WEB SERVICES FOR NOTIFICATION BETWEEN MOTOR VEHICLE REPAIR SHOPS AND THE VIRIYAH INSURANCE COMPANY

SAKAOWRATH CHATJARORNCHAP

MASTER OF SCIENCE
ADVANCED INFORMATION TECHNOLOGY

MAE FAH LUANG UNIVERSITY
2007

WEB SERVICES FOR NOTIFICATION BETWEEN MOTOR VEHICLE REPAIR SHOPS AND THE VIRIYAH INSURANCE COMPANY

SAKAOWRATH CHAJAROENCHAP

**A INDEPENDENT STUDY SUBMITTED TO
MAE FAH LUANG UNIVERSITY IN PARITIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN ADVANCED INFORMATION TECHNOLOGY**

MAE FAH LUANG UNIVERSITY
2007

WEB SERVICES FOR NOTIFICATION BETWEEN MOTOR VEHICLE REPAIR SHOPS AND THE VIRIYAH INSURANCE COMPANY

SAKAOWRATH CHAJAROENCHAP

A INDEPENDENT STUDY HAS BEEN APPROVED
TO BE A PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF THE MASTER OF SCIENCE PROGRAMME
IN ADVANCED INFORMATION TECHNOLOGY
2007

EXAMINING COMMITTEE

……………………………………………………………..CHAIRPERSON
(Asst. Prof. Gp. Capt. Dr.Sanlayut Sawangwan)

…………………Yooyativong. T…………………MEMBER
(Gp. Capt. Dr.Thongchai Yooyativong)

……………………………………………………………..MEMBER
(Lecturer Piyasak Jeatrakul)

……………………………………………………………..MEMBER
(Lecturer Vittayasak Rujivarakul)

ACKNOWLEDGEMENT

I wish to acknowledge the assistance of the following 5 professors. Whose constructive criticism and suggestions helped invaluably in Chaptering the content of the thesis. The successful completing could not have come about without the kindness and strong patience of my former chair advisor associate professor Dr.Thongchai Yooyativong, whom guided me in the right direction. I am in debt to my advisor Mr.Piyasak Jeatrakul who I always look up to as a great mentor and guidance, and supporting the study in so many ways and I would also like to thank Mr.Vittayasak Rujivorakul who thoughtful suggestions me to do this study. Without his insights and supervision, I would not have obtained such valid results.

Special thanks are extended to Dr.Sanlayut Sawangwan, Ms.Paola Di Maio and my associate committee member, for their invaluable feedback, thoughtful suggestions and helpful advice and constructive comments that contributes to the success of this study their continuous guidance, understanding and patience are greatly appreciated.

In addition, I would like to extend my thanks to Mrs.Supatra Thongkhaow CIO and Ms.Aiyarest Saelee Manager in the viriyah Insurance Co, LTD., who is the also contributed to the field of information systems through this project as well as through their participation in the field of insurance functions, hardware and software system. Much credit should go to several individuals who played significant role in this study. Thus, special thanks go to at The Viriyah Insurance Company. Their ideas were invaluable contributions to the successful completion.

Most of all, I wish to thank my family for their patience and understanding during the long days and late nights spent working on this work.

My heartfelt thanks also go to all of my classmate for their experience and support during my study.

Finally, I wish to express my sincere thanks to Mae Fah Luang University and to all my lecturers in The School of Information Technology who provided me with a high standard of education.

Sakaowrath Chatjaroenchap

Independent Study Title	Web Services for Notification Between Motor Vehicle Repair Shops and The Viriyah Insurance Company	
Author	Miss Sakaowrath Chatjaroenchap	
Degree	Master of Science (Advanced Information Technology)	
Supervisory Committee	Gp. Capt. Dr.Thongchai Yooyativong	Chairperson
	Lecturer Yuthana Tra – Ngarn	Member
	Lecturer Vittayasak Rujivarakul	Member

ABSTRACT

The objectives of this independent study is to improve the existing services of notification claim process between motor vehicle repair shops and the Viriyah insurance company by developing the software solution automatically notify claim numbers, resulting to the reduction of operating costs and times.

Web services technology plays a key role in this solution to provide automatically generated or Ad-Hoc notifies claim number. The technology has a strong and well defined role in B2B solutions. It is expected to help reshape even entire industries. More and more companies are realizing the need and rationale to adopt open standards and foster interconnection of value chain participants.

In this solution, web services technology can help and enable the efficiency communication to achieve notification claim process by automating claims process and integrating the motor vehicle repair shop's their existing systems via web-based application. This allows authorized users to use message level security instead of transport level security, real-time access to centralized information and claim status. With one dynamic solution, claim process management enhances the productivity, process claims faster, reduces cycle times and costs, so that customer satisfaction is improved.

Keywords : Web services / Web services and insurance business

CONTENTS

CONTENTS (Cont.)

LIST OF TABLES

LIST OF TABLES

LIST OF ABBREVIATIONS

SOA

Service-Oriented Architecture (SOA) is a software architecture within which all functions are defined as independent services with well-defined invokable interfaces, which can be call in defined sequences to form business processes

SOA, Software architecture is the fundamental organization of a system embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution

SOA follows the ANSI/IEEE Std 1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems [#]. An architectural description conforming to the ANSI/IEEE Std 1471-2000 recommended practice is described by a clause that includes the following six (6) elements:

1. Architectural description identification, version, and overview information
2. Identification of the system stakeholders and their concerns judged to be relevant to the architecture
3. Specifications of each viewpoint that has been selected to organize the representation of the architecture and the rationale for those selections
4. One or more architectural views
5. A record of all known inconsistencies among the architectural description's required constituents
6. A rationale for selection of the architecture

SOAP

SOAP is a standard for exchanging XML-based messages over a computer network, normally using HTTP. SOAP forms the foundation layer of the web services stack, providing a basic messaging framework that more abstract layers can build on.

Web services

A Web services is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web services in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

XML

The 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.

W3C

The World Wide Web Consortium (W3C) is an international consortium where member organizations, a full-time staff and the public work together to develop standards for the World Wide Web. W3C's stated mission is "To lead the World Wide Web to its full potential by developing protocols and guidelines that ensure long-term growth for the Web."[1] W3C also engages in education and outreach, develops software, and serves as an open forum for discussion about the Web. The Consortium is headed by Tim Berners-Lee, the primary author of the URL (Uniform Resource Locator), HTTP (HyperText Transfer Protocol) and HTML (HyperText Markup Language) specifications, the principal technologies that form the basis of the World Wide Web.

WSDL

WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, the only bindings described in this document describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME.

UDDI

UDDI is an acronym for Universal Description, Discovery, and Integration – A platform-independent, XML-based registry for businesses worldwide to list themselves on the Internet. UDDI is an open industry initiative (sponsored by OASIS) enabling businesses to discover each other and define how they interact over the Internet.

Binding

Associating an interface, a valid data format, and a protocol to ensure smooth message transmission.

Client

In any client/server system, the software requesting services or information from the server.

Message

The basic unit of communication containing the data to be transmitted between the client and server.

Service

A facility or application offered by an organization.

XML Namespaces

It is possible for documents to have elements with the same name. XML namespaces are used to differentiate between them to avoid element name conflict.

ebXML

Electronic business XML, an architecture and set of specifications designed to automate business process interaction among trading partners.

Additional technologies

Going beyond the core web services standards to meet requirements for security, reliable messaging, transaction processing, and business process flow so that more complex and critical business application can use them

CHAPTER 1

INTRODUCTION

1.1 Background

The Internet has already evolved well beyond basic connectivity and presentation services to offer rich interactive pages, which are commonly enhanced by client-side or server-side script and other components. To further extend the performance and versatility of the Internet, Web services are being developed that provide new capabilities and deliver a new level of performance.

In essence, they provide a piece of functionality through the Internet. One of the more important and interesting aspects of Web services is the standardization and broad acceptance of protocols such as XML (eXtensible Markup Language, for formatting messages), SOAP (Simple Object Access Protocol, for calling Web services), UDDI (Universal Description, Discovery and Integration, for finding Web services) and WSDL (Web services Description Language, for the description of Web services).

An architecture component that provide the Service Provider makes the service available with its Service Contract and advertises it on the Service Directory, The Service Consumer find the compatible Service and its Service Contract using the Service Directory, The Service Consumer and the Service Provider interact.

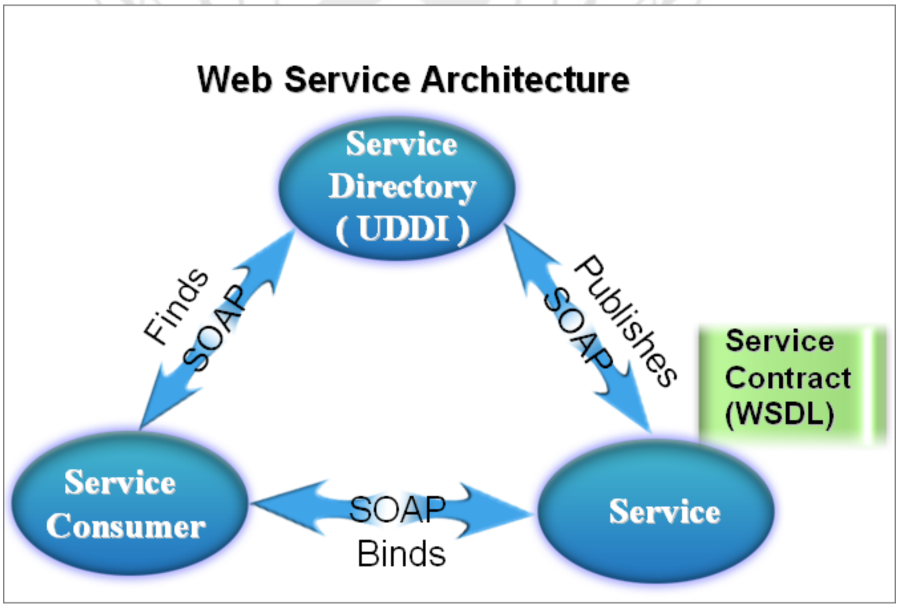


Figure1.1 Web services Architecture

1.1.1 What is a Web service?

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. (W3C Working Group, 2004)

Today's Web services implementations are typically simple and often similar to a client-server model. However, platform-neutral interchange is supported, which allows a diverse range of client implementations to interact with new or legacy code as server functions. Much has been written about the technologies that make such applications straightforward to implement. It is now time to look at the bigger picture of what we can do with them. The author addresses the question of how to move forward from simple models to those that represent real-world business models of arbitrary complexity.

Service-Oriented Architecture (SOA) as key to interoperability and flexibility requirements for its vision of on demand business. SOA supports end-to-end integration across the enterprise and among business partners. This provides a flexible business process model that allows customers to respond quickly to new customer requirements, new business opportunities, and competitive threats.

1.1.2 What is SOA?

SOA presents the big picture of what you can do with Web services. Web services specifications define the details needed to implement services and interact with them. However, SOA is an approach to build distributed systems that deliver application functionality as services to end-user applications or to build other services. SOA can be based on Web services, but it may use other technologies instead. In using SOA to design distributed applications, you can expand the use of Web services from simple client-server models to systems of arbitrary complexity.

Web services are based on invocation using SOAP messages which are described using WSDL over a standard protocol such as HTTP. Use of Web services is a best practice when communicating with external business partners.
(Mark Colan. 2004).

A Web services look like a set of functions sends to the Internet through a Web server. A developer of a Web site or an application has the capability of invoking these functions and making use of the Web services. Web services are gaining popularity due to the many benefits they offer:

1.1.3 Code Reuse

Building applications is simplified using off-the-shelf Web services components.

1. Platform Independence

Because access to Web services uses standard Internet technologies, the consumers of a Web services can utilize any platform.

2. Language Independence

Any programming language can act as a Web services consumer without knowledge of the implementation details of the Web services.

3. Distributed Computing

Web services components can be used to easily build distributed applications that span firewalls.

The power of Web services lies in their ability to provide added functionality to standard Web or application design tools. Web services augment existing programming toolsets. As more programmers adopt Web services as a programming tool, these services become increasingly prevalent in Web and application design.

1.1.4 Web services Publishing with UDDI

A Web services provider exposes selected application functionality for others to share. The provider can use the UDDI publishing methods to enable consumers to find this functionality. Providers publish with a UDDI server in approximately the following development

Steps:

1. Create the core functionality of the service.

2. Write a SOAP wrapper for the core functionality.

3. Create a WSDL service description. It is often possible to do this using an automated service.

4. Install the Web services on a new or existing Web server.

5. Publish the specifications of the Web services in a UDDI server on an intranet, extranet, or on the Internet.

As part of an overall Web services solution, the company implements a service broker using a UDDI registry as a central element. By deploying it within the boundaries of a “DMZ” trusted environment, the company can both isolate interactions from its internal network, as well as limit the exposure of the registry to the outside world. In addition, by establishing subscription-based relationships with partners’ registries in the trading network, the company can ensure that information is fully, but safely, distributed among trading partners. The registry also implements the XML Digital Signatures support in Version 3 of the UDDI specification to ensure the integrity and authenticity of exchanged data. (Organization for the Advancement of Structured Information Standards, 2004)

1.1.5 Insurance Value Chain Integrated Innovation in Insurance with Web services.

Web services give a great deal of agility and serves advanced standards governing Web services security, reliability, and transactions arose. Relation of insurance business and several organizations across the industry responded to this need by authoring a set of specifications referred to collectively as the Web services architecture. The goal of these specifications is to provide a blueprint for advanced functionality while retaining the simplicity of basic Web services.

1.1.6 Benefits Realizations

1. Improve operational efficiencies
   1) Streamline operation process with straight through processing
   2) Improve context sensitive data entry
   3) Instant delivery of claim service with robust and comprehensive rule engine
2. Improve client servicing and satisfaction
   1) Efficient and consistent response to enquiries
   2) Flexible modes of service
   3) Reduce the service turnaround time
   4) Service tracking and automation

Web services technology operate at the most efficient levels possible increasing levels of insurance service reducing operation cost and time.

1.1.7 Overview

The Viriyah Insurance was established on 1947 which provided has run insurance business in Thailand with long established with 59 years and reputable stable institution has been providing premier insurance services for

1. Products
   1) Motor Insurance
   2) Fire Insurance
   3) Marine Cargo Insurance
   4) Miscellaneous Insurance
   5) Property Insurance
   6) Engineering Insurance
   7) Liability Insurance
   8) Professional Indemnity Insurance
   9) Employers Liability
   10) Personal Accident (PA) and Health Insuranc
2. Services
   1) Underwriting Branches, Claims Service Centers and Universal Service Branches
   2) Networking for Medical Service facilities (hospitals) for Health Insurance clients

The Company had 3,421 employees, Claims service centers and Universal Service Branches: With 30 branches and 88 claims service centers overall Thailand.

Ordinarily the business has several organizations to contact in insurance business:

1. Motor vehicle repair shops
2. Agent
3. Broker
4. Other insurance
5. Hospital
6. Supplier parts
7. The viriyah car body
8. Finance

9. Dealer
10. Bank
11. The department of insurance
12. ACORD (Association for Cooperative Operations Research and Development)

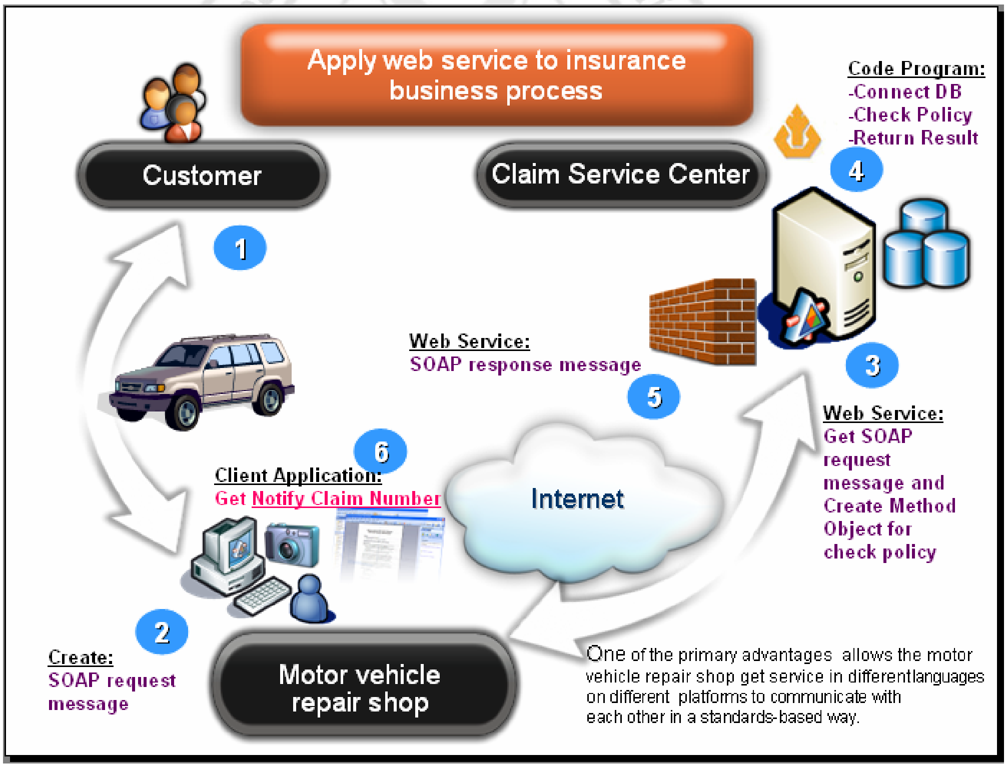
Different organizations collect, store, and process information differently, making data exchange difficult. One of the most time-consuming challenges for developers has been to exchange data between such systems over the Internet.

For this reason computer systems and databases contain data in incompatible formats. One of the most time-consuming challenges for developers has been to exchange data between such systems over the Internet.

1.2 Objectives

This project aims to improve process and automatically generated or Ad-Hoc notifies claim number service process using web services.
Web services help to facilitate the integration of data across different systems, businesses are adopting XML as the standard format for information exchange over HTTP.

For Business to Business interactions, applications need to talk to each other on the Web interoperability is needed which claim service to response with motor vehicle repair shops immediately.



One of the primary advantages allows the motor vehicle repair shop get service in differentlanguages on different platforms to communicate with each other in a standards-based way.

Figure 1.2 Using Web services apply to insurance business process

1.3 Scope

1.3.1 Use web services to fulfill a claim process between motor vehicle repair shops and the company

1.3.2 Build web application support connection for exchange data between motor vehicle repair shops and the company

The scenario implemented in this demonstration is that of a simple claims processing network.

Motor vehicle repair shops make notification available to their customers, and claim service center generate notify number to motor vehicle repair shops.

Motor vehicle repair shops in turn checks the current status and track claim number.

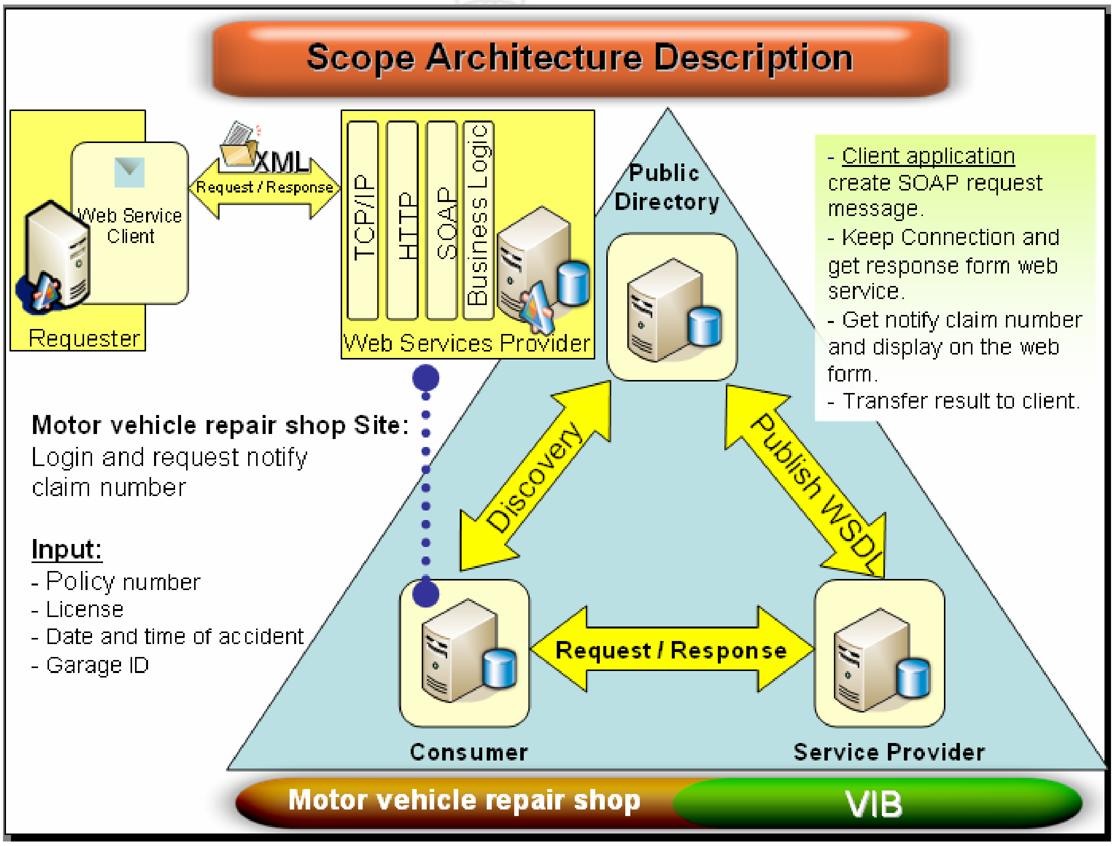

Scope Architecture Description
Web Service Client
Requester
XML
Request / Response
TCP/IP
HTTP
SOAP
Business Logic
Web Services Provider
Public Directory
- Client application create SOAP request message.
- Keep Connection and get response form web service.
- Get notify claim number and display on the web form.
- Transfer result to client.
Motor vehicle repair shop Site:
Login and request notify claim number
Input:
- Policy number
- License
- Date and time of accident
- Garage ID
Discovery
Publish WSDL
Request / Response
Consumer
Service Provider
Motor vehicle repair shop
VIB

Figure 1.3 Web services for notification between motor vehicle repair shops and The viriyah Insurance Company Architecture Description

1.4 Expected benefits

1.4.1 Real-Time Processing – The scenario work to minimize the claim service time between motor vehicle repair shops and claims service centers.

1.4.2 Improve claim process and automatically generated or ad-hoc notify claim service process

1.4.3 Convenience - Make business- to-business easy and make it relatively easy for motor vehicle repair shops to switch from one insurance company to another.

1.4.4 Reduce cost and time in communication such as cost of telephone, fax and paper for proceed.

1.4.5 Improve business agility

1.4.6 Faster software development and Web services delivery

1. Number of people to get benefits from project

1) Motor vehicle repair shops

I	Bangkok Area	638	Offices
II	Other province	1062	Offices

2) The Viriyah Insurance Company – Claim Service Center

I	Officer-Over all the country	686	Persons

In an insurance business where company merge partners shift, staying ahead of the competition means being agile and using change management as a strategic advantage that flexible, scalable IT systems and essential to business agility.

CHAPTER 2

FEASIBILITY STUDY

2.1 Introduction

Today, Insurance business partners and difference application they need technologies to integrate information for service the customer in the real-time what is the way to meet between them. Web services technologies have an idea to help and improve communicate among the whole they can access the internet via many different way of connection as device and platform.

2.2 Problem statement

Nowadays, Lack of solution for response the claim is processing in different organizations as motor vehicle repair shops collect, store, and process information differently, making data exchange difficult.

For motor vehicle repair shops, the seemingly endless round of phone messages and paperwork that follow an accident can make the job of resolving an insurance claim feel as damaging as the accident itself.

For insurance carriers claims processing which involves handling check the policy and more is no less time-consuming.

Waste of time response in notify claim process--Communication Cost as follows:

1. Phone message
2. Paperwork

2.3 Related research and projects

Web services Technology help insurance business meet key business objectives because the vision of business around everything connected is directly in line with where business want to take in future, and Web services deployed the technology deployed will help success.

With Web services, help to develop systems once and design them to be deployable and adaptable all over the world. This approach enables us to be more responsive to company growing business.

To achieve a dynamic, flexible IT architecture, chosen to use web services, discrete units of software that use industry standard protocols to operate across platforms and programming languages.

2.3.1 Third-Party Pricing Configuration.

The pricing configuration Web services enables partners to access real-time information directly from the company online.

2.3.2 Claim Notify Status

The web services create real time of each claim notification as it is fulfilled and can view their status information via internet or be notified by e-mail at agreed-upon intervals.

Rather than go to third-party sites with tracking number to determine where there system is, customers can now obtain all the information they need directly from the company

2.3.3 Interoperability

Like many organizations runs its business with a range of applications that are built with difference programming languages and run on different platforms. Interoperability offered by Web services developed XML connections into each application, which enabled the applications to work together.

2.4 Requirement specifications for the new system

2.4.1 Equipment and Software Required

Table 2.1 An Applications require the environment shown in the following table

Operating System	▪ Microsoft Windows® 2000 Server (SP2 or later). ▪ Microsoft Windows XP.
Applications	▪ Microsoft SQL Server Notification Services. ▪ Microsoft Visual Studio .NET (with Microsoft Visual C#™ components installed). ▪ SQL Server 2005 (Standard, Developer, or Enterprise Edition).
Running Services	▪ Simple Mail Transfer Protocol (SMTP)
Accounts	▪ Valid Windows accounts and SQL Server login accounts for any users who will be working. ▪ One or more accounts under which the instances of Notification Services will run and that you can use to create instances of Notification Services in SQL Server Management Studio.
Services	▪ Microsoft Windows Internet Information Services (IIS) with the Simple Mail Transfer Protocol (SMTP) service.
Connection	▪ Modem Connection

(Microsoft Corporation, 2005)

2.4.2 Comparing Databases

Focus to compare the features and capabilities of different systems to help how it performs and delivers flexible and reliable functionality in comparison with other database offerings to meet your business intelligence (BI), availability, productivity, and security needs.

2.4.3 Comparing SQL Server 2005 and Oracle

The findings of this study demonstrate the real efficiency benefits that are available to developers in their day-to-day activities using SQL Server 2005, relative to Oracle Database 10g. The comparative 31.5 hours required by the SQL Server 2005 developer to perform the same tasks that took 126 hours for the Oracle Database 10g developer are emblematic of the advanced value and functionality afforded by SQL Server 2005.

The maturity of the SQL Server 2005 platform, as well as excellent support for ad-hoc and custom reporting, data synchronization, and replication, were instrumental to the efficiency gains realized by the SQL Server 2005 developer in this study. Increased programmability, efficiency, and value are available for business-critical implementation by enterprise developers today. (3 Leaf LLC, 2006)

2.4.4 Comparing SQL Server 2005 to IBM

SQL Server 2005 continues to surpass IBM DB2 in features, functionality, and innovation at a much lower cost. Organizations that relay on the DB2 system will find that, feature to feature, SQL Server 2005 outperforms DB2, making SQL Server 2005 the right choice for your organization. (Microsoft Corporation, 2005)

2.4.5 Comparing SQL Server 2005 to MySQL

Both products can be used to build stable and efficient system and the stability and effectiveness of your applications and databases depend rather from the experience of the database developers and database administrator than from the database's provider. But SQL Server 2000 has some advantages in comparison with MySQL version 4.1 and vice versa.

The SQL Server 2005 advantages:

1. SQL Server 2005 holds the top TPC-C performance and price performance results.
2. SQL Server 2005 is generally accepted as easier to install, use and manage.
3. Transact-SQL is more powerful language than MySQL dialect.

The MySQL advantages:

1. MySQL supports all known platforms, not only the Windows-based platforms.
2. MySQL requires less hardware resources.
3. You can use MySQL version without any payment under the terms of the GNU-General Public License.

Table 2.2 Platform comparison T-SQL versus MySQL Dialect

Feature	T-SQL	MySQL Dialect
Views	General Views, Indexed Views, Distributed Partitioned Views	Not Supported
Triggers	AFTER triggers, INSTEAD OF triggers	Not Supported
Stored Procedures	T-SQL statements	Not Supported
User-defined functions	Scalar functions, Inline table-valued functions, Multistatement table-valued functions	C, C++ external libraries
Foreign Keys	Supported	Supported for only InnoDB tables
Cursors	Supported	Not Supported
Arrays	Not Supported	Supported

(SSWUG.org, 2005)

2.4.6 Concluding Thoughts

From a database developer's perspective, choosing between a MySQL and SQL Server DBMS is a matter of the scale of the database application. For enterprise-level applications, SQL Server wins hands down. It has advanced set of SQL features, superior replication, clustering, security and management tools.

For lower-tier database applications, MySQL can offer the core functionality you require at a very low cost. Some might argue that the latest offering from MySQL has made the open source database system enterprise "worthy", but this remains to be seen. The advanced functionalities implemented are yet to stabilize and be rationalized across the database engine. What's more, Microsoft has upped the ante with even more advanced features of its own. It's up to MySQL to rise up to the challenge, but at this point in time MySQL is nowhere near the competitive enterprise field of the more established SQL Server 2005.

2.4.7 Budget

Microsoft® Enterprise Agreement makes it easy to acquire Microsoft software licenses for customers who are interested in standardizing on some or the entire Microsoft enterprise platform of products.

2.4.8 Enterprise Platform

Enterprise Agreement offers a flexible way to standardize your company on the following enterprise platform of Microsoft products:

1. Microsoft Office® Professional
2. Microsoft Windows® Professional Desktop Upgrade

3. Microsoft Core Client Access License (CAL)
There are four price levels for each of the enterprise products

Table 2.3 Four price levels for the enterprise products

Level	PC's
A	250-2,399
B	2,400-5,999
C	6,000-14,999
D	15,000+

In this project without cost of hardware and software because exist in the company with Microsoft® Enterprise Agreement License in level B.

In phase of training set cost of material, fare and other estimate 20,000 baths.

2.5 Implementation Techniques

SQL Server 2005 - Database Engine

Microsoft SQL Server 2005 provides native XML Web services through the SQL Server 2005 Database Engine by using the following open standards:

1. Hypertext Transfer Protocol (HTTP)

2. As the core protocol behind the World Wide Web, HTTP provides a platform-neutral Web-based exchange of data.

3. SOAP defines how to use XML and HTTP to access services, objects, and servers regardless of the operating system.

4. Web services Definition Language (WSDL) is an XML document format that can be used to describe Web-based services.

When use Native XML Web services in SQL Server 2005, can send SOAP messaging requests to an instance of SQL Server 2005 over HTTP to run the following:

1. Transact-SQL batch statements, with or without parameters.

2. Stored procedures extended stored procedures, and scalar-valued user-defined functions.

Setting the Server to Listen for Native XML Web services Requests, server computer that is running SQL Server 2005 to help it listen and service HTTP-based requests in a secure manner.

1. Service Broker, a new technology for building database-intensive distributed applications that are secure, reliable, and scalable.

2. Service Broker provides:

1) Facilities for storing message queues in SQL Server databases.

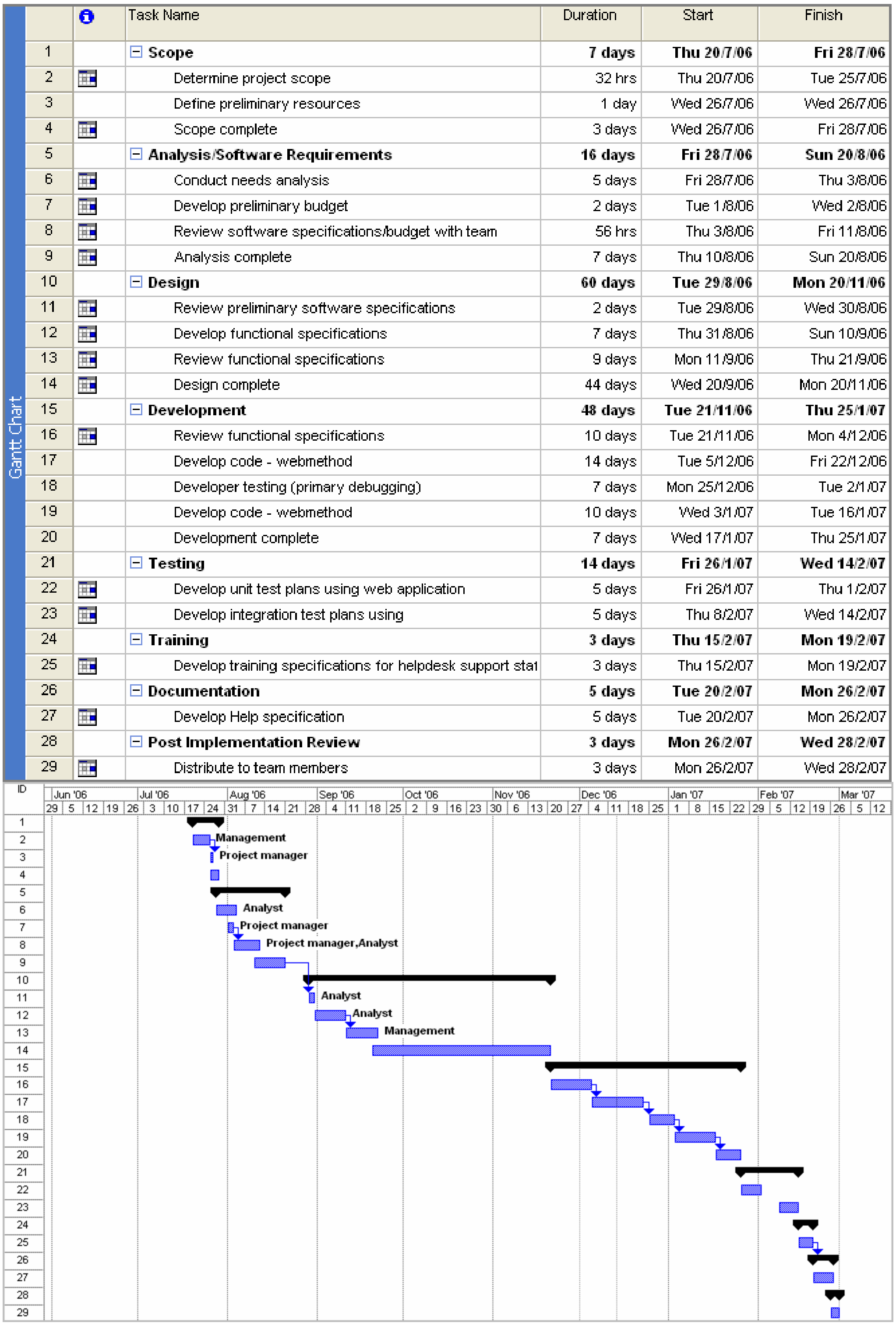
2) New Transact-SQL statements that applications can use to send and receive messages. Each message is part of a dialog: a reliable, persistent communication channel between two participants.

(Microsoft Corporation, 2005)

2.6 Deliverables

1. CD containing source code
2. User Manual
3. System Administrator Manual
4. Documentation
5. Web Application base for connect and interchange insurance business functions
6. Deployment
   Deploying a web service directly to IIS virtual application.

2.7 Implementation Plan



	ℹ	Task Name	Duration	Start	Finish
1		⊟ **Scope**	**7 days**	**Thu 20/7/06**	**Fri 28/7/06**
2	▦	Determine project scope	32 hrs	Thu 20/7/06	Tue 25/7/06
3		Define preliminary resources	1 day	Wed 26/7/06	Wed 26/7/06
4	▦	Scope complete	3 days	Wed 26/7/06	Fri 28/7/06
5		⊟ **Analysis/Software Requirements**	**16 days**	**Fri 28/7/06**	**Sun 20/8/06**
6	▦	Conduct needs analysis	5 days	Fri 28/7/06	Thu 3/8/06
7	▦	Develop preliminary budget	2 days	Tue 1/8/06	Wed 2/8/06
8	▦	Review software specifications/budget with team	56 hrs	Thu 3/8/06	Fri 11/8/06
9	▦	Analysis complete	7 days	Thu 10/8/06	Sun 20/8/06
10		⊟ **Design**	**60 days**	**Tue 29/8/06**	**Mon 20/11/06**
11	▦	Review preliminary software specifications	2 days	Tue 29/8/06	Wed 30/8/06
12	▦	Develop functional specifications	7 days	Thu 31/8/06	Sun 10/9/06
13	▦	Review functional specifications	9 days	Mon 11/9/06	Thu 21/9/06
14	▦	Design complete	44 days	Wed 20/9/06	Mon 20/11/06
15		⊟ **Development**	**48 days**	**Tue 21/11/06**	**Thu 25/1/07**
16	▦	Review functional specifications	10 days	Tue 21/11/06	Mon 4/12/06
17		Develop code - webmethod	14 days	Tue 5/12/06	Fri 22/12/06
18		Developer testing (primary debugging)	7 days	Mon 25/12/06	Tue 2/1/07
19		Develop code - webmethod	10 days	Wed 3/1/07	Tue 16/1/07
20		Development complete	7 days	Wed 17/1/07	Thu 25/1/07
21		⊟ **Testing**	**14 days**	**Fri 26/1/07**	**Wed 14/2/07**
22	▦	Develop unit test plans using web application	5 days	Fri 26/1/07	Thu 1/2/07
23	▦	Develop integration test plans using	5 days	Thu 8/2/07	Wed 14/2/07
24		⊟ **Training**	**3 days**	**Thu 15/2/07**	**Mon 19/2/07**
25	▦	Develop training specifications for helpdesk support stat	3 days	Thu 15/2/07	Mon 19/2/07
26		⊟ **Documentation**	**5 days**	**Tue 20/2/07**	**Mon 26/2/07**
27	▦	Develop Help specification	5 days	Tue 20/2/07	Mon 26/2/07
28		⊟ **Post Implementation Review**	**3 days**	**Mon 26/2/07**	**Wed 28/2/07**
29	▦	Distribute to team members	3 days	Mon 26/2/07	Wed 28/2/07

Figure 2.1 The Implement plan

2.8 Training Phase

2.8.1 Develop training specifications for end users

End users:

1. Officer of motor vehicle repair shops
2. Officer of the viriyah insurance company

2.8.2 Develop training specifications for helpdesk support staff

Training for staff of our company for support this project such when Officer of motor vehicle repair shops leave helpdesk support staff train to new user or get feedback from user.

2.8.3 Training materials complete

2.8.4 Training and publishing the project to the system.

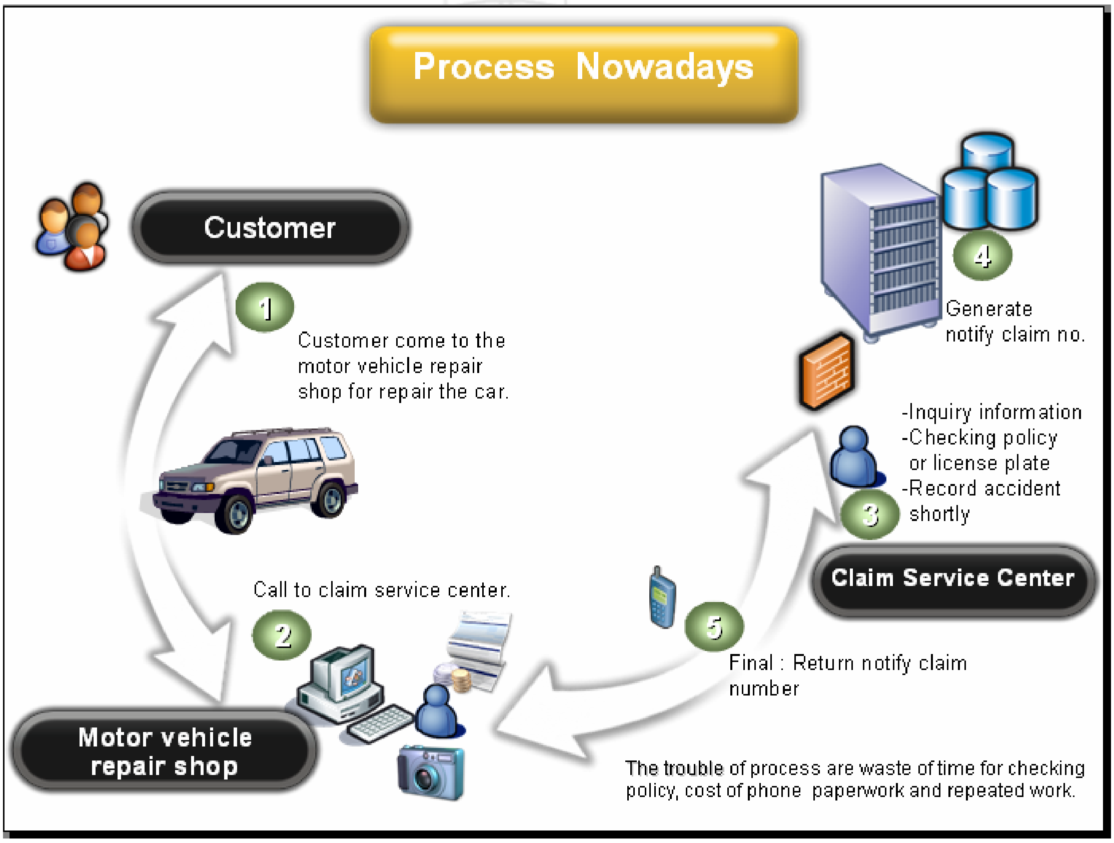
CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 Analysis of the existing system

The project specialized in case of the customer have a policy to cover for repair his motor vehicle.

In such a case, when the customer comes to motor vehicle repair shops, to arrange in order:



Process Nowadays
Customer
1
Customer come to the motor vehicle repair shop for repair the car.
Call to claim service center.
2
Motor vehicle repair shop
4
Generate notify claim no.
-Inquiry information
-Checking policy or license plate
-Record accident shortly
3
Claim Service Center
5
Final : Return notify claim number
The trouble of process are waste of time for checking policy, cost of phone paperwork and repeated work.

Figure 3.1 The process in existing system

Step in claim service process for motor vehicle repair shops:

Step 1: The customer submit service with motor vehicle repair shops which partner of the company insurance.

Step 2: The officer of motor vehicle repair shops call to claim service center about claim information and check policy information.

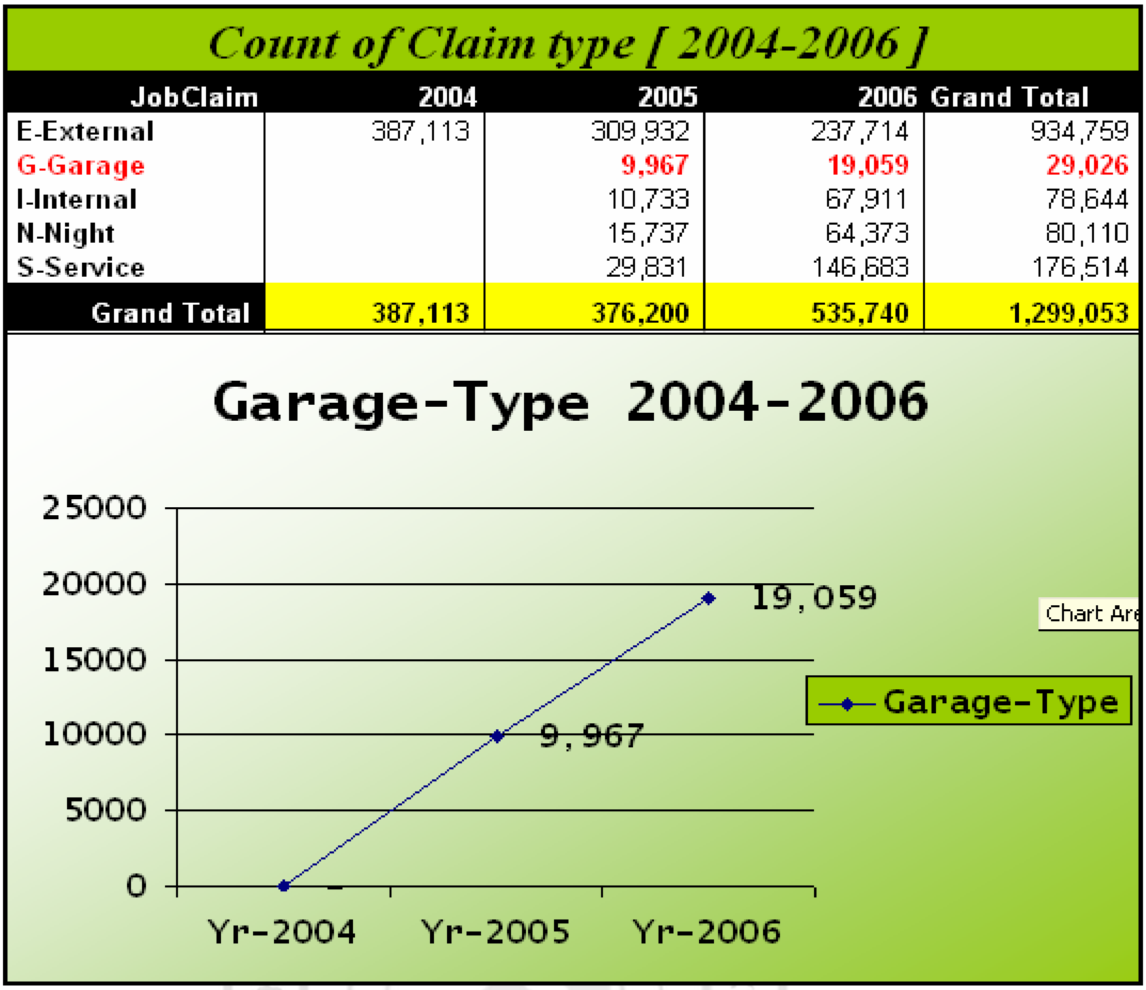
Step 3: Checking information and condition of policy, claim officer record claim information to the claim system

Step 4: Generating notify claim number from claim service system.

Step 5: Sending notify claim number to motor vehicle repair shops site and run process continue such check part detail and estimate expense.

3.2 User requirement analysis

Today, increasing of transaction for notify claim service from motor vehicle repair shops then the requirement of system need real time process. Organization wanted to use electronic capabilities to reach their process faster and more efficiently. By shortening lead time, the time it takes to receive the document number refer to next process, business can lower their time and telephone costs and gain competitive advantage.



Count of Claim type [2004-2006]

JobClaim	2004	2005	2006	Grand Total
E-External	387,113	309,932	237,714	934,759
G-Garage		9,967	19,059	29,026
I-Internal		10,733	67,911	78,644
N-Night		15,737	64,373	80,110
S-Service		29,831	146,683	176,514
Grand Total	387,113	376,200	535,740	1,299,053

Figure 3.2 The chart show number of notify claim service from Motor vehicle repair shops (G-Garage)

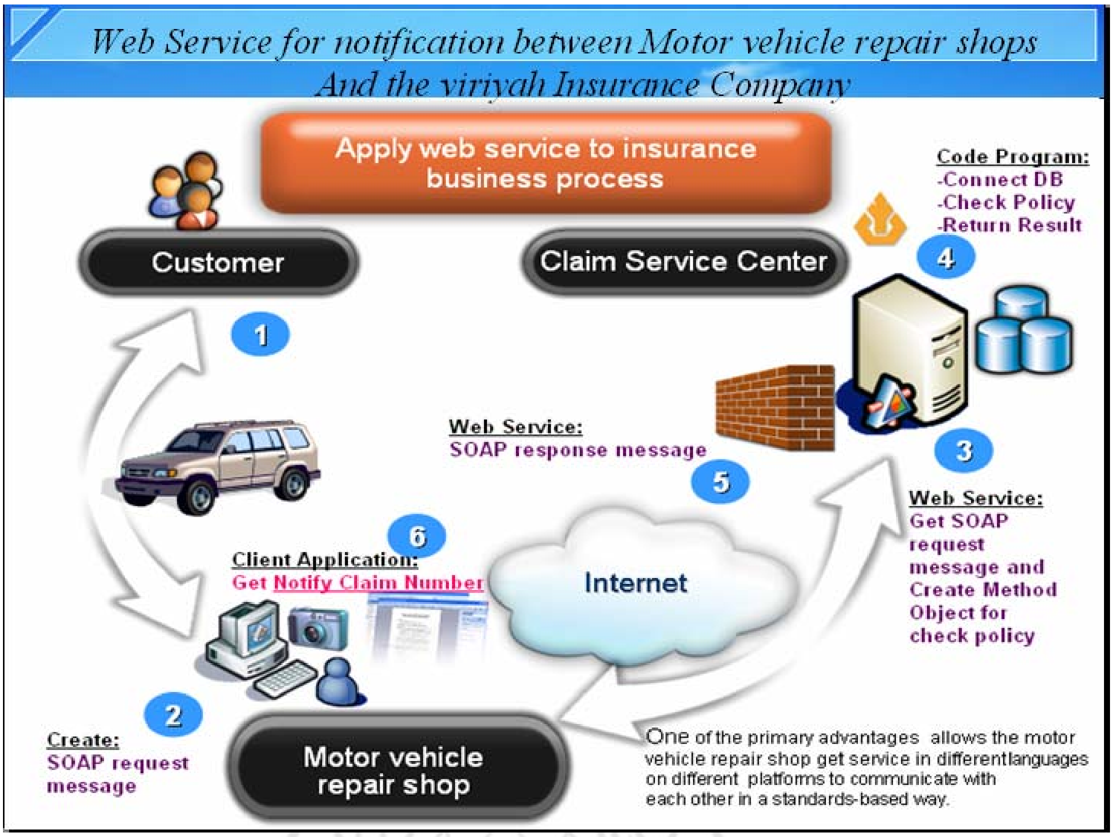

Web Service for notification between Motor vehicle repair shops
And the viriyah Insurance Company
Apply web service to insurance business process
Customer
Claim Service Center
Code Program:
-Connect DB
-Check Policy
-Return Result
4
1
Web Service:
SOAP response message
5
3
Web Service:
Get SOAP request message and Create Method Object for check policy
6
Client Application:
Get Notify Claim Number
Internet
2
Create:
SOAP request message
Motor vehicle repair shop
One of the primary advantages allows the motor vehicle repair shop get service in differentlanguages on different platforms to communicate with each other in a standards-based way.

Figure 3.3 To Apply web services to insurance business process

Using the Insurance claim scenario and Web services:

Step 1: The customer submit service with motor vehicle repair shops which partner of the company insurance.

Step 2: Motor vehicle repair shops – Create SOAP request message

1. Client application create SOAP request message.
2. Call method to check policy information.

Step 3:

1. Web Server get SOAP request message from client application.
2. Web Server create object for call check policy method.

Step 4:

1. Code program in check policy method connect database.
2. A valid input inside of the SOAP request message
3. Check policy, check constraint and condition of policy
   1) Exist in database
   2) Status of policy is active or cancels.
   3) Period of policy is cover an accident date.
4. Generate notify claim number and record in database
5. Return output and pack it's into SOAP response message.
6. If right conditions then return "Notify Claim Number" or return other message such as policy not found, the period of policy is expired.

Step 5:
Web Server pack result into SOAP response message return to requester.
Step 6:

1. Client Application get message in SOAP response message.
2. Display output on web form.

3.3 New System Design

The Microsoft SQL Server 2005 Database Engine is the core service for storing, processing, and securing data. The Database Engine provides controlled access and rapid transaction processing to meet the requirements of the most demanding data consuming applications within this project contains database:

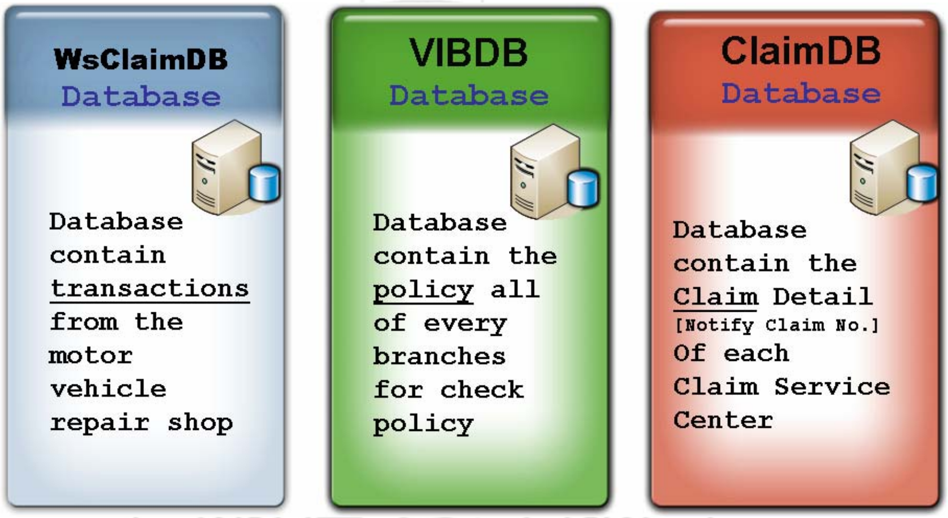

WsClaimDB
Database
Database contain transactions from the motor vehicle repair shop
VIBDB
Database
Database contain the policy all of every branches for check policy
ClaimDB
Database
Database contain the Claim Detail
[Notify Claim No.]
Of each Claim Service Center

Figure 3.4 SQL Server Database and each functional

WsClaimDB: The database design for contains transactions of notify claim service between motor vehicle repair shops and the company

Note: Each the shop depends on each claim service center.

For example: The K.C. garage shop in bangplad area belongs to 101-bangplad claim service Center when the claim officers need to see and check an accident motor for estimate in case of claim more than 20,000 baht.

VIBDB: This database contain overall of policy in company that serve for check policy detail match with an accident motor by policy number, license number and chassis number within period.

ClaimDB: This database to be located in each claim service center that contain overall claim detail specific of each service center only.

For example:

101-Bangplad Claim Service Center has Claimdb-Database, Notify table contain every type of notify claim number that continue running with in.

Type of Notify Claim Number separates the service:

1. E = External Service
2. G = Garage or Motor vehicle repair shops
3. I = Internal Service
4. N = Night Time Service
5. S = Service such as court appointment or parties of lawsuits

In this project provide the automate service to get notify claim number from each claim service center

3.3.1 The sequence diagram show 3 state

State 1: Register

1. The officer of motor vehicle repair shops create internet connect
2. Link to web site and open the default page
3. Select Register function and fill in the contents
4. Submit the register to the insurance system

State 2: Request Notify Claim Number

1. Get requirement from customer for repair the motor vehicle
2. The officer of motor vehicle repair shops
3. Login to connect the system
4. Check the policy exist in the company with policy number or license number or chassis number
5. If exist policy cover an accident date and condition of policy after check policy and the system return the policy number to client application then continue to request number of notify
6. After check policy and get the number then can submit request process
7. The system return notify claim number from their claim service center

State 3: Report of Notify Claim Number

1. Login to then system
2. Select menu for query report of notify claim number
3. Submit and get result of query

3.3.2 The Sequence Diagram

First, in developing the application, several decisions that could potentially affect my other models. This model decision should be reflected by user interface prototype, and verified by subject matter. It also helps our project to identify significant methods/services, such as checking to see if the policy exists in the system, which my system must support.

The reason to called sequence diagrams because obvious: in term database, connection, transactions and shown via the ordering of the activity in each case and easy to review the Return values in each step. As you can see the WsClaimDB Database is a web server in middle tier which to support request and response message between database and web application.

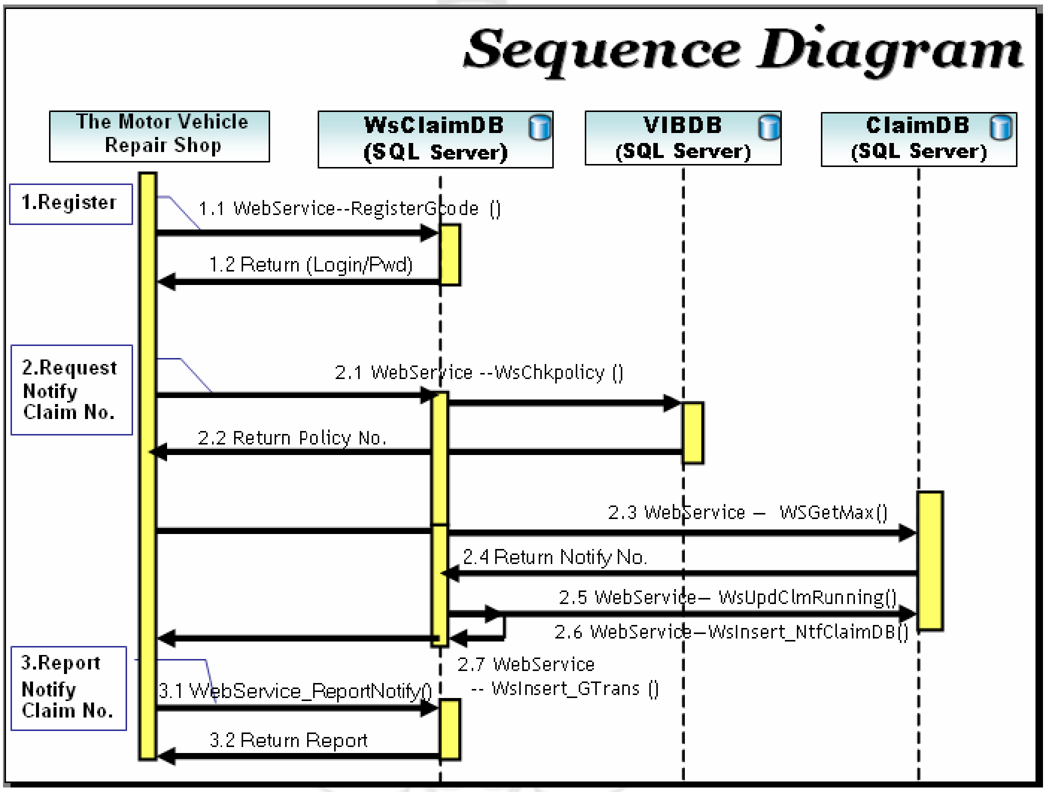

Sequence Diagram
The Motor Vehicle Repair Shop
WsClaimDB (SQL Server)
VIBDB (SQL Server)
ClaimDB (SQL Server)
1.Register
1.1 WebService--RegisterGcode ()
1.2 Return (Login/Pwd)
2.Request Notify Claim No.
2.1 WebService --WsChkpolicy ()
2.2 Return Policy No.
2.3 WebService – WSGetMax()
2.4 Return Notify No.
2.5 WebService– WsUpdClmRunning()
2.6 WebService–WsInsert_NtfClaimDB()
2.7 WebService -- WsInsert_GTrans ()
3.Report Notify Claim No.
3.1 WebService_ReportNotify()
3.2 Return Report

Figure 3.5 Sequence Diagram

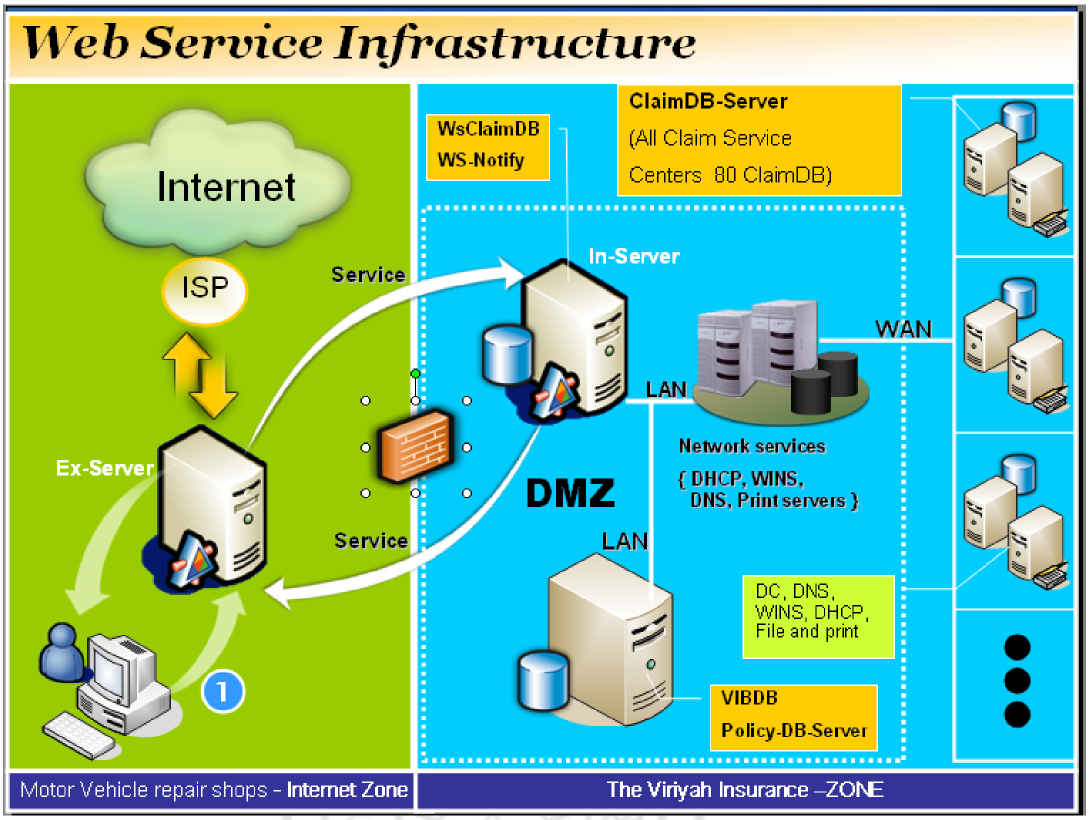


Figure 3.6 Web services Infrastructure

3.3.3 Infrastructure in claim notifies process

1. User Zone:

The officers of the motor vihecle repair shop create connection and connect to the system with web server that locates on ISP.

2. Internet Zone

WsClaimDB is web server and web services that located with in the company and connect with external web server by internet specific port through firewall for motor security.

3. Internal Zone in The Viriyah Insurance Zone

ClaimDB and VIBDB is Database Server located within Local Area Network (LAN) for communication with WsClaimDB and today, security systems managed security domains and accountability.

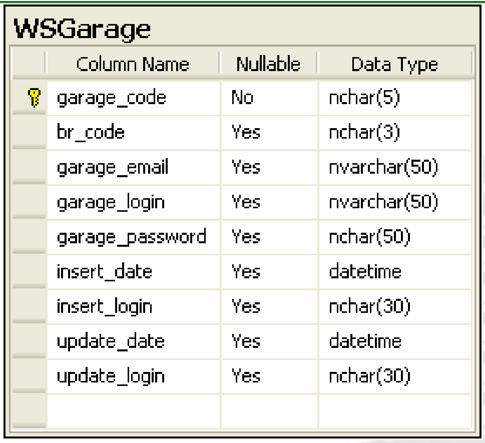
In addition, the security infrastructure will have to provide not only secure transport, but also higher-level access control mechanisms. Web services pose a unique challenge to the design of a distributed security system. At the high level the security system performs the access control, accountability, system integrity and confidentiality.

3.3.4 WSClaimDB - Database design and table

1. Master - WSGarage contain motor vehicle repair shops detail for support connection and authentication via internet.

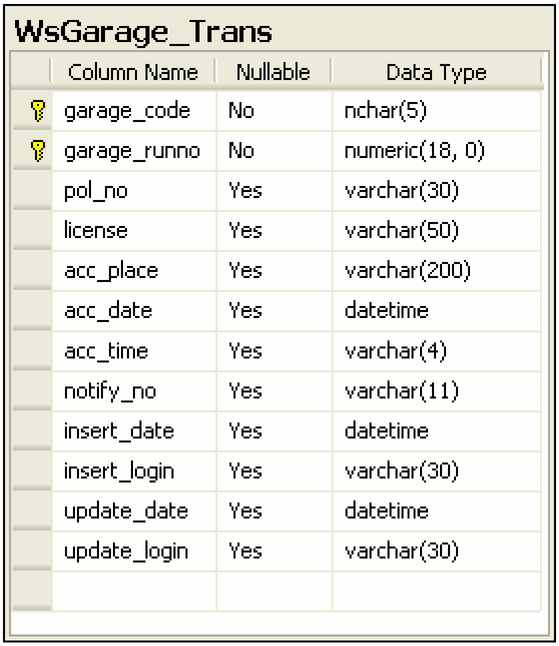
2. Transaction - WsGarage_Trans contain the record of notify claim detail

Master and Transaction



WSGarage

	Column Name	Nullable	Data Type
🔑	garage_code	No	nchar(5)
	br_code	Yes	nchar(3)
	garage_email	Yes	nvarchar(50)
	garage_login	Yes	nvarchar(50)
	garage_password	Yes	nchar(50)
	insert_date	Yes	datetime
	insert_login	Yes	nchar(30)
	update_date	Yes	datetime
	update_login	Yes	nchar(30)



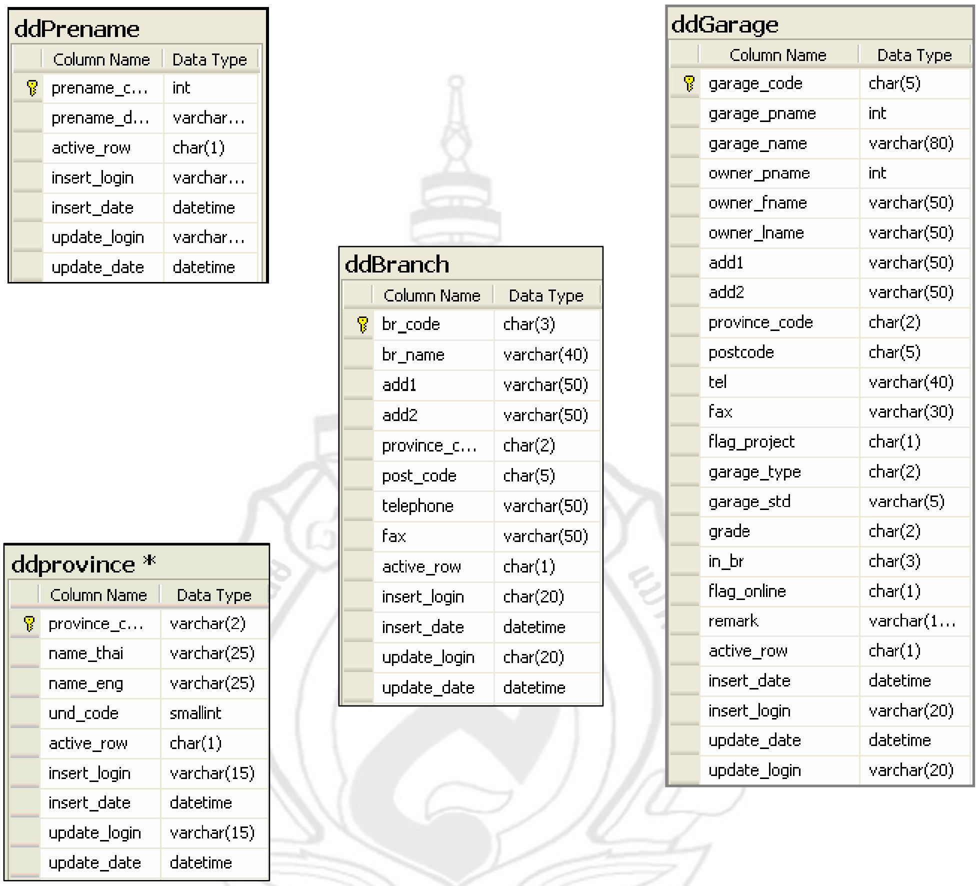
WsGarage_Trans

	Column Name	Nullable	Data Type
🔑	garage_code	No	nchar(5)
🔑	garage_runno	No	numeric(18, 0)
	pol_no	Yes	varchar(30)
	license	Yes	varchar(50)
	acc_place	Yes	varchar(200)
	acc_date	Yes	datetime
	acc_time	Yes	varchar(4)
	notify_no	Yes	varchar(11)
	insert_date	Yes	datetime
	insert_login	Yes	varchar(30)
	update_date	Yes	datetime
	update_login	Yes	varchar(30)

Figure 3.7 WsClaimDB – Master and Transaction Table

WSClaimDB - Database design and table
Table detail

1. ddBranch – the detail of branches or the claim service center
2. ddGarage – the detail of motor vehicle repair shops detail
3. ddPrename – the detail for display prename
4. ddProvince – the detail of province



ddPrename

	Column Name	Data Type
🔑	prename_c...	int
	prename_d...	varchar...
	active_row	char(1)
	insert_login	varchar...
	insert_date	datetime
	update_login	varchar...
	update_date	datetime

ddBranch

	Column Name	Data Type
🔑	br_code	char(3)
	br_name	varchar(40)
	add1	varchar(50)
	add2	varchar(50)
	province_c...	char(2)
	post_code	char(5)
	telephone	varchar(50)
	fax	varchar(50)
	active_row	char(1)
	insert_login	char(20)
	insert_date	datetime
	update_login	char(20)
	update_date	datetime

ddGarage

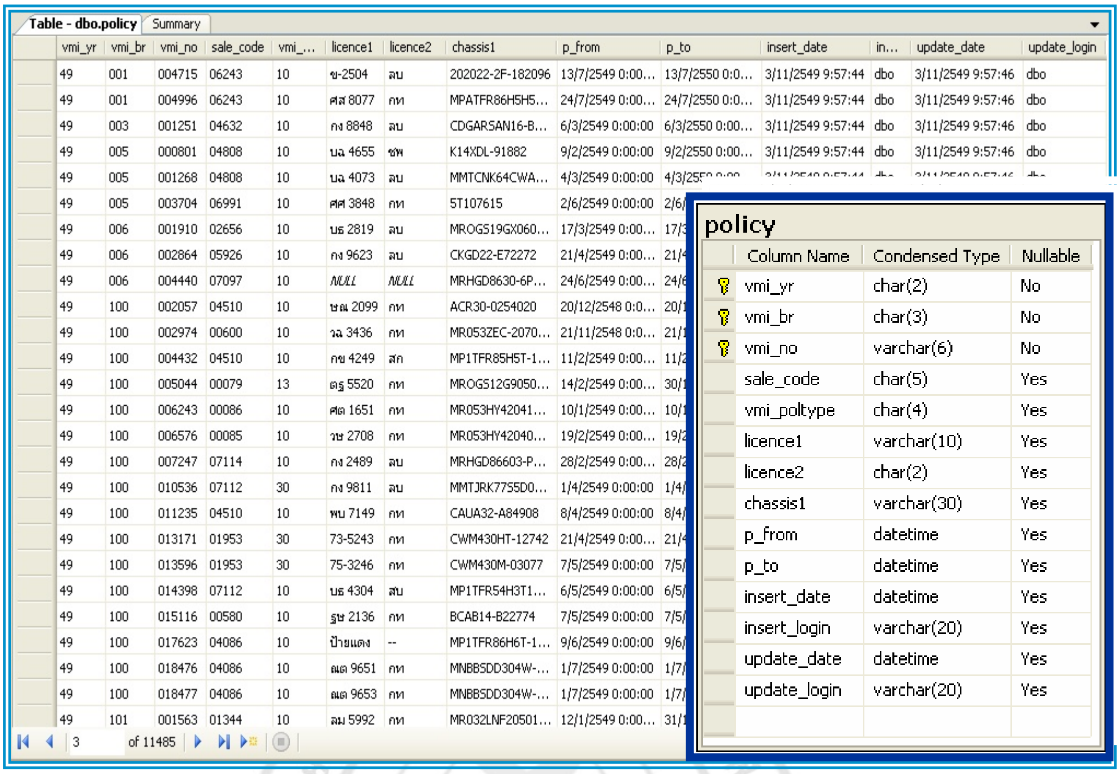
	Column Name	Data Type
🔑	garage_code	char(5)
	garage_pname	int
	garage_name	varchar(80)
	owner_pname	int
	owner_fname	varchar(50)
	owner_lname	varchar(50)
	add1	varchar(50)
	add2	varchar(50)
	province_code	char(2)
	postcode	char(5)
	tel	varchar(40)
	fax	varchar(30)
	flag_project	char(1)
	garage_type	char(2)
	garage_std	varchar(5)
	grade	char(2)
	in_br	char(3)
	flag_online	char(1)
	remark	varchar(1...
	active_row	char(1)
	insert_date	datetime
	insert_login	varchar(20)
	update_date	datetime
	update_login	varchar(20)

ddprovince *

	Column Name	Data Type
🔑	province_c...	varchar(2)
	name_thai	varchar(25)
	name_eng	varchar(25)
	und_code	smallint
	active_row	char(1)
	insert_login	varchar(15)
	insert_date	datetime
	update_login	varchar(15)
	update_date	datetime

Figure 3.8 WsClaimDB – Data Center Table

VIBDB - Database design and table
The policy table in VIBDB contain overall of policy for check policy



vmi_yr	vmi_br	vmi_no	sale_code	vmi_...	licence1	licence2	chassis1	p_from	p_to	insert_date	in...	update_date	update_login
49	001	004715	06243	10	ข-2504	ลบ	202022-2F-182096	13/7/2549 0:00...	13/7/2550 0:0...	3/11/2549 9:57:44	dbo	3/11/2549 9:57:46	dbo
49	001	004996	06243	10	ศส 8077	กท	MPATFR86H5H5...	24/7/2549 0:00...	24/7/2550 0:0...	3/11/2549 9:57:44	dbo	3/11/2549 9:57:46	dbo
49	003	001251	04632	10	กง 8848	ลบ	CDGARSAN16-B...	6/3/2549 0:00:00	6/3/2550 0:00...	3/11/2549 9:57:44	dbo	3/11/2549 9:57:46	dbo
49	005	000801	04808	10	บล 4655	ชพ	K14XDL-91882	9/2/2549 0:00:00	9/2/2550 0:00...	3/11/2549 9:57:44	dbo	3/11/2549 9:57:46	dbo
49	005	001268	04808	10	บล 4073	ลบ	MMTCNK64CWA...	4/3/2549 0:00:00	4/3/25	[illegible]	[illegible]	[illegible]	[illegible]
49	005	003704	06991	10	ศศ 3848	กท	5T107615	2/6/2549 0:00:00	2/6/				
49	006	001910	02656	10	บธ 2819	ลบ	MROGS19GX060...	17/3/2549 0:00...	17/3				
49	006	002864	05926	10	กง 9623	ลบ	CKGD22-E72272	21/4/2549 0:00...	21/4				
49	006	004440	07097	10	*NULL*	*NULL*	MRHGD8630-6P...	24/6/2549 0:00...	24/6				
49	100	002057	04510	10	ษฌ 2099	กท	ACR30-0254020	20/12/2548 0:0...	20/1				
49	100	002974	00600	10	วฉ 3436	กท	MR053ZEC-2070...	21/11/2548 0:0...	21/1				
49	100	004432	04510	10	กข 4249	สก	MP1TFR85H5T-1...	11/2/2549 0:00...	11/2				
49	100	005044	00079	13	ตฐ 5520	กท	MROGS12G9050...	14/2/2549 0:00...	30/1				
49	100	006243	00086	10	ศต 1651	กท	MR053HY42041...	10/1/2549 0:00...	10/1				
49	100	006576	00085	10	วษ 2708	กท	MR053HY42040...	19/2/2549 0:00...	19/2				
49	100	007247	07114	10	กง 2489	ลบ	MRHGD86603-P...	28/2/2549 0:00...	28/2				
49	100	010536	07112	30	กง 9811	ลบ	MMTJRK77S5D0...	1/4/2549 0:00:00	1/4/				
49	100	011235	04510	10	พบ 7149	กท	CAUA32-A84908	8/4/2549 0:00:00	8/4/				
49	100	013171	01953	30	73-5243	กท	CWM430HT-12742	21/4/2549 0:00...	21/4				
49	100	013596	01953	30	75-3246	กท	CWM430M-03077	7/5/2549 0:00:00	7/5/				
49	100	014398	07112	10	บธ 4304	สบ	MP1TFR54H3T1...	6/5/2549 0:00:00	6/5/				
49	100	015116	00580	10	ฐษ 2136	กท	BCAB14-B22774	7/5/2549 0:00:00	7/5/				
49	100	017623	04086	10	ป้ายแดง	--	MP1TFR86H6T-1...	9/6/2549 0:00:00	9/6/				
49	100	018476	04086	10	ฌต 9651	กท	MNBBSDD304W-...	1/7/2549 0:00:00	1/7/				
49	100	018477	04086	10	ฌต 9653	กท	MNBBSDD304W-...	1/7/2549 0:00:00	1/7/				
49	101	001563	01344	10	ลม 5992	กท	MR032LNF20501...	12/1/2549 0:00...	31/1				

policy

	Column Name	Condensed Type	Nullable
🔑	vmi_yr	char(2)	No
🔑	vmi_br	char(3)	No
🔑	vmi_no	varchar(6)	No
	sale_code	char(5)	Yes
	vmi_poltype	char(4)	Yes
	licence1	varchar(10)	Yes
	licence2	char(2)	Yes
	chassis1	varchar(30)	Yes
	p_from	datetime	Yes
	p_to	datetime	Yes
	insert_date	datetime	Yes
	insert_login	varchar(20)	Yes
	update_date	datetime	Yes
	update_login	varchar(20)	Yes

Figure 3.9 VIBDB – Policy Database Design and Table

ClaimDB - Database design and table. The database located with in branches that main database serve to claim process overall time.

Table relate with the project

1. Clmnotify: Table for record notify claim detail

2. ClmRunning: Table for contain number of each document such as notify claim number or claim number or receive number that job of each branches

CHAPTER 4

SYSTEM FUNCTIONALITY

4.1 Introduction

The project designed to make notify claim number fast, easy and improve productivity. In short, the solution can help complete save time and money. Ideally, notification of insurance should occur in “real time.” and cost effective for the claim process for beneficial through a partnership. This system is intended to be drive using web services via internet-based, shared services. Businesses will increase their use of Web services as software that many computer experts believe will usher in a new era of secure but simple interconnections among computer systems at different companies.

4.1.1 Scoping System Functionality:

1. Authentication Process

1) Log in – The process of receiving access to a system (After register process)

2) Registration – Sing up for New Account

2.Notify Claim No. Process

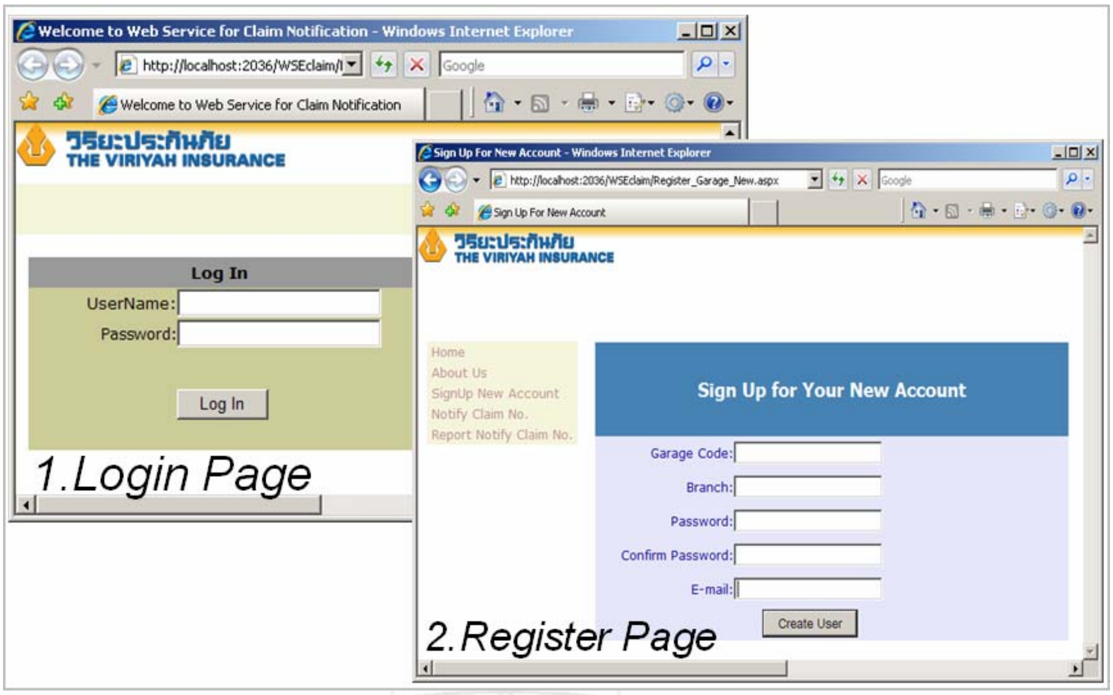
The process request notify claim number to the system within condition motor vehicle repair shops belong to claim service center accessed through a web applications and web services

3. Report of Notify Claim Process

The process request detail of notify claim and accident detail to the insurance system

4.1.2 Webpage Layout

The main reason of design web page layout is the common way based on simple, easy to understand. On top of page layout to show the main structure with in the page.


Welcome to Web Service for Claim Notification - Windows Internet Explorer
http://localhost:2036/WSEclaim/
Google
Welcome to Web Service for Claim Notification
วิริยะประกันภัย
THE VIRIYAH INSURANCE
Log In
UserName:
Password:
Log In
Sign Up For New Account - Windows Internet Explorer
http://localhost:2036/WSEclaim/Register_Garage_New.aspx
Sign Up For New Account
Home
About Us
SignUp New Account
Notify Claim No.
Report Notify Claim No.
Sign Up for Your New Account
Garage Code:
Branch:
Password:
Confirm Password:
E-mail:
Create User

1.Login Page

2.Register Page

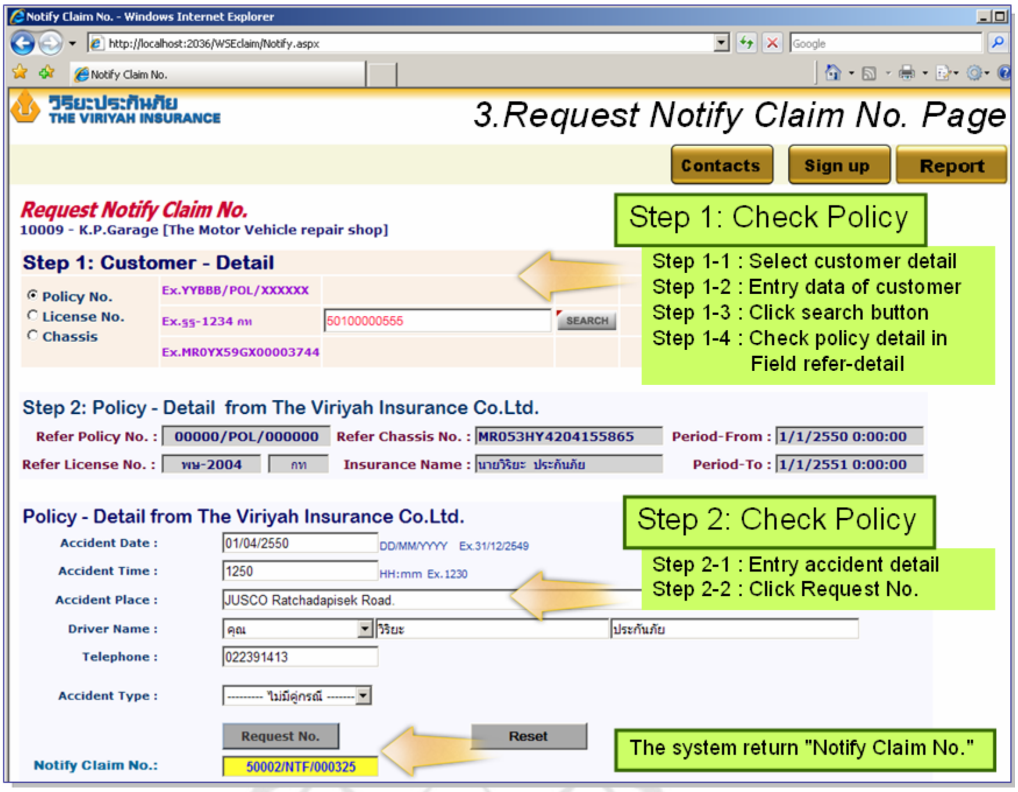
Figure 4.1 Web page design – Login Page and Register Page

4.1.3 Webpage Layout - Request Notify Claim No. Page

Request Notify Claim No. Page - Here is a main process page, this page divide in 2 parts. First is a Customer Detail and Accident Detail.

Table 4.1 Fields Description on Request Notify Claim No. Page

Field Name	Description	Example
Customer Detail		
Policy No.	Enter the number of policy: Year : 2 digit number from right Branch Code : 3 digit number Policy No. : 6 digit number	50 (Ex. 2550) 100 000555
License No.	A vehicle registration plate and One space and Province code	ฐฐ-1234 กท
Chassis No.	Vehicle identification number	MR053HY4204155865
Accident Detail		
Accident Date	Date of Accident	01/03/2550
Accident Time	Time of Accident	1200
Driver Name	Prename, First name and Last name	MR.Viriyah Meemana
Telephone	Phone Number	0819004321
Accident Type	Code of Accident	B01 Single-vehicle collisions



3.Request Notify Claim No. Page

Request Notify Claim No.
10009 - K.P.Garage [The Motor Vehicle repair shop]

Step 1: Customer - Detail

● Policy No. Ex.YYBBB/POL/XXXXXX
○ License No. Ex.ฐฐ-1234 กท 50100000555
○ Chassis Ex.MR0YX59GX00003744

Step 1: Check Policy

Step 1-1 : Select customer detail
Step 1-2 : Entry data of customer
Step 1-3 : Click search button
Step 1-4 : Check policy detail in Field refer-detail

Step 2: Policy - Detail from The Viriyah Insurance Co.Ltd.

Refer Policy No. : 00000/POL/000000
Refer Chassis No. : MR053HY4204155865
Period-From : 1/1/2550 0:00:00
Refer License No. : พษ-2004 กท
Insurance Name : นายวิริยะ ประกันภัย
Period-To : 1/1/2551 0:00:00

Policy - Detail from The Viriyah Insurance Co.Ltd.

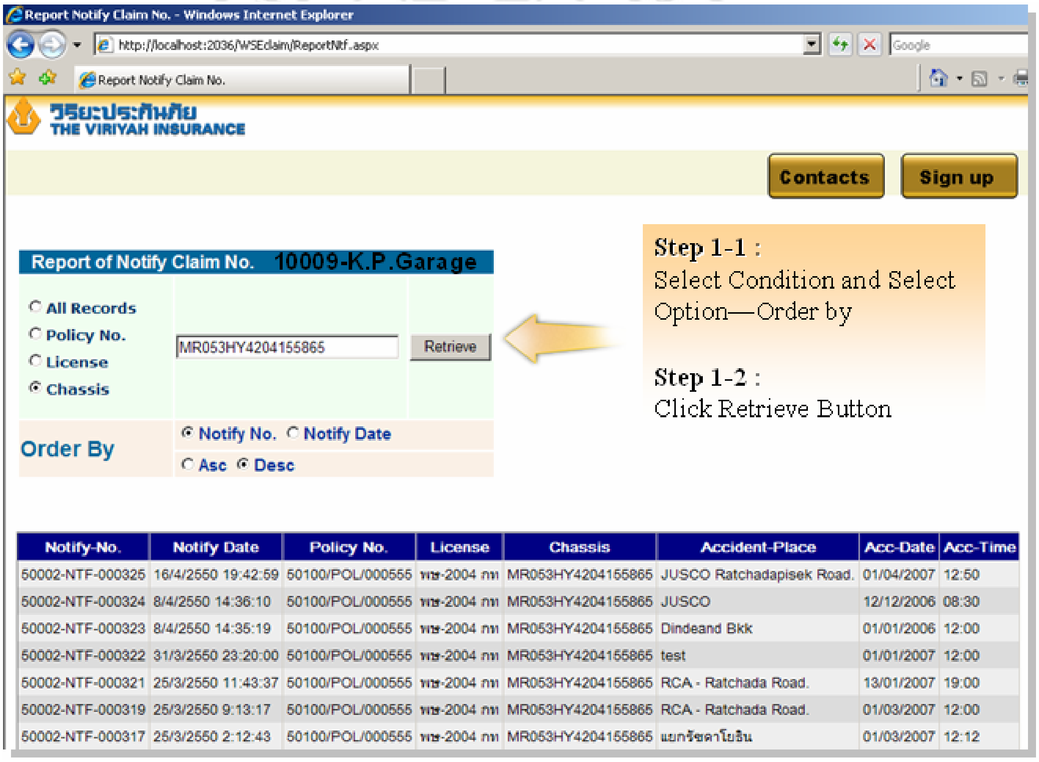
Accident Date : 01/04/2550 DD/MM/YYYY Ex.31/12/2549
Accident Time : 1250 HH:mm Ex.1230
Accident Place : JUSCO Ratchadapisek Road.
Driver Name : คุณ วิริยะ ประกันภัย
Telephone : 022391413
Accident Type : --------- ไม่มีคู่กรณี -------
Notify Claim No.: 50002/NTF/000325

Step 2: Check Policy

Step 2-1 : Entry accident detail
Step 2-2 : Click Request No.

The system return "Notify Claim No."

Figure 4.2 Web page design – Request Notify Claim No.


Report Notify Claim No. - Windows Internet Explorer
http://localhost:2036/WSEclaim/ReportNtf.aspx
Report Notify Claim No.
วิริยะประกันภัย
THE VIRIYAH INSURANCE
Contacts
Sign up
Retrieve

Report of Notify Claim No. 10009-K.P.Garage

○ All Records
○ Policy No.
○ License
● Chassis

MR053HY4204155865

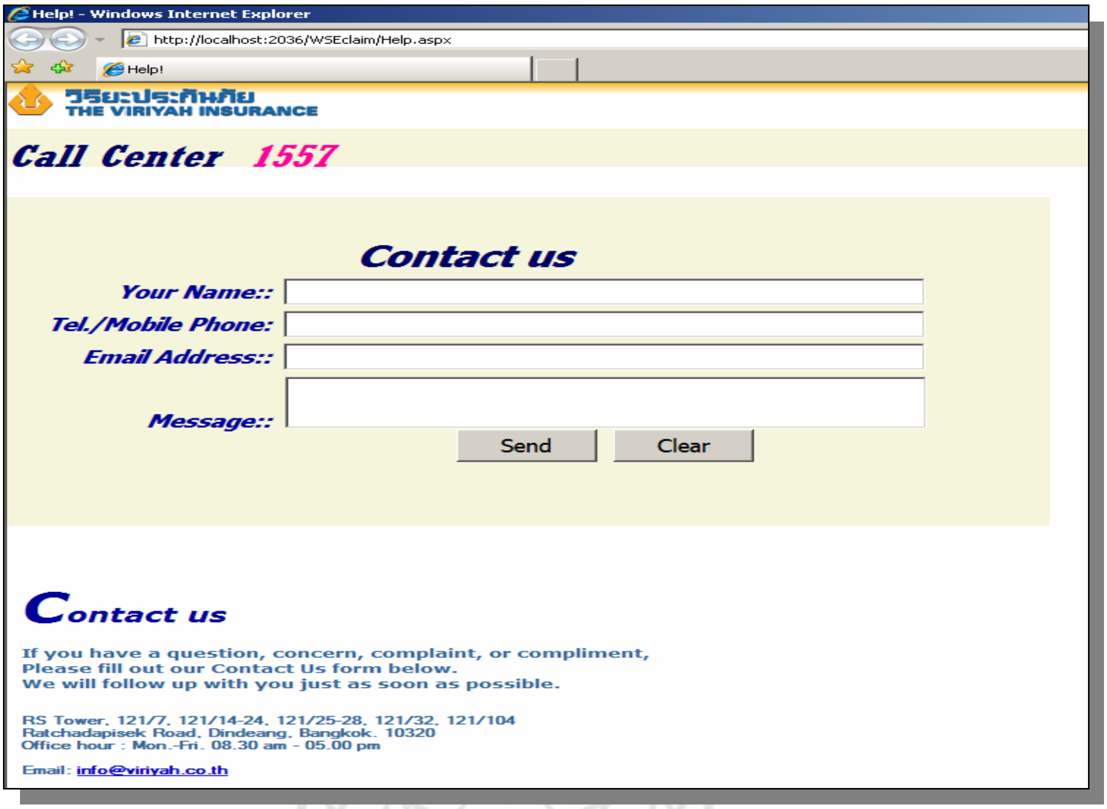
Order By ● Notify No. ○ Notify Date
○ Asc ● Desc

Step 1-1 :
Select Condition and Select Option—Order by

Step 1-2 :
Click Retrieve Button

Notify-No.	Notify Date	Policy No.	License	Chassis	Accident-Place	Acc-Date	Acc-Time
50002-NTF-000325	16/4/2550 19:42:59	50100/POL/000555	พษ-2004 กท	MR053HY4204155865	JUSCO Ratchadapisek Road.	01/04/2007	12:50
50002-NTF-000324	8/4/2550 14:36:10	50100/POL/000555	พษ-2004 กท	MR053HY4204155865	JUSCO	12/12/2006	08:30
50002-NTF-000323	8/4/2550 14:35:19	50100/POL/000555	พษ-2004 กท	MR053HY4204155865	Dindeand Bkk	01/01/2006	12:00
50002-NTF-000322	31/3/2550 23:20:00	50100/POL/000555	พษ-2004 กท	MR053HY4204155865	test	01/01/2007	12:00
50002-NTF-000321	25/3/2550 11:43:37	50100/POL/000555	พษ-2004 กท	MR053HY4204155865	RCA - Ratchada Road.	13/01/2007	19:00
50002-NTF-000319	25/3/2550 9:13:17	50100/POL/000555	พษ-2004 กท	MR053HY4204155865	RCA - Ratchada Road.	01/03/2007	12:00
50002-NTF-000317	25/3/2550 2:12:43	50100/POL/000555	พษ-2004 กท	MR053HY4204155865	แยกรัชดาโยธิน	01/03/2007	12:12

Figure 4.3 Web page design – Report Notify Claim


Help! - Windows Internet Explorer
http://localhost:2036/WSEclaim/Help.aspx
Help!
วิริยะประกันภัย
THE VIRIYAH INSURANCE
Send
Clear

Call Center 1557

Contact us

Your Name::

Tel./Mobile Phone:

Email Address::

Message::

Contact us

If you have a question, concern, complaint, or compliment,
Please fill out our Contact Us form below.
We will follow up with you just as soon as possible.

RS Tower, 121/7, 121/14-24, 121/25-28, 121/32, 121/104
Ratchadapisek Road, Dindeang, Bangkok, 10320
Office hour : Mon.-Fri. 08.30 am - 05.00 pm

Email: info@viriyah.co.th

Figure 4.4 Web page design – Help page

4.2 System Architecture

4.2.1 Functionality

The mission of the Web services to provide a Remote Procedure Call (RPC) interface for client applications to call class methods or web method on the server side (web server). SOAP is an important part of process it's the protocol that's responsible for routing the RPC message form the client to the server and returning the result back to the client application. SOAP is based on XML and follows a relatively simple design that's easy to implement. SOAP's simple protocol has contributed to its widespread support on just about any platform and development environment. (Rick Strahl, 2001)

The three major components behind the web services process:

1. The Web services on the Web Server side
2. The client application calling the Web services via a Web Reference
3. A WSDL Web services description that describes the functionality of the Web services

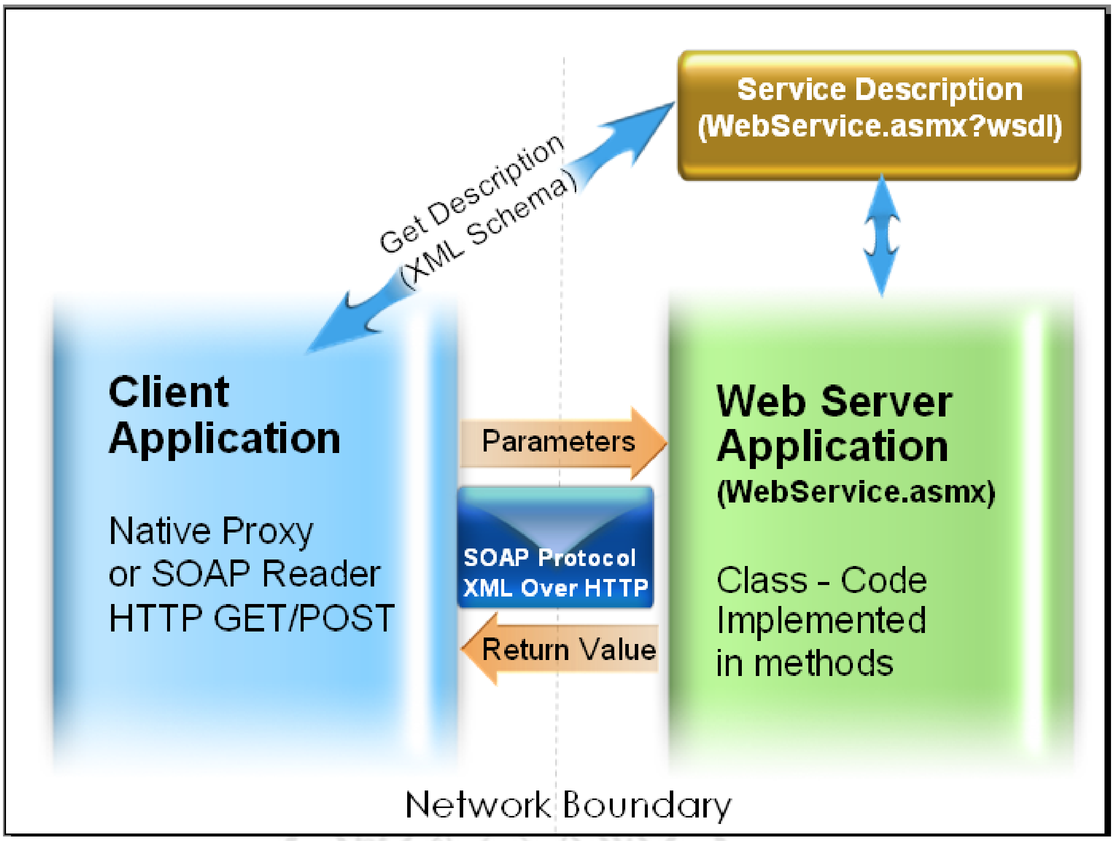


Figure 4.5 Web services use WSDL files to get description of the Web services The proxy calls the Web services using the SOAP protocol passing parameters and returning a return value for the remote method call

Table 4.2 Web Method – Service

Service Name	**WebService_Register()**
Description	To record detail of motor vehicle repair shops
Parameter	Garage_code, Br_code
Database	WsClaimDB
DML	Select and Insert
Service Name	**WebService_chkpolicy()**
Description	Validation Policy exist in the database
Parameter	Garage_code, Policy No.
Database	VibDB
DML	Select
Service Name	**WebService_AddNtf_to_wsdb()**
Description	Insert Notify Claim into WsClaimdb Database on Web Server
Parameter	Notify Claim No., Policy No. , License No., Chassis No.
Database	WsClaimDB
DML	Insert
Service Name	**WebService_AddNtf_to_clmdb()**
Description	Insert into Claimdb Database
Parameter	Notify Claim No., Policy No. , License No., Chassis No.
Database	ClaimDB
DML	Insert
Service Name	**WebService_GetMax_Ntf_ClmDB()**
Description	Select at last of Notify Claim No.
Parameter	Garage_code, Br_code
Database	ClaimDB
DML	Select
Service Name	**WebService_Update_running_clmdb()**
Description	Update Running No.
Parameter	Br_code, Running No.
Database	ClaimDB
DML	Update
Service Name	**WebService_ReportNotify()**
Description	To Report on the claim detail
Parameter	Garage_code, Br_code, Type of report
Database	WsClaimDB
DML	Select

4.3 Test Plan

4.3.1 Test management and testing strategy

Testing is the process of establishing confidence that a program or system does what is supposed to do. Testing of each and every unit/module is done as soon the development part is completed. All the bugs spotted are rectified and reframed so that all the unit or modules can be integrated and final testing can be done. Then the final solution is deployed for client's feedback. In Test Phase, Section the Pilot Testing Performance Indicators. There are three major phases to test project: Unit testing Integration testing and system testing.

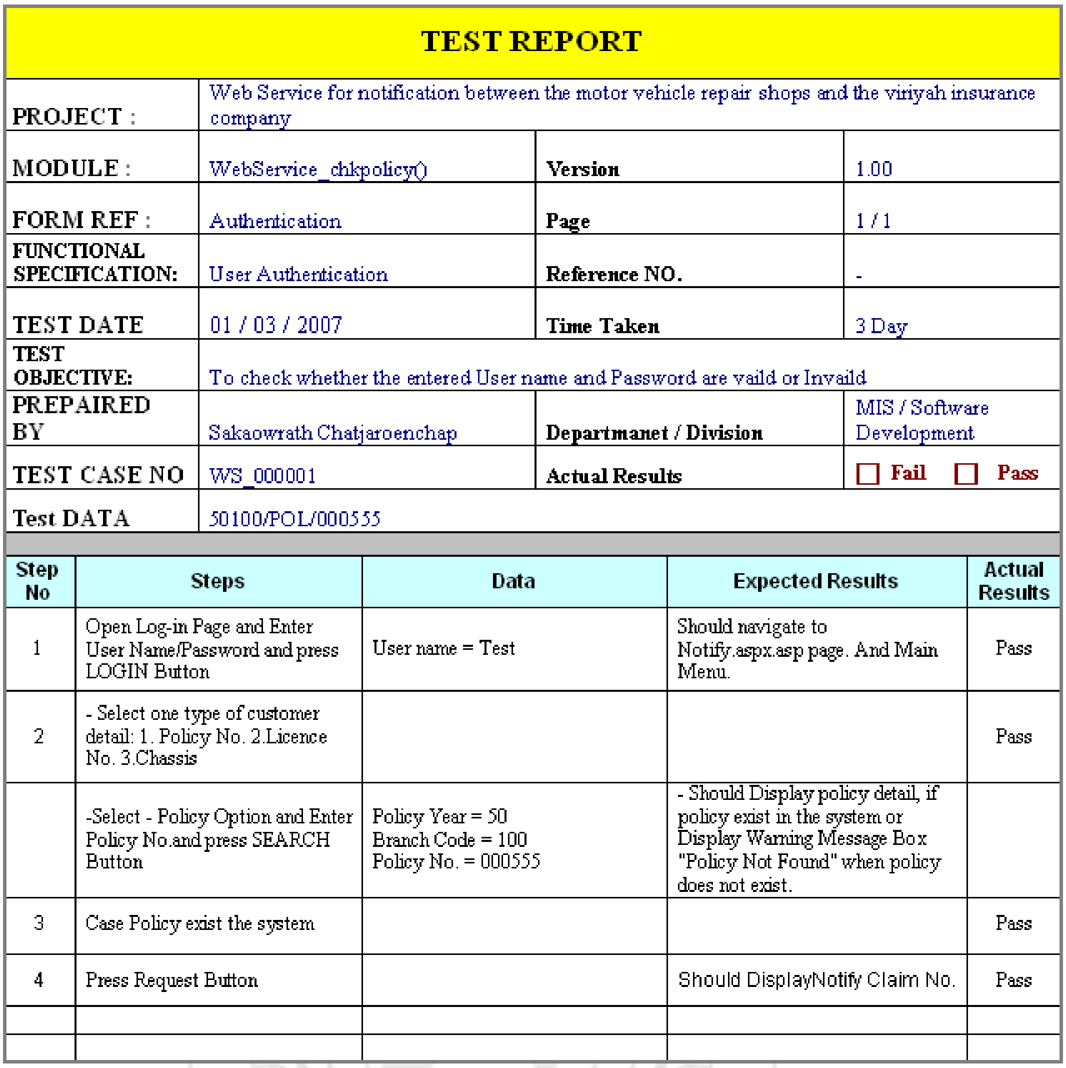
1. Unit testing - Programme Testing

1). Test web page – Review the content, layout, screen design, user interface, format and input field

1. Login Page
2. Register Page
3. Request Notify Claim No. Page
4. Report Notify Claim
5. Help and Contract Page

2). Test Web services and Web Method: Testing of the code in each module

1. WebService_Register
   - I. Method WebService_Register()
2. WebService_RequestNotifyNo()
   - I. Method WebService_chkpolicy()
   - II. Method: WebService_AddNtf_to_wsdb()
   - III. Method: WebService_AddNtf_to_clmdb()
   - IV. Method: WebService_GetMax_Ntf_clmdb()
   - V. Method: WebService_Update_running_clmdb()
3. WebService_ReportNotify()
   - I. Method: WebService_ReportNotify()



TEST REPORT

PROJECT :	Web Service for notification between the motor vehicle repair shops and the viriyah insurance company		
MODULE :	WebService_chkpolicy()	Version	1.00
FORM REF :	Authentication	Page	1 / 1
FUNCTIONAL SPECIFICATION:	User Authentication	Reference NO.	-
TEST DATE	01 / 03 / 2007	Time Taken	3 Day
TEST OBJECTIVE:	To check whether the entered User name and Password are vaild or Invaild		
PREPAIRED BY	Sakaowrath Chatjaroenchap	Departmanet / Division	MIS / Software Development
TEST CASE NO	WS_000001	Actual Results	□ Fail □ Pass
Test DATA	50100/POL/000555		

Step No	Steps	Data	Expected Results	Actual Results
1	Open Log-in Page and Enter User Name/Password and press LOGIN Button	User name = Test	Should navigate to Notify.aspx.asp page. And Main Menu.	Pass
2	- Select one type of customer detail: 1. Policy No. 2.Licence No. 3.Chassis			Pass
	-Select - Policy Option and Enter Policy No.and press SEARCH Button	Policy Year = 50 Branch Code = 100 Policy No. = 000555	- Should Display policy detail, if policy exist in the system or Display Warning Message Box "Policy Not Found" when policy does not exist.	
3	Case Policy exist the system			Pass
4	Press Request Button		Should DisplayNotify Claim No.	Pass

Figure 4.6 Example Test Report of each module test

1. Integration testing

Interoperability is the key that Web services require to open thedoor to mainstream computing. The interoperability of a Web services begins with its WSDL document, which describes everything about a service.

Web services can be created using two methods: top-down development and bottom-up development. The bottom-up approach to Web services development is fast and easy but may produce services with interoperability problems. The best way to maintain interoperability is by following the top-down approach to Web services development and starting your development by authoring your description document.

In this project using the bottom-up approach the developer writes the service implementation in any high-level programming language, such as Visual C#, has some tool generate the WSDL document describing the service and deploys the service to a server.

This approach is commonly used by company that have legacy, coupling systems its run both of business process relate with same running notify claim number of claim service center that applications within organization which want to have communicate with one another such as in this project is motor vehicle repair shops.

A Web services is built on top of the existing infrastructure and then exposed within the organization. In this situation the quick, bottom-up approach works fine. The company is both creating and consuming the services and the environment can be controlled as all of the clients of the service are known. Since both the services and the clients can be created using the same set of tools and run on the same platform, there does not need to be a lot of concern with respect to interoperability. The quickest and easiest way to start developing Web services is the bottom-up approach.

2. System testing - Automated load and performance testing System testing is done to the entire system against the Functional Requirement and an essential part of the process such as when the web services to automate generate the notify claim number (for the vehicle repair shop) the process of claim service center their can continue on-going. Aggregating data is dangerous, particularly on the Internet and support services - 24-hour support, every day of the year in part of Claim Service-Center.

I. Testing strategy – Performance Checking

Performance Checking is a strategy or formal approach to allow everyone involved in a Web application, The following sections are found in a performance test plan:

II. Web services Testing

The performance of the test was monitored on the following counters:

1. Processor: %Processor time
2. Memory: Page faults / Second

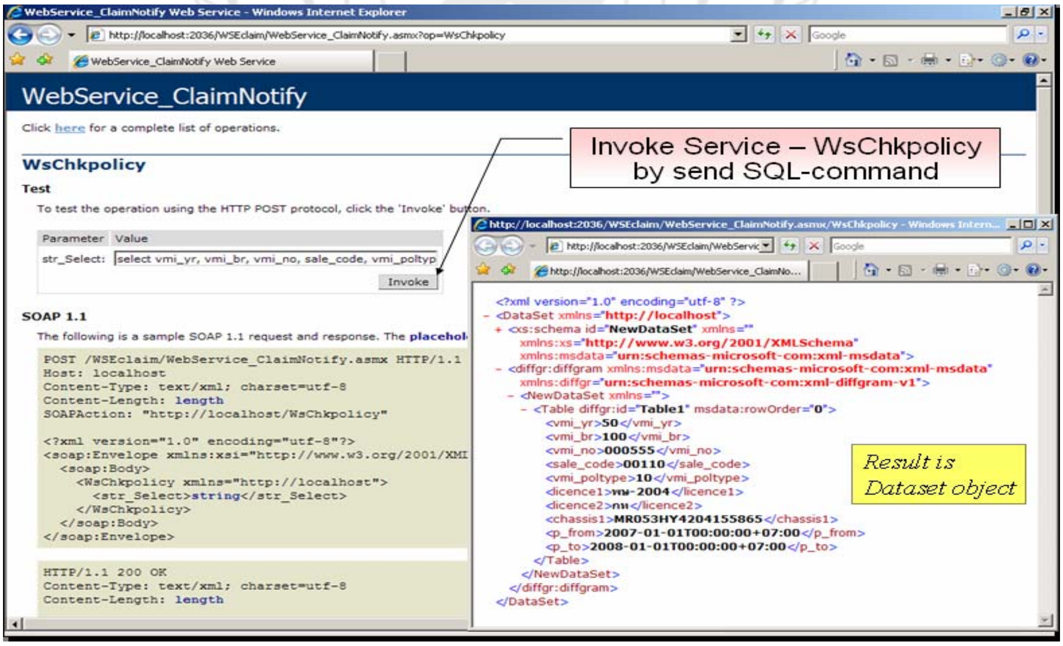
III Available bytes

1. Physical Disk: Average disk queue length
2. Web services: Bytes total / Sec., Current Connections, Get requests / Second and Post requests / Second.

4.4 Test results

Table 4.3 Test results

Item	Module	Result	Comment
1	Register		Should be review the Web services WSDL, Invoke the Web services Test and Review result in XML format and other environment such as Cross OS System and Other Browser.
	▪ WebService_Register()	Pass	
2	Request Notify Claim No.		
	▪ WebService_chkpolicy()	Pass	
	▪ WebService_AddNtf_to_wsdb()	Pass	
	▪ WebService_AddNtf_to_clmdb()	Pass	
	▪ WebService_GetMax_Ntf_ClmDB()	Pass	
	▪ WebService_Update_running_clmdb()	Pass	
3	Reporting of Claim Accident		
	▪ WebService_ReportNotify()	Pass	


WebService_ClaimNotify Web Service - Windows Internet Explorer
http://localhost:2036/WSEclaim/WebService_ClaimNotify.asmx?op=WsChkpolicy
WebService_ClaimNotify Web Service
Invoke

WebService_ClaimNotify

Click here for a complete list of operations.

WsChkpolicy

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
str_Select:	select vmi_yr, vmi_br, vmi_no, sale_code, vmi_poltyp

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placehol**

```
POST /WSEclaim/WebService_ClaimNotify.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://localhost/WsChkpolicy"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMI
  <soap:Body>
    <WsChkpolicy xmlns="http://localhost">
      <str_Select>string</str_Select>
    </WsChkpolicy>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

Invoke Service – WsChkpolicy by send SQL-command

http://localhost:2036/WSEclaim/WebService_ClaimNotify.asmx/WsChkpolicy - Windows Intern...
http://localhost:2036/WSEclaim/WebServic
Google
http://localhost:2036/WSEclaim/WebService_ClaimNo...

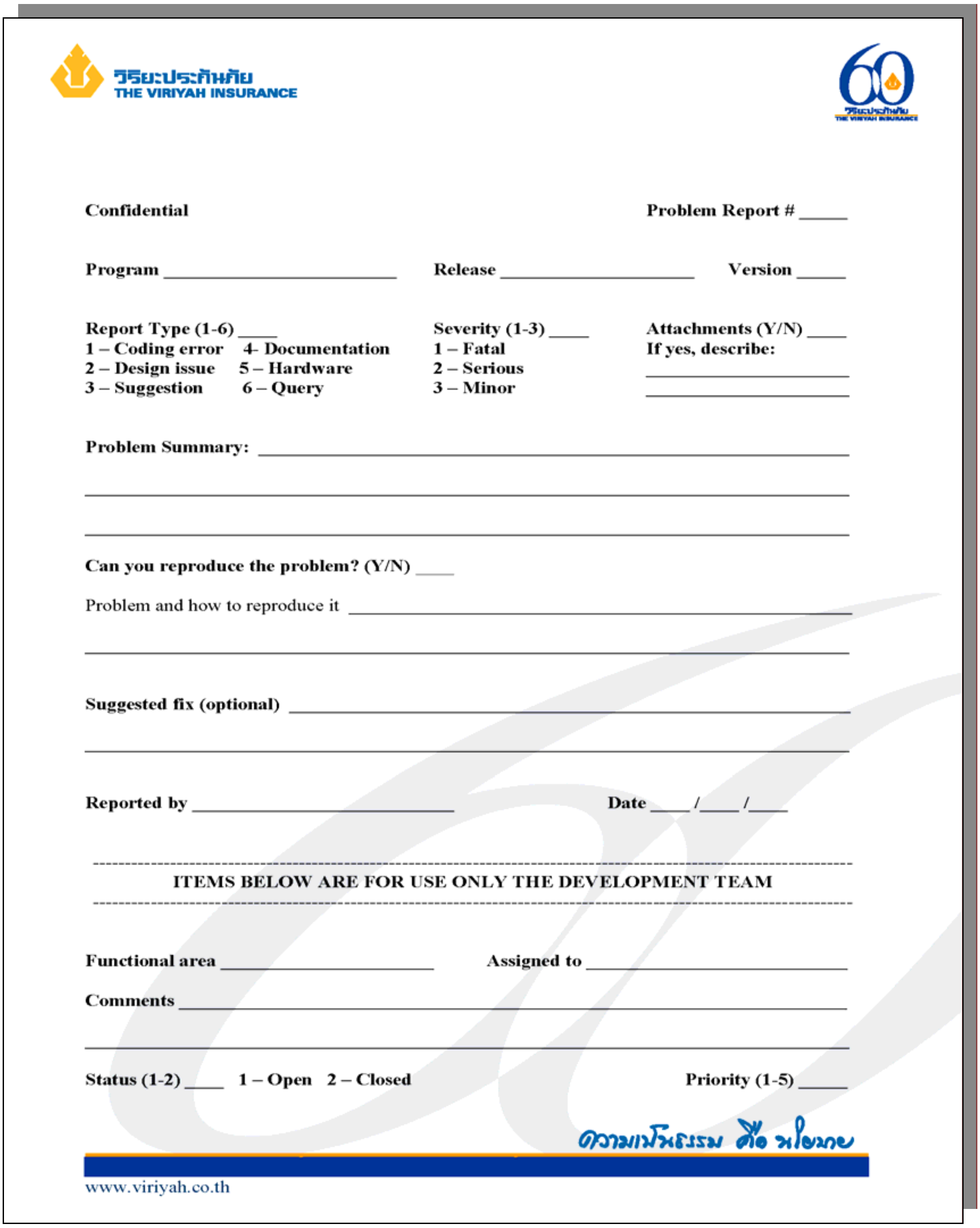
```
  <?xml version="1.0" encoding="utf-8" ?>
- <DataSet xmlns="http://localhost">
  + <xs:schema id="NewDataSet" xmlns=""
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  - <diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
      xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
    - <NewDataSet xmlns="">
      - <Table diffgr:id="Table1" msdata:rowOrder="0">
          <vmi_yr>50</vmi_yr>
          <vmi_br>100</vmi_br>
          <vmi_no>000555</vmi_no>
          <sale_code>00110</sale_code>
          <vmi_poltype>10</vmi_poltype>
          <licence1>ทม-2004</licence1>
          <licence2>กน</licence2>
          <chassis1>MR053HY4204155865</chassis1>
          <p_from>2007-01-01T00:00:00+07:00</p_from>
          <p_to>2008-01-01T00:00:00+07:00</p_to>
        </Table>
      </NewDataSet>
    </diffgr:diffgram>
  </DataSet>
```

Result is Dataset object

Figure 4.7 Result of WebService_ChkPolicy by WsChkPolicy Method in XML Form

4.4.1 Cycle of Testing

In this project provide the report maintained, tracked, controlled, and tested that improve the program, the report of testing to submit for formal testing often crash right away. These are described on problem report forms; In Design Error: How do you know you're in the right program?, How do you know what to do? How do you stop the program? And Coding Error: the sum isn't lined up with the other displayed numbers.



วิริยะประกันภัย
THE VIRIYAH INSURANCE

Confidential | Problem Report # _____

Program ________________________ Release ____________________ Version _____

Report Type (1-6) ____
1 – Coding error 4- Documentation
2 – Design issue 5 – Hardware
3 – Suggestion 6 – Query

Severity (1-3) ____
1 – Fatal
2 – Serious
3 – Minor

Attachments (Y/N) ____
If yes, describe:
____
____

Problem Summary: ____
____
____

Can you reproduce the problem? (Y/N) ____

Problem and how to reproduce it ____
____

Suggested fix (optional) ____
____

Reported by ___________________________ Date ____/____/____

ITEMS BELOW ARE FOR USE ONLY THE DEVELOPMENT TEAM

Functional area ______________________ Assigned to ________________________

Comments ____
____

Status (1-2) ____ 1 – Open 2 – Closed Priority (1-5) _____

ความมีคุณธรรม คือ หัวใจเรา

www.viriyah.co.th

Figure 4.8 The report form of testing to submit for formal testing

CHAPTER 5

SUMMARY AND SUGGESTIONS

From the evolution of internet and web technology effect to develop application platform and cause of open source, cross-platform applications concept that help drive the right business efficiencies, customer connections, and value-added services to advance the speed of innovation and enable business partner to make the decisions that drive business success and best return on investment then this is cause of convergence technology that reason to play changing the property of developer.

The core challenge for this study can be produced web application and web service to serve automated notification claim number process. The ability of web service can be integrated and interoperability between software applications to reduce both developments time and overall costs.

For this study cited indicate that takes advantage of web services to approved claim services which the aim to provide design and management with approachable manner and efficient, dependable claim service process.

5.1 Project Summary

Originally, a concept of web services is nothing new that base on an idea. Web service has two main part, Interoperability improve software development and communication to share function between application, computer, network and platform and the reusable is the key problem with procedural programming, write and rewrite that wastes time and money of business system.

On the evolution, Web services technologies designed to overcome the limited of these, it helps which includes approaching open standard, the way such as web services discovery that description and communication and provide standard of information in form of XML standard then Web services can take advantage and improve distributed computing capability the issue of limited interoperability and take advantage of object-oriented programming enable developer to build application from exist software component.

For this project, web services can improve distributed computing capability and reduce cost of development application and integration between business to business on security and trust communication.

5.2 Problems encountered and solutions

Web services developer should be have convergence in computing system and business environment such as network communication and security, web server and configuration, information and database system not only skill of coding because an application on web services concept that related with everything in computing system when you configured or changing network or something may be effect to application and information the developer should be to analyze the problem and solve to the right way base on convergence computing system.

5.3 Suggestions for further development

In the real computer world which cause of relationship then developer should be understand standard subject that control and manage integration computer environment for integrate business processes.

In the first phase of this project is part of notification claim number and next phase and hope to extend to claim service insurance system, motor insurance system and other part of business process for integrate and link information system and take advantage of reusable of the web services idea.

One Very Important Thought, Security is important for any kind of distributed computing environment. Web services security technique should be select, provides and configured suitable to business environment because of the concept of web services that design the communication over the internet then definition potential security risk and configure system security techniques, the security of network, the operating system, an application should be plug them.

Planning security system, provide the standard choosing the appropriate authentication method for a web application that related close with the performance level when security provide from transportation level its cover to application level and time response of acceptability communication.

Securing web services has several parts such as practical implementations will primarily examine the first, Security concept classifications, Threat classifications, Scope and Network security layers and XML Message Security Concepts that involve SOAP Message Security, XML Digital Signatures and XML Encryption.

In this project provide security in part of authentication method for a web application and security transportation level. Using WSE 3.0, Web services communication can be signed and encrypted to enable this project for web services enhancements, using X.509 certificates and other custom binary and XML-based security tokens. WSE also supports the ability to establish a trust service for retrieval and validation of security tokens, as well as the ability to establish more efficient long-running secure communication via secure conversations.

The fact that, because web services architecture is based on Web architecture, web services have the possibility of taking advantage. These are complex areas that touch on many of the different levels and technologies deployed in the service of web services.

5.4 Conclusions

Web Services leads to the next key concept in the interoperability across heterogeneous systems requires a mechanism to allow commonly understood by web services producers and consumers to enable business-to-business to link information.

In the future, everything comes to open standard for improve and trust and win-win to come together wish in coming soon the next step of web service is to add Just-In-Time Integration then web services applications need effective, standard methods for handling binary data. Web services can allow you to run your business more effectively, efficiently, and or create new business opportunities.

Project wish underline, its aims to bring the advantages of web services to the claim insurance process between multiple parties. It will be exciting to see the applications that emerge as developers and organizations use these systems to create the next generation of Web services-based solutions.

REFERENCES

W3C Working Group. (2004). **Web Services Architecture**. Available from http://www.w3.org/TR/ws-arch (14 March 2007)

Mark Colan. (2004). **Service-Oriented Architecture expands the vision of Web services.** Available from http://www128.ibm.com/developerworks/webservices/ library/ws-soaintro.html (29 April 2007)

Organization for the Advancement of Structured Information Standards. (2004). **Enabling Service-Oriented Architecture.** Available from http://www.uddi.org/whitepapers.html (29 April 2007)

Microsoft Corporation. (2005). **Microsoft Visual Studio 2005 Setup Issues.** Available from http://msdn2.microsoft.com/en-us/vstudio/aa718687.aspx (29 April 2007)

3 Leaf. (2006) **SQL Server 2005 versus Oracle Database 10g.** Available from http://www.microsoft.com/sql/prodinfo/compare/oracle/devprodoracle.mspx (29 April 2007)

Microsoft Corporation. (2005). **Comparing SQL Server 2005 to IBM.** Available from http://www.microsoft.com/sql/prodinfo/compare/default.mspx (29 April 2007)

SSWUG. (2005). **Comparing SQL Server 2000 versus MySQL.** Available from http://www.mssqlcity.com/Articles/Compare/sql_server_vs_mysql.htm (29 April 2007)

Microsoft Corporation. (2005). **Using Native XML Web Services in SQL Server 2005**. Available from: http://msdn2.microsoft.com/en-us/library/ms191274.aspx (29 April 2007)

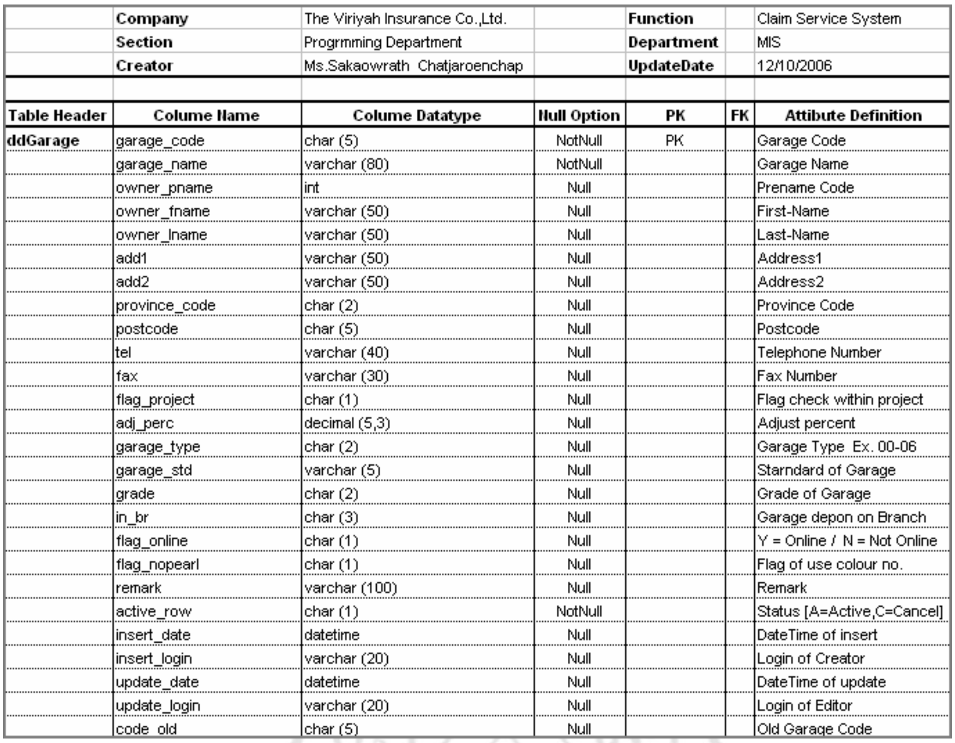
Rick Strahl. (2001). **Creating Web Services with .NET and Visual Studio.NET.** Available from http://www.programmersheaven.com/2/DotNet-Web-Services. (29 April 2007)

Uche Ogbuji. (2000) **Using WSDL in SOAP applications.** Available from http://www-128.ibm.com/developerworks/library/ws-soap (1 May 2007)

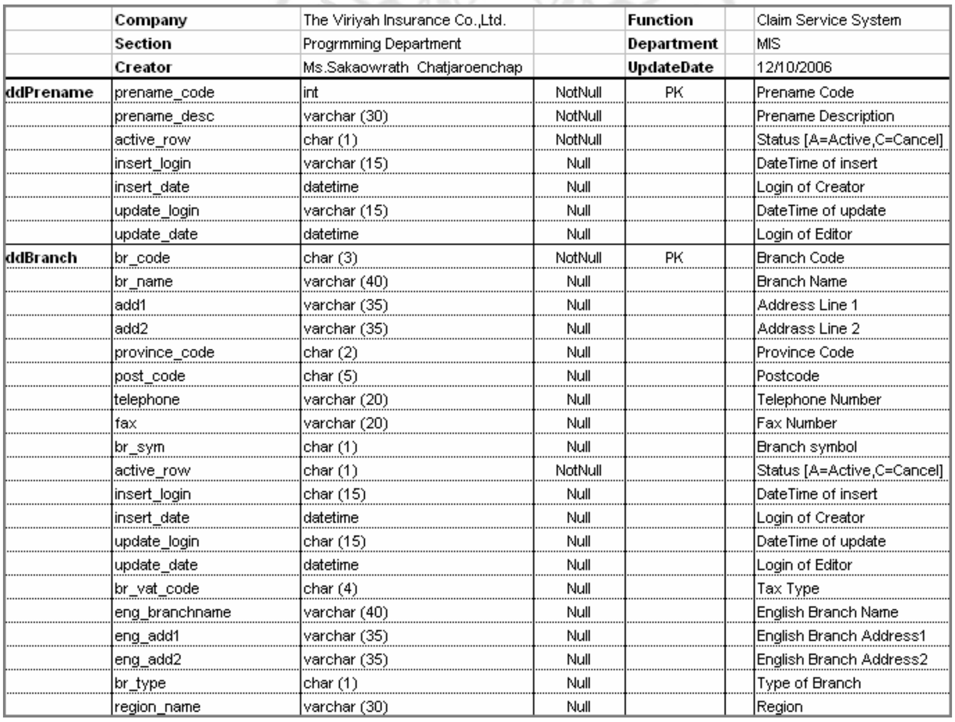
APPENDIX A

DATA DICTIONARY

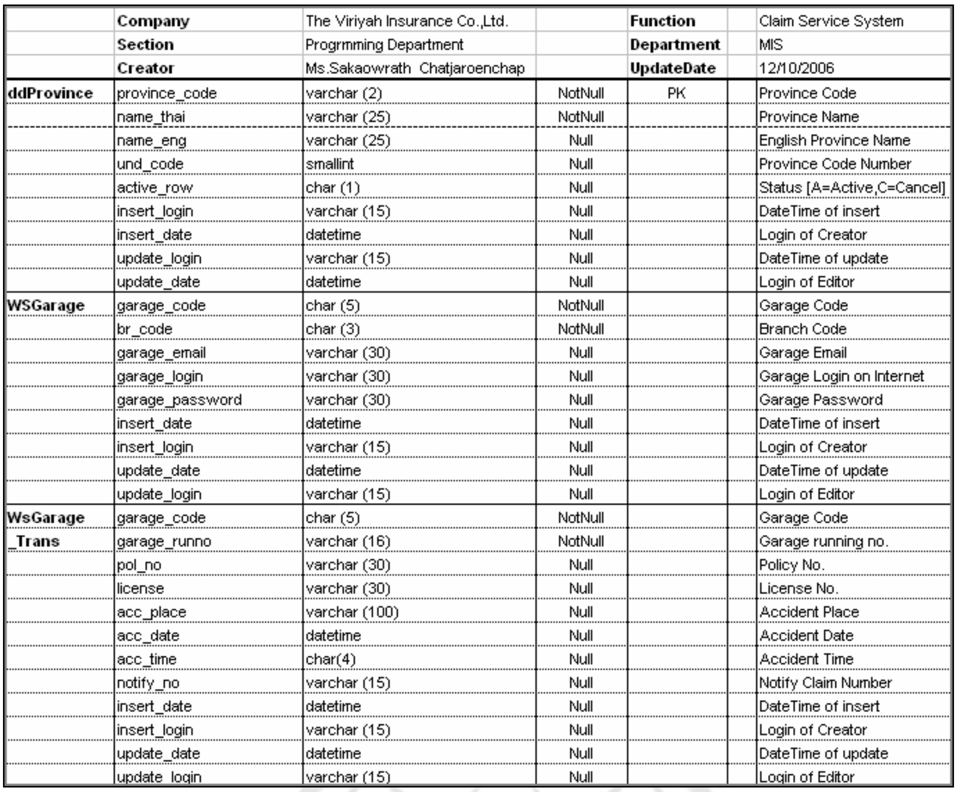
Data Dictionary



	Company	The Viriyah Insurance Co.,Ltd.		Function		Claim Service System
	Section	Progrmming Department		Department		MIS
	Creator	Ms.Sakaowrath Chatjaroenchap		UpdateDate		12/10/2006
Table Header	Colume Name	Colume Datatype	Null Option	PK	FK	Attibute Definition
ddGarage	garage_code	char (5)	NotNull	PK		Garage Code
	garage_name	varchar (80)	NotNull			Garage Name
	owner_pname	int	Null			Prename Code
	owner_fname	varchar (50)	Null			First-Name
	owner_lname	varchar (50)	Null			Last-Name
	add1	varchar (50)	Null			Address1
	add2	varchar (50)	Null			Address2
	province_code	char (2)	Null			Province Code
	postcode	char (5)	Null			Postcode
	tel	varchar (40)	Null			Telephone Number
	fax	varchar (30)	Null			Fax Number
	flag_project	char (1)	Null			Flag check within project
	adj_perc	decimal (5,3)	Null			Adjust percent
	garage_type	char (2)	Null			Garage Type Ex. 00-06
	garage_std	varchar (5)	Null			Starndard of Garage
	grade	char (2)	Null			Grade of Garage
	in_br	char (3)	Null			Garage depon on Branch
	flag_online	char (1)	Null			Y = Online / N = Not Online
	flag_nopearl	char (1)	Null			Flag of use colour no.
	remark	varchar (100)	Null			Remark
	active_row	char (1)	NotNull			Status [A=Active,C=Cancel]
	insert_date	datetime	Null			DateTime of insert
	insert_login	varchar (20)	Null			Login of Creator
	update_date	datetime	Null			DateTime of update
	update_login	varchar (20)	Null			Login of Editor
	code_old	char (5)	Null			Old Garage Code



	Company	The Viriyah Insurance Co.,Ltd.		Function		Claim Service System
	Section	Progrmming Department		Department		MIS
	Creator	Ms.Sakaowrath Chatjaroenchap		UpdateDate		12/10/2006
ddPrename	prename_code	int	NotNull	PK		Prename Code
	prename_desc	varchar (30)	NotNull			Prename Description
	active_row	char (1)	NotNull			Status [A=Active,C=Cancel]
	insert_login	varchar (15)	Null			DateTime of insert
	insert_date	datetime	Null			Login of Creator
	update_login	varchar (15)	Null			DateTime of update
	update_date	datetime	Null			Login of Editor
ddBranch	br_code	char (3)	NotNull	PK		Branch Code
	br_name	varchar (40)	Null			Branch Name
	add1	varchar (35)	Null			Address Line 1
	add2	varchar (35)	Null			Address Line 2
	province_code	char (2)	Null			Province Code
	post_code	char (5)	Null			Postcode
	telephone	varchar (20)	Null			Telephone Number
	fax	varchar (20)	Null			Fax Number
	br_sym	char (1)	Null			Branch symbol
	active_row	char (1)	NotNull			Status [A=Active,C=Cancel]
	insert_login	char (15)	Null			DateTime of insert
	insert_date	datetime	Null			Login of Creator
	update_login	char (15)	Null			DateTime of update
	update_date	datetime	Null			Login of Editor
	br_vat_code	char (4)	Null			Tax Type
	eng_branchname	varchar (40)	Null			English Branch Name
	eng_add1	varchar (35)	Null			English Branch Address1
	eng_add2	varchar (35)	Null			English Branch Address2
	br_type	char (1)	Null			Type of Branch
	region_name	varchar (30)	Null			Region



	Company	The Viriyah Insurance Co.,Ltd.		**Function**		Claim Service System
	Section	Progrmming Department		**Department**		MIS
	Creator	Ms.Sakaowrath Chatjaroenchap		**UpdateDate**		12/10/2006
ddProvince	province_code	varchar (2)	NotNull	PK		Province Code
	name_thai	varchar (25)	NotNull			Province Name
	name_eng	varchar (25)	Null			English Province Name
	und_code	smallint	Null			Province Code Number
	active_row	char (1)	Null			Status [A=Active,C=Cancel]
	insert_login	varchar (15)	Null			DateTime of insert
	insert_date	datetime	Null			Login of Creator
	update_login	varchar (15)	Null			DateTime of update
	update_date	datetime	Null			Login of Editor
WSGarage	garage_code	char (5)	NotNull			Garage Code
	br_code	char (3)	NotNull			Branch Code
	garage_email	varchar (30)	Null			Garage Email
	garage_login	varchar (30)	Null			Garage Login on Internet
	garage_password	varchar (30)	Null			Garage Password
	insert_date	datetime	Null			DateTime of insert
	insert_login	varchar (15)	Null			Login of Creator
	update_date	datetime	Null			DateTime of update
	update_login	varchar (15)	Null			Login of Editor
WsGarage	garage_code	char (5)	NotNull			Garage Code
_Trans	garage_runno	varchar (16)	NotNull			Garage running no.
	pol_no	varchar (30)	Null			Policy No.
	license	varchar (30)	Null			License No.
	acc_place	varchar (100)	Null			Accident Place
	acc_date	datetime	Null			Accident Date
	acc_time	char(4)	Null			Accident Time
	notify_no	varchar (15)	Null			Notify Claim Number
	insert_date	datetime	Null			DateTime of insert
	insert_login	varchar (15)	Null			Login of Creator
	update_date	datetime	Null			DateTime of update
	update_login	varchar (15)	Null			Login of Editor

APPENDIX B

EXAMPLE SOURCE OF WEB METHOD

Example source of Web Method

- **Register section**
  - WebService_Register()
- **Request Notify Claim Number**
  - WebService_chkpolicy()
  - WebService_AddNtf_to_wsdb()
  - WebService_AddNtf_to_clmdb()
  - WebService_GetMax_Ntf_ClmDB()
  - WebService_Update_running_clmdb()
- **Retrieve report - Notify Claim Number**
  - WebService_ReportNotify()

Web Service and Web Method Detail

Filename - WebService_ClaimNotify.cs

```
using System;
using System.Web;
using System.Collections;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Data;
using System.Data.OleDb;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

//For Security-Policy
using Microsoft.Web.Services3;
using Microsoft.Web.Services3.Design;
using Microsoft.Web.Services3.Security;
using Microsoft.Web.Services3.Security.Cryptography;
using Microsoft.Web.Services3.Security.Tokens;
using Microsoft.Web.Services3.Messaging;

[WebService(Namespace = "http://localhost")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
[Policy("ServicePolicy")]
```

```
//[Policy("ClientSideBasicPolicy")]
public class WebService_ClaimNotify : System.Web.Services.WebService
{

   public WebService_ClaimNotify ()
   {
      //InitializeComponent();
   }

   [WebMethod]
   public DataSet WsChkpolicy(string str_Select)
   {
      string str_Conn = "Provider=SQLOLEDB;Data Source=.;Integrated
Security=SSPI;Initial Catalog=VibDB";
      OleDbConnection obj_conn = new OleDbConnection(str_Conn);
      OleDbDataAdapter obj_da = new OleDbDataAdapter(str_Select, str_Conn);
      DataSet obj_ds = new DataSet();
      obj_conn.Open();
      obj_da.Fill(obj_ds);
      obj_conn.Close();
      return obj_ds;
   }

   [WebMethod]
   public DataSet GetReportNotify(string pv_select)
   {
      string pv_ConStr = "Provider=SQLOLEDB;Data Source=.;Integrated
Security=SSPI;Initial Catalog=WSClaimDB;";
      OleDbConnection obj_conn = new OleDbConnection(pv_ConStr);
      OleDbCommand obj_comm = new OleDbCommand(pv_select, obj_conn);
      OleDbDataAdapter obj_da = new OleDbDataAdapter(obj_comm);
      DataSet obj_ds = new DataSet();

      obj_conn.Open();
      obj_conn.Close();
      obj_da.Fill(obj_ds);
      return obj_ds;
   }

   [WebMethod]
   public DataSet RegisterGcode(string pv_select)
   {
      //str_Insert =

      string pv_ConStr = "Provider=SQLOLEDB;Data Source=.;Integrated
Security=SSPI;Initial Catalog=WSClaimDB;";
      OleDbConnection obj_conn = new OleDbConnection(pv_ConStr);
```

```
        OleDbCommand obj_comm = new OleDbCommand(pv_select, obj_conn);
        OleDbDataAdapter obj_da = new OleDbDataAdapter(obj_comm);
        DataSet obj_ds = new DataSet();

        obj_conn.Open();
        obj_conn.Close();
        obj_da.Fill(obj_ds);
        return obj_ds;
    }


    [WebMethod]
    public DataSet WSGetMax(string pv_select)
    {
    string pv_ConStr = "Provider=SQLOLEDB;Data Source=.;Integrated
Security=SSPI;Initial Catalog=ClaimDB";
    //public string str_Select = "select Top1 doc_yr,doc_br, isnull(doc_no,0) +
1'doc_no' from clmrunning where doc_yr = right(year(getdate())+543,2) and doc_br =
'002' and doc_type = 'รจ'";

        OleDbConnection obj_conn = new OleDbConnection(pv_ConStr);
        OleDbDataAdapter obj_da = new OleDbDataAdapter(pv_select, pv_ConStr);
        DataSet obj_ds = new DataSet();
        obj_conn.Open();
        obj_da.Fill(obj_ds);
        obj_conn.Close();
        return obj_ds;
    }


    [WebMethod]
    public DataSet WsUpdClmRunning(string pv_select)
    {
        string pv_ConStr = "Provider=SQLOLEDB;Data Source=.;Integrated
Security=SSPI;Initial Catalog=ClaimDB;";
        OleDbConnection obj_conn = new OleDbConnection(pv_ConStr);
        OleDbCommand obj_comm = new OleDbCommand(pv_select, obj_conn);
        OleDbDataAdapter obj_da = new OleDbDataAdapter(obj_comm);
        DataSet obj_ds = new DataSet();

        obj_conn.Open();
        obj_conn.Close();
        obj_da.Fill(obj_ds);
        return obj_ds;
    }

    [WebMethod]
    public DataSet WsInsert_Gtran(string pv_select)
    {
        string pv_ConStr = "Provider=SQLOLEDB;Data Source=.;UIntegrated
Security=SSPI;Initial Catalog=WsClaimDB;";
        OleDbConnection obj_conn = new OleDbConnection(pv_ConStr);
        OleDbCommand obj_comm = new OleDbCommand(pv_select, obj_conn);
        OleDbDataAdapter obj_da = new OleDbDataAdapter(obj_comm);
        DataSet obj_ds = new DataSet();

        obj_conn.Open();
        obj_conn.Close();
        obj_da.Fill(obj_ds);
        return obj_ds;
    }

    [WebMethod]
    public DataSet WsInsert_NtfClaimDB(string pv_select)
    {
        string pv_ConStr = "Provider=SQLOLEDB;Data Source=.;Integrated
Security=SSPI;Initial Catalog=ClaimDB;";
        OleDbConnection obj_conn = new OleDbConnection(pv_ConStr);
        OleDbCommand obj_comm = new OleDbCommand(pv_select, obj_conn);
        OleDbDataAdapter obj_da = new OleDbDataAdapter(obj_comm);
        DataSet obj_ds = new DataSet();

        obj_conn.Open();
        obj_conn.Close();
        obj_da.Fill(obj_ds);
        return obj_ds;
    }

    public void SetPolicy(Policy policy)
    {
        throw new Exception("The method or operation is not implemented.");
    }

}
```

APPENDIX C

EXAMPLE SOURCE OF WEB APPLICATION

EXAMPLE SOURCE CODE OF WEB APPLICATION

```
    <?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://localhost" 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://localhost"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
- <wsdl:types>
- <s:schema elementFormDefault="qualified" targetNamespace="http://localhost">
- <s:element name="WsChkpolicy">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="str_Select" type="s:string" />
   </s:sequence>
   </s:complexType>
   </s:element>
- <s:element name="WsChkpolicyResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="WsChkpolicyResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
   </s:sequence>
   </s:complexType>
   </s:element>
   </s:sequence>
   </s:complexType>
   </s:element>
- <s:element name="GetReportNotify">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="pv_select" type="s:string" />
   </s:sequence>
   </s:complexType>
   </s:element>
- <s:element name="GetReportNotifyResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="GetReportNotifyResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
    </s:sequence>
    </s:complexType>
    </s:element>
    </s:sequence>
    </s:complexType>
    </s:element>
- <s:element name="RegisterGcode">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="pv_select" type="s:string" />
    </s:sequence>
    </s:complexType>
    </s:element>
- <s:element name="RegisterGcodeResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="RegisterGcodeResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
    </s:sequence>
    </s:complexType>
    </s:element>
    </s:sequence>
    </s:complexType>
    </s:element>
- <s:element name="WSGetMax">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="pv_select" type="s:string" />
    </s:sequence>
    </s:complexType>
    </s:element>
- <s:element name="WSGetMaxResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="WSGetMaxResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
    </s:sequence>
    </s:complexType>
    </s:element>
    </s:sequence>
    </s:complexType>
    </s:element>
- <s:element name="WsUpdClmRunning">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="pv_select" type="s:string" />
    </s:sequence>
    </s:complexType>
    </s:element>
- <s:element name="WsUpdClmRunningResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="WsUpdClmRunningResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
    </s:sequence>
    </s:complexType>
    </s:element>
    </s:sequence>
    </s:complexType>
    </s:element>
- <s:element name="WsInsert_Gtran">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="pv_select" type="s:string" />
    </s:sequence>
    </s:complexType>
    </s:element>
- <s:element name="WsInsert_GtranResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="WsInsert_GtranResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
    </s:sequence>
    </s:complexType>
    </s:element>
    </s:sequence>
    </s:complexType>
    </s:element>
- <s:element name="WsInsert_NtfClaimDB">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="pv_select" type="s:string" />
    </s:sequence>
    </s:complexType>
    </s:element>
- <s:element name="WsInsert_NtfClaimDBResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="WsInsert_NtfClaimDBResult">
- <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="WsChkpolicySoapIn">
  <wsdl:part name="parameters" element="tns:WsChkpolicy" />
    </wsdl:message>
- <wsdl:message name="WsChkpolicySoapOut">
  <wsdl:part name="parameters" element="tns:WsChkpolicyResponse" />
    </wsdl:message>
- <wsdl:message name="GetReportNotifySoapIn">
  <wsdl:part name="parameters" element="tns:GetReportNotify" />
    </wsdl:message>
- <wsdl:message name="GetReportNotifySoapOut">
  <wsdl:part name="parameters" element="tns:GetReportNotifyResponse" />
    </wsdl:message>
- <wsdl:message name="RegisterGcodeSoapIn">
  <wsdl:part name="parameters" element="tns:RegisterGcode" />
    </wsdl:message>
- <wsdl:message name="RegisterGcodeSoapOut">
  <wsdl:part name="parameters" element="tns:RegisterGcodeResponse" />
    </wsdl:message>
- <wsdl:message name="WSGetMaxSoapIn">
  <wsdl:part name="parameters" element="tns:WSGetMax" />
    </wsdl:message>
- <wsdl:message name="WSGetMaxSoapOut">
  <wsdl:part name="parameters" element="tns:WSGetMaxResponse" />
    </wsdl:message>
- <wsdl:message name="WsUpdClmRunningSoapIn">
```

```
 <wsdl:part name="parameters" element="tns:WsUpdClmRunning" />
   </wsdl:message>
- <wsdl:message name="WsUpdClmRunningSoapOut">
 <wsdl:part name="parameters" element="tns:WsUpdClmRunningResponse" />
   </wsdl:message>
- <wsdl:message name="WsInsert_GtranSoapIn">
 <wsdl:part name="parameters" element="tns:WsInsert_Gtran" />
   </wsdl:message>
- <wsdl:message name="WsInsert_GtranSoapOut">
 <wsdl:part name="parameters" element="tns:WsInsert_GtranResponse" />
   </wsdl:message>
- <wsdl:message name="WsInsert_NtfClaimDBSoapIn">
 <wsdl:part name="parameters" element="tns:WsInsert_NtfClaimDB" />
   </wsdl:message>
- <wsdl:message name="WsInsert_NtfClaimDBSoapOut">
 <wsdl:part name="parameters" element="tns:WsInsert_NtfClaimDBResponse" />
   </wsdl:message>
- <wsdl:portType name="WebService_ClaimNotifySoap">
- <wsdl:operation name="WsChkpolicy">
 <wsdl:input message="tns:WsChkpolicySoapIn" />
 <wsdl:output message="tns:WsChkpolicySoapOut" />
   </wsdl:operation>
- <wsdl:operation name="GetReportNotify">
 <wsdl:input message="tns:GetReportNotifySoapIn" />
 <wsdl:output message="tns:GetReportNotifySoapOut" />
   </wsdl:operation>
- <wsdl:operation name="RegisterGcode">
 <wsdl:input message="tns:RegisterGcodeSoapIn" />
 <wsdl:output message="tns:RegisterGcodeSoapOut" />
   </wsdl:operation>
- <wsdl:operation name="WSGetMax">
 <wsdl:input message="tns:WSGetMaxSoapIn" />
 <wsdl:output message="tns:WSGetMaxSoapOut" />
   </wsdl:operation>
- <wsdl:operation name="WsUpdClmRunning">
 <wsdl:input message="tns:WsUpdClmRunningSoapIn" />
 <wsdl:output message="tns:WsUpdClmRunningSoapOut" />
   </wsdl:operation>
- <wsdl:operation name="WsInsert_Gtran">
 <wsdl:input message="tns:WsInsert_GtranSoapIn" />
 <wsdl:output message="tns:WsInsert_GtranSoapOut" />
   </wsdl:operation>
- <wsdl:operation name="WsInsert_NtfClaimDB">
 <wsdl:input message="tns:WsInsert_NtfClaimDBSoapIn" />
 <wsdl:output message="tns:WsInsert_NtfClaimDBSoapOut" />
   </wsdl:operation>
   </wsdl:portType>
```

```
- <wsdl:binding name="WebService_ClaimNotifySoap"
      type="tns:WebService_ClaimNotifySoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="WsChkpolicy">
  <soap:operation soapAction="http://localhost/WsChkpolicy" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="GetReportNotify">
  <soap:operation soapAction="http://localhost/GetReportNotify" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="RegisterGcode">
  <soap:operation soapAction="http://localhost/RegisterGcode" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="WSGetMax">
  <soap:operation soapAction="http://localhost/WSGetMax" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="WsUpdClmRunning">
  <soap:operation soapAction="http://localhost/WsUpdClmRunning" style="document"
      />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
```

```
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="WsInsert_Gtran">
  <soap:operation soapAction="http://localhost/WsInsert_Gtran" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="WsInsert_NtfClaimDB">
  <soap:operation soapAction="http://localhost/WsInsert_NtfClaimDB"
      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="WebService_ClaimNotifySoap12"
      type="tns:WebService_ClaimNotifySoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="WsChkpolicy">
  <soap12:operation soapAction="http://localhost/WsChkpolicy" style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="GetReportNotify">
  <soap12:operation soapAction="http://localhost/GetReportNotify" style="document"
      />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="RegisterGcode">
  <soap12:operation soapAction="http://localhost/RegisterGcode" style="document" />
- <wsdl:input>
```

```
  <soap12:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="WSGetMax">
  <soap12:operation soapAction="http://localhost/WSGetMax" style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="WsUpdClmRunning">
  <soap12:operation soapAction="http://localhost/WsUpdClmRunning"
     style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="WsInsert_Gtran">
  <soap12:operation soapAction="http://localhost/WsInsert_Gtran" style="document"
     />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="WsInsert_NtfClaimDB">
  <soap12:operation soapAction="http://localhost/WsInsert_NtfClaimDB"
     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="WebService_ClaimNotify">
- <wsdl:port name="WebService_ClaimNotifySoap"
      binding="tns:WebService_ClaimNotifySoap">
  <soap:address location="http://localhost/WSEclaim/WebService_ClaimNotify.asmx"
      />
    </wsdl:port>
- <wsdl:port name="WebService_ClaimNotifySoap12"
      binding="tns:WebService_ClaimNotifySoap12">
  <soap12:address
      location="http://localhost/WSEclaim/WebService_ClaimNotify.asmx" />
    </wsdl:port>
    </wsdl:service>
    </wsdl:definitions>
```

CURRICULUM VITAE

NAME	Miss Sakaowrath Chatjaroenchap
DATE OF BIRTH	August 27, 1974
EDUCATION	
Bachelor Degree	Bachelor of Computer Science from Suan Sunandha Rajabhat University 1998
WORK EXPERIENCES	1995 – Present Senior Programmer in MIS Department. The Viriyah Insurance Company.