. The root node in XPath has the role as the document node In the DOM.

2) Element Node: represents every element In an XML document. A well-formed XML document has a single document element, and any number of child elements and descendant elements.

Copyright by Mahidol University

3) Attribute Node: is optional node for elements nodes. It can be represented by a separate attribute node.

4) Text Node: is a sequence of consecutive characters defined in an element's content. Any adjacent text is normalized to form a single text node. It always contains at least one character.

5) Namespace Node: describes a namespace definition within scope for each element.

6) Processing instruction Node: provides information to help the process the XML document. Processing instruction use the <? … ?> syntax.

7) Comment Node: contains the comment text enclosed text between the delimiters <!-- and -->.

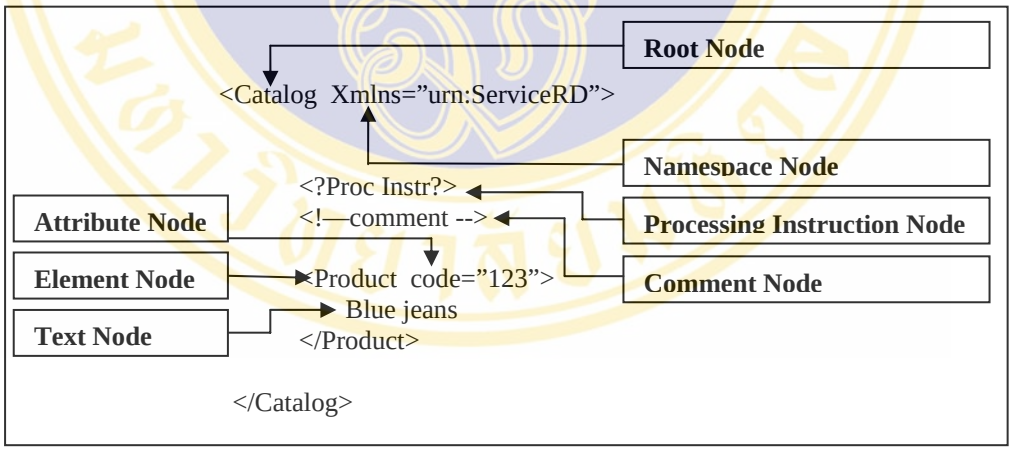


Figure 3.14 Seven Node Types in XPath

To select node in XML document, the special type of XPath expression known as a **location path** is used to select any type of node in XML document, using relative or absolute path. The former defines a location relative to the current node. For example: **product/description**, this relative location path selects child elements named <product>, relative to the current position in the XML document. The location path then selects the <description> child elements of these <product> elements.

Copyright by Mahidol University

For the latter, it defines a location relative to the root node and begins with "/" character. For instance: **/catalog/product**, this example shows an absolute location path. Note that the location path starts with a forward slash character. The location path selects the <catalog> element, relative to the root node. The <product> child elements are then retrieved.

3.1.3.4 UDDI (Universal Description, Discovery, and Integration)

UDDI provides a worldwide registry of web services for advertisement, discovery, and integration purposes. Business analysts and technologists use UDDI to discover available web services by searching for names, identifiers, categories, or the specifications implemented by the web service. UDDI provides a structure for representing business, business relationships, web services, specification metadata, and web service access points.

The UDDI framework defines a data model in XML and SOAP application programming interfaces (APIs) for registering and discovering business information, including the Web Services a business publishes. UDDI is produces by an independent consortium of vendors, founded by Microsoft, IBM, and Ariba, to develop an Internet standard for Web Service description registration and discovery. Microsoft, IBM, Hewlett-Packard, and SAP are hosting the initial deployment of a public UDDI service, which is conceptually patterned after DNS, the internet domain name service that translates Internet host names into TCP addresses. In reality, UDDI is much more like a replicated database service accessible over the Internet.

Business use SOAP to register themselves or others with UDDI; then the registry clients use the SOAP query APIs to search registered information to discover a trading partner. An initial query may return several matched from which a single entry is chosen. Once a business entry is chosen, a final API call is made to obtain the specific contact information for the business [18], [12]. These processes can be depicted in Figure 3.15.

Copyright by Mahidol University

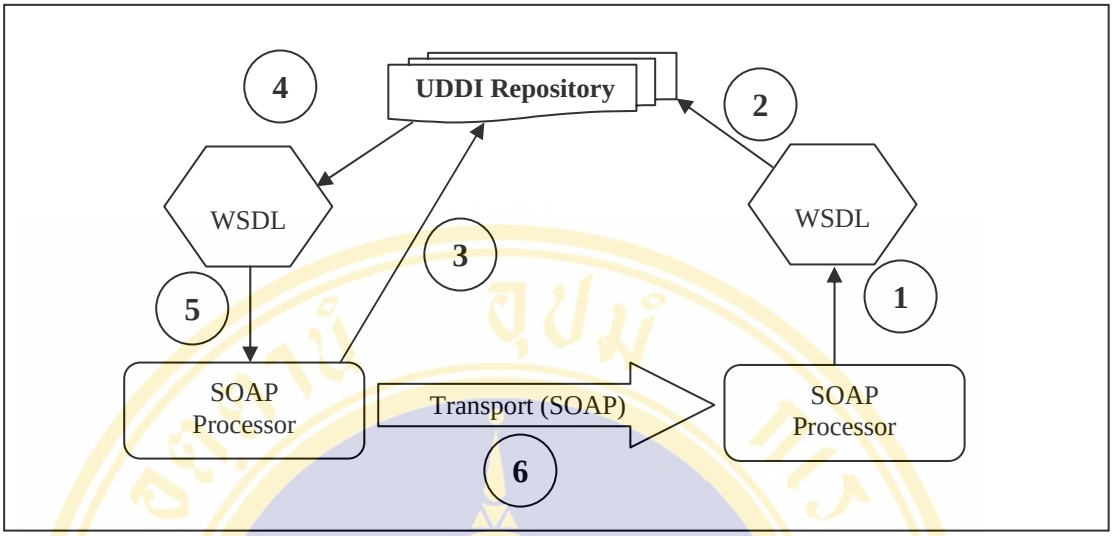


Figure 3.15 The UDDI repository used to discover a Web Service [12]

Figure 3.15 illustrates a business first generates a WSDL file to describe the Web Services supported by its SOAP processor (1) and uses UDDI APIs to register the information with the repository (2). After a business submits its data to the registry, along with other contact information, the registry entry contains a URL that points to the SOAP server site's WSDL or other XML schema file describing the Web Service. Once another business's SOAP processor queries the registry (3) to obtain the WSDL or other schema (4), the client can generate the appropriate message (5) to send to the specified operation over the identified protocol (6). Of course, both client and server have to be able to agree on the same protocol – in this example, SOAP over HTTP – and share the same understanding, or semantic definition of the Web Services, which in this example is represented via WSDL. With the widespread adoption of these fundamental standards, however, this common understanding of WSDL seems ensured.

Copyright by Mahidol University

3.2 Security Technology

3.2.1 Secure Socket Layer (SSL)

The SSL protocol is originally developed by Netscape, to ensure security of data transported and routed through HTTP, LDAP or POP3 application layers [19]. SSL is designed to make use of TCP as a communication layer to provide a reliable end-to-end secure and authenticated connection between two points over a network (for example between the service client and the server). Notwithstanding this SSL can be used for protection of data in transit in situations related to any network service, it is used mostly in HTTP server and client applications.

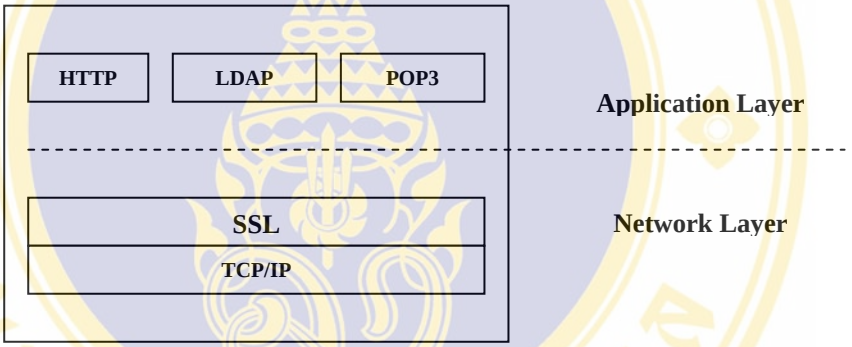


Figure 3.16 SSL between application protocols and TCP/IP [19]

SSL provides services as follows:

- **Authenticating the client and server to each other:** the SSL protocol supports the use of standard key cryptographic techniques (public key encryption) to authenticate the communicating parties to each other. Though the most frequent application consists in authenticating the service client on the basis of a certificate, SSL may also use the same methods to authenticate the client.
- **Ensuring data integrity:** during a session, data cannot be either intentionally or unintentionally tampered with.
- **Securing data privacy:** data in transport between the client and the server must be protected from interception and be readable only by the intended recipient. This prerequisite is necessary for both the data associated with the protocol itself (securing traffic during negotiations) and the application data that is sent during the

Copyright by Mahidol University

session itself. SSL is in fact not a single protocol but rather a set of protocols that can additionally be further divided in two layers:

1) The protocol to ensure data security and integrity: this layer is composed of the SSL Record Protocol,

2) the protocols that are designed to establish an SSL connection: three protocols are used in this layer: the SSL Handshake Protocol, the SSL ChangeCipher SpecPprotocol and the SSL Alert Protocol.

3.2.2 Hash Function

A hash function H is a transformation that takes a variable-size input m and returns a fixed-size string, which is called the hash value h (that is, h = H(m)) [20]. Hash functions with just this property have a variety of general computational uses, but when employed in cryptography the hash functions are usually chosen to have some additional properties.

The basic requirements for a cryptographic hash function are:

- The input can be of any length,
- The output has a fixed length,
- H(x) is relatively easy to compute for any given x,
- H(x) is one-way,
- H(x) is collision-free.

A hash function H is said to be one-way if it is hard to invert, where "hard to invert" means that given a hash value h, it is computationally infeasible to find some input x such that H(x) = h.

The algorithm takes a message of less than 264 bits in length and produces a 160-bit message digest. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.

Copyright by Mahidol University

SHA1 (Secure Hash Algorithm 1) is message-digest algorithm, which takes an input message of any length < 2^64 bits and produces a 160-bit output as the message digest.

Based on the SHA1 RFC document, the SHA-1 is called secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest. Any change to a message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify.

SHA1 algorithm consists of 6 tasks:

Task 1 Appending Padding Bits. The original message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512. The padding rules are:

- The original message is always padded with one bit "1" first.
- Then zero or more bits "0" are padded to bring the length of the message up to 64 bits fewer than a multiple of 512.

Task 2 Appending Length. 64 bits are appended to the end of the padded message to indicate the length of the original message in bytes. The rules of appending length are:

- The length of the original message in bytes is converted to its binary format of 64 bits. If overflow happens, only the low-order 64 bits are used.
- Break the 64-bit length into 2 words (32 bits each).
- The low-order word is appended first and followed by the high-order word

Task 3 Preparing Processing Functions. SHA1 requires 80 processing functions defined as:

```
f(t;B,C,D) = (B AND C) OR ((NOT B) AND D)          (0 <= t <= 19)
f(t;B,C,D) = B XOR C XOR D                         (20 <= t <= 39)
f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D)   (40 <= t <= 59)
f(t;B,C,D) = B XOR C XOR D                         (60 <= t <= 79)
```

Copyright by Mahidol University

Task 4 Preparing Processing Constants. SHA1 requires 80 processing constant words defined as:

```
K(t) = 0x5A827999        (0 <= t <= 19)
K(t) = 0x6ED9EBA1        (20 <= t <= 39)
K(t) = 0x8F1BBCDC        (40 <= t <= 59)
K(t) = 0xCA62C1D6        (60 <= t <= 79)
```

Task 5 Initializing Buffers. SHA1 algorithm requires 5 word buffers with the following initial values:

```
H0 = 0x67452301
H1 = 0xEFCDAB89
H2 = 0x98BADCFE
H3 = 0x10325476
H4 = 0xC3D2E1F0
```

Task 6 Processing Message in 512-bit Blocks. This is the main task of SHA1 algorithm, which loops through the padded and appended message in blocks of 512 bits each. For each input block, a number of operations are performed. This task can be described in the following pseudo code slightly modified from the RFC 3174's method 1:

Input and predefined functions:

```
M[1, 2, ..., N]: Blocks of the padded and appended message
f(0;B,C,D), f(1,B,C,D), ..., f(79,B,C,D): Defined as above
K(0), K(1), ..., K(79): Defined as above
H0, H1, H2, H3, H4, H5: Word buffers with initial values
```

Algorithm:

```
For loop on k = 1 to N
 (W(0),W(1),...,W(15)) = M[k] /* Divide M[k] into 16 words */
 For t = 16 to 79 do:
     W(t) = (W(t-3) XOR W(t-8) XOR W(t-14) XOR W(t-16)) <<< 1
     A = H0, B = H1, C = H2, D = H3, E = H4
     For t = 0 to 79 do:
         TEMP = A<<<5 + f(t;B,C,D) + E + W(t) + K(t)
         E = D, D = C, C = B<<<30, B = A, A = TEMP
     End of for loop
     H0 = H0 + A, H1 = H1 + B, H2 = H2 + C, H3 = H3 + D, H4 = H4 + E
 End of for loop
```

Output:

```
H0, H1, H2, H3, H4, H5: Word buffers with final message digest
```

Copyright by Mahidol University

Output: The contents in H0, H1, H2, H3, H4, H5 are returned in sequence the message digest.

3.3 Web Services Technology in Revenue Departments of Other Countries

The revenue department in other countries tries to enhance services for providing to both business and taxpayer using web technologies, web services technologies and security technologies. As a result, they can provide efficient services to business and taxpayer.

3.3.1 Canada Revenue Agency (CRA)

CRA provides many e-services for individual customer and business. To serve customer more efficiently, CRA gives Goods and Services Tax/Harmonized Sales Tax (GST/HST) filing and remitting options. Taxpayer's GST/HST return and remittance information can be filed electronically using electronic data interchange (EDI) technology. Nowadays, individuals and business can use CRA's secure, automated self-service channels to meet their tax obligations. Internet users can interact with CRA directly or link to CRA's Web site from other government sites. In addition, CRA provide e-form on CRA's website for taxpayers and business. They can download those forms via the Internet. In security issue, CRA has adopted a Public Key Infrastructure (PKI) to preserve the security and integrity of electronic transactions over the Internet. PKI is being implemented for several projects throughout the CRA. Online registration will enable participants to register for PKI certificates over the Internet [21].

3.3.2 HM Revenue & Customs (HMRC)

HMRC provides online services that are quick and convenient for Customer/Business to deal with a whole range of tax, excise and benefit related issues. The example of HMRC's online service is VAT online services, Electronic Binding Tariff Information (eBTI), New Computerized Transit System (NCTS) web service, etc. Customer/Business can register for VAT, make VAT Returns and request

Copyright by Mahidol University

changes to your registration details online via VAT online service. Traders can use eBTI to request a ruling from Customs on the correct Tariff classification on their goods. The NCTS is developed to allow electronic declarations to be made for Community and Common transit. NCTS web service provides the ability to input NCTS declarations and arrivals online. In addition, NCTS web service send data via Secure Socket Layer to protect data from unauthorized user. Moreover, HMRC provides e-form for customer to download on HMRC's website. These services can facilitate Customer/Business more convenient and easier to contact with HMRC [22].

3.3.3 Inland Revenue Authority of Singapore (IRAS)

IRAS provides several e-services to customer for example e-filing, e-stamping, e-Services Authorization System (EASY) -an online business authorization system that allows organizations to authorize staff or third party organizations to access government e-services on their behalf- and MyTaxPortal- an Internet portal that is organized around taxpayer's needs, to perform taxpayer's tax-related transactions with IRAS. MytaxPortal is to allow customer to interact with IRAS with greater ease and convenience as part of IRAS efforts to partner customer in nation building. Before the staff of the business or a third party (e.g. tax agent) can access the e-Services available under MytaxPortal, the company must first authorize the staff or the third party via EASY [23].

3.3.4 Internal Revenue Service (IRS) of the United States

IRS provided Internet e-filing to modernize e-Filing system. This is an important component of IRS's re-engineering efforts to improve service, enhance enforcement, and modernize technology and work processes. It is an internet-based system that uses standardized Extensible Mark-Up Language (XML) constructs that provides corporations the capability to electronically file Forms. Later, IRS enhances existing e-filing system to One-Stop e-Filing, immediate acknowledgement (within 48 hours of e-filing). After IRS deploys One-Stop e-Filing, the number of e-filing increase sharply [24].

Copyright by Mahidol University

3.3.5 DG Tax & Customs (DG)

DG realizes that the efficiency financial report data exchange is the most important thing for the government, business, and taxpayer. Since, there are the various format of financial report. Government departments must entry data to the system when they receive financial report. This may cause the data mistake. To solve this problem, Dutch Taxonomy Project (DTP) is founded. DTP is responsible for developing Financial Reporting Chain by using XBRL, a language for the electronic communication of business and financial data which is revolutionizing business reporting around the world [25] DTP can generate the standard format of financial report in Netherland and Europe. This will save cost and time for financial report data exchange and enhance the efficiency of financial report. In addition, DTP makes the process of data processing and report generating easier and faster. Moreover, DG provides e-import, e-export, and VAT e-filing for business and taxpayer.

3.3.6 The Revenue Department of Thailand (RD)

RD provides several e-services to Customer/Business for example e-Filing, e-Form, Web Services etc. Customer/Business can file personal income tax, corporate income tax, Value Added Tax and Specific Business Tax payment online via the RD website. Moreover, the RD provides electronic forms such as PND90, PND91, PND50, PP30, etc. that Customer/Business can download and print. Customer/Business can use paper forms printed from the RD website to file at area revenue branches. This can help the problems in remote area revenue branches that have insufficient form for Customer/Business. In addition, the RD provides eight Web Services [26] for Customer/Business as follows:

1) **TIN/PIN Verification Service:** Check the validation and existence of Tax Identification Number (TIN) and Personal Identification Number (PIN).

2) **VAT Registrant Info Service:** Provide information of taxpayer who is the Value-added tax registrants. This can help business to validate whether its partner can legally issue the tax invoice in accordance with the tax law. Moreover this WS can verify the existence of the business partner.

Copyright by Mahidol University

3) VAT Return to Tourist Service: Provide information on business that joins VAT refund for tourists. This can prevent forging of tax invoices which can effectively increase tourists' confidence in Thai business.

4) VAT Exempted Jewelry Business Service: Provide the information of jewelry importers registering with the RD who are exempted from VAT. This is useful for the Customs Department to verify the information with the taxpayer and checking those VAT exempted companies, thereby making VAT collection correct and complete.

5) Personal Income Tax Calculation Service: Provide an automatic calculation of Personal Income Tax liability (PND 91).

6) Tax Collection Summary Service: Provide information of the total tax collection of the RD from all tax types and tax return forms.

7) Joint Venture and Foreign Company Service: Provide information on business which have not registered with the Department of Business Development but have registered with the RD. This service will allow agencies to be able to investigate and verify such existence of business.

8) Corporate Income Taxpayer Info Service: Provide basic information of business. The information is grouped by industry sector. In addition a business can get a list of prospective business partners by means of the service.

Presently, these web services are now available on the RD website already. All web services provide information for requestor and they are independently. Customer/Business can develop application to request service from these web services [26].

Copyright by Mahidol University

CHAPTER IV
SYSTEM ANALYSIS AND DESIGN

This chapter presents in details system architecture, approach, and method for developing the e-Tax Invoice System.

4.1 e-Tax Invoice System Architecture

As illustrated in Figure 4.1, the e-Tax Invoice System architecture presents seven core components: User Authenticator (UA), Data Provider (DP), Data Collector (DC), Login Manager (LM), Report Generator (RG), Log Report Generator (LRG), and User Manager (UM). The system is separated into two parts 1) Front-end application for serving Customer/Business 2) Back-end application for supporting RD officer's work processes.

In the first part, there are three Web Services: UA, DP, and DC. From these Web Services, there are two Web Services for Customer/Business: DP and DC. The process starts with Customer/Business sending request message to the system. UA authenticates incoming message using data from RD's User Access List and RD's User Account. If message is authorized UA will record user's usage information to the RD's Log. Afterward, UA sends authorized message to requested web service. Customer/business can develop applications to request tax invoice data from DP for generating tax invoice slip online. The DC provides for only business. Business can send tax invoice data to the system online. These two components use SOAP message to communicate with Customer/Business. For security aspect, message is sent in Secure Socket Layer (SSL) channel to encapsulate data from unauthorized user. Furthermore, these two web services have interaction with existing RD's Web Services: TIN/PINService and VATService.

Copyright by Mahidol University

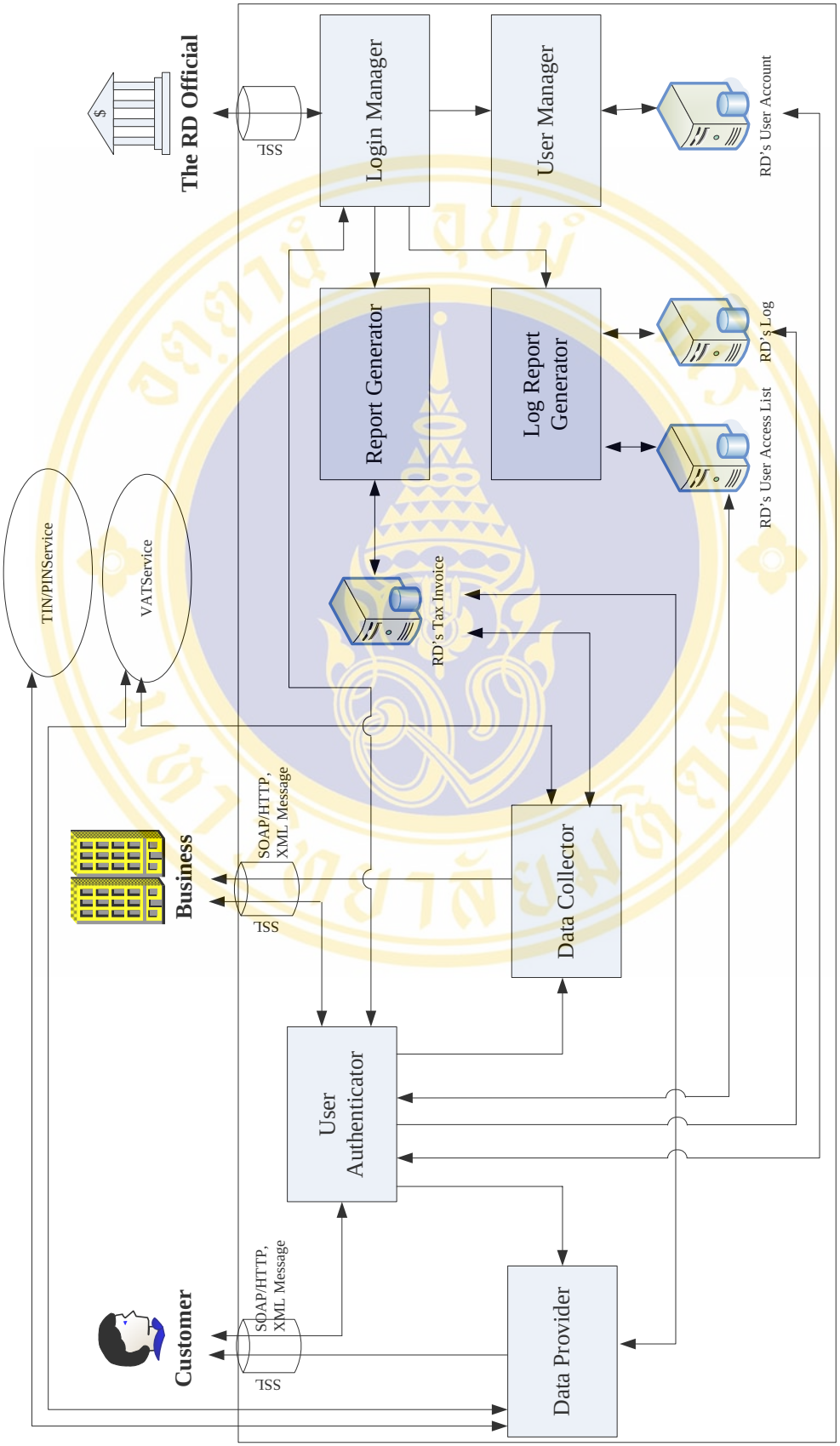


Figure 4.1 e-Tax Invoice System Architecture

Copyright by Mahidol University

For the second part, there are two Web Services - RG and LRG - and two components - LM and UM -. These are used by RD officer. LM allows RD officer to login to the system. When RD officer has login already, RD officer can choose operation: view report provided by RG, view log report provided by LRG and add/delete/update user account provided by UM.

4.2 Structure Chart

The structure chart of e-Tax Invoice System is shown in Figure 4.2. The system composes of seven modules: User Authenticator, Data Collector, Data Provider, Login Manager, Report Generator, Log Report Generator, and User Manager, as illustrated below:

4.2.1 User Authenticator (UA)

User Authenticator is responsible for authenticating client and logging user's usage information. The process of UA is described as follows:

Starting with client: customer/business sends request message to the system. UA authenticates username and password. UA will verify that user has the right to access requested web services. If user has the right to access UA will send requested message to web service specified in requested message. Otherwise UA will return denied message to the client

4.2.2 Data Collector (DC)

Data Collector is responsible for collecting tax invoice data from allied business and sending the collected status message back to client. The process of DC is described as follows:

Copyright by Mahidol University

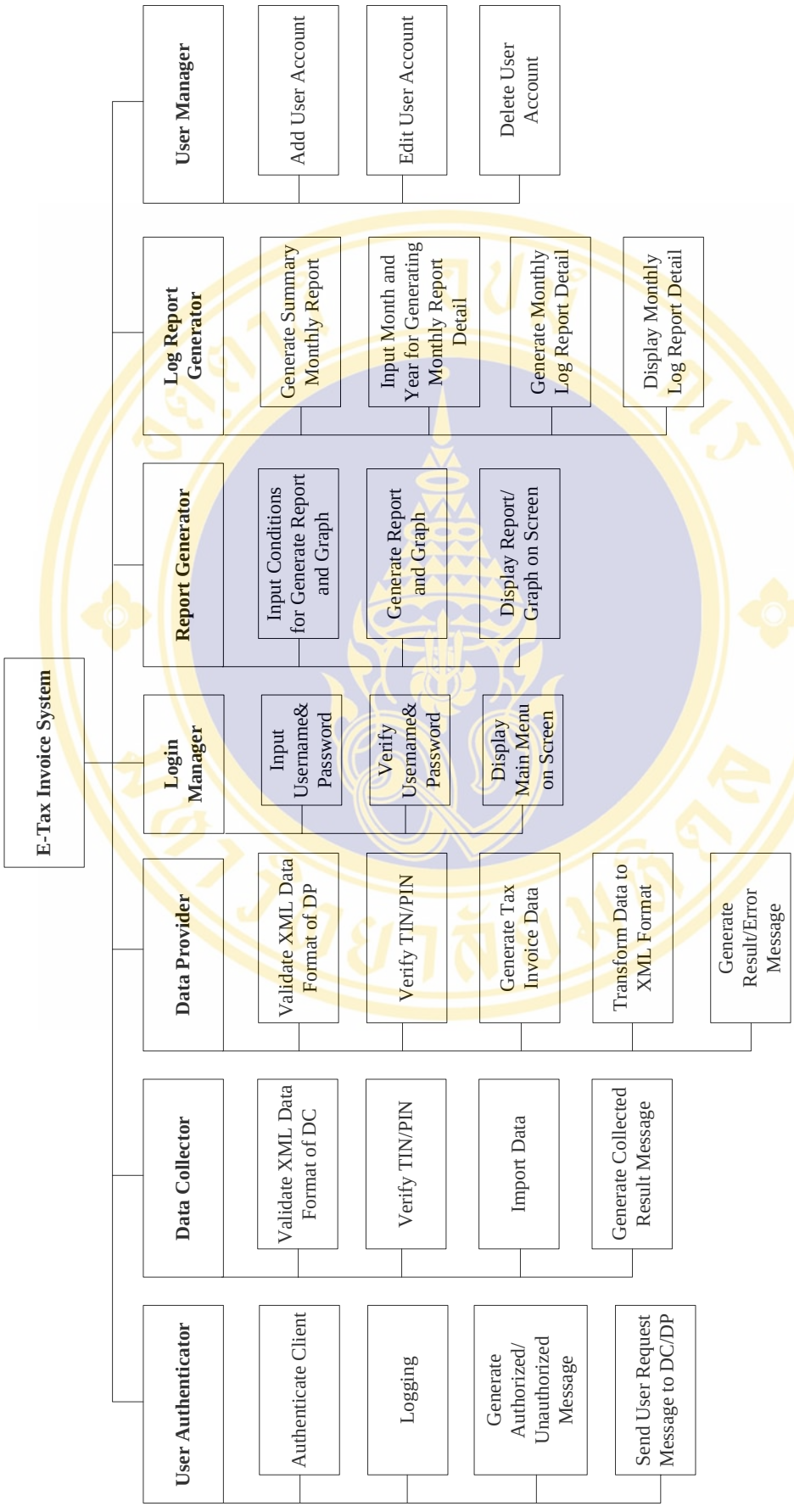


Figure 4.2 e-Tax Invoice System Structure Chart

Copyright by Mahidol University

Starting with business browses to the RD's website and download DC's WSDL document that describes DC's services description, and message format to communicate with DC. Afterward, business implements application for sending tax invoice transaction data online. When DC received request, DC validates message format. If valid, DC transforms data to dataset and imports to the RD's Tax Invoice database. After that, DC generates collected result message and returns to the client. Otherwise, DC generates denied message and returns to client.

Remark: Message is SOAP message and its format satisfies defined message format in WSDL document published by The Revenue Department.

4.2.3 Data Provider (DP)

Data Provider is responsible for generating tax invoice data that corresponds to specified criteria and sending back to client in XML format. The process of DP is described as follows:

Starting with customer browses to the RD's website and download DP's WSDL document. Afterward, customer implements application to request tax invoice data for generating tax invoice slip online. When DP received message from client already, DP validates data format. If valid, DC queries data using criteria specified in message and generates data as response message. After that, DC return message to client. Otherwise DC generates denied message, and returns to client.

4.2.4 Login Manager (LM)

Login Manager is responsible for user login. The process of LM is described as follows:

Starting with, RD officer inputs username and password to login to the system, LM verifies username and password. If username and password are valid, UM authorizes RD officer to access the system. Otherwise UM show login form with denied message for user to input username and password again.

Copyright by Mahidol University

4.2.5 Report Generator (RG)

Report Generator is responsible for creating tax invoice report and shows it on screen. Moreover, RG can generate graph and display on screen. Report provided by RG can support the RD's work operations more efficient. The process of RG is described as follows:

Starting with, RD officer specifies criteria for generating reports and graphs. When RG receives conditions, RG queries data that corresponded to specified criteria. Afterward, RG generates reports and graphs and shows on screen. These reports and graphs can help RD officers make a decision in tax payment auditing.

4.2.6 Log Report Generator (LRG)

Log Report Generator is responsible for generating log report and display on screen. The process of LRG is described as follows:

Starting with, RD officer clicks on log menu, LRG shows summary monthly log report on screen. If RD officer wants to see detail of each month, he/she can click at month's name to view log detail.

4.2.7 User Manager (UM)

User Manager is responsible for user manipulation. RD officer can add/delete/update user account in the system. The process of UM is described as follows:

Starting with, RD officer clicks on user menu. UM shows all users list on screen. There are three operations that RD officer can choose: 1) Add user account 2) Edit user account and 3) Delete user account. Only authorized RD officer has the right to use these operations.

Copyright by Mahidol University

4.3 Use Case Diagram

There are ten use cases in the e-Tax invoice system. These use cases are illustrated in Figure 4.3

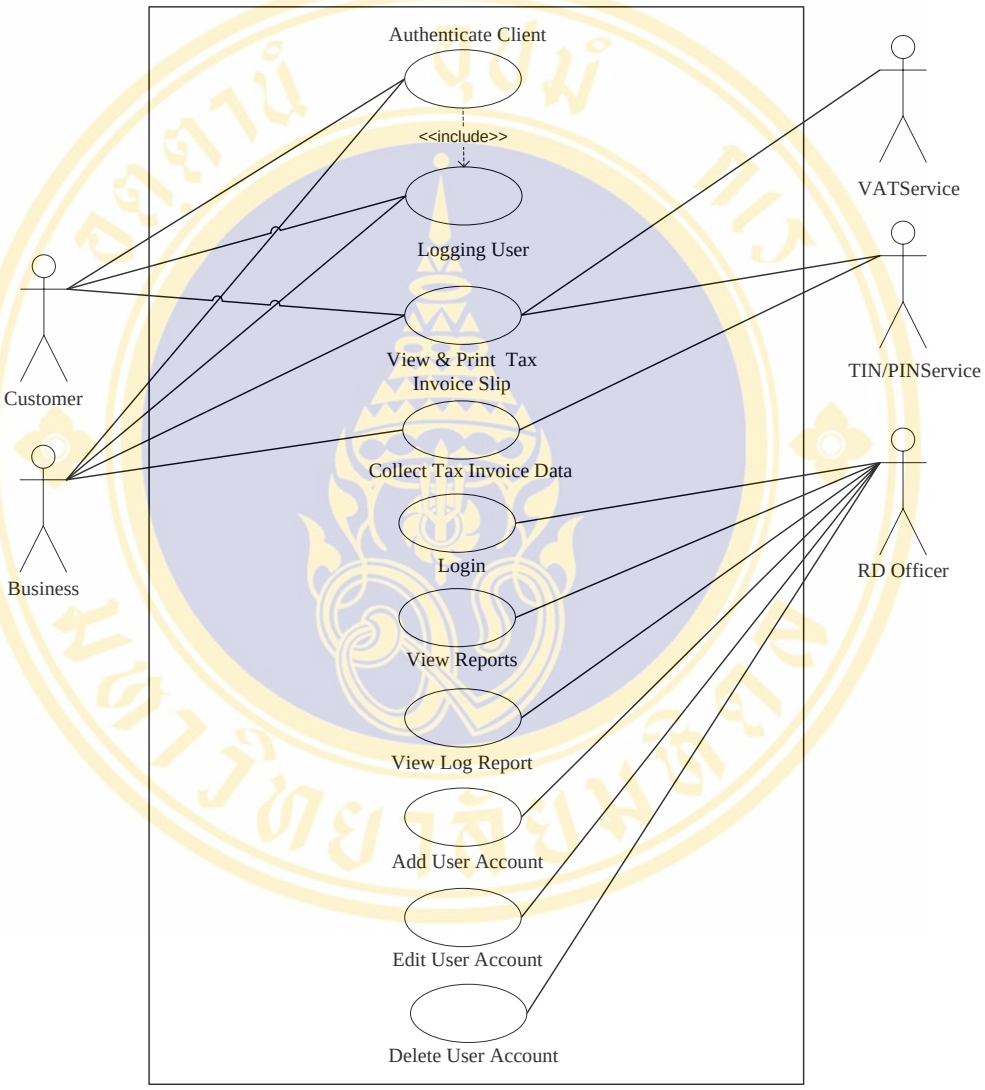


Figure 4.3 e-Tax Invoice System Use Case Diagram

Copyright by Mahidol University

Table 4.1 Authenticate Client Use Case

Use case name	**Logging User**
Participating actors	Initiated by Customer/Business
Flow of event	1. Customer/Business sends a request message to request service from the system. 2. The system authenticates the request from Customer/Business. 3. If the request is authorized, the system returns authorized message. Otherwise the system returns unauthorized message.
Entry condition	Customer/Business sends a request message to the system
Exit Condition	Request from Customer/Business is authenticated by the system.
Quality requirements	-

Table 4.2 Logging User Use Case

Use case name	**Logging User**
Participating actors	Initiated by Customer/Business
Flow of event	1. When Customer/Business is authorized the system will log user usage to database for generating log report.
Entry condition	Customer/Business is authorized by the system.
Exit Condition	The system logs user usage to database.
Quality requirements	-

Copyright by Mahidol University

Table 4.3 View & Print Tax Invoice Slip Use Case

Use case name	**View & Print Tax Invoice Slip**
Participating actors	Initiated by Customer/Business Communicates with VATService, TIN/PINService
Flow of event	1. Customer/Business develops application to send a request message composed of username, password, and conditions for querying tax invoice data from the system. 2. The system validates received XML data format. 3. After that, the system sends the request to TIN/PINService for verifying whether received TIN or PIN is valid or not. 4. If above two step are valid, the system queries tax invoice data corresponded to received conditions and sends request to VATService to get business's information. 5. The system converts data to XML format. Otherwise, the system generates denied message. Finally the system returns results to the caller to be used for generate tax invoice slip.
Entry condition	Customer/Business requests for tax invoice data
Exit Condition	Customer/Business receives result message from the system.
Quality requirements	-

Copyright by Mahidol University

Table 4.4 Collect Tax Invoice Data Use Case

Use case name	**Collect Tax Invoice Data**
Participating actors	Initiated by Business Communicates with TIN/PINService
Flow of event	1. Business generates collected tax invoice data and converts to XML format. And then business sends it as a request message to the system. 2. When the system is activated by the request message from business, the system validates data format. If valid, the system transforms data to dataset and imports to database. After that, the system generates collected result message and returns to the caller.
Entry condition	Business sends request message to the system.
Exit Condition	Business receives collected result message.
Quality requirements	-

Table 4.5 Login Use Case

Use case name	**Login**
Participating actors	Initiated by RD officer
Flow of event	1. RD officer browses to the system. The system displays login page. 2. RD officer fills in user and password and presses submit button. 3. The system authenticates user and check user's right. If user is authorized and has the right to login to the system, the system will show main menu for user to select operations: view report, view log report and mange user account. Otherwise the system will show

Copyright by Mahidol University

Table 4.5 Login Use Case (Cont.)

Use case name	**Login**
Flow of event	login page with denied message for user to fill in username and password again.
Entry condition	RD officer browses to the system.
Exit Condition	Authorized RD officer can login to the system.
Quality requirements	-

Table 4.6 View Reports Use Case

Use case name	**View Reports**
Participating actors	Initiated by RD officer
Flow of event	1. RD officer chooses operations from main menu to view reports and graphs. 2. RD officer inputs criteria to generate reports and graphs and presses search button. 3. The system queries tax invoice data corresponded to input criteria and generates reports and graphs. 4. The system shows reports and graphs on screen.
Entry condition	RD officer inputs criteria for generating reports and graphs.
Exit Condition	RD officer views reports/graphs.
Quality requirements	-

Copyright by Mahidol University

Table 4.7 View Log Report Use Case

Use case name	**View Log Report**
Participating actors	Initiated by RD officer
Flow of event	1. RD officer chooses log menu. 2. The system displays summary monthly report on screen. After that RD officer can choose month and year to view in detail of each month.
Entry condition	RD officer selects log menu.
Exit Condition	RD officer views log report.
Quality requirements	-

Table 4.8 Add User Account Use Case

Use case name	**Add User Data**
Participating actors	Initiated by RD officer
Flow of event	1. RD officer clicks on add user menu. 2. The system shows add user form. 3. RD officer fills in form with user information and then submits form. 4. The system records user account to database.
Entry condition	RD officer clicks add user menu.
Exit Condition	The system records user account to database.
Quality requirements	-

Copyright by Mahidol University

Table 4.9 Edit User Account Use Case

Use case name	**Edit User Data**
Participating actors	Initiated by RD officer
Flow of event	1. RD officer clicks on user account for editing. 2. The system displays form with selected user information. 3. RD officer edits user information and submits form. 4. The system saves changed user information to database.
Entry condition	RD officer clicks on user account for updating data.
Exit Condition	The system saves updated user detail to database.
Quality requirements	-

Table 4.10 Delete User Account Use Case

Use case name	**Delete User Data**
Participating actors	Initiated by RD officer
Flow of event	1. RD officer clicks on delete icon on the user account row. 2. The system shows message dialog to confirm user deletion. 3. The system removes user account from database if RD officer chooses to confirm user removal. Otherwise not doing anything
Entry condition	RD officer clicks on delete icon on the user account row.
Exit Condition	The system removes user account from database.
Quality requirements	-

Copyright by Mahidol University

4.4 Activity Diagram

Activity diagram describes activities in the system. There are nine activities diagram in the e-Tax invoice system illustrates follows:

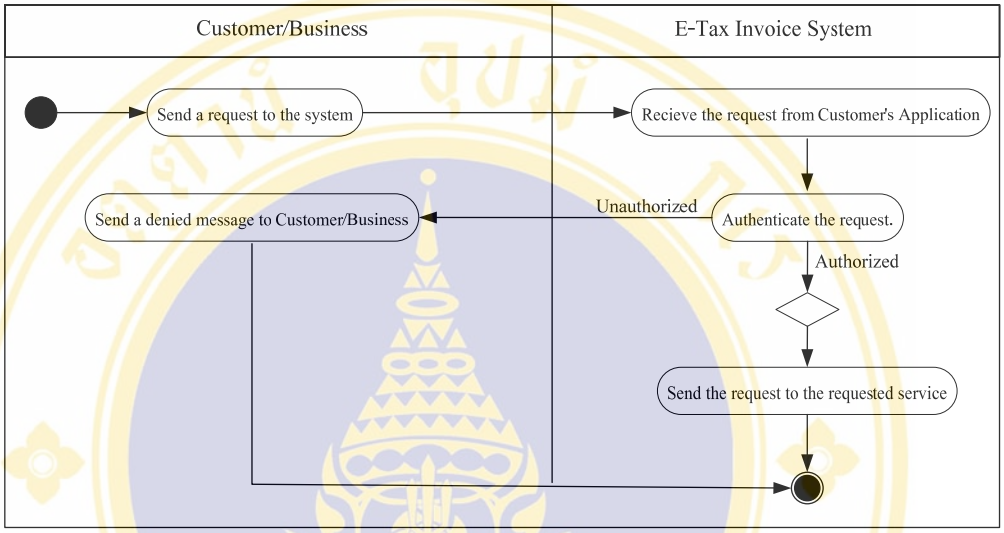


Figure 4.4 Authenticate Client Activity Diagram

Figure 4.4 illustrates activities in Authenticate Client. Starting with, customer/business sends a request to the system. When the system received the request from customer/business, if the request is authenticated, the system will send request to the desired service. Otherwise, the system will generate denied message and send back to customer/business.

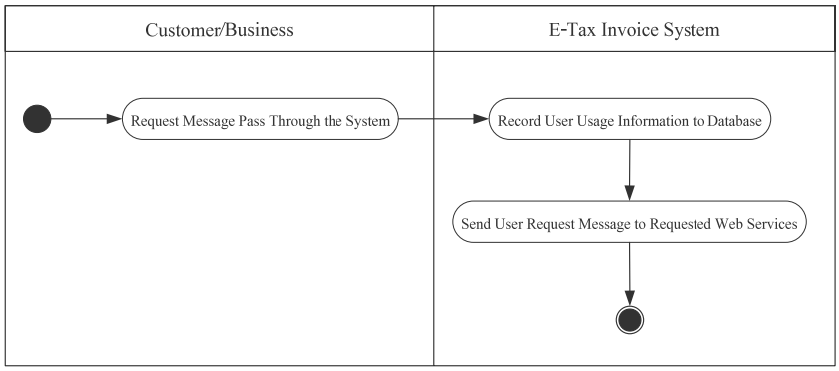


Figure 4.5 Logging User Activity Diagram

Copyright by Mahidol University

Figure 4.5 illustrates activities in Logging Report. Starting with, Customer/ Business sends request to the system. When message is authorized by the system, the system saves user's usage information to RD's Log database. After that, this system sends message to service specified in message.

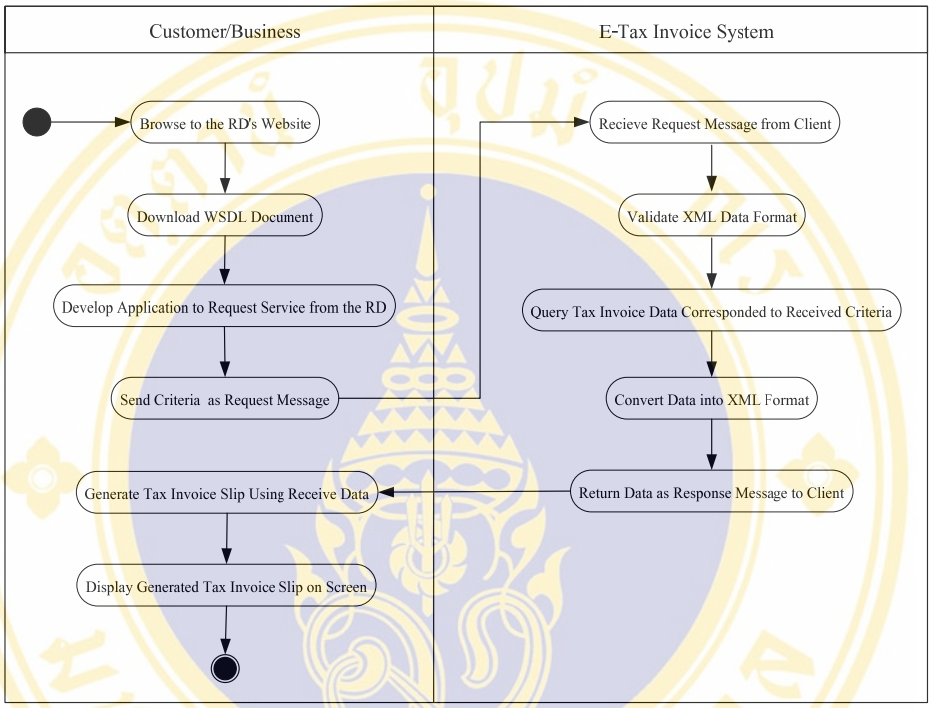


Figure 4.6 View & Print Tax Invoice Activity Diagram

Figure 4.6 illustrates activities in View & Print Tax Invoice. Starting with, customer/business browses to the RD's website and downloads WSDL. Afterward, customer/business develops application to request service from RD's web service. When the system received request message from customer/business, the system validates XML data format with defined XML data format. If valid data format, the system queries tax invoice data from the RD's tax invoice database, converts data into XML format and sends back to client as response message. Otherwise, the system generates denied message and return to sender.

Copyright by Mahidol University

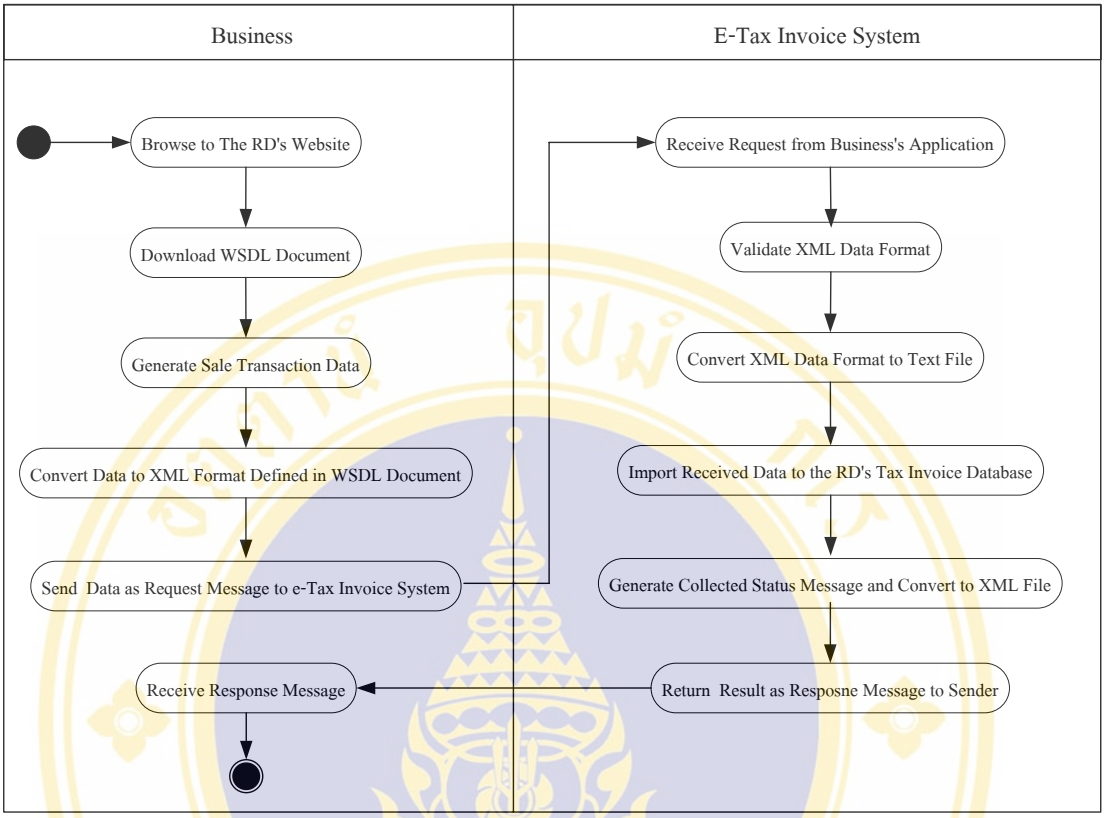


Figure 4.7 Collect Tax Invoice Data Activity Diagram

Figure 4.7 illustrates activities in Collect Tax Invoice Data. First, the business browses to the RD's website and downloads WSDL. Afterward the business develops application to generate and send collected tax invoice data to the RD online. When the system received message the system validates data format. If valid, the system transforms data into dataset and imports to the RD's tax invoice database and generate collected result message. Otherwise the system generates uncompleted message and sends back to sender.

Copyright by Mahidol University

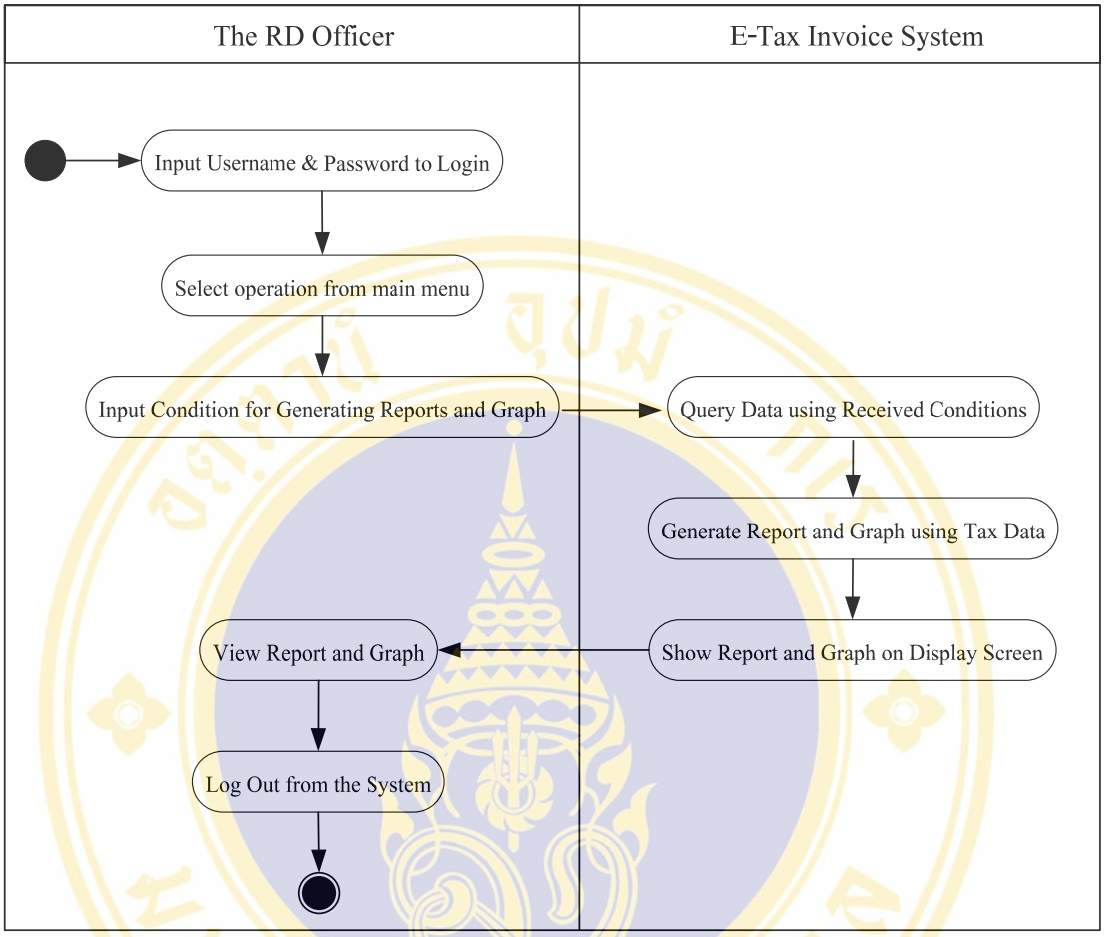


Figure 4.8 View Reports Activity Diagram

Figure 4.8 illustrates activities in View Reports. First, RD officer logins to the system. Afterward, RD officer selects operation from main menu. Then, RD officer inputs criteria for generating reports and graphs. The system queries tax invoice data using received conditions and generates reports and graphs. Then the system shows reports and graphs on screen.

Copyright by Mahidol University

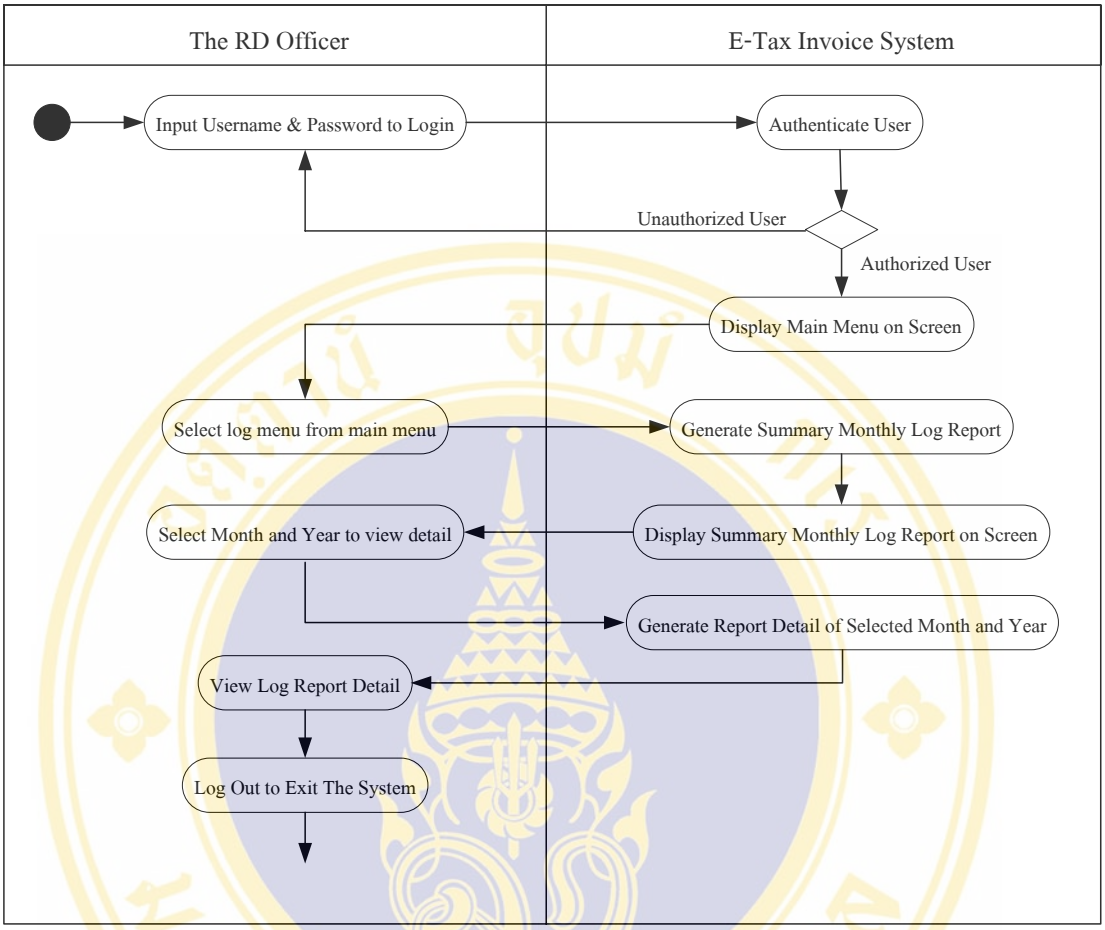


Figure 4.9 View Log Report Activity Diagram

Figure 4.9 illustrates activities in View Log Report. Starting with, RD officer clicks on log menu. The system shows summary monthly report on screen. RD officer can clicks on month name to view detail of each month.

Copyright by Mahidol University

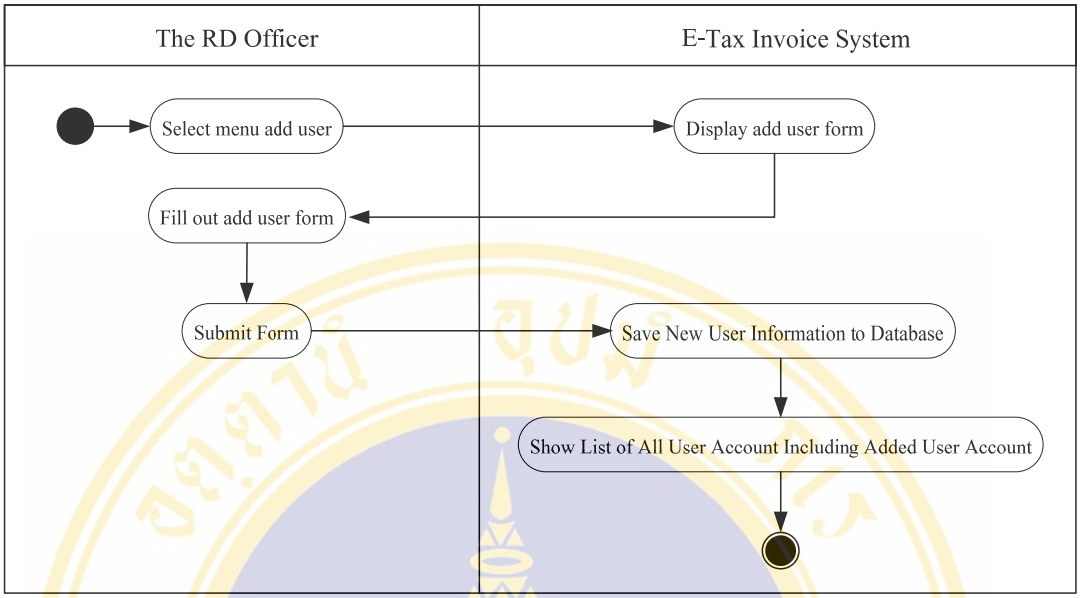


Figure 4.10 Add User Account Activity Diagram

Figure 4.10 illustrates activities in Add User Account Starting with, RD officer clicks on add icon. The system will show add user form. After that, RD officer fills in user information and submits form. The system saves new user information to RD's User Account database.

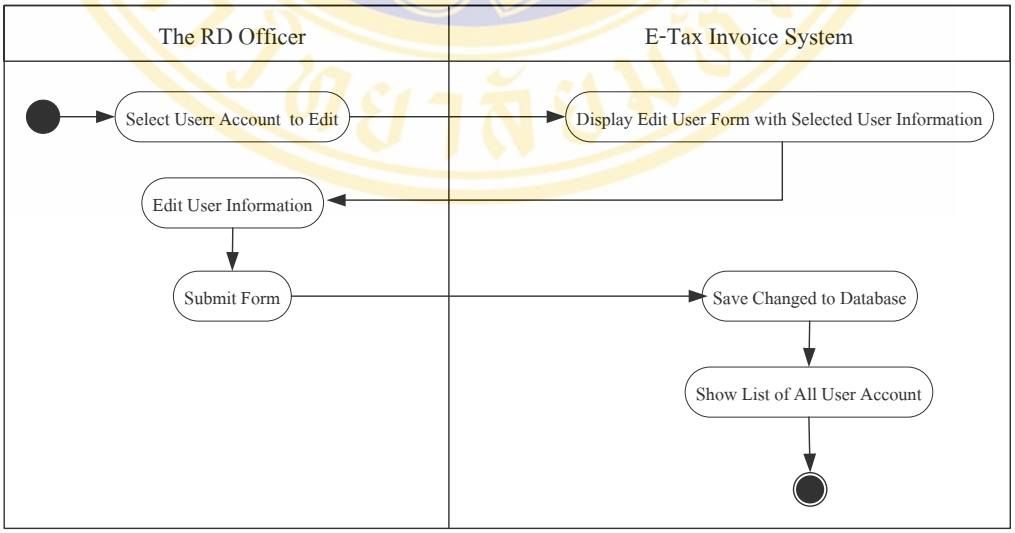


Figure 4.11 Edit User Account Activity Diagram

Copyright by Mahidol University

Figure 4.11 illustrates activities in Edit User Account Starting with, RD officer clicks on username. The system will show edit user form with selected user information. After that, RD officer edits user information and submits form. The system saves change to RD's User Account database.

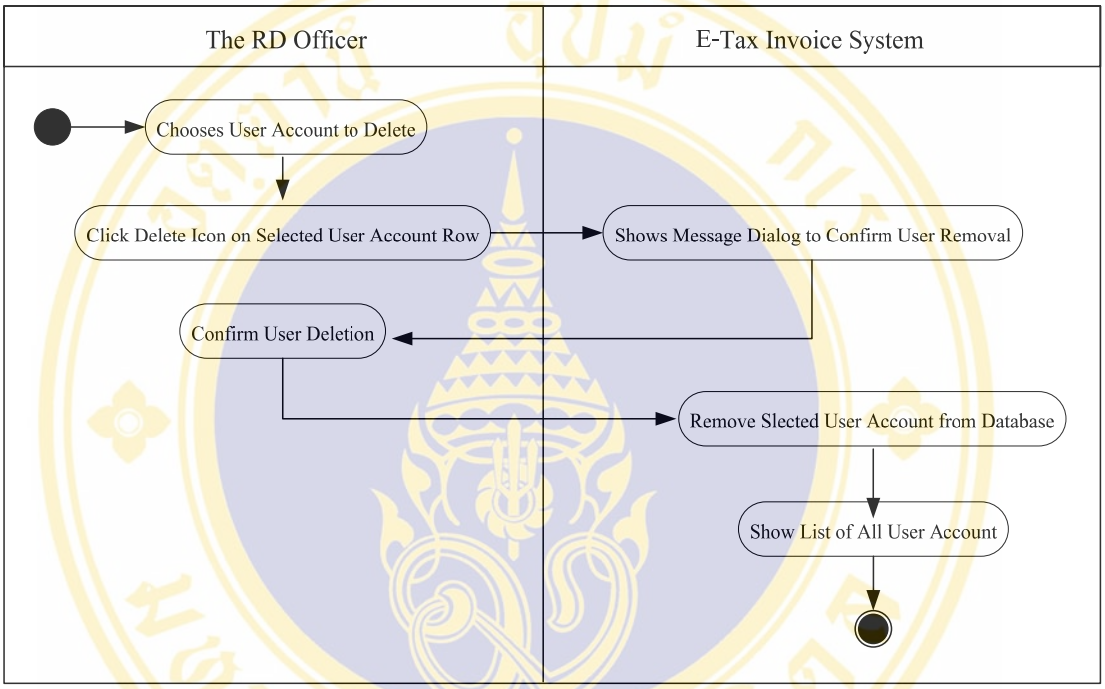


Figure 4.12 Delete User Account Activity Diagram

Figure 4.12 illustrates activities in Delete User Account Starting with, RD officer clicks delete icon on user row. The system shows message dialog to confirm user removal. In case of RD officer confirms to delete. The system removes selected user account from RD's User Account database and shows list of all user account.

Copyright by Mahidol University

4.5 Class Diagram

There are seven data tables used in e-Tax Invoice System described as follows:

Table 4.11 List of Data Tables

Table Name	Description
1. Invoice	Contains the information of tax invoice slip sent from business. It contains tax invoice description used to show in generated tax invoice.
2. InvoiceProduct	Contains the information of product in tax invoice slip. It contains product description used to shows in generated tax invoice.
3. BusinessProfile	Contains the information of business that applied to use e-Tax Invoice System. It contains business description used to shows in generated tax invoice.
4. UserAuthentication	Contains the information of user. It contains username and password used to authenticate user.
5. UserAccess	Contains the information of user's access right. It used to check user's access right.
6. WService	Contains the information of Web Services in the system.
7. WSLog	Contains the information of user's usage log. It is used to generate report for RD officer.

These tables can be depicted in Class Diagram as shown in Figure 4.13

Copyright by Mahidol University

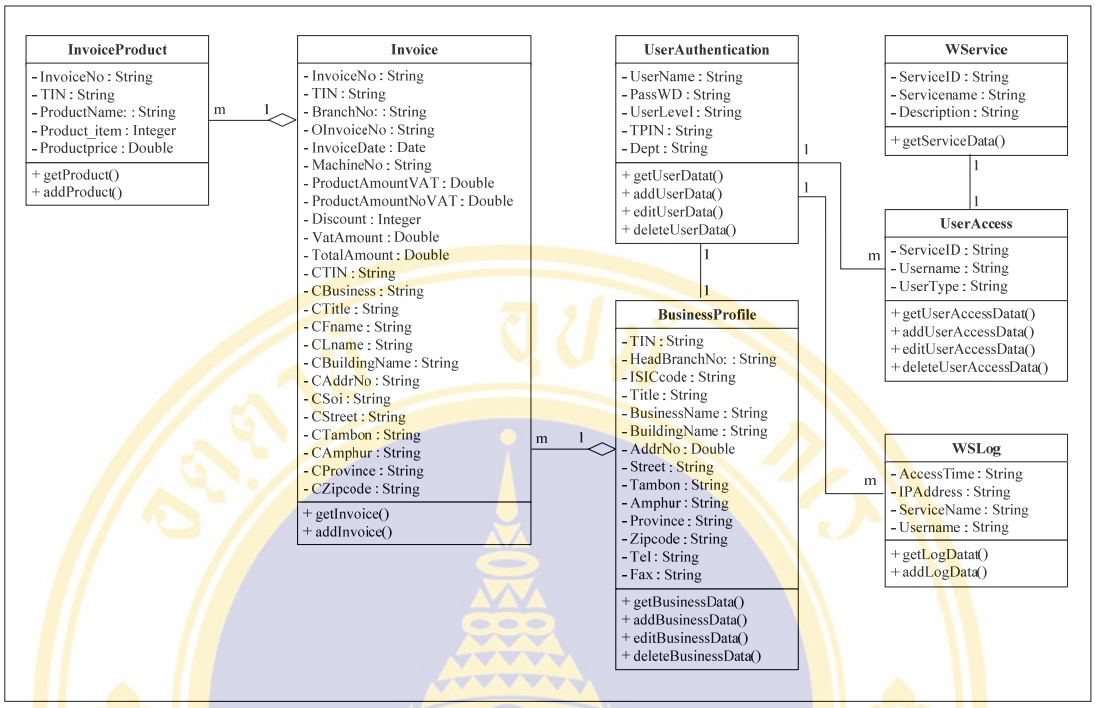


Figure 4.13 Class Diagram of e-Tax Invoice System

Attributes of these tables are given in Table 4.12 - 4.18 as follows:

Table 4.12 Invoice

No	Field	Description	Type	Size
1.	InvoiceNo	Invoice Number	Character	6
2.	TIN	Tax Identification Number	Character	10
3.	BranchNo	Branch Code	Character	6
4.	OInvoiceNo	Old Invoice Number	Character	6
5.	InvoiceDate	Invoice Date	Character	50
6.	MachineNo	Machine Number	Character	20
7.	ProductAmountVAT	Product Amount VAT Include	Float	-
8.	ProductAmountNoVAT	Product Amount no VAT Include	Float	-

Copyright by Mahidol University

Table 4.12 Invoice (Cont.)

No	Field	Description	Type	Size
9.	Discount	Discount	Float	-
10.	VATAmount	Total VAT	Float	-
11.	TotalAmount	Total Amount	Float	-
12.	CTIN	Customer's TIN	Character	10
13.	CBusinessName	Customer's Business Name	Character	150
14.	CTitle	Customer's Title	Character	100
15.	CFname	Customer's Name	Character	100
16.	CLname	Customer's Last Name	Character	100
17.	CBuildingName	Customer's Building Name	Character	100
18.	CAddrNo	Customer's Address Number	Character	50
19.	CSoi	Customer's Soi Name	Character	100
20.	CStreet	Customer's Street Name	Character	100
21.	CTambon	Customer's Sub District	Character	100
22	CAmphur	Customer's District	Character	100
23.	CProvince	Customer's Province Name	Character	100
24.	CZipcode	Customer's Zipcode	Character	5

Table 4.13 InvoiceProduct

No	Field	Description	Type	Size
1.	InvoiceNo	Tax Invoice Serial Number	Character	6
2.	TIN	Tax Identification Number	Character	10
3.	ProductName	Product's Name	Character	200
4.	Product_Item	Product's price	Integer	-
5.	ProductPrice	Total Product's price (not include VAT)	Float	-

Copyright by Mahidol University

Table 4.14 BusinessProfile

No	Field	Description	Type	Size
1.	TIN	Business' Tax Identification Number	Character	10
2.	HeadBranchNo	Business's Head Office Code	Character	6
3.	ISICcode	Business Type Code	Character	6
4.	Title	Business's Title	Character	100
5.	BusinessName	Business's Name	Character	150
6.	BuildingName	Business's Building Number	Character	100
7.	AddrNo	Business's Address number	Character	50
8.	Street	Business's Street Name	Character	100
9.	Tambon	Business's Sub District	Character	100
10.	Amphur	Business's District	Character	100
11.	Province	Business's Province	Character	100
12.	Zipcode	Business's Zipcode	Character	5
13.	Tel	Business's Telephone	Character	30
14.	Fax	Business's Email	Character	30

Table 4.15 UserAuthentication

No	Field	Description	Type	Size
1.	UserName	Username	Character	20
2.	PassWD	Digested Password	Character	50
3.	userLevel	User Level: a –Admin u – General User	Character	1
4.	TPIN	Tax Identification Number /Personal Identification Number	Character	13
5.	Dept	Department Name	Character	150

Copyright by Mahidol University

Table 4.16 UserAccess

No	Field	Description	Type	Size
1.	ServiceID	Service Identification	Character	5
2.	Username	Username	Character	20
3.	User_type	Type of user: b – Business p – Person	Character	1

Table 4.17 WService

No	Field	Description	Type	Size
1.	ServiceID	Service Identification	Character	5
2.	ServiceName	Service Name	Character	200
3.	Description	Service Description	Text	-

Table 4.18 WSLog

No	Field	Description	Type	Size
1.	AccessTime	Date & Time Access	Character	80
2.	IpAddress	User's IP	Character	50
3.	ServiceName	Web Services Name	Character	200
4.	Username	Date & Time Log in	Character	20

4.6 Business Process Diagram

The e-Tax Invoice System can be represented in business process diagram as shown in Figure 4.14. Operations in this system are divided into two groups: Process and Web Services. These operations are shown as follows:

Copyright by Mahidol University

4.6.1 Web Services

There are five Web Services in e-Tax Invoice System presented as follows:

1. User Authenticator (Figure 4.15)
2. Data Provider (Figure 4.16)
3. Data Collector (Figure 4.17)
4. Report Generator (Figure 4.18)
5. Log Report Generator (Figure 4.19)

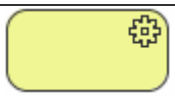
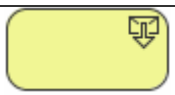
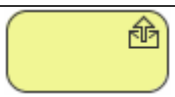
4.6.2 Processes

In Business Process Diagram of e-Tax Invoice System, there are four processes illustrated as follows:

1. Logging User Usage Process (Figure 4.20)
2. Generate Report Process (Figure 4.21)
3. Generate Log Report Process (Figure 4.22)
4. Manage User Account Process (Figure 4.23)

Business Process Diagram symbols of **Business Process Visual Architecture Analyst Edition Version 2.1** [27] are described in Table 4.19.

Table 4.19 Business Process Diagram Symbols

No	Symbol	Name	Description
1.		User Task	Represent a user task.
2.		Service Task	Represent a service task.
3.		Receive Task	Represent a receive task.
4.		Send Task	Represent a send task.

Copyright by Mahidol University

Table 4.19 Business Process Diagram Symbols (Cont.)

No	Symbol	Name	Description
5.		Script Task	Represent a script task.
6.		Process	Represent a process.
7.		Start Event	Represent a starting point.
8.		End Event	Represent a ending point.
9.		Error End Event	Represent a error message with ending point.
10.		Message End Event	Represent message with ending point.
11.		Decision	Represent a decision.
12.		Parallel Fork/Join	Represent parallel fork/join.
13.		Sequence Flow	Represent a direction of flow.
14.		Message Flow	Represent a direction of message.

Copyright by Mahidol University

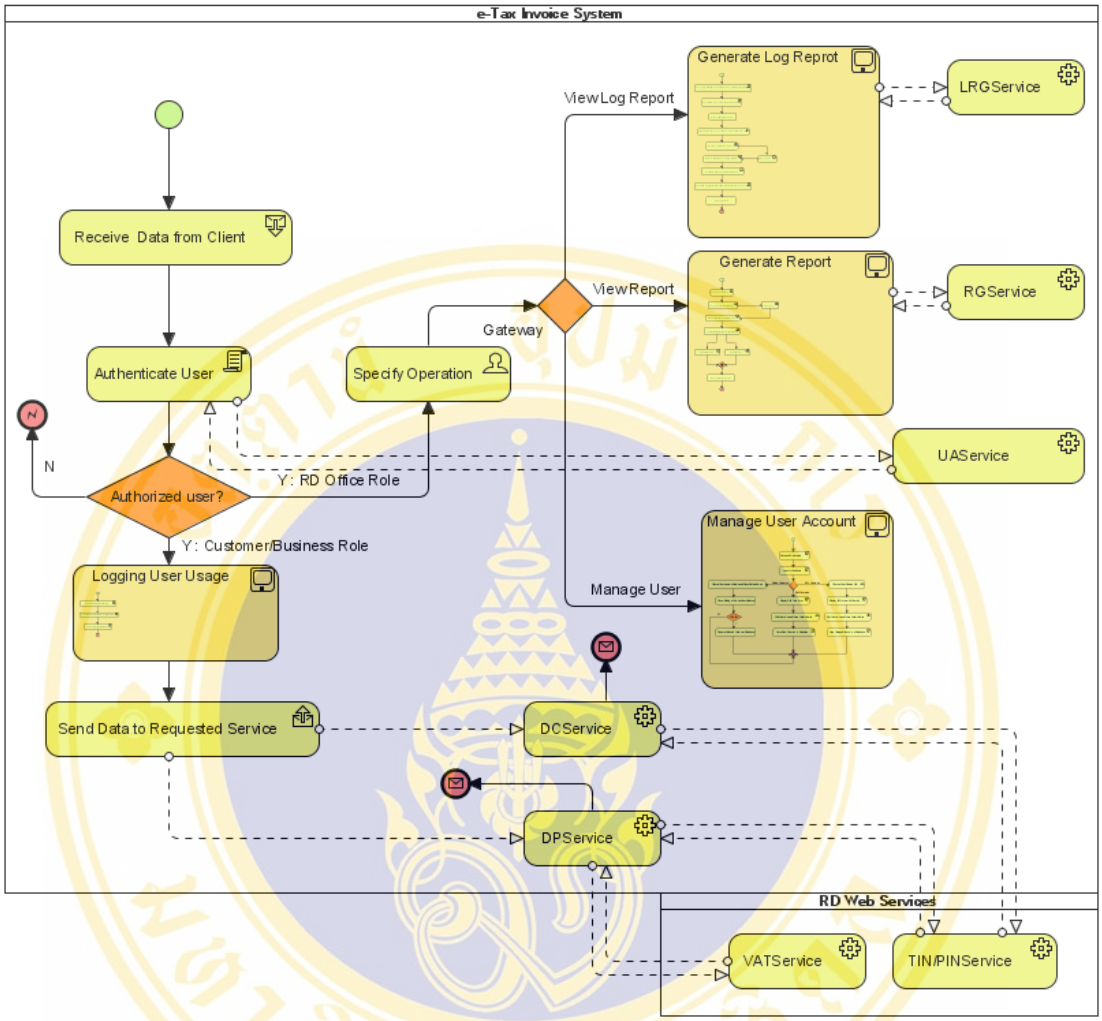


Figure 4.14 Business Process Diagram of The e-Tax Invoice System

Copyright by Mahidol University

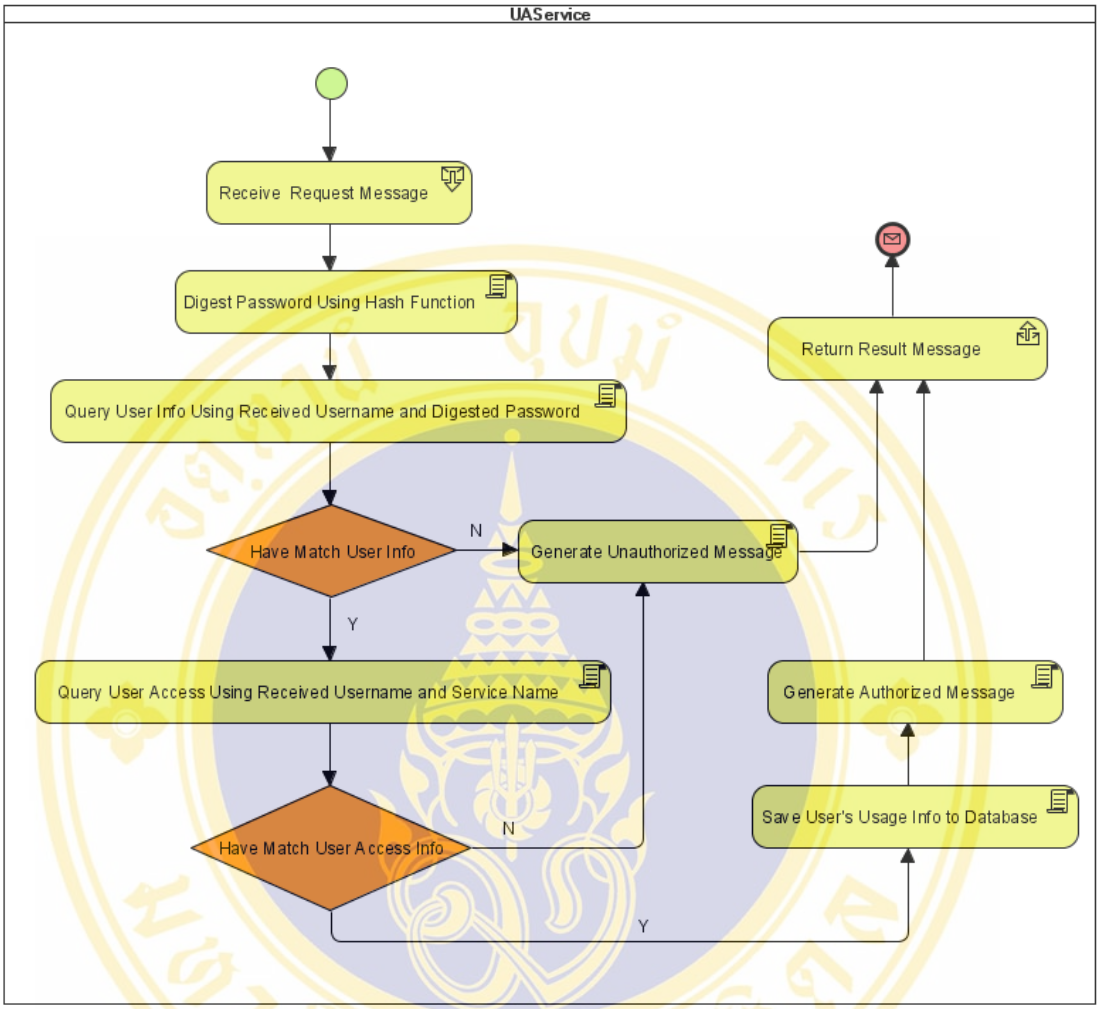


Figure 4.15 Business Process Diagram of User Authenticator

Copyright by Mahidol University

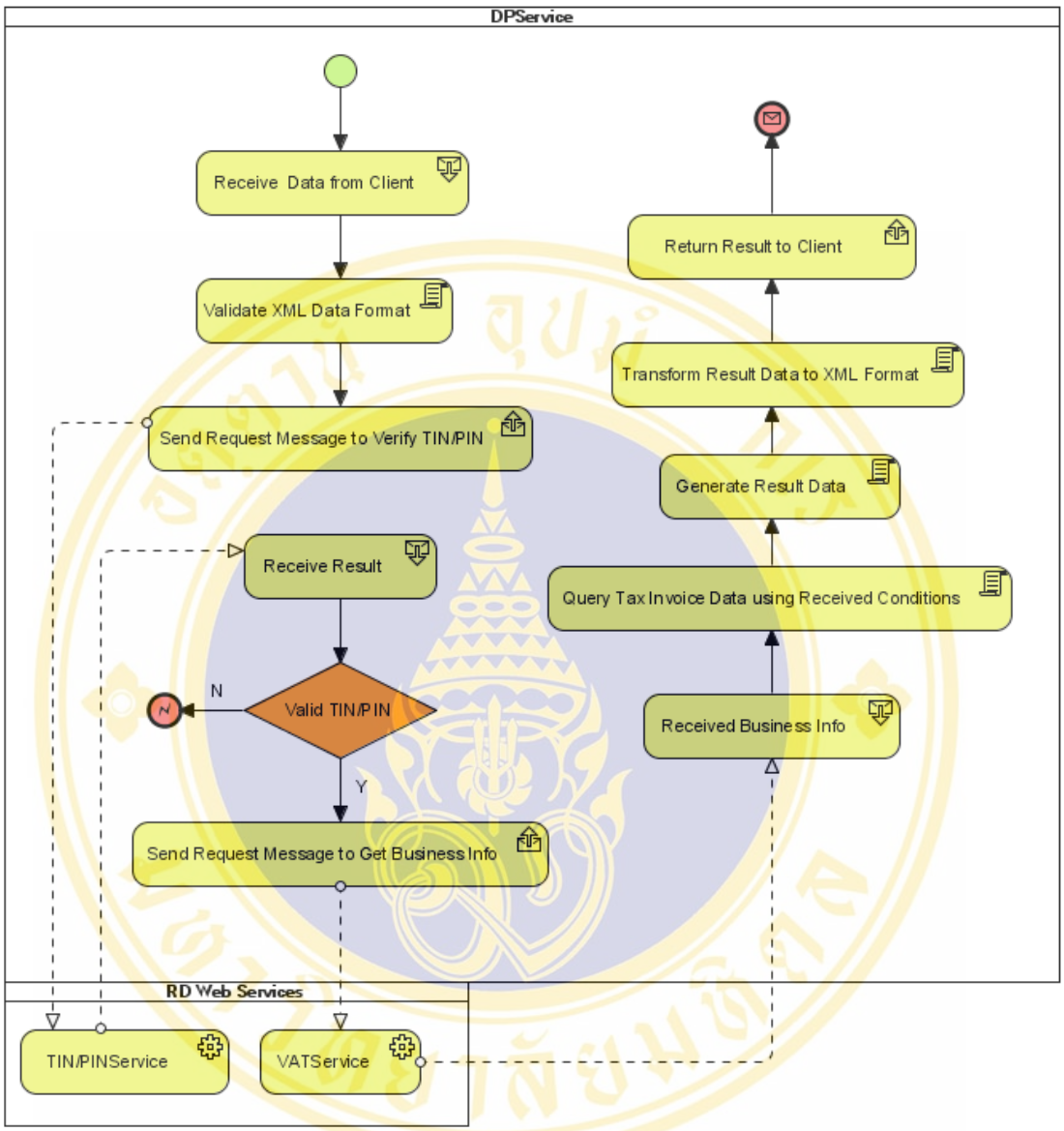


Figure 4.16 Business Process Diagram of Data Provider

Copyright by Mahidol University

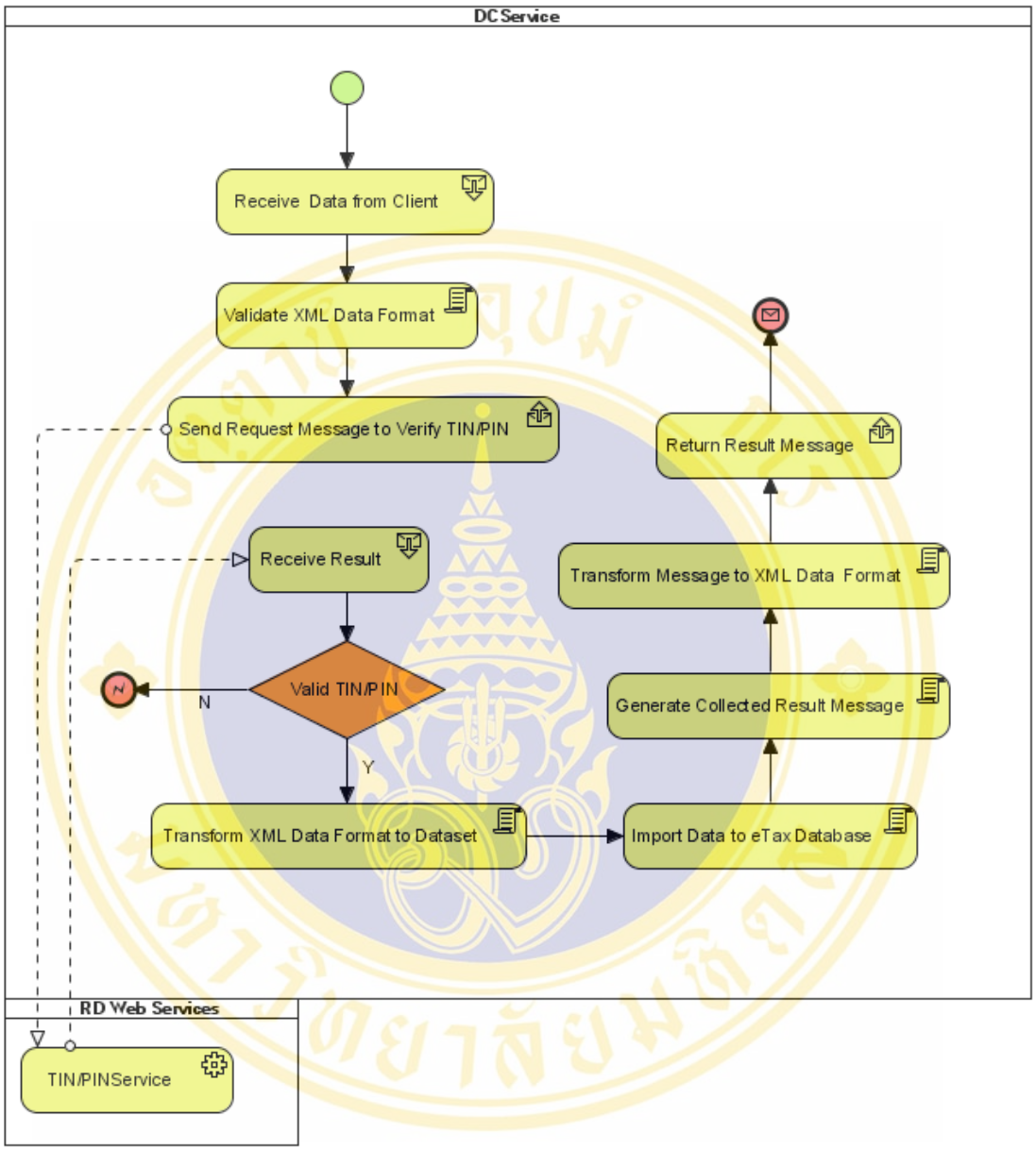


Figure 4.17 Business Process Diagram of Data Collector

Copyright by Mahidol University

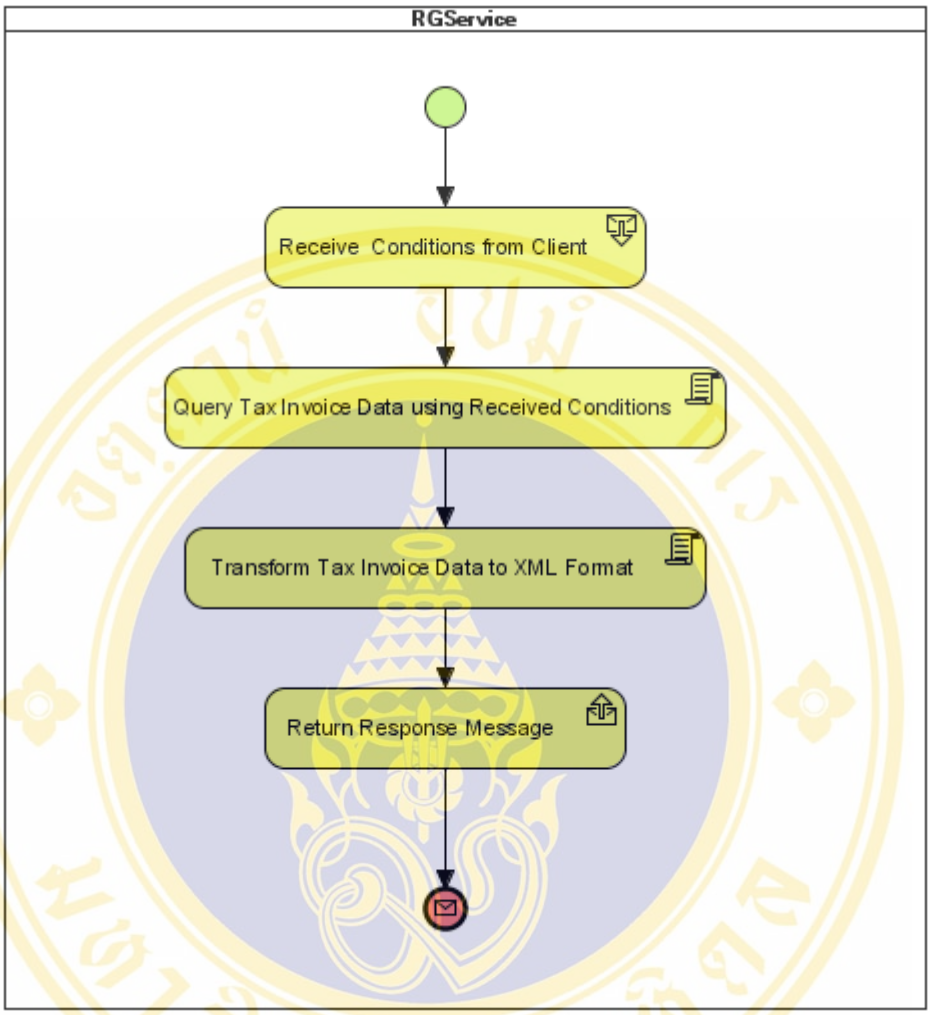


Figure 4.18 Business Process Diagram of Report Generator

Copyright by Mahidol University

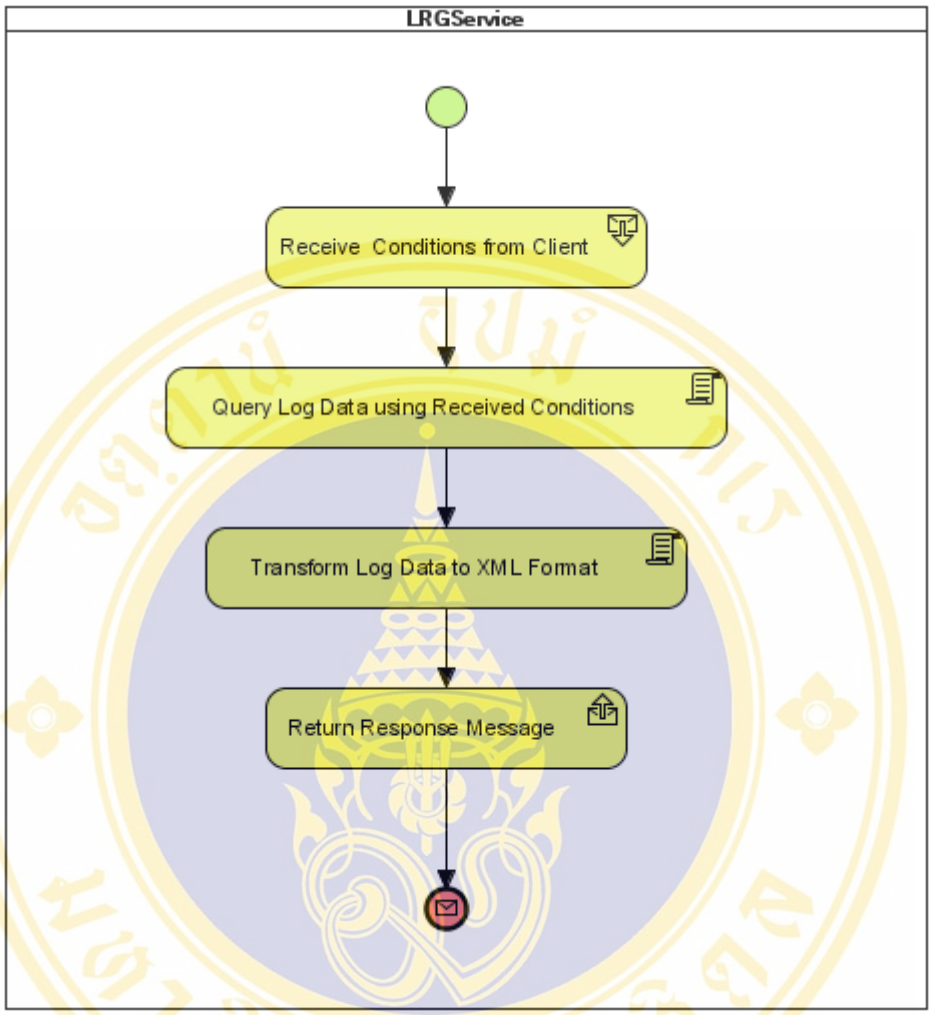


Figure 4.19 Business Process Diagram of Log Report Generator

Copyright by Mahidol University

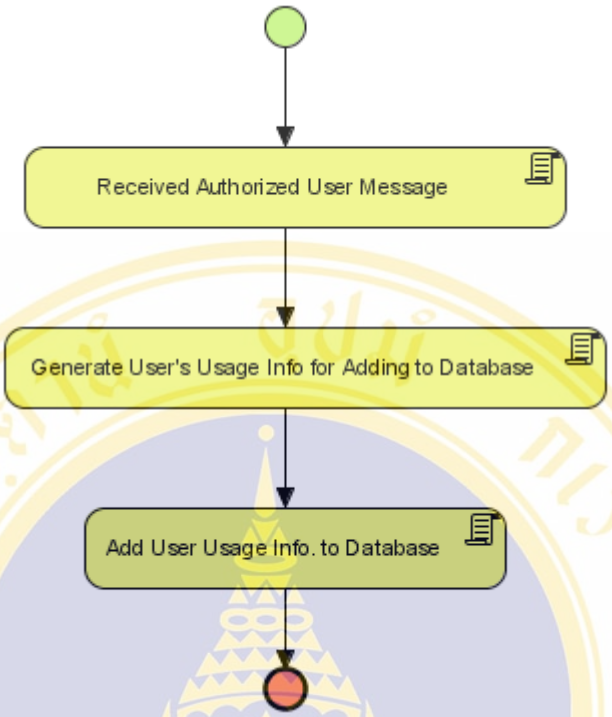


Figure 4.20 Business Process Diagram of Logging User Usage Process

Copyright by Mahidol University

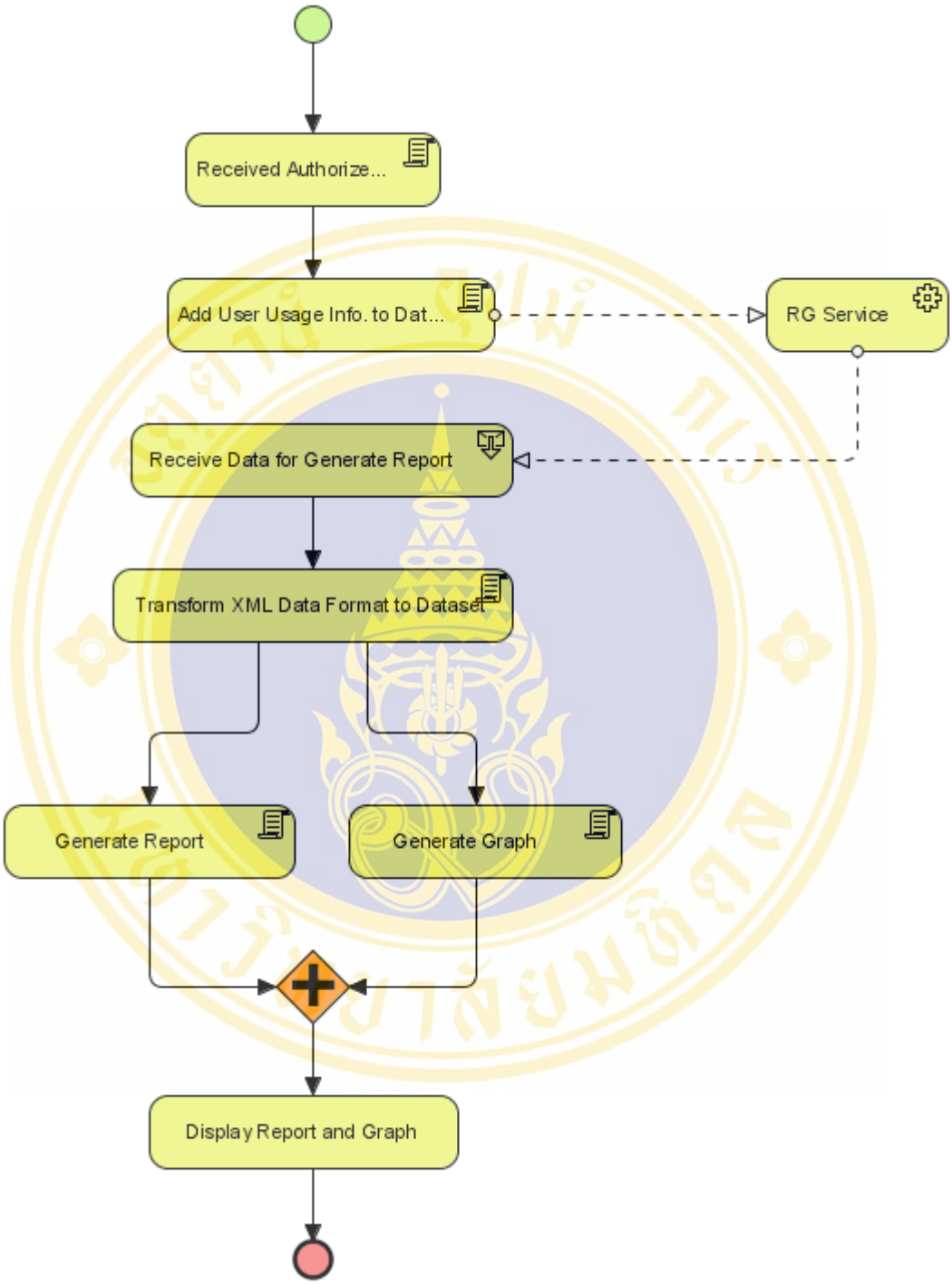


Figure 4.21 Business Process Diagram of Generate Report Process

Copyright by Mahidol University

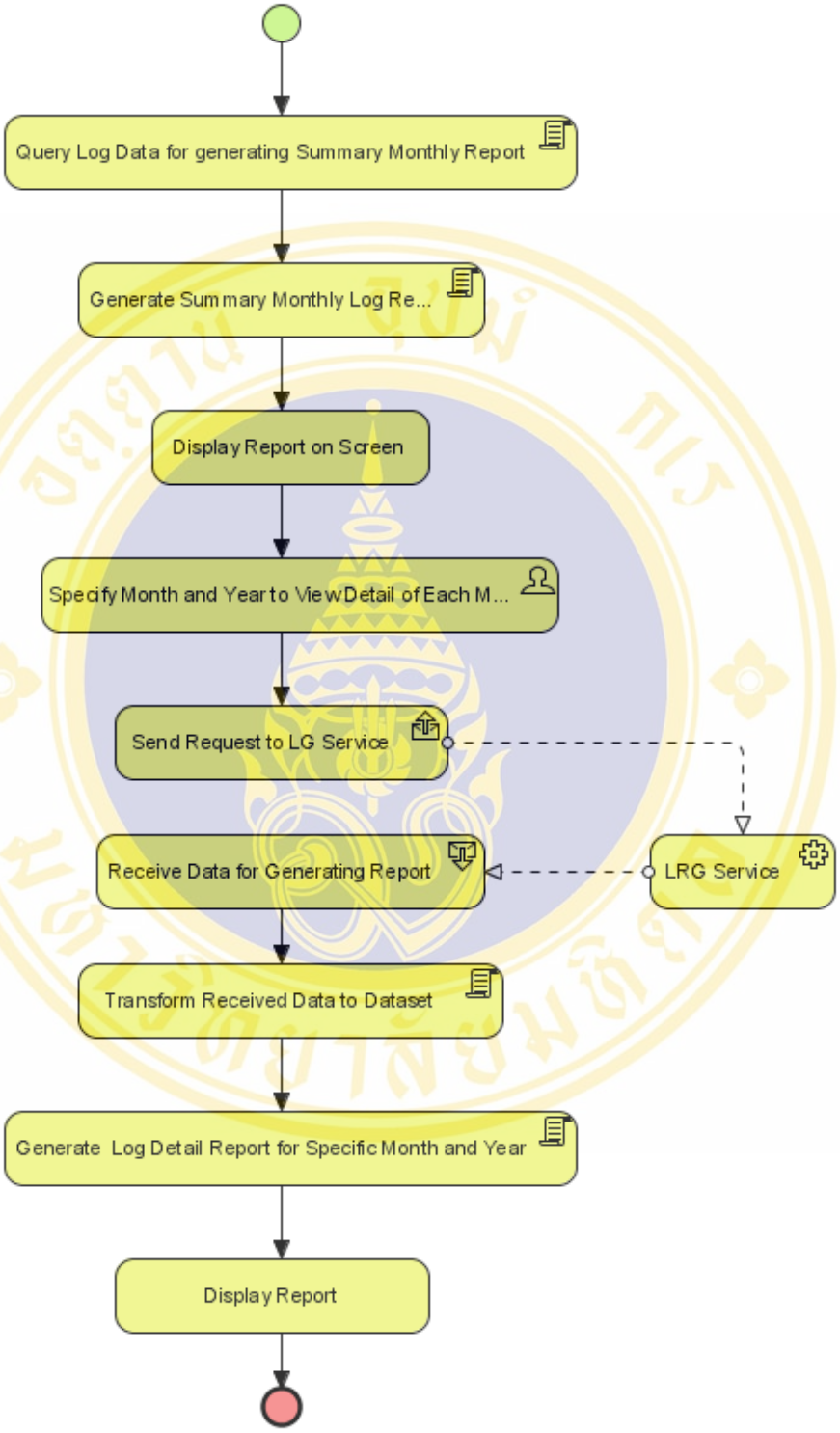


Figure 4.22 Business Process Diagram of Generate Log Report Process

Copyright by Mahidol University

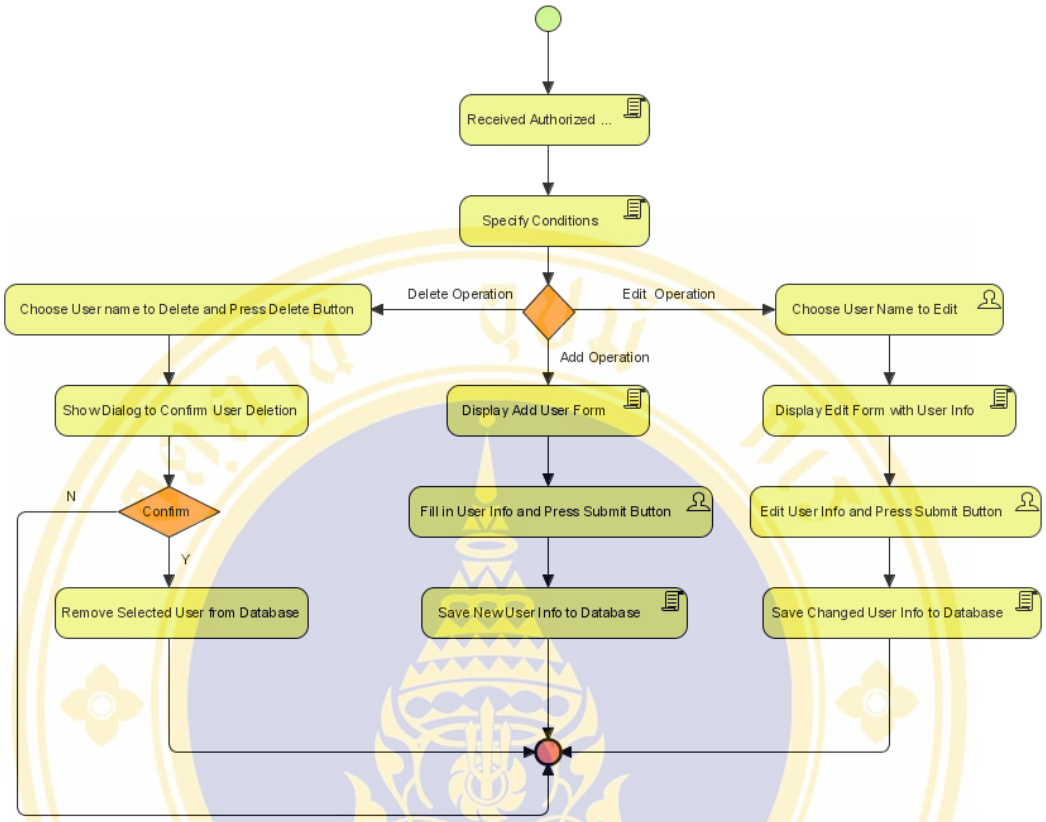


Figure 4.23 Business Process Diagram of Manage User Account Process

4.7 Sequence Diagram

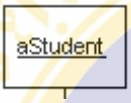
The process in the e-Tax Invoice System can be represented in sequence diagram as follows:

1. User Authenticator (Figure 4.24)
2. Data Provider (Figure 4.25)
3. Data Collector (Figure 4.26)
4. Login Manager (Figure 4.27)
5. Report Generator (Figure 4.28)
6. Log Report Generator (Figure 4.29)
7. Log User Manager (Figure 4.30)

Copyright by Mahidol University

Sequence Diagram symbols according to the UML standard [28] are described in Table 4.20

Table 4.20 Sequence Diagram Symbols

No	Symbol	Name	Description
1.		Actor	Represents an external person or entity that interacts with the system
2.		Object	Represents an object in the system or one of its components
3.		Message	Represent an interaction between two objects. It is performed as a message sent from one object to another
4.		Lifeline	Represents the object's life during the interaction
5.		Activation Box	Represent focus of control or activation.
6.		Return Value	Represent return value.

Copyright by Mahidol University

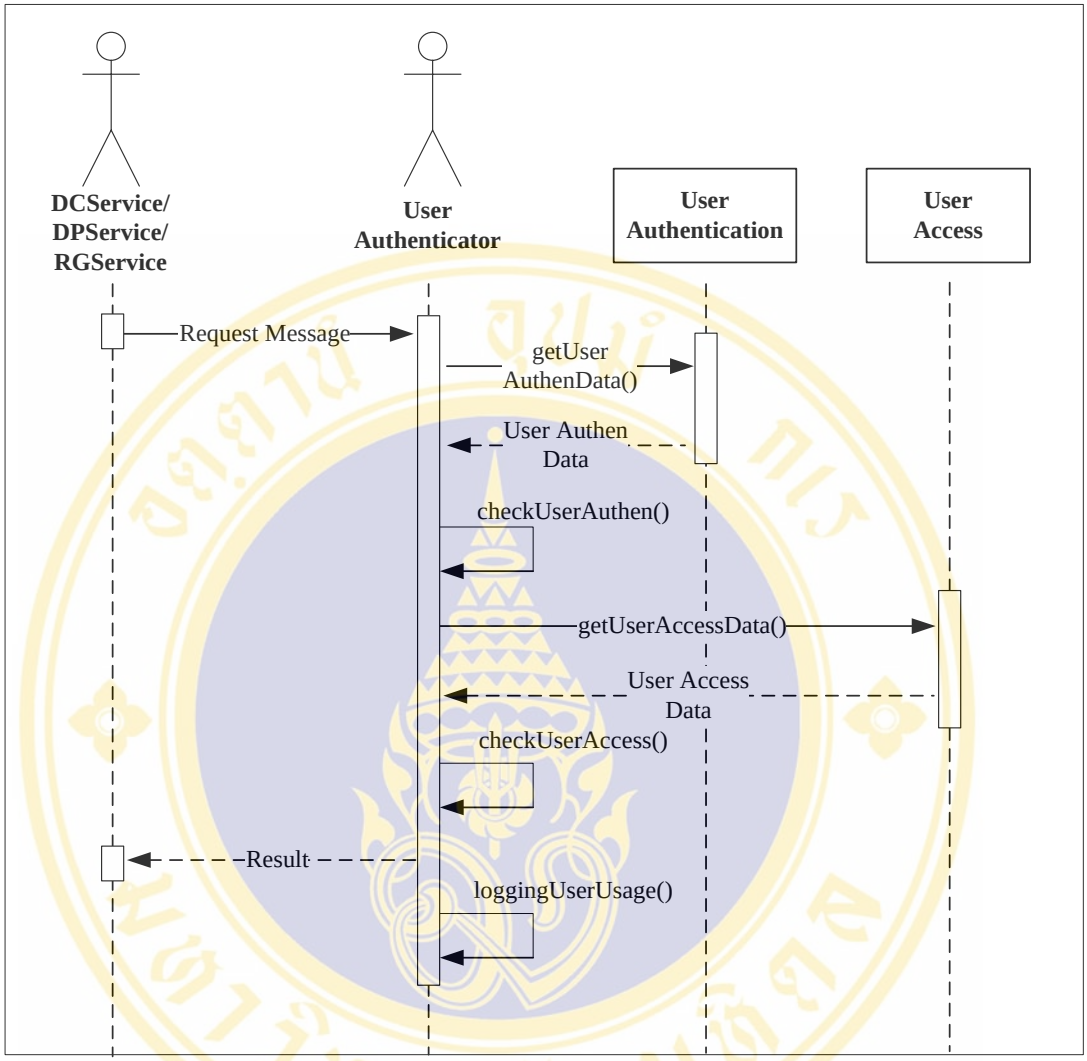


Figure 4.24 User Authenticator Sequence Diagram

Copyright by Mahidol University

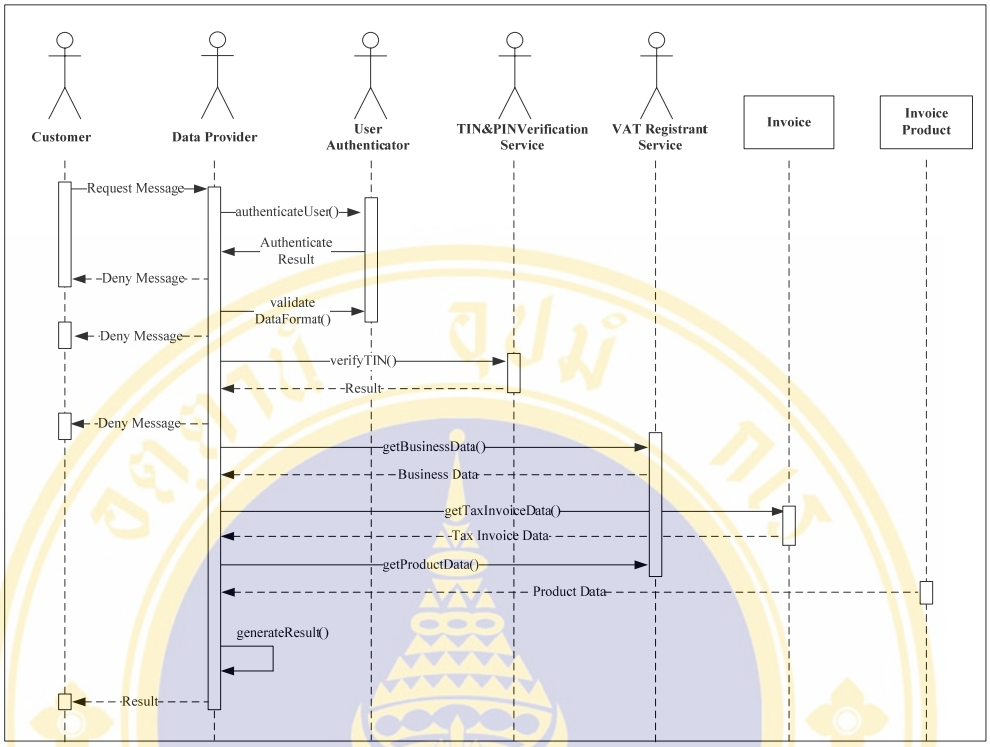


Figure 4.25 Data Provider Sequence Diagram

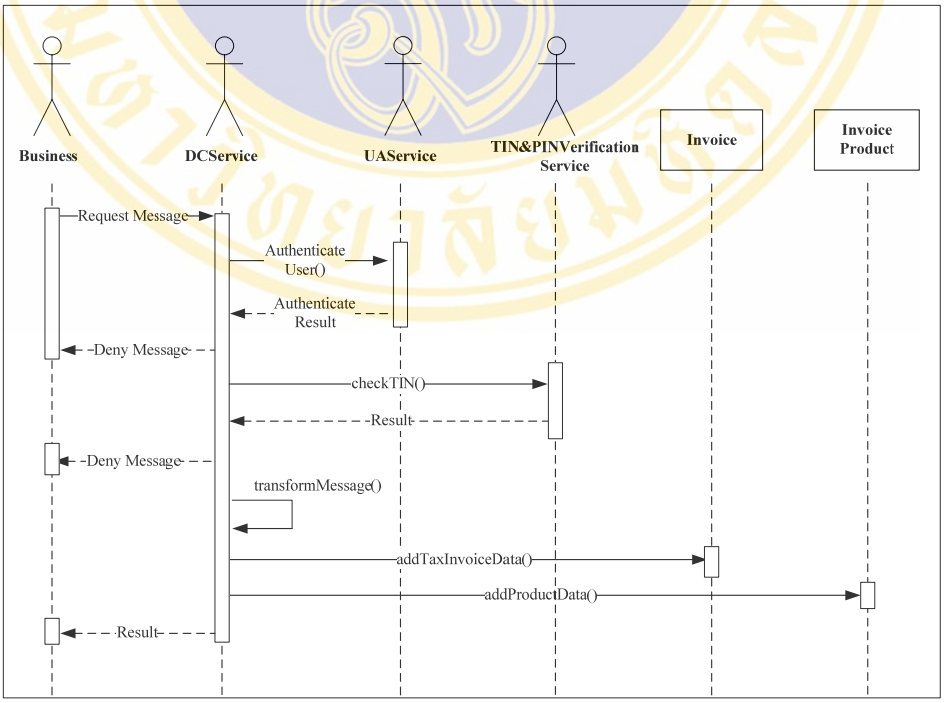


Figure 4.26 Data Collector Sequence Diagram

Copyright by Mahidol University

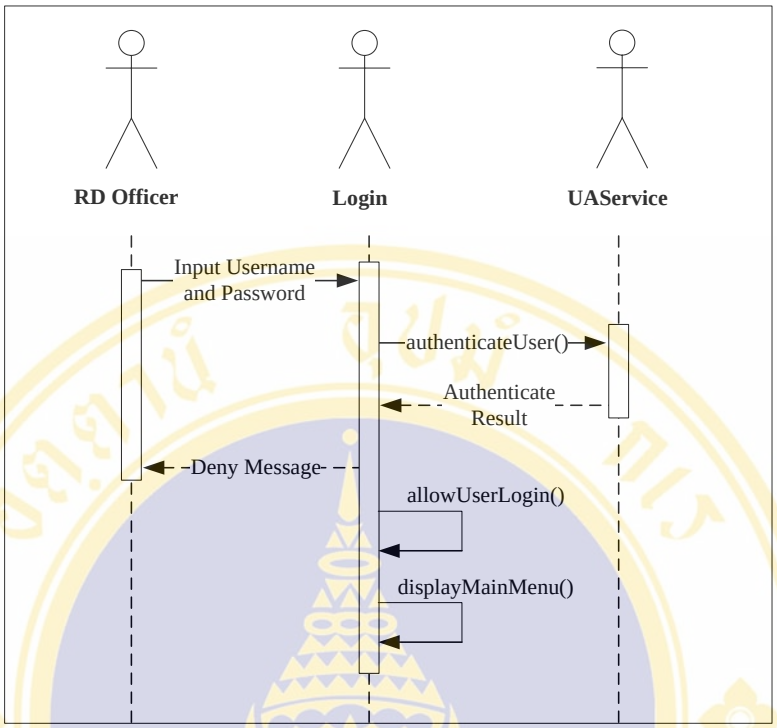


Figure 4.27 Login Manager Sequence Diagram

Copyright by Mahidol University

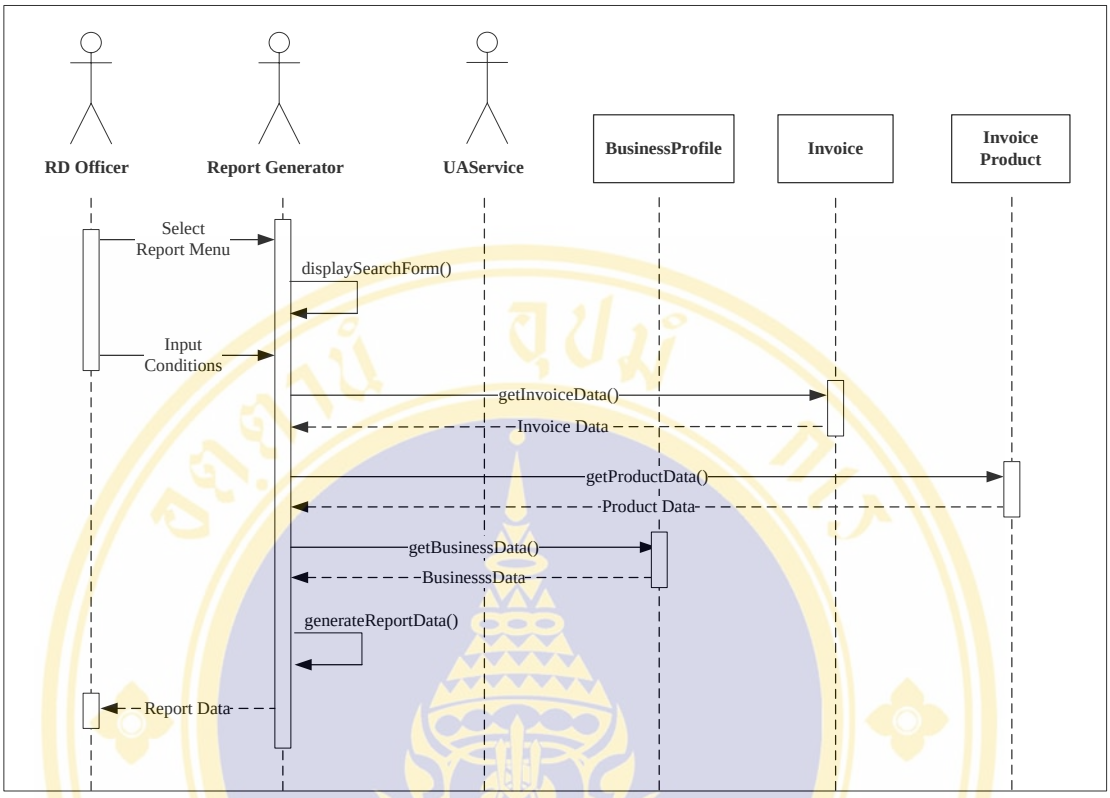


Figure 4.28 Report Generator Sequence Diagram

Copyright by Mahidol University

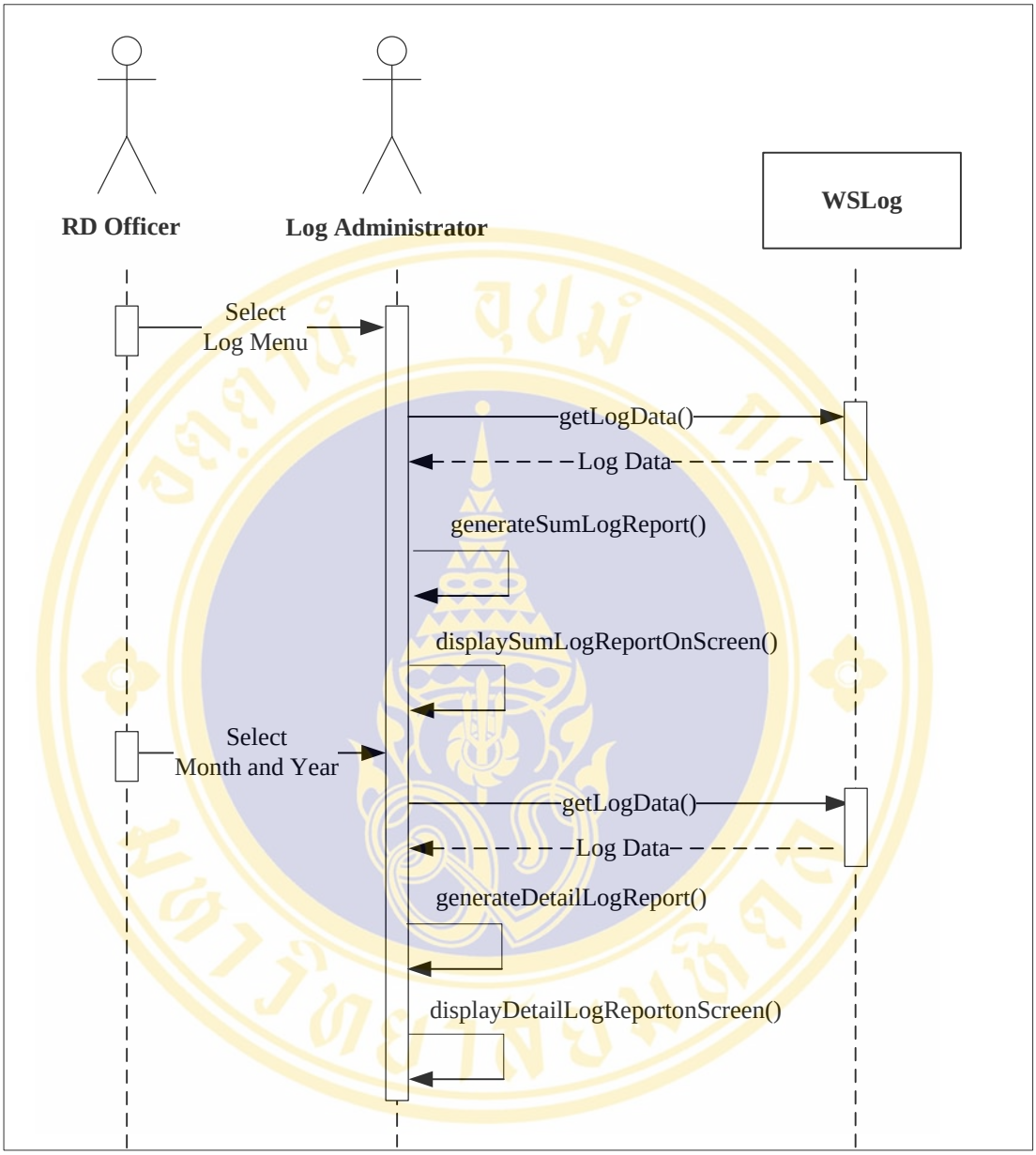


Figure 4.29 Log Report Generator Sequence Diagram

Copyright by Mahidol University

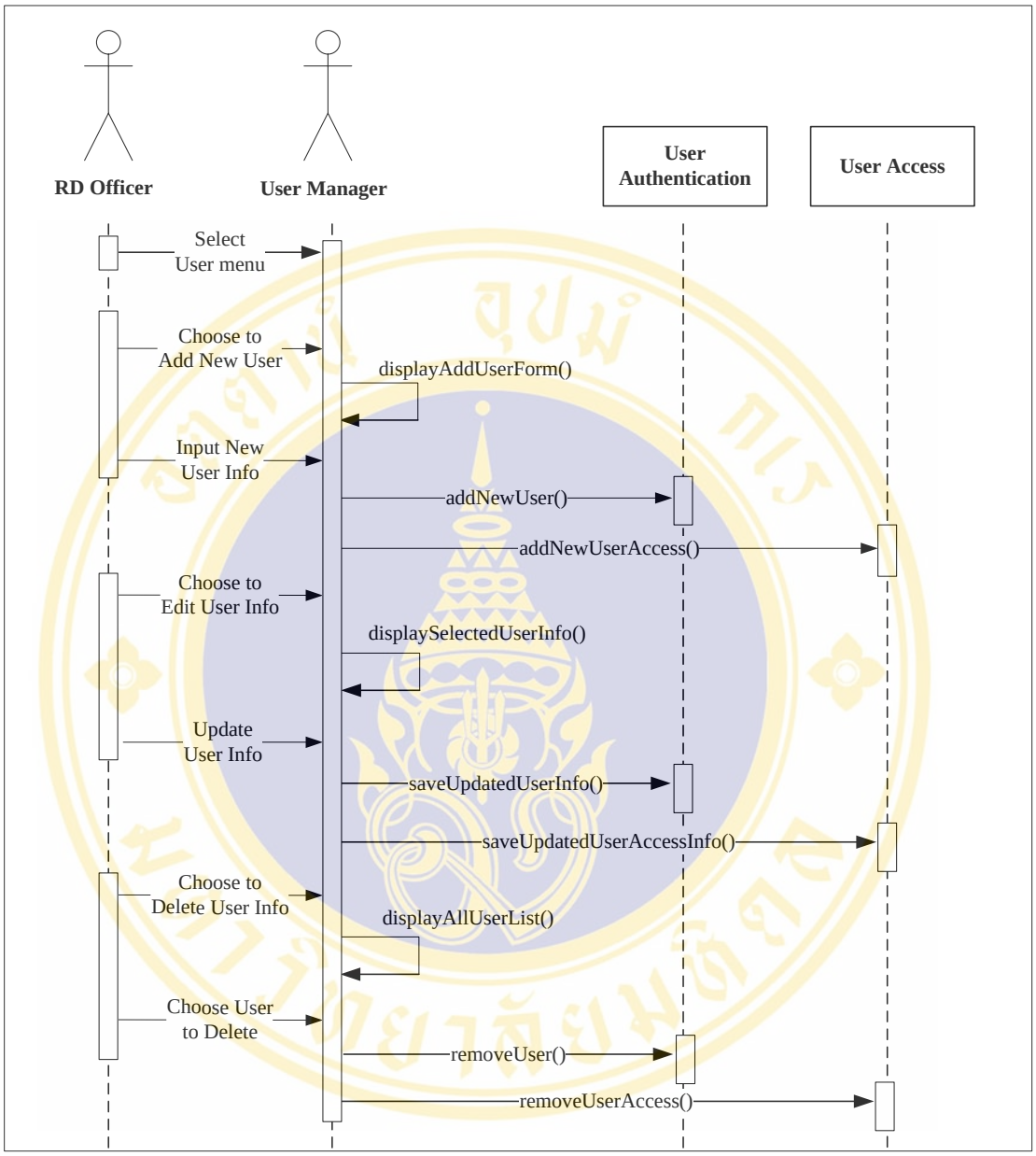


Figure 4.30 Log User Manager Sequence Diagram

Copyright by Mahidol University

CHAPTER V
IMPLEMENTATION

5.1 Hardware and Software Requirements

The e-Tax Invoice System environment is illustrated in Figure 5.1.

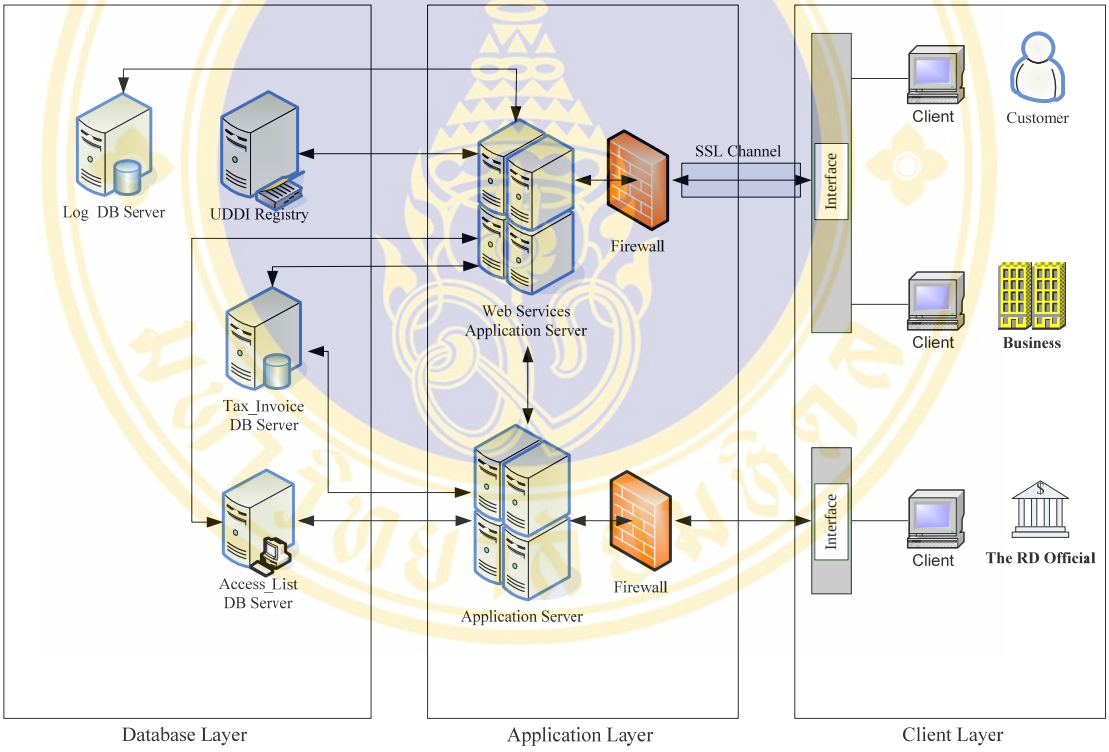


Figure 5.1 e-Tax Invoice System Environment

The e-Tax invoice system environment is separated into 3 layers:

5.1.1 Database Layer

The e-Tax Invoice system composes of four database servers described as follows:

Copyright by Mahidol University

1) Log Database Server contains user access log file that must have high capacity hard disk for collected daily user access log. The server specification is described as follows:

- Processor : Intel Xeon 2.4 GHz
- RAM : 2 GB
- Hard Disk : 80 GB
- Operating System : Windows 2003 Server
- Software : IIS 6.0, CA 509

2) UDDI Registry contains WSDL documents of Web Services provided by the RD. The server specification is described as follows:

- Processor : Intel Xeon 2.4 GHz
- RAM : 2 GB
- Hard Disk : 80 GB
- Operating System : Windows 2003 Server
- Software : XML SPY, Internet Explorer

3) Tax Invoice Database Server contains tax invoice data collected from businesses. The server specification is described as follows:

- Processor : Intel Xeon 2.4 GHz
- RAM : 2 GB
- Hard Disk : 80 GB
- Operating System : Windows 2003 Server
- Software : SQL Server 2005

4) Access List Database Server contains username, password, capabilities, and labels: the possession of which may be used to indicate access rights. The server specification is described as follows:

- Processor : Intel Xeon 2.4 GHz
- RAM : 2 GB
- Hard Disk : 80 GB
- Operating System : Windows 2003 Server

Copyright by Mahidol University

- Software : SQL Server 2005

5.1.2 Application Layer

The e-Tax Invoice system consists of two application servers described as follows:

1) Web Services Application Server contains Web Services provided by the RD. The server specification is described as follows:

- Processor : Intel Xeon 2.4 GHz
- RAM : 2 GB
- Hard Disk : 80 GB
- Operating System : Windows 2003 Server
- Software : IIS 6.0, CA 509

2) Application Server contains application that provides to the RD Officer for support the RD Officer's operations. The server specification is described as follows:

- Processor : Intel Xeon 2.4 GHz
- RAM : 2 GB
- Hard Disk : 80 GB
- Operating System : Windows 2003 Server
- Software : IIS 6.0

5.1.3 Client Layer

In the e-Tax Invoice system, clients that request service from the system are divided into three groups 1) Customer 2) Business 3) The RD Officer. All clients that communicate with the system must have minimum computer specification described as follows:

- Processor : Intel Pentium III
- RAM : 256 MB
- Hard Disk : 10 GB
- Operating System : Windows NT /2000/XP/2003
- Software : SQL Server 2005, Microsoft Visual Studio .NET , Internet Explorer

Copyright by Mahidol University

For the implementation, Computer specification for implementing the system is

- Processor : Intel Pentium III
- RAM : 512 MB
- Hard Disk : 80 GB
- Operating System : Windows 2000
- Software : SQL Server 2005, Microsoft .NET, JAVA, Internet Explorer, XML Spy, Adobe Photoshop

This project uses Microsoft Visual Studio .NET 2005 to develop the system, SQL Server 2005 to store database, XML spy to generate XML schema, and Adobe Photoshop to edit/create images. Moreover, this system will use communication channel bandwidth 1000 Mbps and SSL channel to communicate with each other. Furthermore, the system has firewall to screen between external clients and servers. As a result, the new system can provide effective and secure communication among stakeholders. The new system will be developed in simulated environment. When the system is developed and tested completely, it will be deployed on the RD server.

5.2 Prototype

The prototype of the e-Tax Invoice System is described as follows:

5.2.1 User Interface

User interface of the e-tax Invoice System can be divided into three parts: Customer, Business and RD Officer.

1) Customer

- **View Tax Invoice Data Page** is shown in Figure 5.2. In this page user can specify conditions: Customer's name, Customer's surname, customer's TIN, Business's name and Date corresponded to tax invoice slip that they want to view and press "Search Button" to view tax invoice data related with specified conditions. Tax invoice data search form shown in Figure 5.2 is generated by using XSLT file to transform received xml document to html document. The example of XSLT file, XML document, and HTML result are depicts in Figure 5.3, 5.4 and 5.5 respectively.

Copyright by Mahidol University

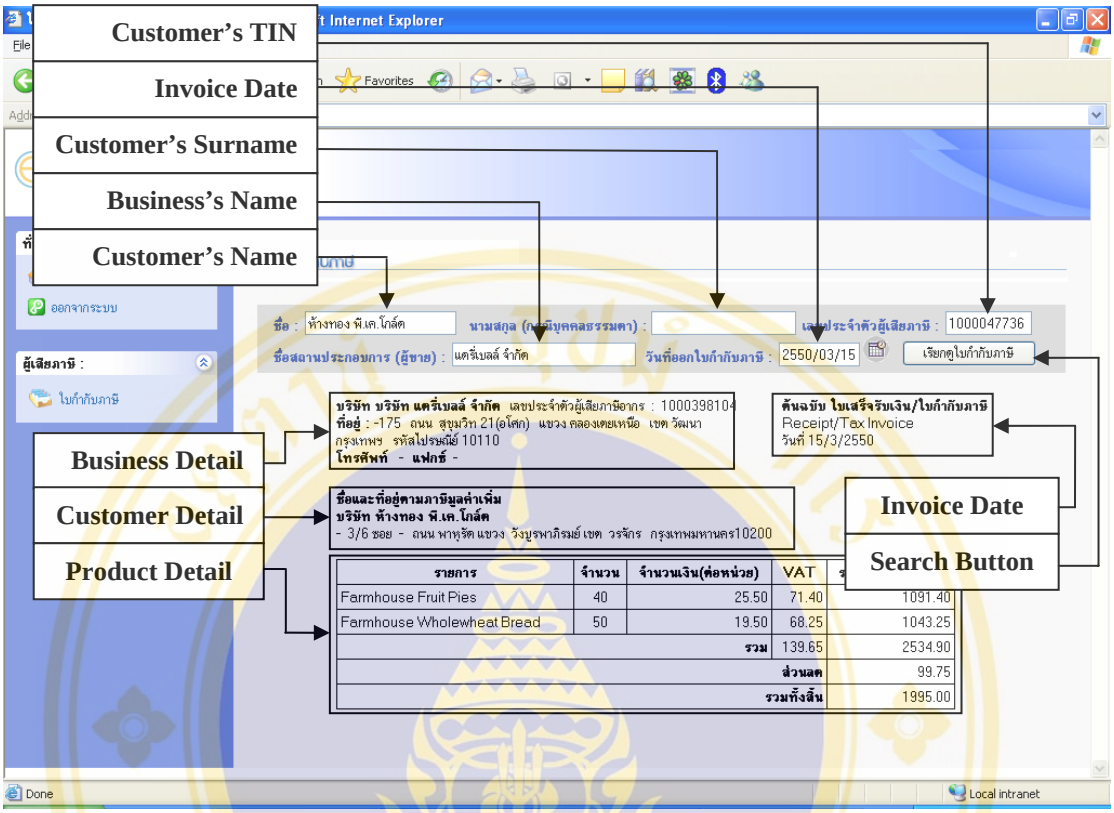


Figure 5.2 View Tax Invoice Page

```
<?xml version="1.0" encoding="utf-8"?>
 <xsl:stylesheet version="1.0"    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
      <html>
        <head>
          <style type="text/css">
            A
              {
               COLOR: #0A57A4;
               FONT-FAMILY: MS Sans Serif, Arial;
               FONT-SIZE: 10pt;
               TEXT-DECORATION: none
              }
          TD
              {
               COLOR: #000000;
               FONT-FAMILY: MS Sans Serif, Arial;
               FONT-SIZE: 10pt;
               TEXT-DECORATION: none
              }.lpink
```

Figure 5.3 An Example of XSLT in e-Tax Invoice System

Copyright by Mahidol University

```
            {
             COLOR: #664e38;
             FONT-FAMILY: Ms sans serif;
             FONT-SIZE: 10pt;
             FONT-WEIGHT: regular
            }
          pink
            {
             COLOR: #777777;
             FONT-FAMILY: Ms sans serif;
             FONT-SIZE: 10pt;
             FONT-WEIGHT: regular
            }
         </style>
       </head>
       <body>
         <xsl:apply-templates select="//Invoices"/>
       </body>
      </html>
    </xsl:template>
    <xsl:template match="Invoice">
     <table border="0" cellpadding="3" cellspacing="0">
      <tr>
       <td valign="top"><b>บริษัท  <xsl:value-of select="BusinessName"/></b>
            เลขประจำตัวผู้เสียภาษีอากร  :  <xsl:value-of select="TIN"/><br/>
          <b>ที่อยู่ : </b>
          <xsl:value-of select="BuildingName"/>
          <xsl:value-of select="AddrNo"/>   ถนน   
          <xsl:value-of select="Street"/>  
          แขวง  <xsl:value-of select="Tambon"/>   เขต
          <xsl:value-of select="Amphur"/>
          <br/><xsl:value-of select="Province"/>   รหัสไปรษณีย์
          <xsl:value-of select="Zipcode"/>
           <br/><b>โทรศัพท์ </b> 
          <xsl:value-of select="Tel"/>   <b>แฟกซ์</b> 
          <xsl:value-of select="Fax"/>
       </td>
       <td width="30"></td>
      <td valign="top">
          <b>ต้นฉบับ ใบเสร็จรับเงิน/ใบกำกับภาษี</b>
          <br/>Receipt/Tax Invoice
          <br/>วันที่ <xsl:value-of select="InvoiceDate"/>
      </td>
     </tr>
    <tr>
      <td colspan="3"><br/><b>ชื่อและที่อยู่ตามภาษีมูลค่าเพิ่ม</b><br/>
          <b>บริษัท <xsl:value-of select="CBusinessName"/></b>  
          <xsl:value-of select="CTitle"/>  <xsl:value-of select="CFname"/>  
          <xsl:value-of select="CLname"/><br/>
```

Figure 5.3 An Example of XSLT in e-Tax Invoice System (Cont.)

Copyright by Mahidol University

```
          <xsl:value-of select="CBuildingName"/>  <xsl:value-of select="CAddrNo"/>
          ซอย   <xsl:value-of select="CSoi"/>
            ถนน  <xsl:value-of select="CStreet"/> แขวง   
          <xsl:value-of select="CTambon"/> เขต  
          <xsl:value-of select="CAmphur"/>
             <xsl:value-of select="CProvince"/> <xsl:value-of select="CZipcode"/>
         <br/><br/>
      <table border="0" cellpadding="3" cellspacing="1" width="95%" bgcolor="#000000">
            <tr>
              <td align="center" bgcolor="#ffffff"><b>รายการ</b></td>
              <td align="center" bgcolor="#ffffff"><b>จำนวน</b></td>
              <td align="center" bgcolor="#ffffff"><b>จำนวนเงิน(ต่อหน่วย)</b></td>
              <td align="center" bgcolor="#ffffff"><b>VAT</b></td>
              <td align="center" bgcolor="#ffffff"><b>รวมเงิน/TOTAL</b></td>
            </tr>
<!-- Start Product List -->
          <xsl:for-each select="Products/Product">
            <tr>
              <td bgcolor="#ffffff"><xsl:value-of select="Product_name"/></td>
              <td align="center" bgcolor="#ffffff"><xsl:value-of select="Product_item"/></td>
              <td align="right" bgcolor="#ffffff">
                <xsl:value-of select="format-number(Product_price,'#.00')"/></td>
              <td align="right" bgcolor="#ffffff">
               <xsl:value-of select="format-number(((Product_price*Product_item)*0.07)
                ,'#.00')"/></td>
                <td align="right" bgcolor="#ffffff">
                 <xsl:value-of select="format-number((((Product_price*Product_item)*0.07)
                  +(Product_price*Product_item)),'#.00')"/></td>
            </tr>
          </xsl:for-each>
<!-- End Product List -->
            <tr>
              <td align="right" bgcolor="#ffffff" colspan="3"><b>รวม</b></td>
              <td align="right" bgcolor="#ffffff">
                  <xsl:value-of select="format-number(VATamount,'#.00')"/></td>
              <td align="right" bgcolor="#ffffff">
                 <xsl:value-of select="format-number(TotalAmount,'#.00')"/></td>
            </tr>
            <tr>
              <td align="right" bgcolor="#ffffff" colspan="4"><b>ส่วนลด</b></td>
              <td align="right" bgcolor="#ffffff"> 
                 <xsl:value-of select="format-number(Discount,'#.00')"/>
              </td>
            </tr>
            <tr>
              <td align="right" bgcolor="#ffffff" colspan="4"><b>รวมทั้งสิ้น</b></td>
              <td align="right" bgcolor="#ffffff">
                <xsl:value-of select="format-number(ProductAmountVAT,'#.00')"/></td>
            </tr>
          </table>
```

Figure 5.3 An Example of XSLT in e-Tax Invoice System (Cont.)

Copyright by Mahidol University

```
                </td></tr>
            </table>
              <br/><br/>
</xsl:template>
</xsl:stylesheet>
```

Figure 5.3 An Example of XSLT in e-Tax Invoice System (Cont.)

```
<?xml version="1.0"?>
<!--Generate Invoices Data in XML format-->
<Invoices>
  <Invoice>
    <TIN>1000398104</TIN>
    <Invoiceno>000365</Invoiceno>
    <Invoicedate>10/21/2007</Invoicedate>
    <ISICcode>333333</ISICcode>
    <OInvoiceNo>-</OInvoiceNo>
    <BranchNo>000000</BranchNo>
    <Machineno>00332128</Machineno>
    <Products>
      <Product>
        <Product_name>กระดาษ A4 80 แกรม</Product_name>
        <Product_item>20</Product_item>
        <Product_price>57.50</Product_price>
      </Product>
      <Product>
        <Product_name>ปากกาน้ำเงิน</Product_name>
        <Product_item>30</Product_item>
        <Product_price>52.25</Product_price>
      </Product>
    </Products>
    <ProductAmountVAT>2717.5</ProductAmountVAT>
    <ProductAmountNoVAT>0</ProductAmountNoVAT>
    <Discount>135.875</Discount>
    <VATamount>190.225</VATamount>
    <Totalamount>2771.85</Totalamount>
    <CTIN>1000063224</CTIN>
    <Ctitle />
    <Cbsuinessname>ที อาร์ เอฟ อิมปอร์ต แอนด์ เอ็กซ์ปอร์ต</Cbsuinessname>
    <Cfname />
    <Clname />
    <Cbuildingname>-</Cbuildingname>
    <Caddrno>448/1</Caddrno>
    <Csoi>พหลโยธิน 30</Csoi>
    <Cstreet>พหลโยธิน</Cstreet>
    <Cdistrict>พญาไท</Cdistrict>
    <Csubdistrict>สามเสนใน</Csubdistrict>
    <Cprovince>กรุงเทพมหานคร</Cprovince>
    <Czipcode>10500</Czipcode>
  </Invoice>
</Invoices>
```

Figure 5.4 An Example of Tax Invoice XML Document

Copyright by Mahidol University

```
<html>
 <head>
  <META http-equiv="Content-Type" content="text/html; charset=utf-8">
  <style type="text/css">
    A
    {
     COLOR: #0A57A4;
     FONT-FAMILY: MS Sans Serif, Arial;
     FONT-SIZE: 10pt;
     TEXT-DECORATION: none
    }
    TD
    {
     COLOR: #000000;
     FONT-FAMILY: MS Sans Serif, Arial;
     FONT-SIZE: 10pt;
     TEXT-DECORATION: none
    }
    .lpink
    {
     COLOR: #664e38;
     FONT-FAMILY: Ms sans serif;
     FONT-SIZE: 10pt;
     FONT-WEIGHT: regular
    }
    .pink
    {
     COLOR: #777777;
     FONT-FAMILY: Ms sans serif;
     FONT-SIZE: 10pt;
     FONT-WEIGHT: regular
    }
  </style>
 </head>
 <body>
<table border="0" cellpadding="3" cellspacing="0">
    <tr>
     <td valign="top">
       <b>บริษัท  บริษัท แดรี่เบลล์ จำกัด</b>เลขประจำตัวผู้เสียภาษีอากร  :  1000398104<br>
       <b>ที่อยู่ : </b>-175 ถนน สุขุมวิท 21(อโศก) แขวง  คลองเตยเหนือ เขต วัฒนา
          <br>กรุงเทพฯ รหัสไปรษณีย์ 10110<br><b>โทรศัพท์ </b> -  <b>แฟกซ์</b> -</td>
     <td width="30"></td>
     <td valign="top"><b>ต้นฉบับ ใบเสร็จรับเงิน/ใบกำกับภาษี</b><br>
        Receipt/Tax Invoice <br>วันที่ 21/10/2550</td>
    </tr>
    <tr>
     <td colspan="3"><br><b>ชื่อและที่อยู่ตามภาษีมูลค่าเพิ่ม</b><br>
       <b>บริษัท ที อาร์ เอฟ อิมปอร์ต แอนด์ เอ็กซ์ปอร์ต</b><br>-  448/1 ซอย  พหลโยธิน 30
          ถนน  พหลโยธิน แขวง  สามเสนใน เขต  พญาไท          กรุงเทพมหานคร10500<br><br>
```

Figure 5.5 An Example of HTML Document Result

Copyright by Mahidol University

```
        <table border="0" cellpadding="3" cellspacing="1" width="95%" bgcolor="#000000">
          <tr>
            <td align="center" bgcolor="#ffffff"><b>รายการ</b></td>
            <td align="center" bgcolor="#ffffff"><b>จำนวน</b></td>
            <td align="center" bgcolor="#ffffff"><b>จำนวนเงิน(ต่อหน่วย)</b></td>
            <td align="center" bgcolor="#ffffff"><b>VAT</b></td>
            <td align="center" bgcolor="#ffffff"><b>รวมเงิน/TOTAL</b></td>
          </tr>
          <tr>
            <td bgcolor="#ffffff">กระดาย A4 80 แกรม</td>
            <td align="center" bgcolor="#ffffff">20</td>
            <td align="right" bgcolor="#ffffff">57.50</td>
            <td align="right" bgcolor="#ffffff">80.50</td>
            <td align="right" bgcolor="#ffffff">1230.50</td>
          </tr>
          <tr>
            <td bgcolor="#ffffff">ปากกาน้ำเงิน</td>
            <td align="center" bgcolor="#ffffff">30</td>
            <td align="right" bgcolor="#ffffff">52.25</td>
            <td align="right" bgcolor="#ffffff">109.73</td>
            <td align="right" bgcolor="#ffffff">1677.23</td>
          </tr>
          <tr>
            <td align="right" bgcolor="#ffffff" colspan="3"><b>รวม</b></td>
            <td align="right" bgcolor="#ffffff">190.23</td>
            <td align="right" bgcolor="#ffffff">2771.85</td>
          </tr>
          <tr>
            <td align="right" bgcolor="#ffffff" colspan="4"><b>ส่วนลด</b></td>
            <td align="right" bgcolor="#ffffff">135.88</td>
          </tr>
          <tr>
            <td align="right" bgcolor="#ffffff" colspan="4"><b>รวมทั้งสิ้น</b></td>
            <td align="right" bgcolor="#ffffff">2717.50</td>
          </tr>
        </table>
      </td>
    </tr>
  </table>
  <br><br>
 </body>
</html>
```

Figure 5.5 An Example of HTML Document Result (Cont.)

2) Business

- **Send Tax Invoice Data Page** is shown in Figure 5.6. In this page user can fill in tax invoice slip detail: Customer's detail and Product's detail and click on "Send Button" for sending data to RD online. When user presses "Send Button" to

Copyright by Mahidol University

send data to RD, this page will display Sending Result Message at the bottom of the page.

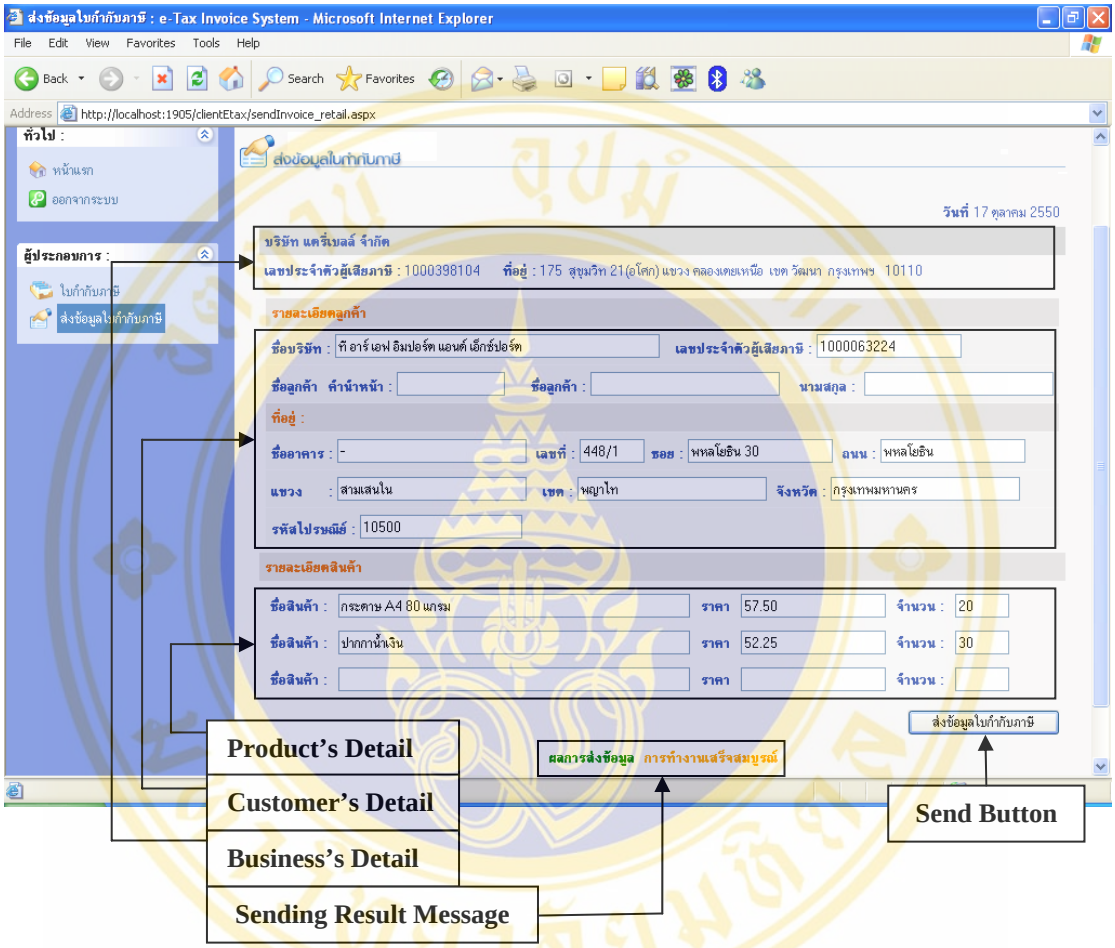


Figure 5.6 Send Tax Invoice Data Page

3) RD Officer

- **Login Page** is shown in Figure 5.7. In this page user can fill in username and password for login to the system. This page can verify username and password and display Main Menu Page shown in Figure 5.8 if username and password are valid. Otherwise this page will display login form with error massage for user to fill in username and password again.

- **Main Menu Page** is shown in Figure 5.8. In this page user can choose operation: User Account Menu, Log Menu, Report Menu, Invoice Menu, and Send Invoice Data Menu by click on left menu or icon menu on the right of page.

Copyright by Mahidol University

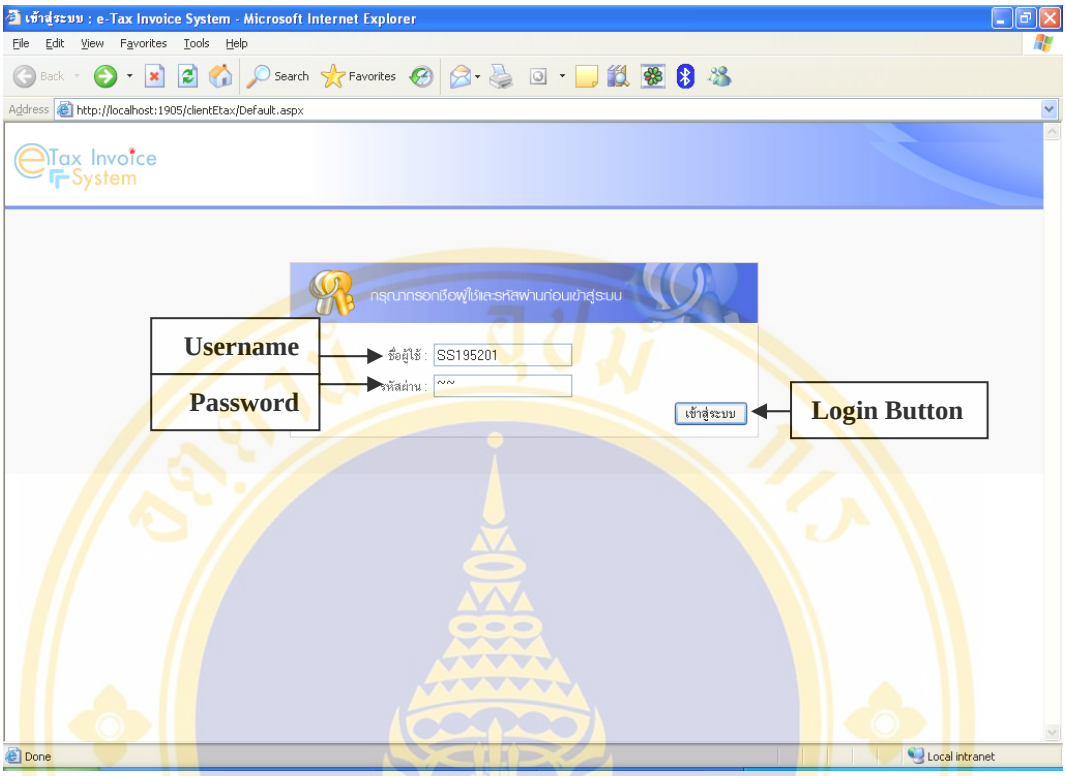


Figure 5.7 Login Page

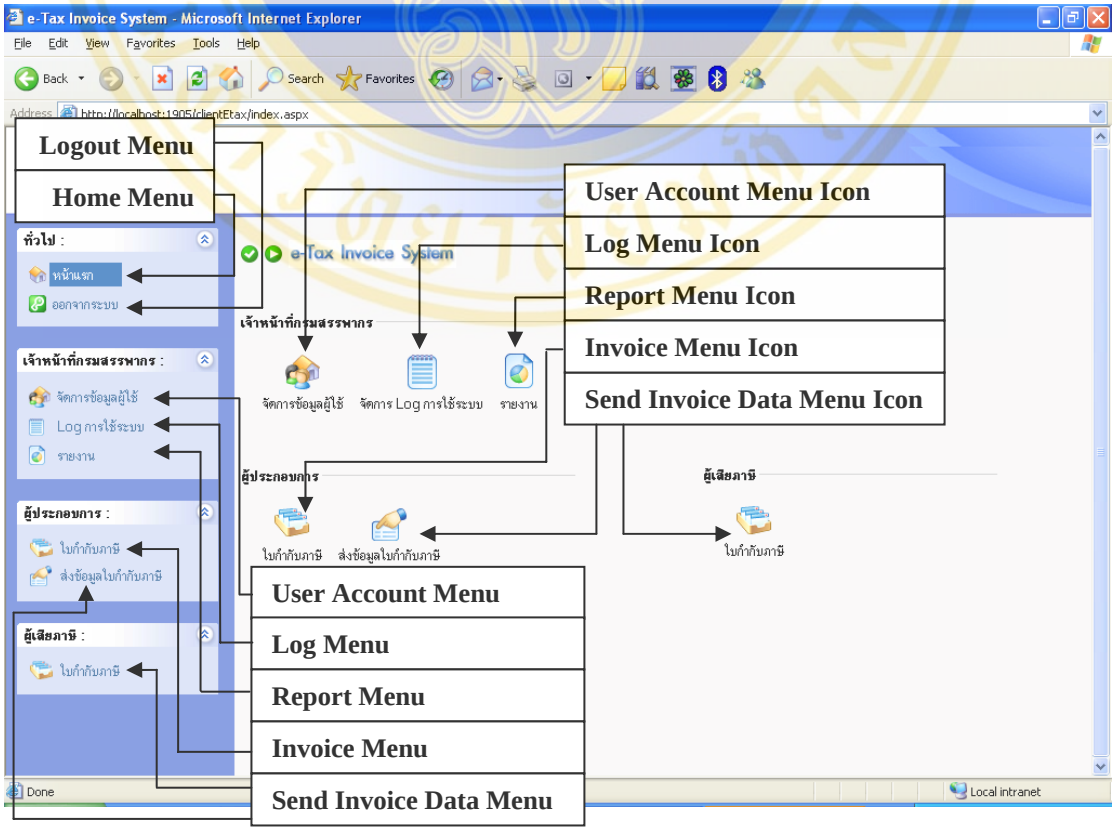


Figure 5.8 Main Menu Page

Copyright by Mahidol University

- **User Account Main Page** is shown in Figure 5.9. In this page user can view list of all user accounts. Moreover user can click on “Add Icon” to display Add User Account Page (Figure 5.10) for adding new user account, on “Edit Icon” to display Edit User Account Page (Figure 5.11) for update user and on “Delete Icon” for delete user account. If user clicks on delete icon, Confirm User Removal Message Dialog shown in Figure 5.12 will be displayed. If user wants to confirm the user removal, user can click on “OK Button”, otherwise click on “Cancel Button”.

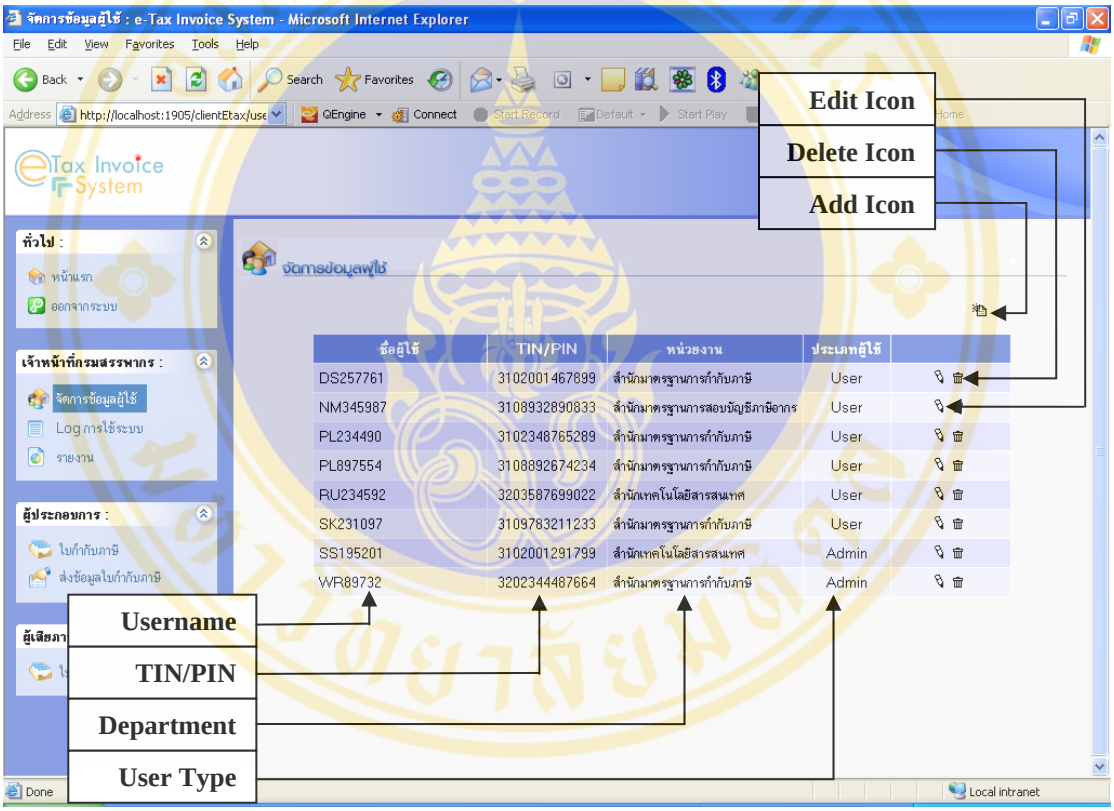


Figure 5.9 User Account Main Page

Copyright by Mahidol University

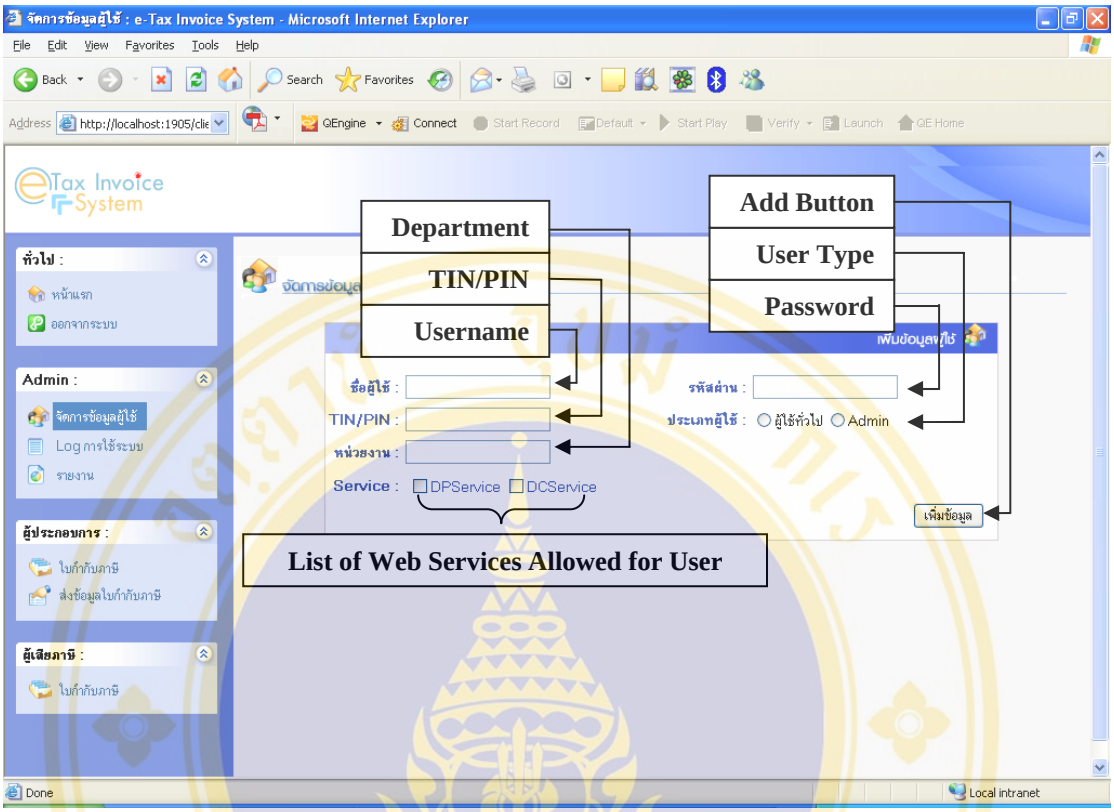


Figure 5.10 Add User Account Page

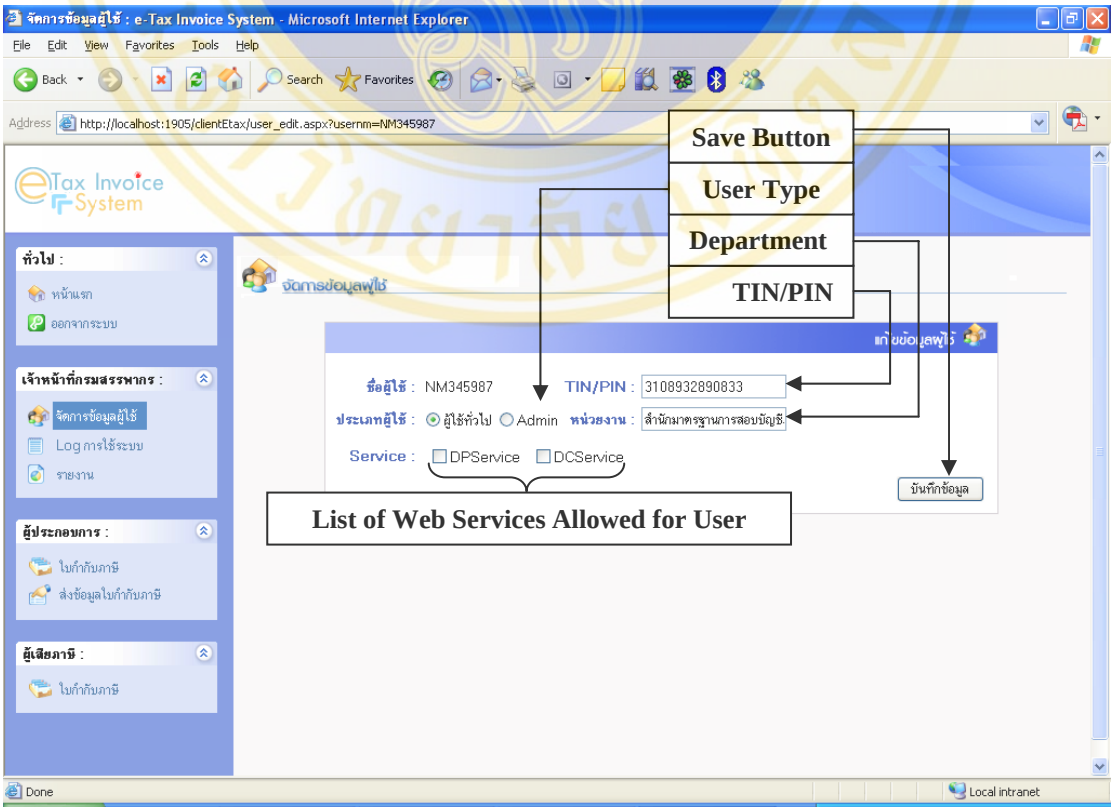


Figure 5.11 Edit User Account Page

Copyright by Mahidol University

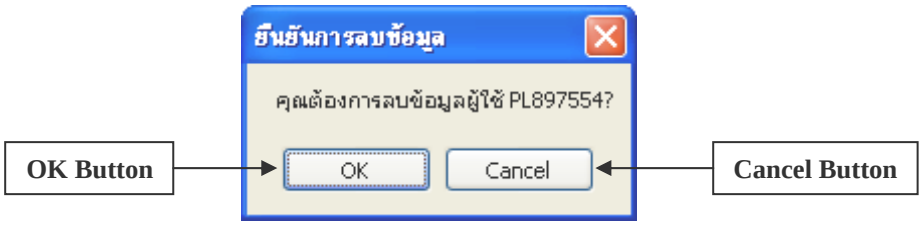


Figure 5.12 Confirm User Removal Message Dialog

- **Summary Monthly Log Report Page** is shown in Figure 5.13. In this page user can view summary monthly log report. Moreover user can clicks on month-year name to display Log Detail Page shown in Figure 5.14 for viewing log detail report related to selected month and year.

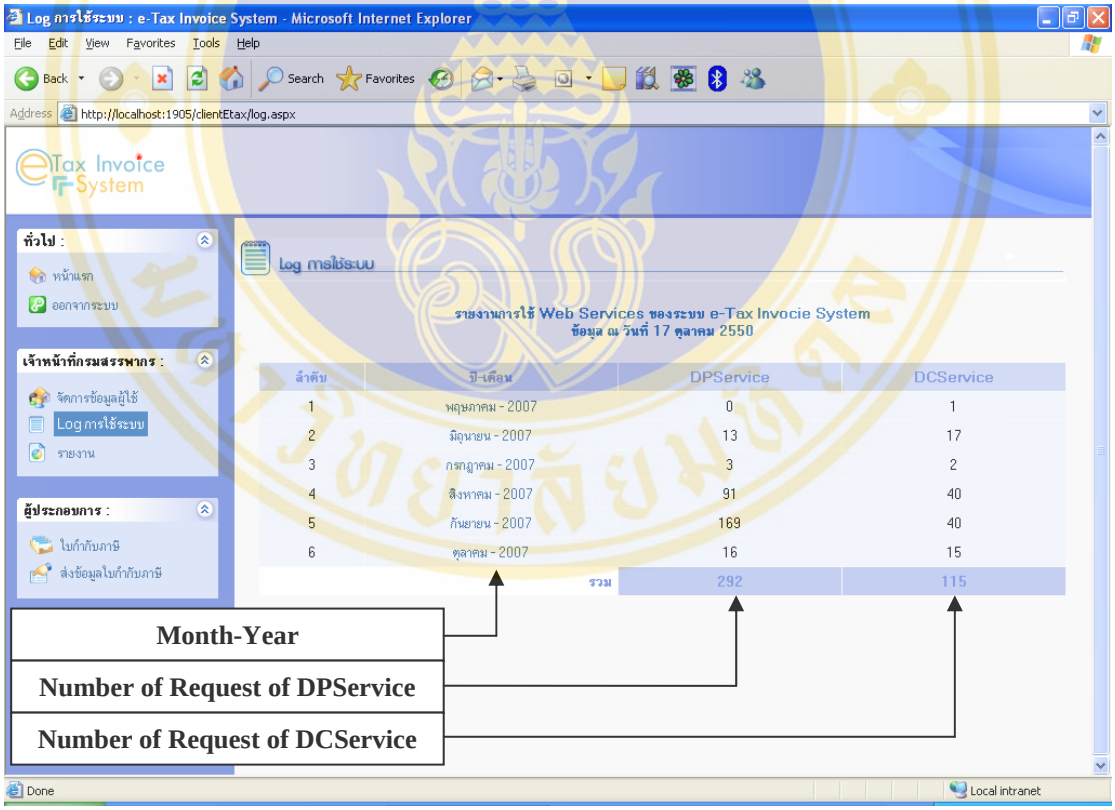


Figure 5.13 Summary Monthly Log Report Page

- **Log Detail Page** is shown in Figure 5.14. In this page user can view detail of log report corresponding to selected month and year.

Copyright by Mahidol University

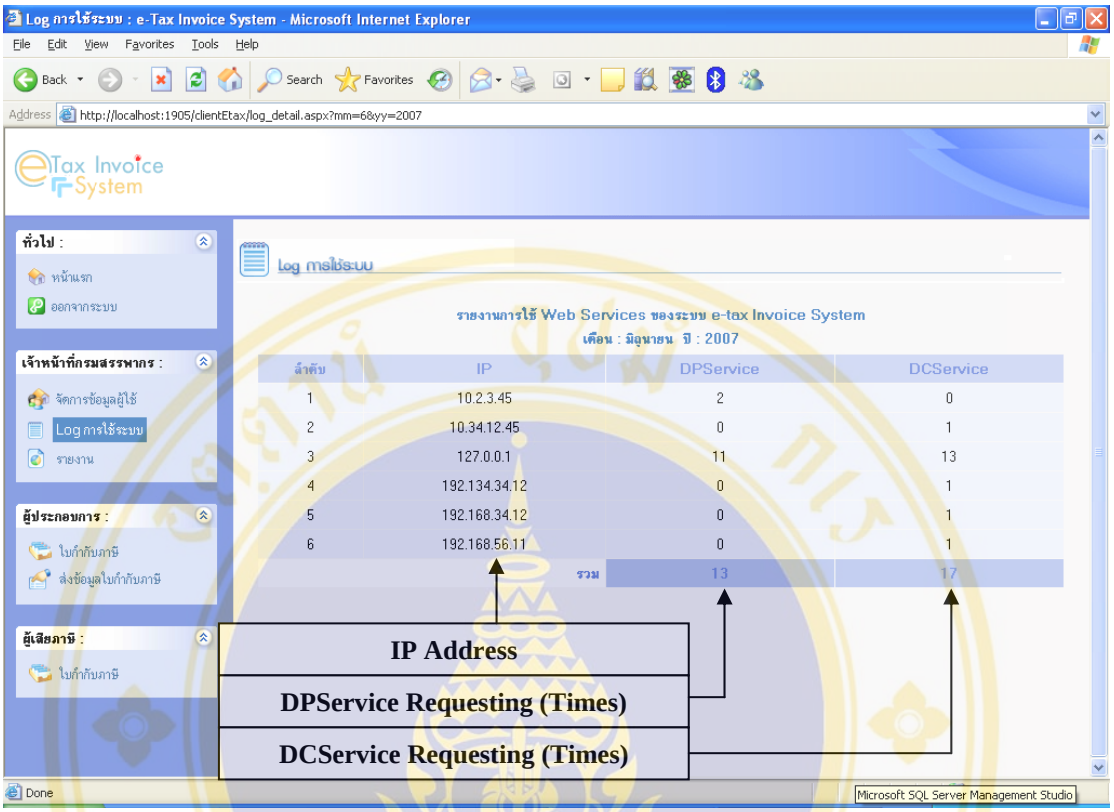


Figure 5.14 Log Detail Page

- **Report Main Page** is shown in Figure 5.15. In this page user can choose type of report user want to view. The system equips two report types: Purchase/Sales Summary Report and Total Sales Summary Report. To view the former type of report user can click on Purchase/Sales Summary Report Menu to display Purchases/Sales Summary Report Page shown in Figure 5.16. To view the latter type of report, user can click on Total Sales Summary Report Menu to show Total Sales Summary Report Page shown in Figure 5.17.

- **Purchase/Sales Summary Report Page** is shown in Figure 5.16. In this page user can perceive monthly summary report of business's sales and purchase. This report can help user especially tax supervisors to examine business's tax payment and tax return. To view report, user can specify criteria: Business's name, Business's TIN, Business's Branch Code, Customer's name, Customer's TIN, and Sales date used to generated report and click on "Search Button" to view report corresponded to specified conditions.

Copyright by Mahidol University

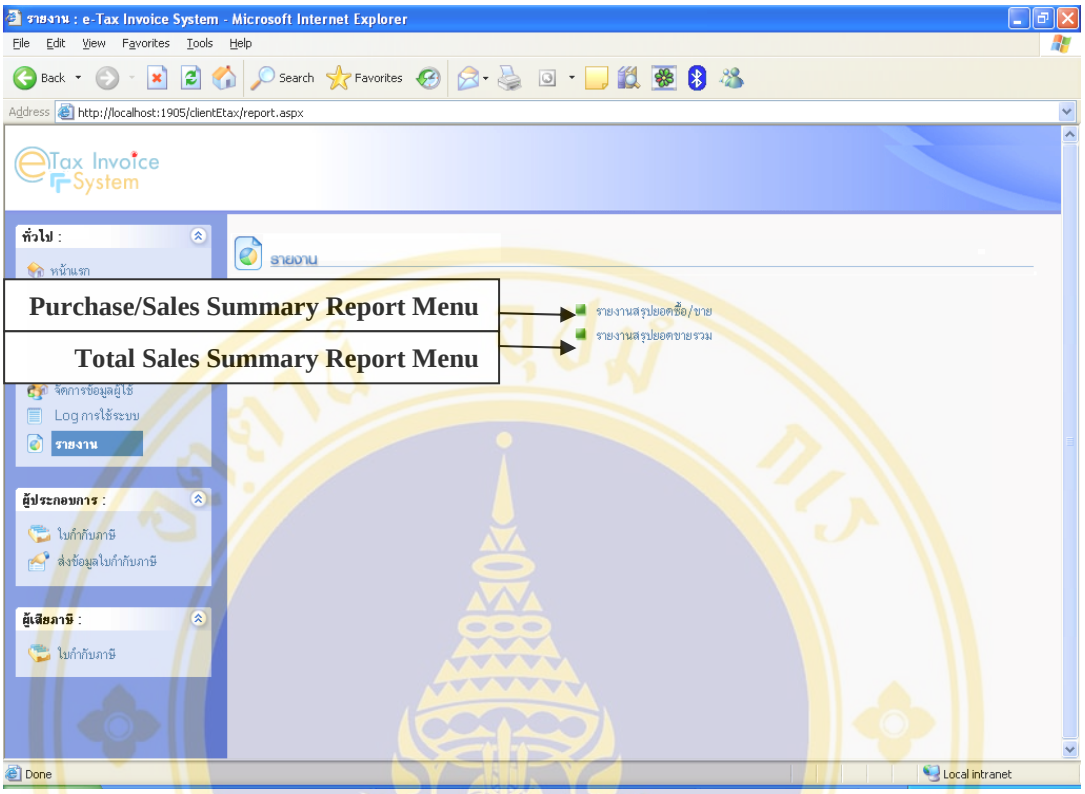


Figure 5.15 Report Main Page

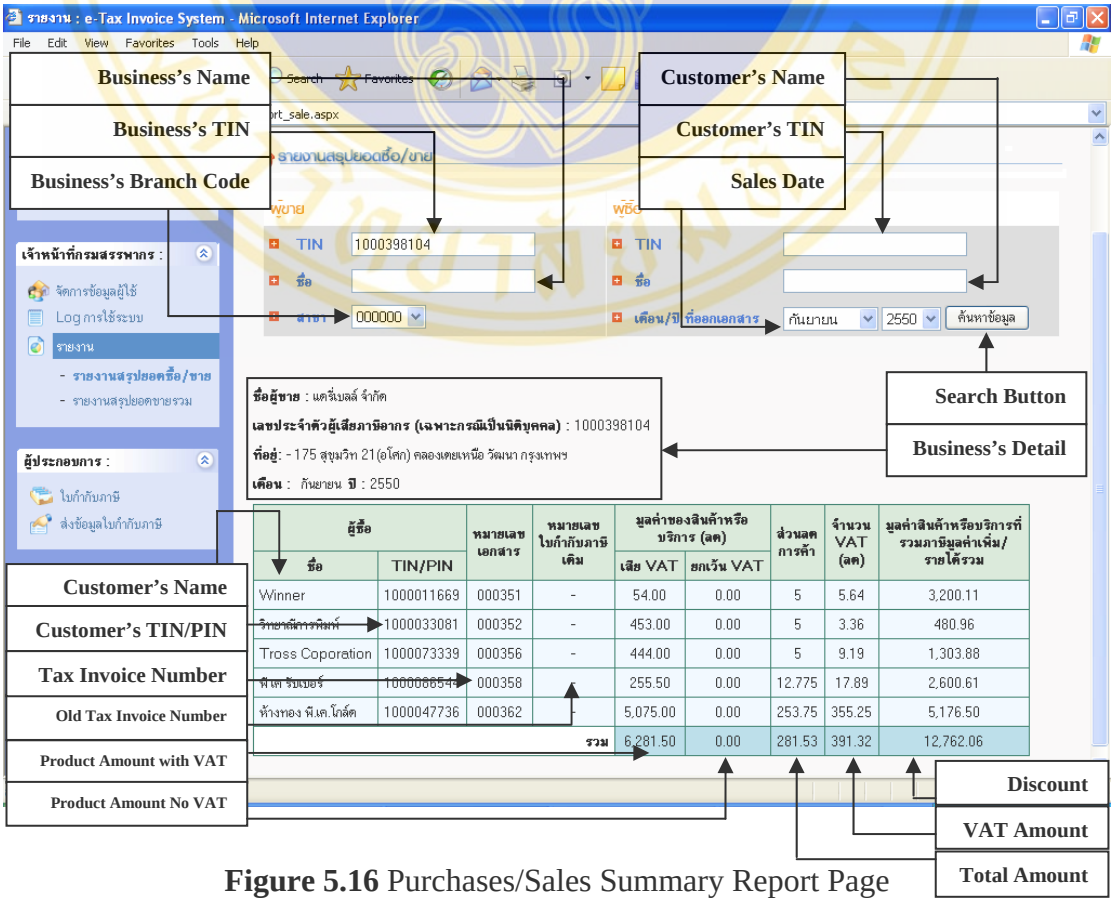


Figure 5.16 Purchases/Sales Summary Report Page

Copyright by Mahidol University

- **Total Sales Summary Report Page** is shown in Figure 5.17. In this page user can view Total Sales Summary Report of business's sales and purchase. This report can be generated to graph. This will facilitate user to analyze data easier. To view the report user can specify criteria: Business's name, Business's TIN, Business's Branch Code, and range of Sales date used to created report and click on "Search Button" to view report corresponding to specified conditions. Moreover user can view graph shown in Figure 5.18 corresponded to generated report.

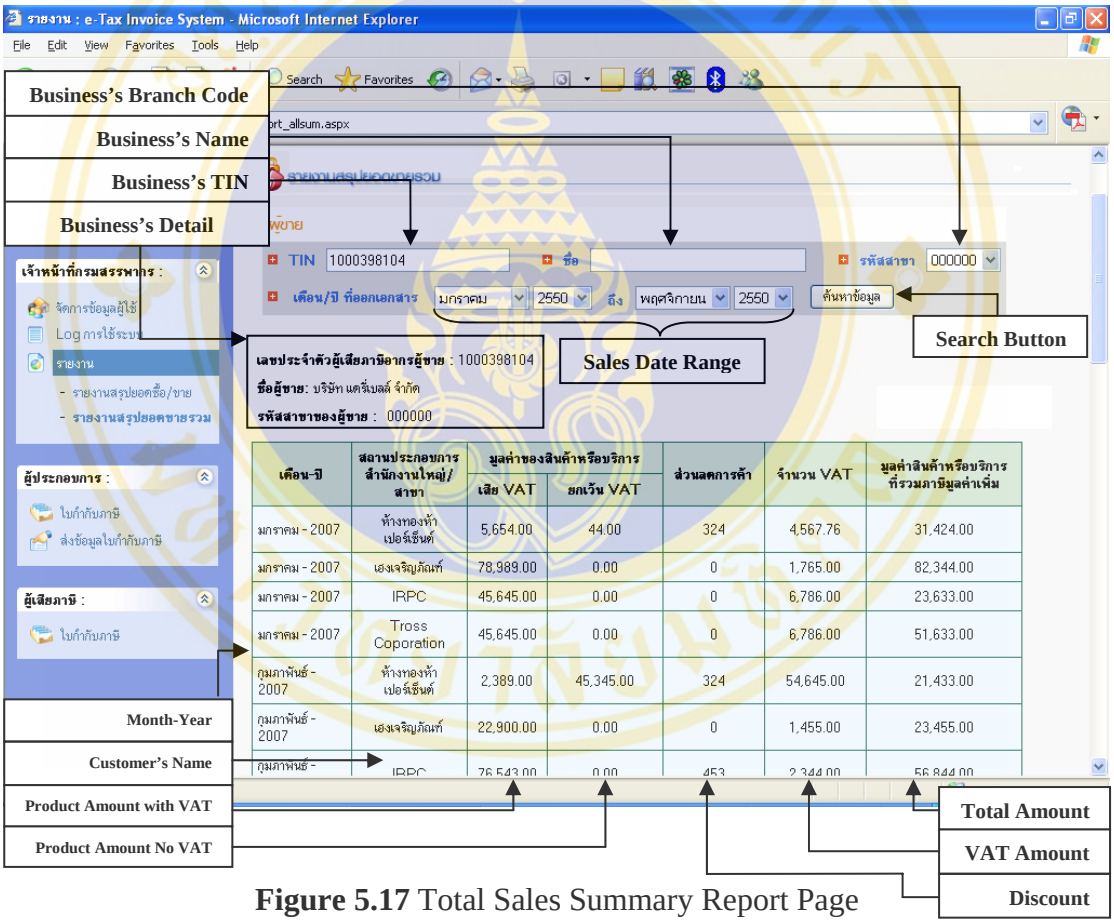


Figure 5.17 Total Sales Summary Report Page

- **Total Sales Summary Report Graph Page** is shown in Figure 5.18. In this page user can see graph corresponded to generated report. In X axis is month and year and Y axis is sales amount. Moreover the box at the top right of page shows business's name in graph that related with color.

Copyright by Mahidol University

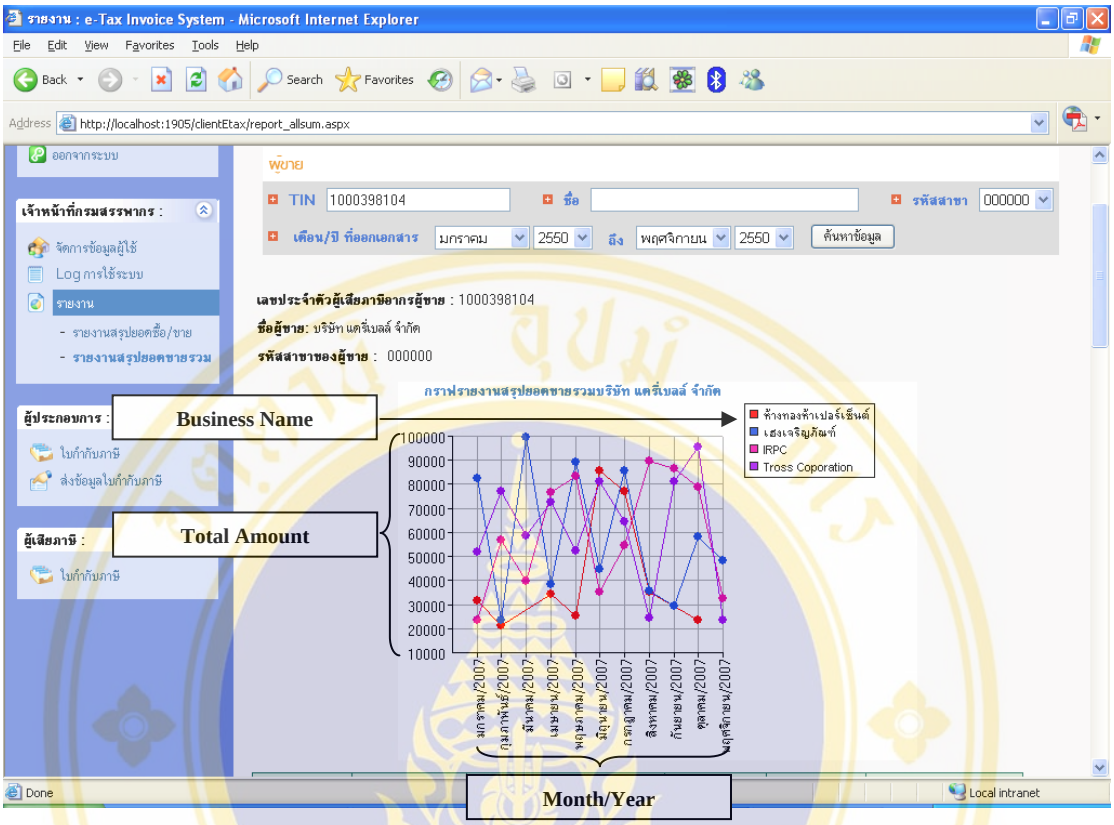


Figure 5.18 Total Sales Summary Report Graph Page

5.2.2 Web Services

In e-Tax Invoice System, there are five web services: UAService, DPService, DCService, RGService, and LRGService. These Web Services are categorized into two groups: Front-end and Back-end as follows:

5.2.2.1 Front-end Group

There are three Web Services in this group provided for business/customer. These Web Services are described below:

1. **UAService:** is responsible for authenticate user. This Web Service is called by DPService and DCService. Detail of WSDL, Input Schema and Output Schema of UAService is described as follows**:**

- **WSDL of UAService**

WSDL of UAService is represented in Figure 5.19

Copyright by Mahidol University

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="ServiceRD"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" targetNamespace="ServiceRD"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
 <wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="ServiceRD">
   <s:element name="UserAuthentication">
    <s:complexType>
     <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="username" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="password" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="typeofservice" type="s:string" />
     </s:sequence>
    </s:complexType>
   </s:element>
   <s:element name="UserAuthenticationResponse">
    <s:complexType>
     <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="UserAuthenticationResult">
       <s:complexType>
        <s:sequence>
         <s:element ref="s:schema" />
         <s:any />
        </s:sequence>
       </s:complexType>
      </s:element>
     </s:sequence>
    </s:complexType>
   </s:element>
  </s:schema>
 </wsdl:types>
 <wsdl:message name="UserAuthenticationSoapIn">
  <wsdl:part name="parameters" element="tns:UserAuthentication" />
 </wsdl:message>
 <wsdl:message name="UserAuthenticationSoapOut">
  <wsdl:part name="parameters" element="tns:UserAuthenticationResponse" />
 </wsdl:message>
 <wsdl:portType name="user_authenSoap">
  <wsdl:operation name="UserAuthentication">
   <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Check username and
        Password</wsdl:documentation>
   <wsdl:input message="tns:UserAuthenticationSoapIn" />
   <wsdl:output message="tns:UserAuthenticationSoapOut" />
  </wsdl:operation>
 </wsdl:portType>
```

Figure 5.19 WSDL of UAService

Copyright by Mahidol University

```
  <wsdl:binding name="user_authenSoap" type="tns:user_authenSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="UserAuthentication">
      <soap:operation soapAction="ServiceRD/UserAuthentication" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="user_authenSoap12" type="tns:user_authenSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="UserAuthentication">
      <soap12:operation soapAction="ServiceRD/UserAuthentication" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="user_authen">
    <wsdl:port name="user_authenSoap" binding="tns:user_authenSoap">
      <soap:address location="http://localhost:1045/etax_web2/user_authen.asmx" />
    </wsdl:port>
    <wsdl:port name="user_authenSoap12" binding="tns:user_authenSoap12">
      <soap12:address location="http://localhost:1045/etax_web2/user_authen.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Figure 5.19 WSDL of UAService (Cont.)

- **Input/Output Schema of UAService**

Input/Output Schema of UAService can be represented in two styles: Design View (Generated from XML Spy 4.0 [29]) and Code View in Figure 5.20 and 5.21 respectively.

Copyright by Mahidol University

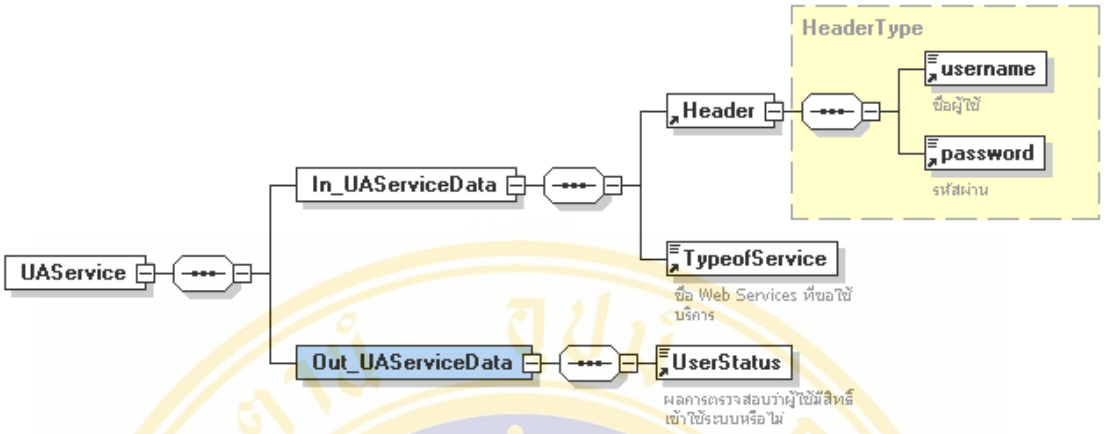


Figure 5.20 Input/Output Schema of UAService: Design View

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="UAService">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="In_UAServiceData">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="Header"/>
                            <xs:element ref="TypeofService"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Out_UAServiceData">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="UserStatus"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Header" type="HeaderType"/>
    <xs:element name="username" type="xs:string"/>
    <xs:element name="password" type="xs:string/">
    <xs:element name="TypeofService" type="xs:string"/>
    <xs:element name="UserStatus" type="xs:string"/>
    <xs:complexType name="HeaderType">
        <xs:sequence>
            <xs:element ref="username"/>
            <xs:element ref="password"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

Figure 5.21 Input/Output Schema of UAService: Code View

Copyright by Mahidol University

2. DPService: is responsible for provide tax invoice corresponded to specified conditions. This web service called by Business/Customer. Detail of WSDL, Input Schema and Output Schema of DPService is described as follows**:**

- **WSDL of DPService**

WSDL of DPService is represented in Figure 5.22

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="ServiceRD"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" targetNamespace="ServiceRD"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
 <wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="ServiceRD">
   <s:element name="InvoiceProvider">
    <s:complexType>
     <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="username" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="passwd" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="TIN" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="CBusinessname" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="CLname" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="InvDate" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="BusinessName" type="s:string" />
     </s:sequence>
    </s:complexType>
   </s:element>
   <s:element name="InvoiceProviderResponse">
    <s:complexType>
     <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="InvoiceProviderResult">
       <s:complexType mixed="true">
        <s:sequence>
         <s:any />
        </s:sequence>
       </s:complexType>
      </s:element>
     </s:sequence>
    </s:complexType>
   </s:element>
  </s:schema>
 </wsdl:types>
 <wsdl:message name="InvoiceProviderSoapIn">
  <wsdl:part name="parameters" element="tns:InvoiceProvider" />
 </wsdl:message>
 <wsdl:message name="InvoiceProviderSoapOut">
  <wsdl:part name="parameters" element="tns:InvoiceProviderResponse" />
 </wsdl:message>
```

Figure 5.22 WSDL of DPService

Copyright by Mahidol University

```
  <wsdl:portType name="DPServiceSoap">
    <wsdl:operation name="InvoiceProvider">
      <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Provide Invoice data to
customer or business</wsdl:documentation>
      <wsdl:input message="tns:InvoiceProviderSoapIn" />
      <wsdl:output message="tns:InvoiceProviderSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="DPServiceSoap" type="tns:DPServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="InvoiceProvider">
      <soap:operation soapAction="ServiceRD/InvoiceProvider" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="DPServiceSoap12" type="tns:DPServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="InvoiceProvider">
      <soap12:operation soapAction="ServiceRD/InvoiceProvider" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="DPService">
    <wsdl:port name="DPServiceSoap" binding="tns:DPServiceSoap">
      <soap:address location="http://localhost:1045/etax_web2/DPService.asmx" />
    </wsdl:port>
    <wsdl:port name="DPServiceSoap12" binding="tns:DPServiceSoap12">
      <soap12:address location="http://localhost:1045/etax_web2/DPService.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Figure 5.22 WSDL of DPService (Cont.)

- **Input/Output Schema of DPService**

Input/Output Schema of DPService can be represented in two styles: Design View and Code View in Figure 5.23, and 5.28 respectively. In addition, Invoice Schema, Product Schema, CFullname Schema, and Caddr Schema are illustrated in Figure 5.24, 5.25, 5.26, and 5.27 respectively.

Copyright by Mahidol University

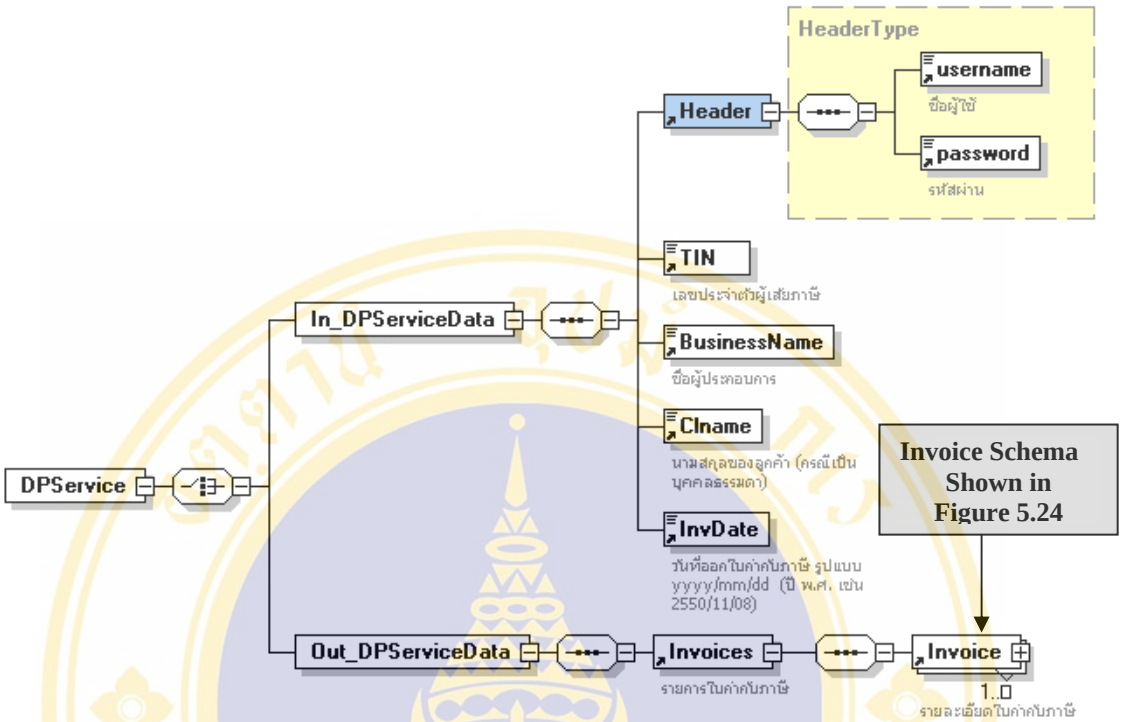


Figure 5.23 Input/Output Schema of DPService: Design View

Copyright by Mahidol University

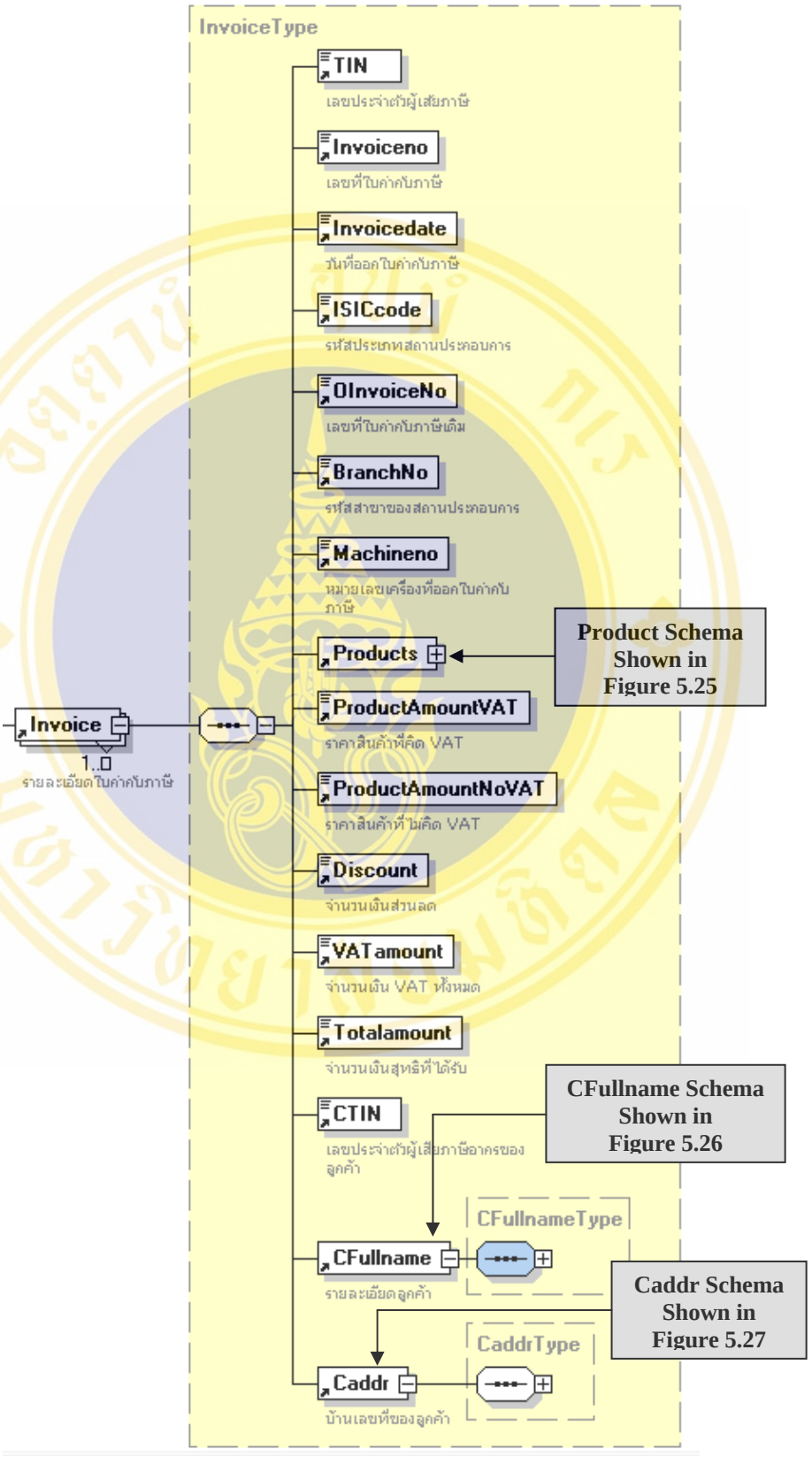


Figure 5.24 Invoice Schema: Design View

Copyright by Mahidol University

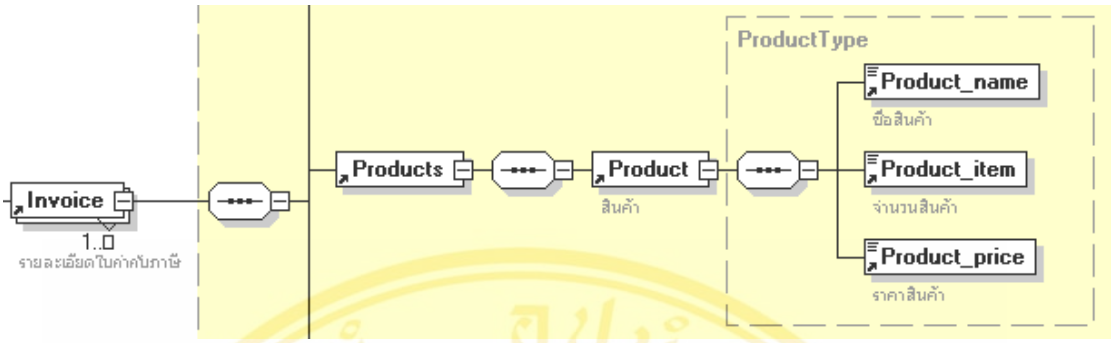


Figure 5.25 Product Schema: Design View

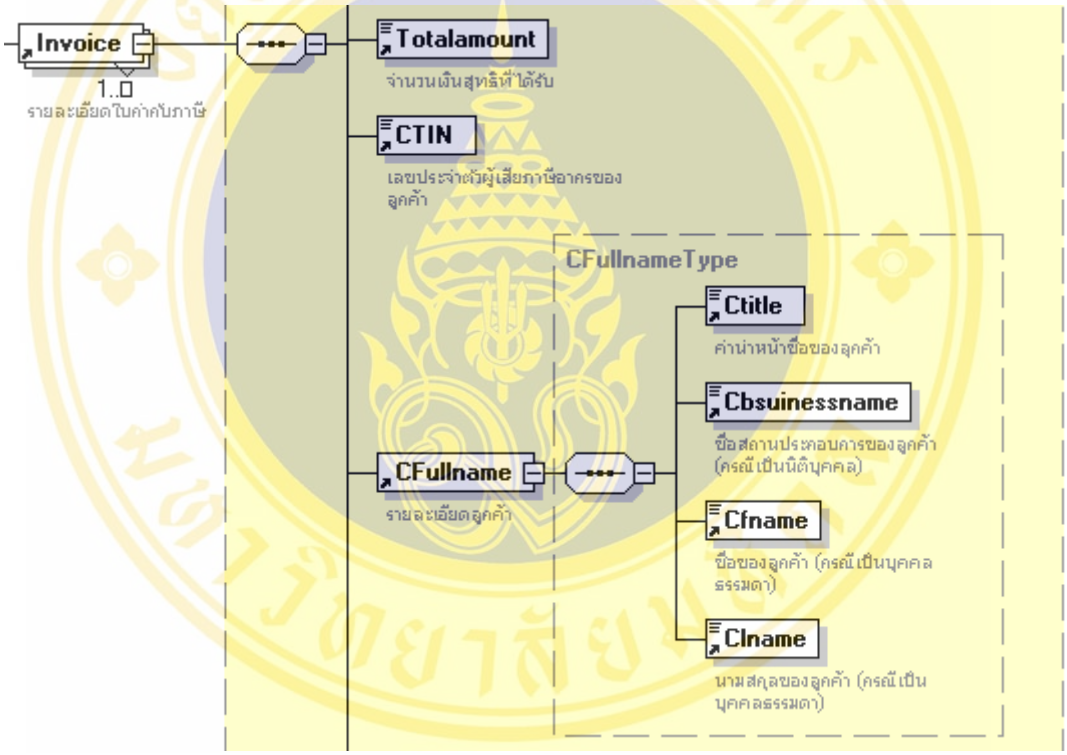


Figure 5.26 CFullname Schema: Design View

Copyright by Mahidol University

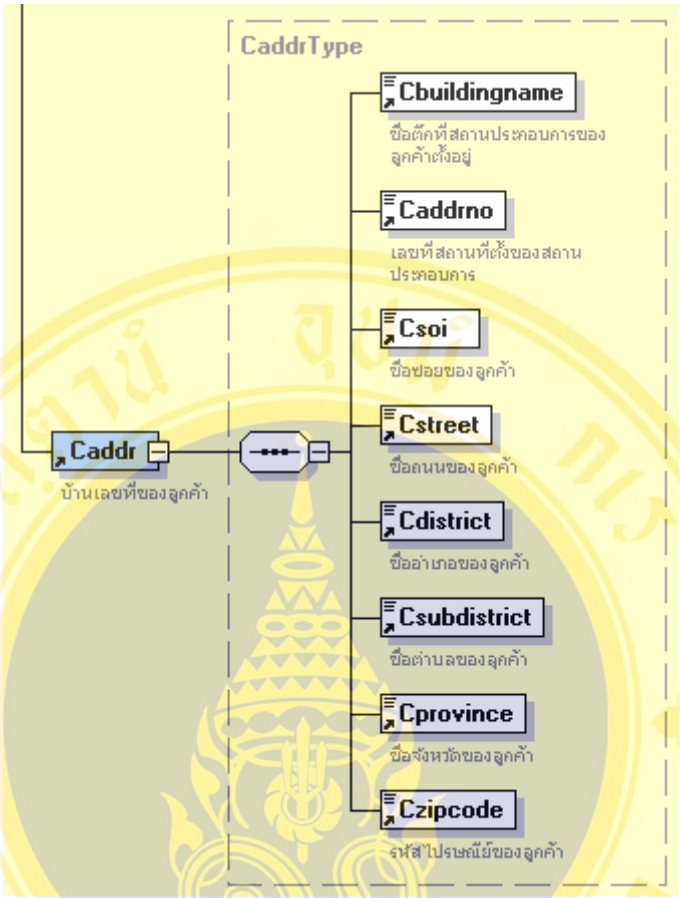


Figure 5.27 Caddr Schema: Design View

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="DPService">
        <xs:complexType>
            <xs:choice>
                <xs:element name="In_DPServiceData">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="Header"/>
                            <xs:element ref="TIN"/>
                            <xs:element ref="BusinessName"/>
                            <xs:element ref="Clname"/>
                            <xs:element ref="InvDate"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Out_DPServiceData">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="Invoices"/>
                        </xs:sequence>
```

Figure 5.28 Input/Output Schema of DPService: Code View

Copyright by Mahidol University

```
                </xs:complexType>
            </xs:element>
        </xs:choice>
    </xs:complexType>
</xs:element>
<xs:element name="Header" type="HeaderType"/>
<xs:element name="username" type="xs:string"/>
<xs:element name="password" type="xs:string"/>
<xs:element name="TIN" type="TINType"/>
<xs:element name="BusinessName" type="xs:string"/>
<xs:element name="Clname" type="xs:string"/>
<xs:element name="InvDate" type="InvDateType"/>
<xs:element name="Invoices">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Invoice" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Invoice" type="InvoiceType"/>
<xs:element name="Invoiceno" type="InvoiceNoType"/>
<xs:element name="Invoicedate" type="xs:string"/>
<xs:element name="ISICcode" type="ISICType"/>
<xs:element name="OInvoiceNo" type="xs:string"/>
<xs:element name="BranchNo" type="BranchCodeType"/>
<xs:element name="Machineno" type="xs:string"/>
<xs:element name="Products">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Product"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Product" type="ProductType"/>
<xs:element name="Product_name" type="xs:string"/>
<xs:element name="Product_item" type="xs:positiveInteger"/>
<xs:element name="Product_price" type="xs:decimal"/>
<xs:element name="ProductAmountVAT" type="xs:decimal"/>
<xs:element name="ProductAmountNoVAT" type="xs:decimal"/>
<xs:element name="Discount" type="xs:decimal"/>
<xs:element name="VATamount" type="xs:decimal"/>
<xs:element name="Totalamount" type="xs:decimal"/>
<xs:element name="CFullname" type="CFullnameType"/>
<xs:element name="Caddr" type="CaddrType"/>
<xs:element name="CTIN" type="TINType"/>
<xs:element name="Ctitle" type="xs:string"/>
<xs:element name="Cbsuinessname" type="xs:string"/>
<xs:element name="Cfname" type="xs:string"/>
<xs:element name="Clname" type="xs:string"/>
<xs:element name="Cbuildingname" type="xs:string"/>
<xs:element name="Caddrno" type="xs:string"/>
<xs:element name="Csoi" type="xs:string"/>
```

Figure 5.28 Input/Output Schema of DPService: Code View (Cont.)

Copyright by Mahidol University

```
<xs:element name="Cstreet" type="xs:string"/>
<xs:element name="Cdistrict" type="xs:string"/>
<xs:element name="Csubdistrict" type="xs:string"/>
<xs:element name="Cprovince" type="xs:string"/>
<xs:element name="Czipcode" type="xs:string"/>
<xs:complexType name="HeaderType">
    <xs:sequence>
        <xs:element ref="username"/>
        <xs:element ref="password"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="InvoiceType">
    <xs:sequence>
        <xs:element ref="TIN"/>
        <xs:element ref="Invoiceno"/>
        <xs:element ref="Invoicedate"/>
        <xs:element ref="ISICcode"/>
        <xs:element ref="OInvoiceNo"/>
        <xs:element ref="BranchNo"/>
        <xs:element ref="Machineno"/>
        <xs:element ref="Products"/>
        <xs:element ref="ProductAmountVAT"/>
        <xs:element ref="ProductAmountNoVAT"/>
        <xs:element ref="Discount"/>
        <xs:element ref="VATamount"/>
        <xs:element ref="Totalamount"/>
        <xs:element ref="CTIN"/>
        <xs:element ref="CFullname"/>
        <xs:element ref="Caddr"/>
    </xs:sequence>
 </xs:complexType>
<xs:complexType name="ProductType">
    <xs:sequence>
        <xs:element ref="Product_name"/>
        <xs:element ref="Product_item"/>
        <xs:element ref="Product_price"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="CFullnameType">
    <xs:sequence>
        <xs:element ref="Ctitle"/>
        <xs:element ref="Cbsuinessname"/>
        <xs:element ref="Cfname"/>
        <xs:element ref="Clname"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="CaddrType">
    <xs:sequence>
        <xs:element ref="Cbuildingname"/>
        <xs:element ref="Caddrno"/>
        <xs:element ref="Csoi"/>
        <xs:element ref="Cstreet"/>
        <xs:element ref="Cdistrict"/>
        <xs:element ref="Csubdistrict"/>
```

Figure 5.28 Input/Output Schema of DPService: Code View (Cont.)

Copyright by Mahidol University

```
            <xs:element ref="Cprovince"/>
            <xs:element ref="Czipcode"/>
        </xs:sequence>
    <xs:simpleType name="TINType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{10}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="ISICType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{6}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="BranchCodeType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{6}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="InvoiceNoType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{6}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="InvDateType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{4}[/][0-9]{2}[/][0-9]{2}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

Figure 5.28 Input/Output Schema of DPService: Code View (Cont.)

3. DCService: is responsible for collect tax invoice data. This Web Service is called by business. Business can immediately send tax invoice data to RD after customer make a payment by requesting this web service. Detail of WSDL, Input Schema and Output Schema of DCService is described as follows**:**

- **WSDL of DCService**

WSDL of DCService is represented in Figure 5.29

Copyright by Mahidol University

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://rdws.rd.go.th/ServiceRD/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://rdws.rd.go.th/ServiceRD/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
 <wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace=
        "http://rdws.rd.go.th/ServiceRD/">
   <s:element name="ReceiveInvoice">
    <s:complexType>
     <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="username" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="password" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string" />
     </s:sequence>
    </s:complexType>
   </s:element>
   <s:element name="ReceiveInvoiceResponse">
    <s:complexType>
     <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="ReceiveInvoiceResult">
       <s:complexType>
        <s:sequence>
         <s:element ref="s:schema" />
         <s:any />
        </s:sequence>
       </s:complexType>
      </s:element>
     </s:sequence>
    </s:complexType>
   </s:element>
  </s:schema>
 </wsdl:types>
 <wsdl:message name="ReceiveInvoiceSoapIn">
  <wsdl:part name="parameters" element="tns:ReceiveInvoice" />
 </wsdl:message>
 <wsdl:message name="ReceiveInvoiceSoapOut">
  <wsdl:part name="parameters" element="tns:ReceiveInvoiceResponse" />
 </wsdl:message>
 <wsdl:portType name="DCServiceSoap">
  <wsdl:operation name="ReceiveInvoice">
   <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Online import
       business's invoice data to database</wsdl:documentation>
   <wsdl:input message="tns:ReceiveInvoiceSoapIn" />
   <wsdl:output message="tns:ReceiveInvoiceSoapOut" />
  </wsdl:operation>
 </wsdl:portType>
```

Figure 5.29 WSDL of DCService

Copyright by Mahidol University

```
  <wsdl:binding name="DCServiceSoap" type="tns:DCServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="ReceiveInvoice">
      <soap:operation soapAction="http://rdws.rd.go.th/ServiceRD/ReceiveInvoice"
            style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="DCServiceSoap12" type="tns:DCServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="ReceiveInvoice">
      <soap12:operation soapAction="http://rdws.rd.go.th/ServiceRD/ReceiveInvoice"
          style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="DCService">
    <wsdl:port name="DCServiceSoap" binding="tns:DCServiceSoap">
      <soap:address location="http://localhost:1045/etax_web2/DCService.asmx" />
    </wsdl:port>
    <wsdl:port name="DCServiceSoap12" binding="tns:DCServiceSoap12">
      <soap12:address location="http://localhost:1045/etax_web2/DCService.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Figure 5.29 WSDL of DCService (Cont.)

- **Input/Output Schema of DCService**

Input/Output Schema of DCService can be represented in two styles: Design View and Code View in Figure 5.30 and 5.31 respectively.

Copyright by Mahidol University

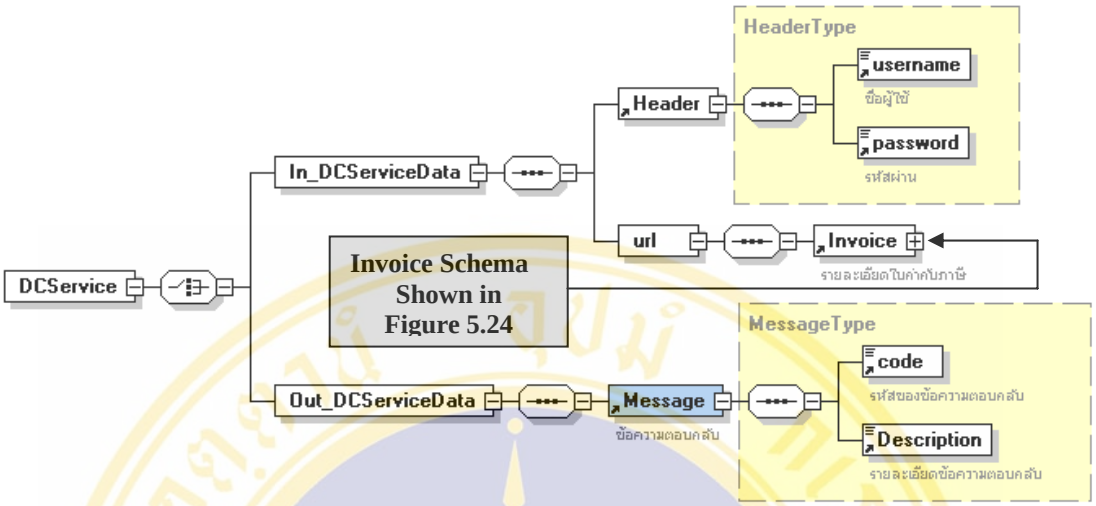


Figure 5.30 Input/Output Schema of DCService: Design View

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="DCService">
        <xs:complexType>
            <xs:choice>
                <xs:element name="In_DCServiceData">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="Header"/>
                            <xs:element name="url">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element ref="Invoice"/>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Out_DCServiceData">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="Message"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:element name="Header" type="HeaderType"/>
    <xs:element name="username" type="xs:string"/>
    <xs:element name="password" type="xs:string"/>
```

Figure 5.31 Input/Output Schema of DCService: Code View

Copyright by Mahidol University

```
<xs:element name="TIN" type="TINType"/>
<xs:element name="BusinessName" type="xs:string"/>
<xs:element name="Clname" type="xs:string"/>
<xs:element name="InvDate" type="InvDateType"/>
<xs:element name="Invoices">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Invoice" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Invoice" type="InvoiceType"/>
<xs:element name="Invoiceno" type="InvoiceNoType"/>
<xs:element name="Invoicedate" type="xs:string"/>
<xs:element name="ISICcode" type="ISICType"/>
<xs:element name="OInvoiceNo" type="xs:string"/>
<xs:element name="BranchNo" type="BranchCodeType"/>
<xs:element name="Machineno" type="xs:string"/>
<xs:element name="Products">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Product"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Product" type="ProductType"/>
<xs:element name="Product_name" type="xs:string"/>
<xs:element name="Product_item" type="xs:positiveInteger"/>
<xs:element name="Product_price" type="xs:decimal"/>
<xs:element name="ProductAmountVAT" type="xs:decimal"/>
<xs:element name="ProductAmountNoVAT" type="xs:decimal"/>
<xs:element name="Discount" type="xs:decimal"/>
<xs:element name="VATamount" type="xs:decimal"/>
<xs:element name="Totalamount" type="xs:decimal"/>
<xs:element name="CFullname" type="CFullnameType"/>
<xs:element name="Caddr" type="CaddrType"/>
<xs:element name="CTIN" type="TINType"/>
<xs:element name="Ctitle" type="xs:string"/>
<xs:element name="Cbsuinessname" type="xs:string"/>
<xs:element name="Cfname" type="xs:string"/>
<xs:element name="Clname" type="xs:string"/>
<xs:element name="Cbuildingname" type="xs:string"/>
<xs:element name="Caddrno" type="xs:string"/>
<xs:element name="Csoi" type="xs:string"/>
<xs:element name="Cstreet" type="xs:string"/>
<xs:element name="Cdistrict" type="xs:string"/>
<xs:element name="Csubdistrict" type="xs:string"/>
<xs:element name="Cprovince" type="xs:string"/>
<xs:element name="Czipcode" type="xs:string"/>
```

Figure 5.31 Input/Output Schema of DCService: Code View (Cont.)

Copyright by Mahidol University

```
<xs:complexType name="HeaderType">
    <xs:sequence>
        <xs:element ref="username"/>
        <xs:element ref="password"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="InvoiceType">
    <xs:sequence>
        <xs:element ref="TIN"/>
        <xs:element ref="Invoiceno"/>
        <xs:element ref="Invoicedate"/>
        <xs:element ref="ISICcode"/>
        <xs:element ref="OInvoiceNo"/>
        <xs:element ref="BranchNo"/>
        <xs:element ref="Machineno"/>
        <xs:element ref="Products"/>
        <xs:element ref="ProductAmountVAT"/>
        <xs:element ref="ProductAmountNoVAT"/>
        <xs:element ref="Discount"/>
        <xs:element ref="VATamount"/>
        <xs:element ref="Totalamount"/>
        <xs:element ref="CTIN"/>
        <xs:element ref="CFullname"/>
        <xs:element ref="Caddr"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ProductType">
    <xs:sequence>
        <xs:element ref="Product_name"/>
        <xs:element ref="Product_item"/>
        <xs:element ref="Product_price"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="CFullnameType">
    <xs:sequence>
        <xs:element ref="Ctitle"/>
        <xs:element ref="Cbsuinessname"/>
        <xs:element ref="Cfname"/>
        <xs:element ref="Clname"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="CaddrType">
    <xs:sequence>
        <xs:element ref="Cbuildingname"/>
        <xs:element ref="Caddrno"/>
        <xs:element ref="Csoi"/>
        <xs:element ref="Cstreet"/>
        <xs:element ref="Cdistrict"/>
        <xs:element ref="Csubdistrict"/>
        <xs:element ref="Cprovince"/>
        <xs:element ref="Czipcode"/>
    </xs:sequence>
```

Figure 5.31 Input/Output Schema of DCService: Code View (Cont.)

Copyright by Mahidol University

```
    <xs:simpleType name="TINType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{10}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="ISICType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{6}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="BranchCodeType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{6}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="InvoiceNoType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{6}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="InvDateType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{4}[/][0-9]{2}[/][0-9]{2}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

Figure 5.31 Input/Output Schema of DCService: Code View (Cont.)

5.2.2.2 Back-end Group

There are two Web Services in this group provided for RD Officer. These Web Services are described below:

1. **RGService**: is responsible for provide Sales/Purchase Amount data used to generating two types of report in this system. This web service is requested by this system. Detail of WSDL, Input Schema and Output Schema of RGService is presented as follows**:**

- **WSDL of RGService**

WSDL of RGService is represented in Figure 5.32

Copyright by Mahidol University

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://rdws.rd.go.th/SERVICERD/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://rdws.rd.go.th/SERVICERD/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://rdws.rd.go.th/SERVICERD/">
      <s:element name="ReportGenerator">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="username" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="password" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="BTIN" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="BName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="BranchNo" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="CName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="CTIN" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="InvDateFrom" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="InvDateTo" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="reporttype" type="s:int" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ReportGeneratorResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ReportGeneratorResult">
              <s:complexType>
                <s:sequence>
                  <s:element ref="s:schema" />
                  <s:any />
                </s:sequence>
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="ReportGeneratorSoapIn">
    <wsdl:part name="parameters" element="tns:ReportGenerator" />
  </wsdl:message>
  <wsdl:message name="ReportGeneratorSoapOut">
    <wsdl:part name="parameters" element="tns:ReportGeneratorResponse" />
  </wsdl:message>
  <wsdl:portType name="RGServiceSoap">
    <wsdl:operation name="ReportGenerator">
```

Figure 5.32 WSDL of RGService

Copyright by Mahidol University

```
      <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Generate data for
create sale/purchase report</wsdl:documentation>
      <wsdl:input message="tns:ReportGeneratorSoapIn" />
      <wsdl:output message="tns:ReportGeneratorSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="RGServiceSoap" type="tns:RGServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="ReportGenerator">
      <soap:operation soapAction="http://rdws.rd.go.th/SERVICERD/ReportGenerator"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="RGServiceSoap12" type="tns:RGServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="ReportGenerator">
      <soap12:operation soapAction="http://rdws.rd.go.th/SERVICERD/ReportGenerator"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="RGService">
    <wsdl:port name="RGServiceSoap" binding="tns:RGServiceSoap">
      <soap:address location="http://localhost:1045/etax_web2/RGService.asmx" />
    </wsdl:port>
    <wsdl:port name="RGServiceSoap12" binding="tns:RGServiceSoap12">
      <soap12:address location="http://localhost:1045/etax_web2/RGService.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Figure 5.32 WSDL of RGService (Cont.)

- **Input/Output Schema of RGService**

Input/Output Schema of RGService can be represented in two styles: Design View and Code View in Figure 5.33 and 5.37 respectively. In addition, ReportCond Schema, Sum_Report Schema, and Tot_Sum_Report Schema are illustrated in Figure 5.34, 5.35, and 5.36 respectively.

Copyright by Mahidol University

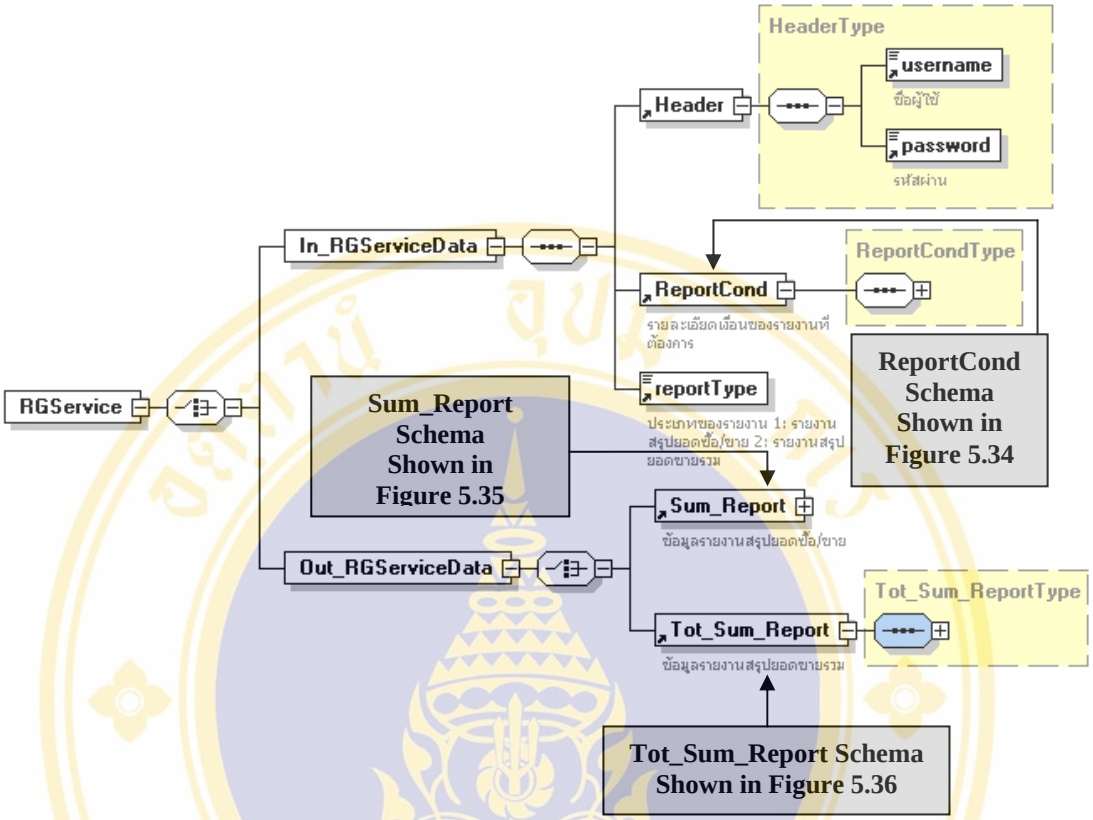


Figure 5.33 Input/Output Schema of RGService: Design View

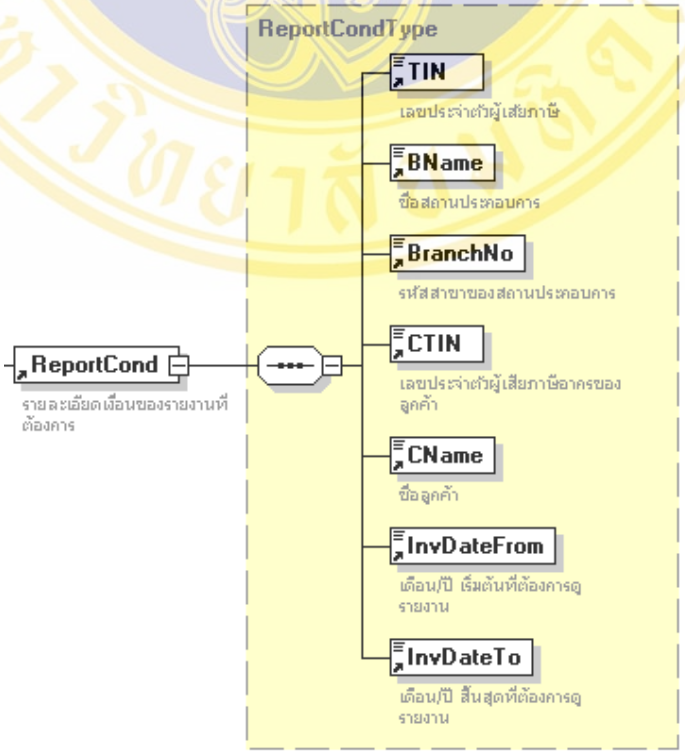


Figure 5.34 ReportCond Schema: Design View

Copyright by Mahidol University

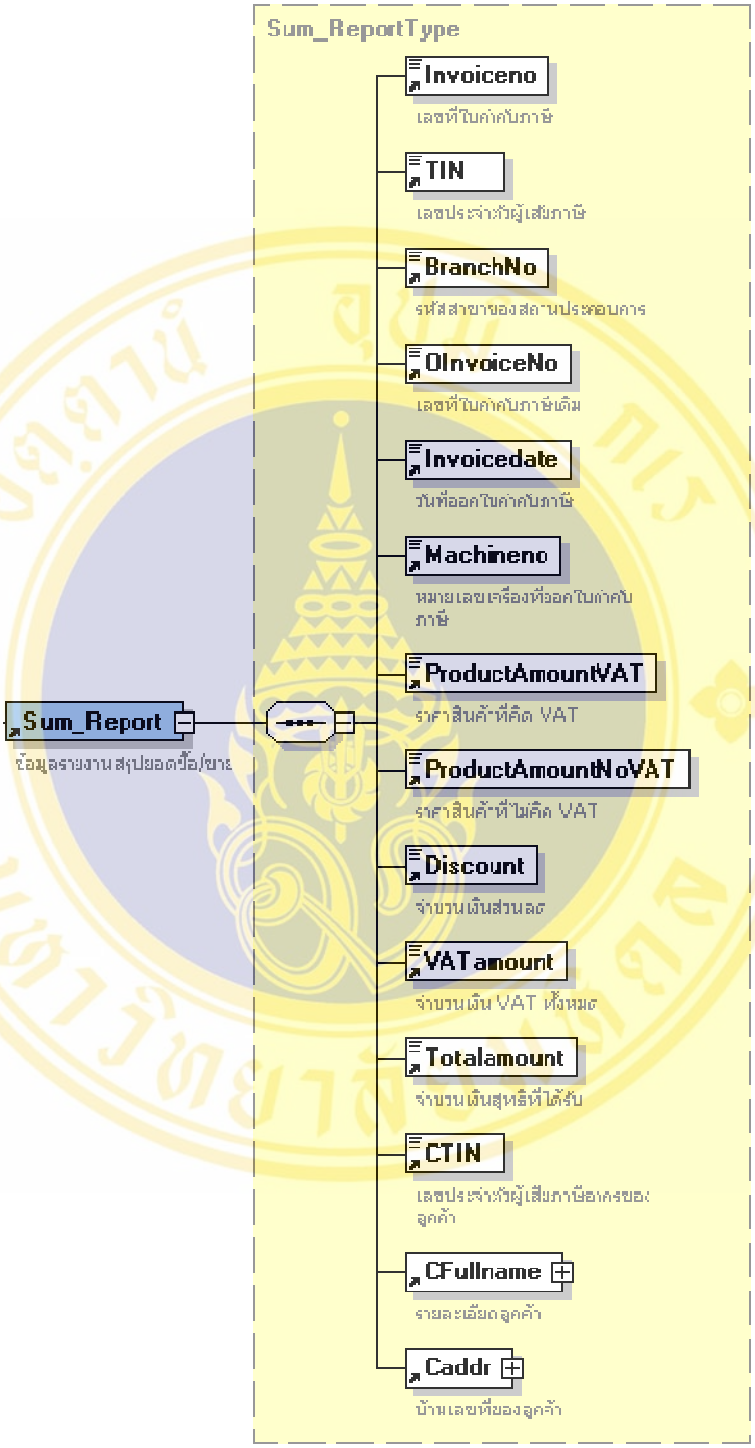


Figure 5.35 Sum_Report Schema: Design View

Copyright by Mahidol University

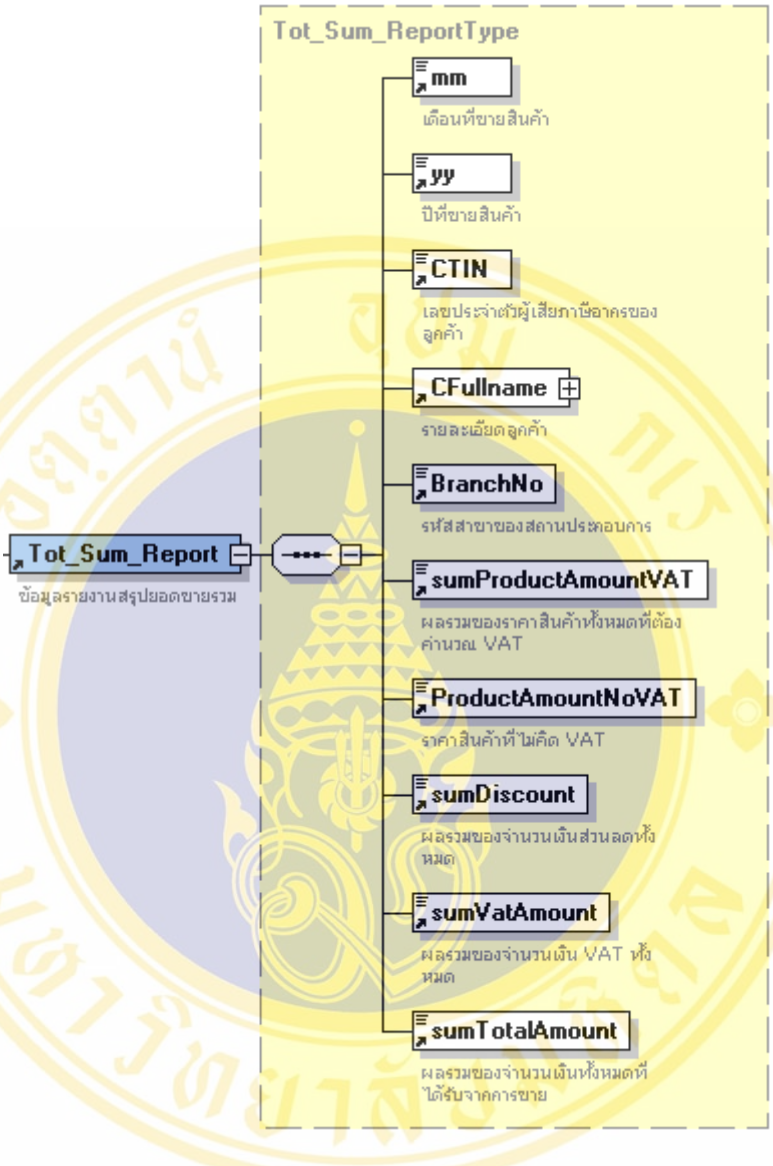


Figure 5.36 Tot_Sum_Report Schema: Design View

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="RGService">
        <xs:complexType>
            <xs:choice>
                <xs:element name="In_RGServiceData">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="Header"/>
                            <xs:element ref="ReportCond"/>
                            <xs:element ref="reportType"/>
                        </xs:sequence>
```

Figure 5.37 Input/Output Schema of RGService: Code View

Copyright by Mahidol University

```
                </xs:complexType>
            </xs:element>
            <xs:element name="Out_RGServiceData">
                <xs:complexType>
                    <xs:choice>
                        <xs:element ref="Sum_Report"/>
                        <xs:element ref="Tot_Sum_Report"/>
                    </xs:choice>
                </xs:complexType>
            </xs:element>
        </xs:choice>
    </xs:complexType>
</xs:element>
<xs:element name="Header" type="HeaderType"/>
<xs:element name="username" type="xs:string"/>
<xs:element name="password" type="xs:string"/>
<xs:element name="ReportCond" type="ReportCondType"/>
<xs:element name="reportType" type="reporttypeType">
<xs:element name="Sum_Report" type="Sum_ReportType"/>
<xs:element name="Tot_Sum_Report" type="Tot_Sum_ReportType"/>
<xs:element name="sumProductAmountVAT" type="xs:decimal"/>
<xs:element name="sumProductAmountNoVAT" type="xs:decimal"/>
<xs:element name="sumDiscount" type="xs:decimal"/>
<xs:element name="sumVatAmount" type="xs:decimal"/>
<xs:element name="sumTotalAmount" type="xs:decimal"/>
<xs:element name="TIN" type="TINType"/>
<xs:element name="BName" type="xs:string"/>
<xs:element name="BranchNo" type="BranchCodeType"/>
<xs:element name="CTIN" type="TINType"/>
<xs:element name="CName" type="xs:string"/>
<xs:element name="InvDateFrom" type="xs:date"/>
<xs:element name="InvDateTo" type="xs:date"/>
<xs:element name="Invoiceno" type="InvoiceNoType"/>
<xs:element name="Invoicedate" type="xs:string"/>
<xs:element name="ISICcode" type="ISICType"/>
<xs:element name="OInvoiceNo" type="xs:string"/>
<xs:element name="BranchNo" type="BranchCodeType"/>
<xs:element name="Machineno" type="xs:string"/>
<xs:element name="ProductAmountVAT" type="xs:decimal"/>
<xs:element name="ProductAmountNoVAT" type="xs:decimal"/>
<xs:element name="Discount" type="xs:decimal"/>
<xs:element name="VATamount" type="xs:decimal"/>
<xs:element name="Totalamount" type="xs:decimal"/>
<xs:element name="CTIN" type="TINType"/>
<xs:element name="CFullname" type="CFullnameType"/>
<xs:element name="Caddr" type="CaddrType"/>
```

Figure 5.37 Input/Output Schema of RGService: Code View (Cont.)

Copyright by Mahidol University

```
<xs:element name="mm" type="xs:positiveInteger"/>
<xs:element name="yy" type="xs:positiveInteger"/>
<xs:complexType name="HeaderType">
    <xs:sequence>
        <xs:element ref="username"/>
        <xs:element ref="password"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ReportCondType">
    <xs:sequence>
        <xs:element ref="TIN"/>
        <xs:element ref="BName"/>
        <xs:element ref="BranchNo"/>
        <xs:element ref="CTIN"/>
        <xs:element ref="CName"/>
        <xs:element ref="InvDateFrom"/>
        <xs:element ref="InvDateTo"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Sum_ReportType">
    <xs:sequence>
        <xs:element ref="Invoiceno"/>
        <xs:element ref="TIN"/>
        <xs:element ref="BranchNo"/>
        <xs:element ref="OInvoiceNo"/>
        <xs:element ref="Invoicedate"/>
        <xs:element ref="Machineno"/>
        <xs:element ref="ProductAmountVAT"/>
        <xs:element ref="ProductAmountNoVAT"/>
        <xs:element ref="Discount"/>
        <xs:element ref="VATamount"/>
        <xs:element ref="Totalamount"/>
        <xs:element ref="CTIN"/>
        <xs:element ref="CFullname"/>
        <xs:element ref="Caddr"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Tot_Sum_ReportType">
    <xs:sequence>
        <xs:element ref="mm"/>
        <xs:element ref="yy"/>
        <xs:element ref="CTIN"/>
        <xs:element ref="CFullname"/>
        <xs:element ref="BranchNo"/>
        <xs:element ref="sumProductAmountVAT"/>
        <xs:element ref="ProductAmountNoVAT"/>
        <xs:element ref="sumDiscount"/>
        <xs:element ref="sumVatAmount"/>
        <xs:element ref="sumTotalAmount"/>
    </xs:sequence>
</xs:complexType>
```

Figure 5.37 Input/Output Schema of RGService: Code View (Cont.)

Copyright by Mahidol University

```
    <xs:simpleType name="TINType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{10}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="ISICType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{6}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="BranchCodeType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{6}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="InvoiceNoType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{6}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="InvDateType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{4}[/][0-9]{2}[/][0-9]{2}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="reporttypeType">
        <xs:restriction base="xs:integer">
            <xs:enumeration value="1"/>
            <xs:enumeration value="2"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

Figure 5.37 Input/Output Schema of RGService: Code View (Cont.)

2. LRGService: is responsible for provide log detail data used to generate log report in this system. This web service is requested by the system. Detail of WSDL, Input Schema and Output Schema of LRGService is presented as follows**:**

- **WSDL of LRGService**

WSDL of LRGService is represented in Figure 5.38

Copyright by Mahidol University

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://rdws.rd.go.th/SERVICERD/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://rdws.rd.go.th/SERVICERD/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://rdws.rd.go.th/SERVICERD/">
      <s:element name="LogReportGenerator">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="username" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="password" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="LogMonth" type="s:int" />
            <s:element minOccurs="1" maxOccurs="1" name="LogYear" type="s:int" />
            <s:element minOccurs="0" maxOccurs="1" name="LogIP" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="LogServiceName" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="LogReportGeneratorResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="LogReportGeneratorResult"
type="s:int" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="LogReportGeneratorSoapIn">
    <wsdl:part name="parameters" element="tns:LogReportGenerator" />
  </wsdl:message>
  <wsdl:message name="LogReportGeneratorSoapOut">
    <wsdl:part name="parameters" element="tns:LogReportGeneratorResponse" />
  </wsdl:message>
  <wsdl:portType name="LRGServiceSoap">
    <wsdl:operation name="LogReportGenerator">
      <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Generate data for
create log report</wsdl:documentation>
      <wsdl:input message="tns:LogReportGeneratorSoapIn" />
      <wsdl:output message="tns:LogReportGeneratorSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
```

Figure 5.38 WSDL of LRGService

Copyright by Mahidol University

```
  <wsdl:binding name="LRGServiceSoap" type="tns:LRGServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="LogReportGenerator">
      <soap:operation soapAction="http://rdws.rd.go.th/SERVICERD/LogReportGenerator"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="LRGServiceSoap12" type="tns:LRGServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="LogReportGenerator">
      <soap12:operation soapAction="http://rdws.rd.go.th/SERVICERD/LogReportGenerator"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="LRGService">
    <wsdl:port name="LRGServiceSoap" binding="tns:LRGServiceSoap">
      <soap:address location="http://localhost:1045/etax_web2/LRGService.asmx" />
    </wsdl:port>
    <wsdl:port name="LRGServiceSoap12" binding="tns:LRGServiceSoap12">
      <soap12:address location="http://localhost:1045/etax_web2/LRGService.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Figure 5.38 WSDL of LRGService (Cont.)

- **Input/Output Schema of LRGService**

Input/Output Schema of LRGService can be represented in two styles: Design View and Code View in Figure 5.39 and 5.40 respectively.

Copyright by Mahidol University

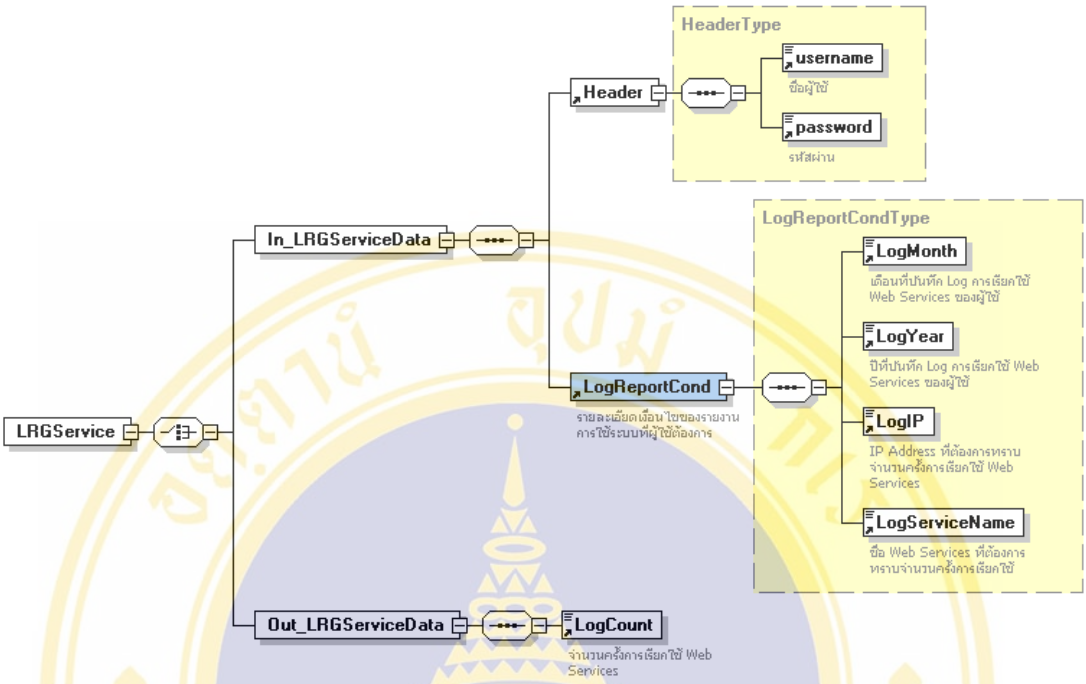


Figure 5.39 Input/Output Schema of LRGService: Design View

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="LRGService">
        <xs:complexType>
            <xs:choice>
                <xs:element name="In_LRGServiceData">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="Header"/>
                            <xs:element ref="LogReportCond"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Out_LRGServiceData">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="LogCount"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:choice>
        </xs:complexType>
    </xs:element>
```

Figure 5.40 Input/Output Schema of LRGService: Code View

Copyright by Mahidol University

```
    <xs:element name="Header" type="HeaderType"/>
    <xs:element name="username" type="xs:string"/>
    <xs:element name="password" type="xs:string"/>
    <xs:element name="LogReportCond" type="LogReportCondType"/>
    <xs:element name="LogMonth" type="xs:integer"/>
    <xs:element name="LogYear" type="xs:integer"/>
    <xs:element name="LogIP" type="xs:string"/>
    <xs:element name="LogServiceName" type="xs:string"/>
    <xs:element name="LogCount" type="xs:integer"/>
    <xs:complexType name="LogReportCondType">
        <xs:sequence>
            <xs:element ref="LogMonth"/>
            <xs:element ref="LogYear"/>
            <xs:element ref="LogIP"/>
            <xs:element ref="LogServiceName"/>
        </xs:sequence>
     </xs:complexType>
    <xs:simpleType name="TINType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{10}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

Figure 5.40 Input/Output Schema of LRGService: Code View (Cont.)

Furthermore, the system calls two existing Web Services of RD [26] as follows:

1. TIN/PINService: is responsible for checking validation and existence of Tax Identification Number (TIN) and Personal Identification Number (PIN) to prevent forging.

2. VATService: is responsible for providing comprehensive information of taxpayer who is the Value-added tax registrant. This will help the system to verify whether its partner can legally issue the tax invoice in accordance with the tax law in order to check if the tax invoice received is legal for his input tax credit. The service can also verify the existence of the business partners.

Copyright by Mahidol University

CHAPTER VI
SYSTEM EVALUATION AND RESULT

To evaluate the quality of e-Tax Invoice System, the researcher employs two main tests: usability testing and comparative analysis. The former aims to discover what user thinks and feels about using the system and to assess the perceived quality of use. The researcher develops questionnaire as a tool to measure user satisfaction. The latter intents to compare between the existing system and the new system to determine how the new system is better than the existing system.

6.1 Usability Testing

Usability Testing is the means to evaluate the degree to which users can interact effectively with the system and the degree to which the system guides user action, provides meaningful feedback, and enforces a consistent interaction approach. Rather than focusing intently on the semantics of some interactive objective, usability reviews and tests are designed to determine the degree to which the system interface makes the users easy to interact with the system [30].

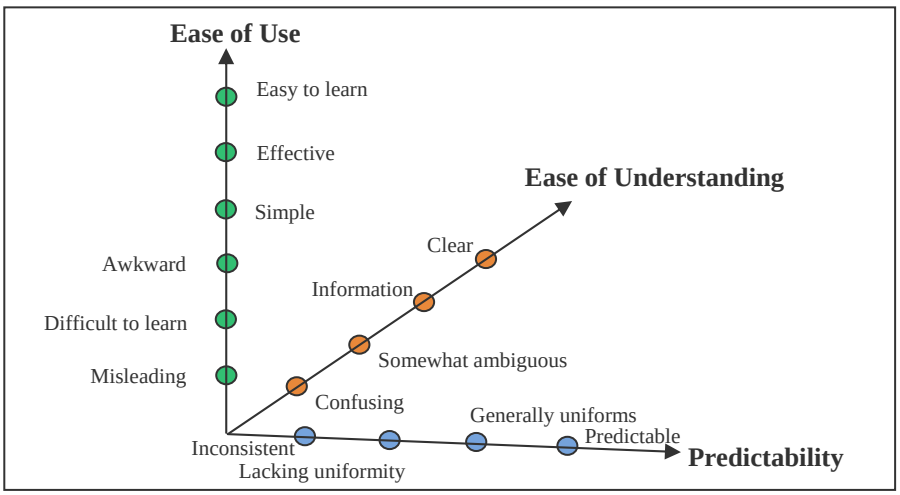


Figure 6.1 Qualitative Assessment of Usability [30]

Copyright by Mahidol University

Figure 6.1 depicts a possible set of assessment “grades” that can be selected by users. These grades are applied to each feature individually to complete the system as a whole.

In addition, measurable usability goal must be set for performing usability testing. Measurable usability goal is the definition of successful usability of the system for a specific set of users [31]. Typically, usability goals include:

- **Time:** the overall time the user will take to carry out a task in the system.
- **Accuracy:** the accuracy with which the user carries out the task in the system.
- **Overall Success:** the user can do a task in the system successful.
- **Satisfaction:** the user satisfies to use the system.

6.1.1 Evaluation Objective

Based on some qualitative assessment of usability testing and usability goals above, the researcher generates questionnaire to measure the degree of user satisfaction of the system in two aspects -Ease of Use and Ease of Understanding-. For the first aspect, users are measured to determine that the system is easy, effective and simple to use and for the second aspect, users are measured to determine that the system can give enough and clear information for user to use the system.

Table 6.1 System Evaluation Criteria

Level		**Meaning**
Qualitative Criteria	**Quantitative Criteria**	
Excellent	4.50 – 5.00	Quality Level is Excellent.
Good	3.50 - 4.49	Quality Level is Good.
Fairly Good	2.50 – 3.49	Quality Level is Fairly Good.
Fairly Poor	1.50 – 2.49	Quality Level is Fairly Poor.
Poor	1.00 – 1.49	Quality Level is Poor.

In additions, to evaluate the quality of the system using questionnaire, the criteria for system evaluation must be defined for rating the score. The researcher generates Qualitative Criteria and Quantitative Criteria. Both criteria have five levels.

Copyright by Mahidol University

After rating the result of experiment, the researcher calculates the mean average. To classify the quality level, the mean average is matched with a rate in Table 6.1

6.1.2 Evaluation Methodology

To evaluate a quality of the system, Black-Box testing is proceeded by the researcher. A set of questions have been employed to measure quality evaluation. Black-Box testing is employed to measure usability quality of the system.

Data Analysis Method

This section will describe the principles employed for data analysis and result interpretation. The Black-Box testing is employed to quantitatively measure the quality of the e-Tax Invoice System and which respect to the user satisfaction. The methodology used in the Black-Box testing is mean average.

Mean Average

Formula

$$\bar{x} = \frac{\sum x}{n}$$

Where

$\bar{x}$ = Mean Average

$\sum x$ = Summation of all

n = Participants

6.1.3 Testing Environment

The following is the explanation of the environment for usability testing.

6.1.3.1 Preparation and Method

This section will elucidate about the preparation and method to perform the usability testing as below:

Copyright by Mahidol University

The steps for the preparation of the testing are described as follows:

1) Install program to server and client machines.

2) Import simulated tax invoice data into the database of the prototype. These tax invoice data are used to generate report for RD Officer and tax invoice data for Customer in usability testing.

3) Test programs both in server side and client side.

After the researcher performs the preparation already, the selected RD Officers are invited to have a brief of system overview and test the system. Then, the researcher performs questionnaire to obtain feedback from RD Officers.

6.1.3.2 User Group

The main objective of usability testing is to measure user satisfaction. There are three user groups in e-Tax Invoice System, but only two groups: Customer and RD Officer are tested. This is because the researcher cannot contact business to test the system. Since the rights to access the system for these two user groups are different, the researcher develops two sets of questionnaire for each group.

1) Customer: are people who make a payment for products or services and must use tax invoice slip for financial transaction. Since users in this group are very large, the researcher defines user specification to scope this user group. In this experiment, the researcher focused on people who work at Bureau of Information Technology Building. A total of people are approximately 422 persons. The researcher employs random sampling on these RD Officers and 30 persons are selected for testing.

2) RD Officer: is RD officer who is responsible for auditing business's tax payment known as supervisor. In this experiment, the researcher focuses on supervisors who work at the head branch office. A total of supervisors who work at the head branch office are 461 persons. The researcher employs random sampling on these supervisors and 30 supervisors are selected for testing.

Copyright by Mahidol University

6.1.4 Questionnaire

There are two set of questionnaire to measure level of user satisfaction. Questions in these questionnaires for Customer and RD Officer are described in Table 6.2 and Table 6.3 respectively.

Table 6.2 Questionnaire for Customer

No.	Question
1.	E-Tax Invoice System is able to verify user authorization correctly and safely.
2.	E-Tax Invoice System is able to prioritize user right to access each Web Services correctly.
3.	The speed of performance of E-Tax Invoice System starting from sending request to receiving result is satisfied.
4.	Web Services provided by E-Tax Invoice System is always available.
5.	Which level of confidence in security technology in e-Tax Invoice System: Secure Socket Layer (SSL).
6.	From overall performance of the developed e-Tax Invoice System, which level of satisfactory is noticed by you
7.	Web Services provided by E-Tax Invoice System is easy to connect.
8.	E-Tax Invoice System is able to help you search tax invoice data.

Copyright by Mahidol University

Table 6.3 Questionnaire for RD Officer

No.	Question
1.	E-Tax Invoice System is able to verify user authorization correctly and safely.
2.	E-Tax Invoice System user interface is usable.
3.	E-Tax Invoice System is able to provide accurate reports with up-to-date data.
4.	E-Tax Invoice System is able to reduce your work process.
5.	E-Tax Invoice System is able to save your cost and time to work.
6.	From overall performance of E-Tax Invoice System, which level of satisfactory is noticed by you.
7.	E-Tax Invoice System is able to provide sales and purchase report which conforms to specific conditions.
8.	E-Tax Invoice System can generate and display total summary report with line graph which help you to compare data easier.

To facilitate RD Officer and Customer to fill in the questionnaire, the researcher implemented questionnaires in web application. Customers and RD Officer can fill in to answer the question online via browser. These web applications of Questionnaire for Customer and RD Officer are depicted in Figure 6.2 and 6.3 respectively.

Copyright by Mahidol University

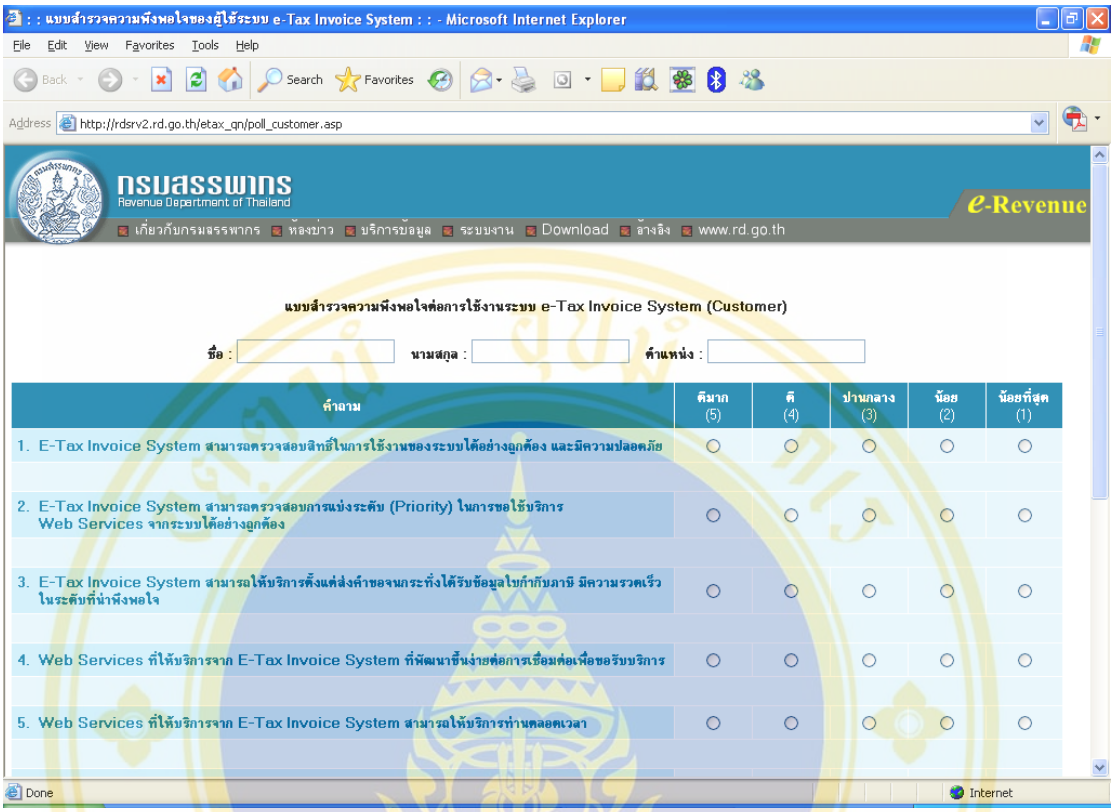


Figure 6.2 Questionnaire for Customer in Web Application

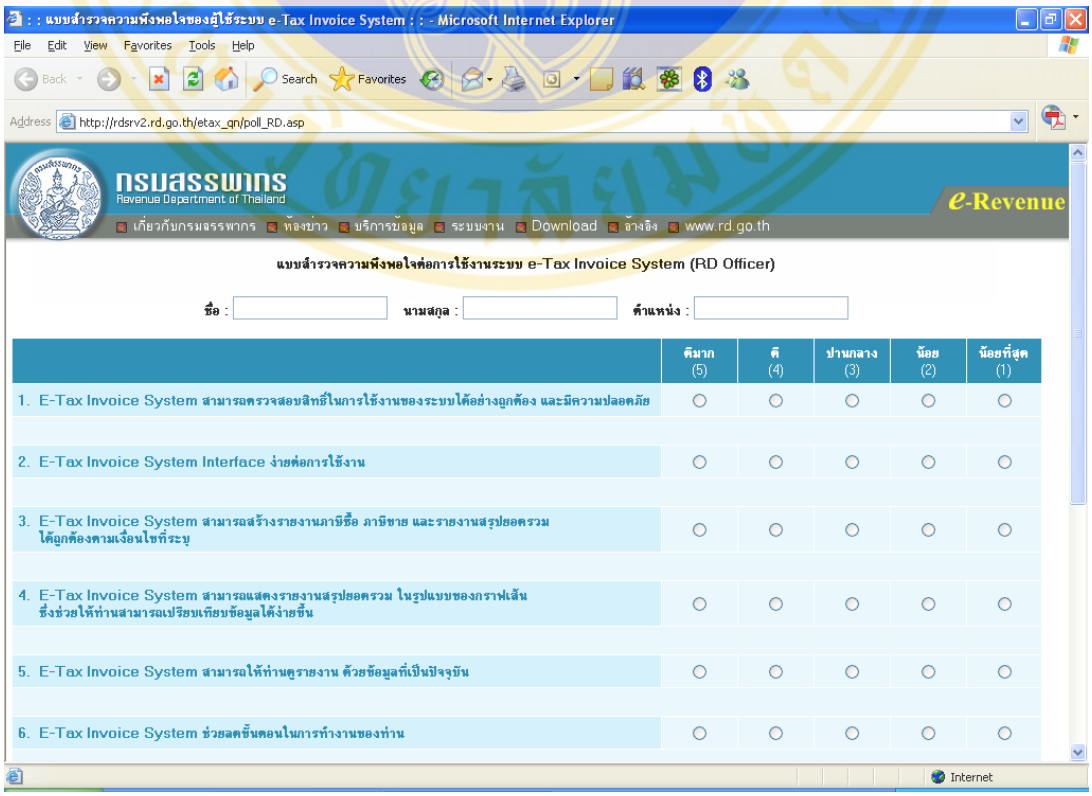


Figure 6.3 Questionnaire for RD Officer in Web Application

Copyright by Mahidol University

6.1.5 Usability Testing Result

From the score received from user, the research calculates mean average of each question described as follows:

Table 6.4 Customer Usability Test Result

No.	Excellent (%)	Good (%)	Fairly Good (%)	Average	Quality
1.	43	57	0	4.43	Good
2.	33	67	0	4.33	Good
3.	0	80	20	3.80	Good
4.	3	67	30	3.73	Good
5.	0	83	17	3.83	Good
6.	0	97	3	3.97	Good
7.	0	70	30	3.70	Good
8.	7	90	3	4.03	Good
Total Average				3.98	Good

Table 6.5 RD Officer Usability Test Result

No.	Excellent (%)	Good (%)	Fairly Good (%)	Average	Quality
1.	0	97	3	3.96	Good
2.	24	76	0	4.23	Good
3.	70	27	3	4.67	Excellent
4.	47	53	0	4.47	Good
5.	37	57	6	4.30	Good
6.	20	70	10	4.22	Good
7.	67	33	0	4.67	Excellent
8.	73	27	0	4.73	Excellent
Total Average				4.41	Good

In this experiment, the researcher defines that if user satisfies to use the system, user must give response at least in Good level. The usability testing results in Table 6.4 and 6.5 can be described respectively as follows:

Copyright by Mahidol University

1) Customer Usability Test Result

- 43% of Customers give a positive response in Excellent level of user satisfaction and 57% of Customers give a positive response in Good level for the user authorization of the system.
- 33% of Customers give a positive response in Excellent level of user satisfaction and 67% of Customers give a positive response in Good level of user satisfaction for the user prioritization for each Web Services of the system.
- 80% of Customers give a positive response in Good level of user satisfaction for the speed of performance of the system.
- 3% of Customers give a positive response in Excellent level of user satisfaction and 67% of Customers give a positive response in Good level of user satisfaction for the availability of WS of the system.
- 83% of Customers give a positive response in Good level of user satisfaction for the security of the system.
- 97% of Customers give a positive response in Good level of user satisfaction for the overall performance of the system.
- 70% of Customers give a positive response in Good level of user satisfaction for the ease to connect to WS provided by the system.
- 7% of Customers give a positive response in Excellent level of user satisfaction and 90% of Customers give a positive response in Good level of user satisfaction for the search system of the system.

2) RD Officer Usability Test Result

- 97% of RD Officers give a positive response in Good level of user satisfaction for the user authorization of the system.
- 24% of RD Officers give a positive response in Excellent level of user satisfaction and 76% of RD Officers give a positive response in Good level of user satisfaction for the user interface of the system.
- 70% of RD Officers give a positive response in Excellent level of user satisfaction and 30% of RD Officers give a positive response in Good level of user satisfaction for reports of the system.

Copyright by Mahidol University

- 47% of RD Officers give a positive response in Excellent level of user satisfaction and 53% of RD Officers give a positive response in Good level of user satisfaction about that the system can help them to reduce work process..
- 37% of RD Officers give a positive response in Excellent level of user satisfaction and 57% of RD Officers give a positive response in Good level of user satisfaction about that the system can save cost and time in work process.
- 20% of RD Officers give a positive response in Excellent level of user satisfaction and 70% of RD Officers give a positive response in Good level of user satisfaction for the overall performance of the system.
- 67% of RD officers give a positive response in Excellent level of user satisfaction and 33% of RD Officers give a positive response in Good level of user satisfaction for that the system is able to provide sales and purchase report which conforms to specific conditions.
- 73% of RD Officers give a positive response in Excellent level of user satisfaction and 27% of RD Officers give a positive response in Good level of user satisfaction for that the system can generate and display total summary report with line graph which help them to compare data easier.

From the above results, the researcher can conclude that most of users in both user groups satisfy to use the system. This indicates that the system can achieve user requirements. Therefore the system is suitable for deploy in actual situation.

6.2 Comparative Analysis

This section, will give the comparison between the existing system and the new system to show the benefit of the new system. This starting from, described processes in both systems. The researcher will discuss about the advantages provided by the new system.

Copyright by Mahidol University

Table 6.6 Comparison of activities in Existing System and new system

User Group	Existing System	New system
Customer	1. Make a payment. 2. Request tax invoice slip from business owner. 3. Wait for tax invoice slip. 4. Keep tax invoice slip for using in financial transaction.	1. Make a payment. 2. Optional to request tax invoice slip. 3. Search tax invoice data via browser. 4. View and print tax invoice slip online.
Business	1. Store tax invoice data to database. 2. Print two copies of slips. 3. Give master copy to customer. 4. Keep copy of tax invoice slip for RD Officer. 5. Generate report for RD Officer.	1. Send tax invoice data to RD immediately the transaction occurs. 2. Optional to print tax invoice slip for customer. 3. No need to print and keep tax invoice slip for RD Officer.
RD Officer	1. Go to business location. 2. Request for reports. 3. Wait for report. 4. Review report for helping to the decision in business's tax payment auditing.	1. Log in to e-Tax Invoice System. 2. View report online with up-to-date data.

In Table 6.6, the researcher compares processes between the existing system and the new system for three user groups: Customer, Business, and RD Officer.

For Customer, Customer can search tax invoice data generated since the past three years ago. This facilitates Customer to search tax invoice data via browser rather than find tax invoice from received tax invoice slips. From this, they can search tax invoice data faster and easier.

Copyright by Mahidol University

For Business, Business can send tax invoice data to RD immediately the transaction once occurs. As a result, it is not necessary for business to print and keep the copied version of tax invoice slip for RD officer because business have sent tax invoice data to RD already. This will help business to save cost of paper for printing tax invoice slip that have large amount of transactions per year. For example, the statistical discovered from the survey at Metropolitan Waterworks Authority (MWA) indicates that MWA has approximately 1,607,921 transactions per year. [4] This is the huge number of transactions. WMA must print two copied of tax invoice slip for each transaction so that Business must print tax invoice 3,215,842 slips. In contrast, with the adoption of the new system WMA print tax invoice slip only 1,607,921 slips. This the number of paper reduced from the existing system.

For the RD officer, presently when supervisors at RD want to view report for audit business's tax payment, they must go to business location and request it from Business. For this, RD officer must use at least three hours starting from requests report until receives report. From this, it consumes time and cost in this process for RD officer. In contrast, with the implementation of the new system, when RD officer wants to view report, they only login to the system and specify criteria: business/customer's name, TIN, date used for generating report. From this, RD officer spends only approximately five minutes for this operation. Moreover, Since Business had sent tax invoice data to RD online, data in the system is up-to-date. So that, report generated by the system are generated with up-to-date data. Therefore report can help RD officer to make a decision in Business's tax payment auditing more efficient. This is the great benefit for the RD officer.

From these explanations that describes the comparison between the processes in existing system and the new system. The researcher finds that the new system can solve the problem in the existing tax invoice system. Moreover, the system can facilitate all stakeholders in the system.

Copyright by Mahidol University

The result of two tests: usability testing and comparative analysis indicate that the new system can solve the problem in existing system and facilitate customer, business, and RD officer in work process. Moreover, the system can achieve user requirements of both Customer and RD officer. These imply that the new system is proper to deploy in actual situation to eradicate the problem in existing tax invoice system and to enhance work process in tax invoice system more efficient.

Copyright by Mahidol University

CHAPTER VII
CONCLUSION, DISCUSSION AND FUTURE WORKS

7.1 Conclusion

This research project aims to study and analyze the problems of the existing tax invoice system and develop e-Tax Invoice System for solving these problems. Web Services technologies: WSDL, SOAP, XML, and UDDI and security technologies: SSL and Hash Function are adopted to develop the new system. By using these complementary technologies, the e-Tax Invoice System developed can provide the great contribution to all stakeholders of the system which are: Customer, Business, and RD officer.

The e-Tax Invoice System is implemented to provide tax invoice data for Customer and Business. They can search tax invoice data online. Moreover, the new system provides service to collect tax invoice data immediately once sales transaction occurs from Business Hence, it is not necessary for Business to print tax invoice slips and keep them for RD's supervisor. From this, Business can save cost to purchase paper for printing tax invoice slip. Furthermore, the system provides accurate reports with up-to-date data for RD officer. Additionally, the system can generate line graphs corresponded to report which help RD officer to compare data easier. These reports and graphs can support RD officer to make a decision in tax payment auditing.

The experimental result denotes that e-Tax Invoice System can achieve user requirements and ready for deployment. The system can provide services for Customer and Business and accurate reports for RD officer with the good level of user satisfaction. Moreover the system advocates the management decision making process.

Copyright by Mahidol University

This system is expected to enhance efficiency of work process in RD that will increase efficiency and transparency in the tax administration. It used to solve the problem in existing tax invoice system for example control and anti-fraud in tax payment and tax return. Furthermore it can provide faster and easier services for Business and Customer.

7.2 Discussion

According to the implementation and evaluation of the e-Tax Invoice System, the new system is designed and implemented with a certain level of security by using SSL and Hash Function to encapsulate tax invoice data from unauthorized users. However there are two vulnerabilities of the system. The first is that business may edits/changes tax invoice data before sending to the system. Business may use this vulnerability to gain benefit dishonestly. To alleviate this vulnerability, RD should have the system used to cross check between input and output tax of each business to detect any fraud tax invoice data. The second is that an intruder tries to access the database. To harden it, database security should be a serious concern. RD must apply security technologies that complement and extend traditional database solutions to ensure that data remain in the hands of those authorized access and keep accessed log for database auditing.

7.3 Future Works

The following issues are the further extension of the e-Tax Invoice System:

1. Increase the quality of report by adding component to link with e-Supervision system for requesting actual tax payment data. The system must be modified to generate the new report from data received from e-Supervision system. The system will show that comparison of two types of data: tax invoice data and actual tax payment data. The new report can show the comparison of two data types in line

Copyright by Mahidol University

graph. This will make the great benefit for the RD officer to use the report for making the decision in tax payment auditing.

2. Forecast the tax collection by analyzing tax invoice data collected in current year to forecast tax collection of VAT in next year.

3. Cross check between Input Tax and Output Tax of each business using accumulated tax invoice data. This will help supervisor to make the decision in tax payment auditing more efficient.

4. Enhance the performance of the system by performing performance testing after the system is deployed in actual situations. If the performance of the system is in unsatisfied level, the system must be analyzed again to improve performance of the system.

Copyright by Mahidol University

REFERENCES

1. Bureau of Information Technology, “รายชื่อผู้เสียภาษีที่อยู่ภายใต้ระบบการบริหารการจัดเก็บภาษีอากรของ ภญ.”, Available from: http://rdsrv.rd.go.th/13719.0.html, Created Date: 30 March 2004, Last Updated Date: 17 March 2006, Accessed Date: 27 June 2006.
2. Bureau of Large Business Tax Administration, “สถิติจำนวนรายผู้ประกอบการจดทะเบียนภาษีมูลค่าเพิ่ม”, Available from: http://lto.rd.go.th/html/tik/publish/27.0.html, Created Date: 19 December 2003, Last Updated Date: 28 June 2006, Accessed Date: 28 June 2006
3. Available from: http://www.7eleven.co.th/Franchise/franchise.html, Last Update: Not Available, Create Date: Not Available, Accessed Date: 2 June 2006.
4. Available from: http://www.mwa.co.th/download/, Last Update: Not Available, Create Date: Not Available, Accessed Date: 25 June 2006.
5. The Revenue Department, “ประมวลรัษฎากร มาตรา 86/4”, Available From: http://www.rd.go.th/publish/5208.0.html#mata86_4, Created Date: 17 July 2002, Last Updated Date: 14 October 2005, Accessed Date: 22 June 2006.
6. The Revenue Department, “ประมวลรัษฎากร มาตรา 86/4”, Available From: http://www.rd.go.th/publish/5208.0.html#mata86_6, Created Date: 17 July 2002, Last Updated Date: 14 October 2005, Accessed Date: 22 June 2006.
7. Daniel Austin, Abbie Barbir, Christopher Ferris, Sharad Garg, “Web Services Architecture Requirements”, Available From: http://www.w3.org/TR/wsa-reqs, Created Date: 11 February 2004, Last Updated Date: Not Available, Accessed Date: 5 February 2006
8. Kreger, H., “Web Services Conceptual Architecture (WSCA 1.0)”, IBM Software Group, Available From: http://www.ibm.com/software/solutions/webservices/resources.htm, 2001.

Copyright by Mahidol University

9. Chuang-Cheng Chiu and Chieh-Yuan Tsai, “A Web Services-Based Collaborative Scheme for Credit Card Fraud Detection”, Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE’04), pp.177-181, March 2004.
10. Min-Hsiung Hung, Fan-Tien Cheng, and Sze-Chien Yeh., “Development of a Web-Services-Based e-Diagnostics Framework for Semiconductor Manufacturing Industry”, IEEE Transactions on Semiconductor Manufacturing Industry, Vol. 18 NO.1, pp. 122-135, February 2005.
11. IBM Services Architecture Team , “Web Services architecture overview. The next stage of evolution for e-business”, Available From: http://www.128.ibm.com/developerworks/webservices/library/wovr/#h3, Created Date: 3 September 2000, Last Updated Date: Not Available, Accessed Date: 5 February 2006.
12. Gustavo Alonso,Fabio Casati, Harumi Kuno, and Vijay machiraju, *Web Services Concepts, Architectures and Applications*, Springer-Verlag Berlin Heidelberg, Printed in Germany 2004.
13. Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, and Henrik Frystyk Nielsen , “SOAP Version 1.2 Part 1: Messaging Framework”, Available From: http://www.w3.org/TR/soap12, Created Date: 24 June 2003, Last Updated Date: 24 June 2003, Accessed Date: 4 February 2006.
14. http://www.w3schools.com/wsdl/wsdl_intro.asp, Last Update: Not Available, Create Date: Not Available, Accessed Date: 27 January 2007.
15. http://www.w3schools.com/schema/schema_simple.asp, Last Update: Not Available, Create Date: Not Available, Accessed Date: 1 October 2007.
16. http://looselycoupled.com/glossary/XSLT, Last Update: Not Available, Create Date: Not Available, Accessed Date: 25 January 2007.
17. Microsoft, *1913A: Exchangeing and Transforming Data Using AML and XSLT*, Microsoft Coporation, Edition 02/2001.
18. “Universal Description, Discovery and Integration”, Available From: http://www.uddi.org, Created Date: Not Available, Last Updated Date: Not Available, Accessed Date: 2 June 2006.

Copyright by Mahidol University

19. Tomasz Onyszko, “Secure Socket Layer”, Available From: http://www.windowsecurity.com, Created Date: 16 July 2002, Last Updated Date: 22 July 2004. Accessed Date: 5 July 2006.
20. http://www.herongyang.com/crypto/message_digest_sha1.html, Last Update: Not Available, Create Date: Not Available, Accessed Date: 27 January 2007.
21. Available From: http://www.cra-arc.gc.ca/menu-e.html, Last Update: Not Available, Create Date: Not Available, Accessed Date: 17 July 2006.
22. Available From: http://www.inlandrevenue.gov.uk, Last Update: Not Available, Create Date: Not Available, Accessed Date: 17 July 2006.
23. Available From: http://www.iras.gov.sg/ESVPortal/our_eservices/index.asp, Last Update: Not Available, Create Date: Not Available, Accessed Date: 17 July 2006.
24. Internal Revenue Service of the United State., “IRS e-file for Large Taxpayers Filing Their Own Corporate Income Tax”, February 2006.
25. http://www.xbrl.org/Home/, Last Update: Not Available, Create Date: Not Available, Accessed Date: 27 September 2007.
26. http://www.rd.go.th/webservice/, Last Update: Not Available, Create Date: Not Available, Accessed Date: 27 September 2007.
27. http://www.visual-paradigm.com/product/bpva//, Last Update: Not Available, Create Date: Not Available, Accessed Date: 27 September 2007.
28. http://www.sequencediagrameditor.com/uml/sequence-diagram.htm, Last Update: Not Available, Create Date: Not Available, Accessed Date: 27 September 2007.
29. http://www.altova.com/download.html, Last Update: Not Available, Create Date: Not Available, Accessed Date: 12 February 2007.
30. Roger S. Pressman, Software Engineering: A Practitioner’s Approach, Sixth Edition, McGraw-Hill, International Edition 2005, Printed in Singapore.
31. http://www.usability.gov/analyze/goals.html, Last Update: Not Available, Create Date: Not Available, Accessed Date: 12 October 2007.

Copyright by Mahidol University

BIOGRAPHY

NAME	Miss Sukanya Suwisuthikasem
DATE	January 18, 1979
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Chiang Mai University, 1995-1998: Bachelor of Science (Computer Science) Mahidol University, 2004-2007: Master of Science (Computer Science)
POSITION & OFFICE	2001 – Present, Computer programmer, The Revenue Department, Thailand
ADDRESS	3/5 Mui 9 Soi Changransanitwong 35, Changransanitwong Road, Bangprom, Talingchung, Bangkok 10170, Thailand
E-MAIL	Pukies.su@gmail.com

Copyright by Mahidol University