

**DEVELOPMENT OF A CURRICULUM INFORMATION
RETRIEVAL SYSTEM USING WEB SERVICE
TECHNOLOGY: A CASE STUDY OF
THE NATIONAL CURRICULUM DATABASE**



POJ THAMMATHIWAT

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
(TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2011**

COPYRIGHT OF MAHIDOL UNIVERSITY

Copyright by Mahidol University

Thesis
entitled

**DEVELOPMENT OF A CURRICULUM INFORMATION
RETRIEVAL SYSTEM USING WEB SERVICE
TECHNOLOGY: A CASE STUDY OF
THE NATIONAL CURRICULUM DATABASE**

Poj Thammathiwat

.....
Mr. Poj Thammathiwat
Candidate

M. Chulasamaya

.....
Assoc. Prof. Montree Chulasamaya,
Ph.D., M.D.
Major advisor

Warakorn Charoensuk

.....
Asst. Prof. Warakorn Charoensuk, Ph.D.
Co-advisor

B. Mahaisavariya

.....
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

Rawin Raviwongse

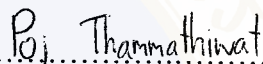
.....
Asst. Prof. Rawin Raviwongse, Ph.D.
Program Director
Master of Science Program in
Technology of Information System
Management
Faculty of Engineering,
Mahidol University

Thesis
entitled


**DEVELOPMENT OF A CURRICULUM INFORMATION
RETRIEVAL SYSTEM USING WEB SERVICE
TECHNOLOGY: A CASE STUDY OF
THE NATIONAL CURRICULUM DATABASE**


was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Science
(Technology of Information System Management)

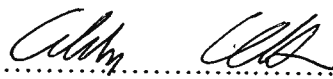
on
May 9, 2011

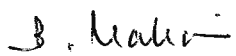

.....
Mr. Poj Thammathiwat
Candidate



.....
Assoc. Prof. Panya Kaimuk, M.D.
Chair


.....
Assoc. Prof. Montree Chulasamaya,
Ph.D., M.D.
Member


.....
Lect. Worasit Choochaiwattana, Ph.D.
Member


.....
Asst. Prof. Warakorn Charoensuk, Ph.D.
Member


.....
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedies
Dean
Faculty of Graduate Studies
Mahidol University


.....
Asst. Prof. Rawin Raviwongse, Ph.D.
Dean
Faculty of Engineering
Mahidol University

ACKNOWLEDGEMENTS

I would like to express my profound gratitude and appreciation to Asst. Prof. Thanakorn Uan-on who is my major advisor for his assistance leading to meet with success of this thesis.

I would like to express my gratitude to Assoc. Prof. Panya Kaimuk, Assoc. Prof. Montree Chulasamaya and Asst. Prof. Warakorn Charoensuk. I profoundly appreciate them for their valuable advices and comments. I also appreciate to Dr. Worasit Choochaiwattana, who was the external examiner of the thesis, for his .

I would like thank to Mr. Patiwat Wisedsukol for his knowledge of Service Oriented Architecture (SOA) and Representational State Transfer (REST) web service. My appreciation is also to my friends for their kind supports and suggestions, which made me feel that every time we joined a class was such a great time.

Last but not least, I am grateful to my parents for their support, care, encouragement and love.

Poj Thammathiwat

**DEVELOPMENT OF A CURRICULUM INFORMATION RETRIEVAL SYSTEM
USING WEB SERVICE TECHNOLOGY: A CASE STUDY OF THE NATIONAL
CURRICULUM DATABASE**

POJ THAMMATHIWAT 4936775 EGTI / M

M.Sc. (TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)

**THESIS ADVISORY COMMITTEE: MONTREE CHULASAMAYA, Ph.D., M.D.,
WARAKORN, Ph.D.**

ABSTRACT

Curriculum information is an important factor for students making decisions about learning at a higher level. Curriculum information is often presented by individual universities and there is not a centralized service that provides this information. This means that it often takes a long time for students to get the required curriculum information.

The objective of this research was to develop a curriculum information retrieval system. The system can be used as a one-stop service for providing curriculum information and was designed with web service technology. The system was designed using a relational database, and was developed using PHP, HTML, XML for language programming, and MySQL as a database management system.

A prototype of a search engine to retrieve graduate curriculum information was developed. The system is divided into three sections; curriculum searching, web service API management, and curriculum information management. The authority for this application is classified to three levels; administrator, staff, and general user. The results of system testing show the searching results of this system are quite accurate and the system runs efficiently overall.

**KEY WORDS: CURRICULUM INFORMATION RETRIEVAL/ WEB SERVICE/
REPRESENTATIONAL STATE TRANSFER/ SEARCH ENGINE**

86 pages

การพัฒนาาระบบสืบค้นข้อมูลหลักสูตรการศึกษาโดยเทคโนโลยีเว็บเซอร์วิส กรณีศึกษาระบบ
ฐานข้อมูลหลักสูตรแห่งชาติ

DEVELOPMENT OF A CURRICULUM INFORMATION RETRIEVAL SYSTEM USING
WEB SERVICE TECHNOLOGY: A CASE STUDY OF THE NATIONAL CURRICULUM
DATABASE

พจน์ ธรรมาธิวัฒน์ 4936775 EGTI/M

วท.ม. (เทคโนโลยีการจัดการระบบสารสนเทศ)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์: มนต์รี จุลสมัย, Ph.D., M.D., วรากร เจริญสุข, Ph.D.

บทคัดย่อ

ข้อมูลหลักสูตรเป็นสิ่งสำคัญต่อการตัดสินใจศึกษาต่อของนักศึกษา ในปัจจุบันมหาวิทยาลัย
เริ่มเปิดการเรียนการสอนในระดับบัณฑิตศึกษามากขึ้นการค้นหาข้อมูลหลักสูตรที่ต้องการจึงใช้เวลามาก
เนื่องจากนักศึกษาต้องค้นหาข้อมูลจากหลายแหล่งข้อมูลผ่านทางเว็บไซต์ของมหาวิทยาลัยต่าง ๆ

งานวิจัยนี้จึงมีจุดประสงค์เพื่อพัฒนาระบบค้นหาข้อมูลหลักสูตรระดับบัณฑิตศึกษา ซึ่งจะ
รวบรวมข้อมูลหลักสูตรระดับบัณฑิตศึกษาไว้ในที่เดียวหรือเป็นการให้บริการแบบ One Stop Service โดย
การพัฒนาาระบบสืบค้นข้อมูลหลักสูตรการศึกษาโดยเทคโนโลยีเว็บเซอร์วิส (Web Service) ใช้หลักการ
ออกแบบของฐานข้อมูลเชิงสัมพันธ์ พัฒนาระบบโดยใช้ภาษาพีเอชพี, ภาษาเอชทีเอ็มแอล และภาษาเอ็กซ์
เอ็มแอล ใช้ MySQL ในระบบการจัดการฐานข้อมูล

ผลที่ได้จากการพัฒนาระบบนั้นจะได้ต้นแบบของระบบค้นหาข้อมูลหลักสูตรระดับบัณฑิต
ศึกษา โดยแบ่งระบบออกเป็นสามส่วนได้แก่ ระบบค้นหาข้อมูลหลักสูตร, ระบบจัดการ Web Service API
และระบบจัดการฐานข้อมูลหลักสูตร โดยการกำหนดสิทธิ์การเข้าถึง สิทธิ์ในการเข้าใช้ระบบแบ่งออกเป็น
สามระดับ คือ ผู้ดูแลระบบ, เจ้าหน้าที่ที่ได้รับสิทธิ์จากมหาวิทยาลัยและผู้ใช้ จากการทดสอบระบบ ระบบที่
พัฒนาขึ้นมีประสิทธิภาพโดยรวมอยู่ในระดับดี และสามารถนำไปใช้งานได้อย่างมีประสิทธิภาพ

86 หน้า

CONTENTS

	Page
ACKNOWLEDGEMENTS	iii
ABSTRACT (ENGLISH)	iv
ABSTRACT (THAI)	v
LIST OF TABLES	viii
LIST OF FIGURES	x
CHAPTER I INTRODUCTION	1
1.1 Background and Statements of Problems	1
1.2 Objectives	2
1.3 Scopes of Work	2
CHAPTER II LITERATURE REVIEW	3
2.1 Information and Information System Concept	3
2.2 Service Oriented Architecture (SOA)	4
2.3 Web Service	9
2.4 Representational State Transfer (REST)	9
2.5 Extensible Markup Language (XML)	13
2.6 Related Researches	15
CHAPTER III MATERIALS AND METHODS	18
3.1 Step of Research Methodology	18
3.1.1 Preliminary Investigation	19
3.1.2 Determination of System Requirements	19
3.1.3 System Analysis and Design	20
3.1.4 Development	22
3.1.5 Testing	22
3.2 Materials	22
3.2.1 Hardware	22
3.2.2 Tool and Software	22

CONTENTS (cont.)

	Page
3.3 Research Schedule	23
CHAPTER IV SYSTEM IMPLEMENTATION AND RESULTS	24
4.1 Use Case Diagram	24
4.2 Sequence Diagram	32
4.3 Database Design	34
4.4 The Results of System Development	36
4.4.1 Curriculum search system	36
4.4.2 REST API	40
4.4.3 Service Management	42
4.5 Testing	51
CHAPTER V DISCUSSIONS	62
5.1 The Advantage of System	63
5.2 Limitation of System	63
CHAPTER VI CONCLUSIONS AND RECOMMENDATIONS	65
6.1 Conclusions	65
6.2 Recommendations	67
REFERENCES	68
APPENDICES	70
Appendix A DATA DICTIONARY	71
Appendix B TESTING WEB SERVICE API	81
BIOGRAPHY	86

LIST OF TABLES

Table	Page
2.1 Relationships between SQL and HTTP Verbs	11
3.1 Research Schedule	23
4.1 Description of Search use case	25
4.2 Description of View information use case	25
4.3 Description of View XML use case	26
4.4 Description of View XML Schema use case	27
4.5 Description of Test use case	27
4.6 Description of Manage Data use case	28
4.7 Description of Manage User use case	29
4.8 Description of Manage Reference Data use case	30
4.9 Description of Manage Web Service use case	31
4.10 Test program	51
4.11 Test ref-inter-type	52
4.12 Test curriculum	53
4.13 Test ref-univ	54
4.14 Test ref-fac	55
4.15 Test ref-lev	56
4.16 Test ref-curr	57
4.17 Test curr-type	58
4.18 Test ref-study-time	59
4.19 Test subject	60
4.20 Test ref-iscsd	61
A.1 Table UOC_CURR	71
A.2 Table REF_UNIV	72

LIST OF TABLES (cont.)

Table	Page
A.3 Table REF_UNIV_ADDRESS	72
A.4 Table REF_CURR	73
A.5 Table REF_PROGRAM	73
A.6 Table REF_ISCED	73
A.7 Table REF_LEV	73
A.8 Table REF_CURR_TYPE	74
A.9 Table REF_FAC	74
A.10 Table REF_STUDY_TYPE	74
A.11 Table REF_STUDY_TIME	74
A.12 Table REF_CURR_INTER	74
A.13 Table REF_SUBJECT	75
A.14 Table REF_SUBJECT_TYPE	75
A.15 Table API_REQUEST_LOG	75
A.16 Table API_PROFILE	76
A.17 Table API_SCHEMA	77
A.18 Table SYS_APPLICATION_LOG	78
A.19 Table USER	78
A.20 Table GROUP	79
A.21 Table RESOURCE	79
A.22 Table GROUP_RIGHT	80

LIST OF FIGURES

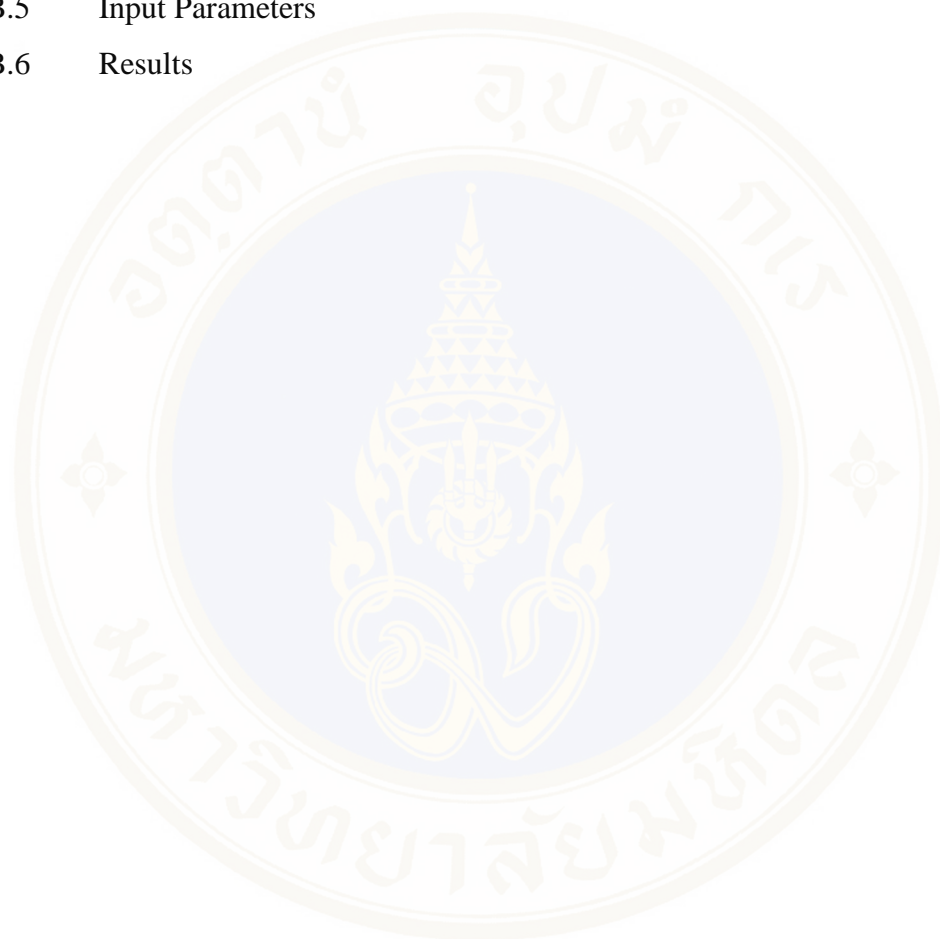
Figures	Page
2.1 Information system model	3
2.2 Three business processes within one company duplicating functionality	5
2.3 Three business processes repurposing one service for common tasks	6
2.4 Service-Oriented Architecture	7
2.5 REST-style Workflow	10
3.1 Step of Waterfall Model	18
3.2 System architecture	21
4.1 Use case diagrams: Search	24
4.2 Use case diagrams: Rest API	26
4.3 Use case diagrams: Curriculum Management System	28
4.4 UML Sequence Diagram: Search	32
4.5 UML Sequence Diagram: View	33
4.6 ER-Diagram: Curriculum	34
4.7 ER-Diagram: Web Service API	35
4.8 ER-Diagram: User	35
4.9 Main Page	36
4.10 Search by Keyword	37
4.11 Search by Filters	37
4.12 Search Result	37
4.13 Flow Chart Search by Keyword	38
4.14 Detail of curriculum	39
4.15 University Map	39
4.16 Detail of subject	40
4.17 Rest API	41

LIST OF FIGURES (cont.)

Figures	Page
4.18 XML	41
4.19 Service Attribute	42
4.20 Test API	42
4.21 Log in	43
4.22 Main Page of Authorized user	43
4.23 Edit University Data	44
4.24 Edit Curriculum Data	44
4.25 Main Page of Administrator	45
4.26 University management	45
4.27 Curriculum management	46
4.28 Service API	46
4.29 Request Log	47
4.30 Manage Reference Data	47
4.31 User Account Management	48
4.32 Create New User	48
4.33 Edit User	49
4.34 User Group Management	49
4.35 Edit User Group	50
4.36 Group Permissions	50
4.37 System AppLog	50
5.1 Web service vs. Web application	64
B.1 Menu SoapUI	81
B.2 New soapUI Project	82
B.3 New REST Service	83
B.4 New REST Resource	83

LIST OF FIGURES (cont.)

Figures	Page
B.5 Input Parameters	84
B.6 Results	85



CHAPTER I

INTRODUCTION

1.1 Background and Statements of Problems

The Office of the Higher Education Commission (OHEC) is under directed of the Higher Education Commission. The Higher Education Commission has the authority to propose higher education policies and educational standards that must be corresponding to the National Economic and Social Development Plan and the National Education Plan. OHEC responsibilities on education are composed of analysis in the education directions, research problems on development higher education, development on education policies and educational standard. OHEC focuses on the further education information. These was information are scattered. It causes of inconvenience for students that are require the information of higher education.

The Office of the Higher Education Commission oversees altogether 166 higher education institutions. They are 13 limited admission public universities, 2 open admission universities, 13 autonomous universities; 69 private higher education institutions; 40 Rajabhat Universities; 9 Rajamangala Universities of Technology; Pathumwan Institute of Technology and 19 Community Colleges. Education information has been presented on different web site in each of the universities.

At present, the data have been collecting into database consisting of the based database such as curriculum data, student data and staff data. However, those data was not utilization. The application of information retrieval system for national curriculum can reduce the time for searching the further education data and more comfortable to student or who decided additional education by finding on one site.

Nowadays, there is various desire of collegian to search for the data to further education and use too much time to looking for resources of education data in decision support because they have no aggregating additional education data on one site.

The information retrieval system for national curriculum has been implemented Web Service approach in this research. In part of searching, Information Retrieval approach is used in query process. The system is consisting of services as transmitting data, searching data, and editing data. Moreover, Information Retrieval approach is used to support the process of curriculum comparison.

This research proposed the development of information searching system for curriculum database of The Office of the Higher Education Commission. The system could provide curriculum information by web service. The web service is popular used in data service providing that through network or internet with no platform. In addition, there is secure in data transmission of web service because user can access only exist service section.

1.2 Objectives

Main objectives of the proposed system are as follows:

- 1.2.1 To develop prototype of website for publicize the post graduate curriculums.
- 1.2.2 To develop web service for publicize the postgraduate curriculums.
- 1.2.3 To develop web service database.
- 1.2.4 To apply search engine for search data form CHE database.

1.3 Scopes of Work

The scope of this work will;

- 1.3.1 Analyze, design information retrieval system from CHE database.
- 1.3.2 To develop the web information retrieval system from national curriculum. PHP is tool for development web application and use MySQL for develop web server.

CHAPTER II

LITERATURE REVIEW

2.1 Information and Information System Concept[1]

Data are facts and figures that are relatively meaningless to users. Data are the raw material of information that is transformed into information by an information process.

Information is a collection of facts which is arranged, and processed by sorting, classifying, merging, computing or summarizing in such a way that it has additional value beyond the value of the facts themselves. They are simply useful data via used in the decision making process. For instance, sales forecasting data can be processed to produce information that helps determine production strategies.

System is a set of elements or components that interact to accomplish goals. The element themselves and the relationships among them determine how system work. System has inputs, processing mechanisms, outputs and feedback.

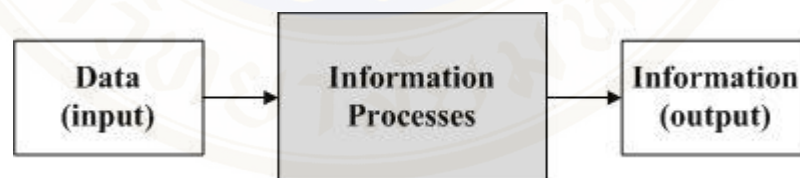


Figure 2.1 Information system model

In figure 2.1, Information System (IS) is set of interrelated elements or components that collect (input), manipulate, store (process) and disseminate (output) data or information and provide a feedback mechanism. IS designed for operations managers are valuable if they can provide information to help in controlling operations in a timely.

2.2 Service Oriented Architecture (SOA)[2]

Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains and implemented using various technology stacks. In general, entities (people and organizations) create capabilities to solve or support a solution for the problems they face in the course of their business. It is natural to think of one person's needs being met by capabilities offered by someone else; or, in the world of distributed computing, one computer agent's requirements being met by a computer agent belonging to a different owner. The term owner here may be used to denote different divisions of one business or perhaps unrelated entities in different countries.

There is not necessarily a one-to-one correlation between needs and capabilities; the granularity of needs and capabilities vary from fundamental to complex, and any given need may require a combination of numerous capabilities while any single capability may address more than one need. One perceived value of SOA is that it provides a powerful framework for matching needs and capabilities and for combining capabilities to address those needs by leveraging other capabilities. One capability may be repurposed across a multitude of needs.

SOA is a "view" of architecture that focuses in on services as the action boundaries between the needs and capabilities in a manner conducive to service discovery and repurposing.

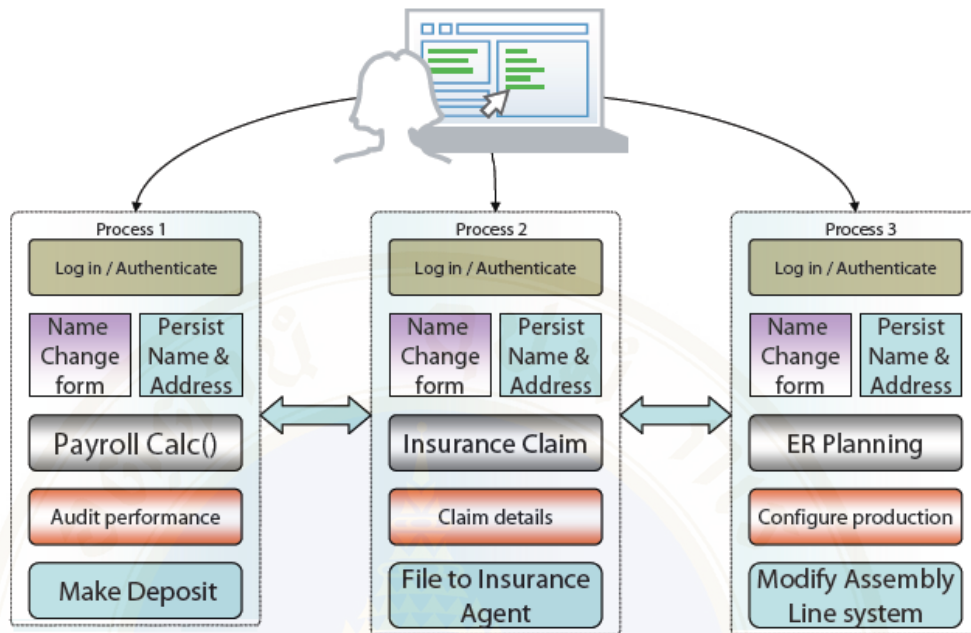


Figure 2.2 Three business processes within one company duplicating functionality

Figure 2.2 shows an example of an information system scenario that could benefit from a migration to SOA. Within one organization, three separate business processes use the same functionality, each encapsulating it within an application. In this scenario, the login function, the ability to change the user name, and the ability to persist it are common tasks implemented redundantly in all three processes. This is a suboptimal situation because the company has paid to implement the same basic functionality three times.

Moreover, such scenarios are highly inefficient and introduce maintenance complexity within IT infrastructures. For example, consider an implementation in which the state of a user is not synchronized across all three processes. In this environment users might have to remember multiple login username/password tokens and manage changes to their profiles in three separate areas. Additionally, if a manager wanted to deny a user access to all three processes, it is likely that three different procedures would be required (one for each of the applications). Corporate IT workers managing such a system would be effectively tripling their work –and spending more for software and hardware systems.

In a more efficient scenario, common tasks would be shared across all three processes. This can be implemented by decoupling the functionality from each process or application and building a standalone authentication and user management application that can be accessed as a service. In such a scenario, the service itself can be repurposed across multiple processes and applications and the company owning it only has to maintain the functionality in one central place. This would be a simple example of Service Oriented Architecture in practice. The resultant IT infrastructure would resemble Figure 2.3.

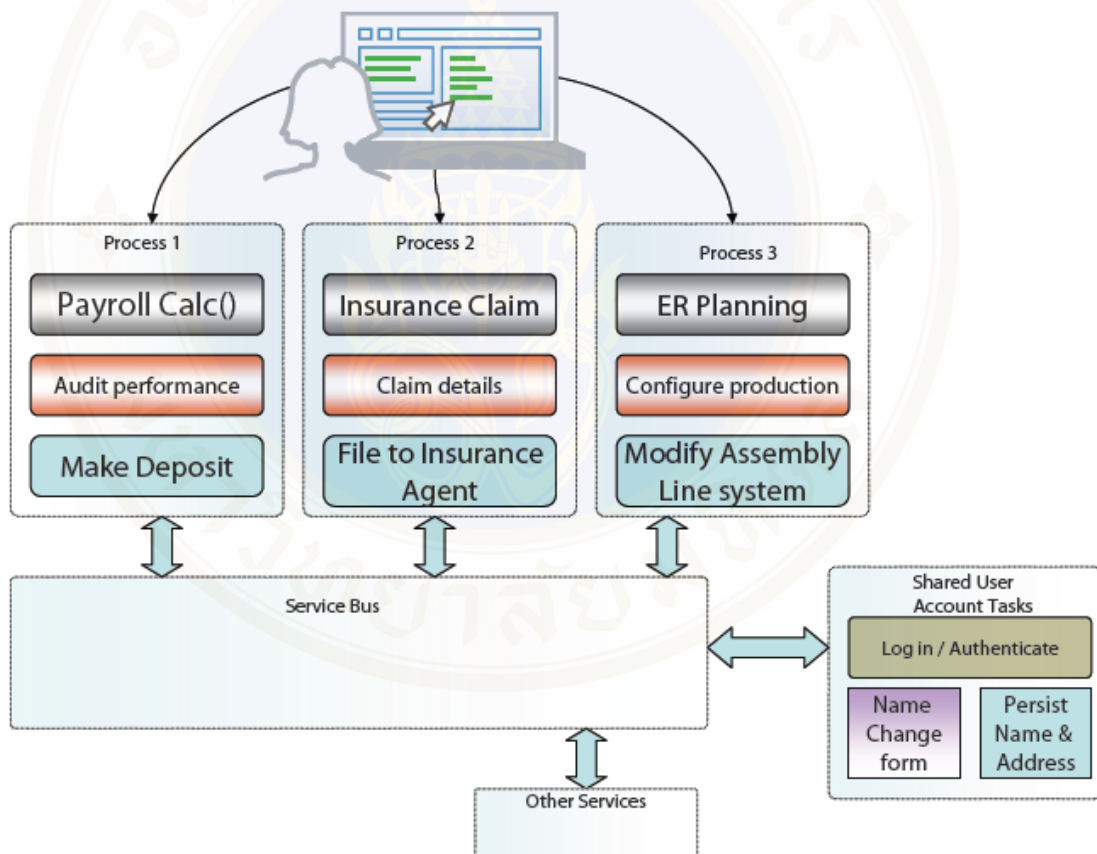


Figure 2.3 Three business processes repurposing one service for common tasks.

In figure 2.3, the shared user account tasks have been separated from each process and implemented in a way that enables other processes to call them as a service. This allows the shared functions to be repurposed across all three processes. The common service bus is really a virtual environment whereby services are made available to all potential consumers on a fabric. This is typically referred to as an

Enterprise Service Bus (ESB) and has a collection of specialized subcomponents including naming and lookup directories, registry-repositories, and service provider interfaces (for connecting capabilities and integrating systems) as well as a standardized collection of standards and protocols to make communications seamless across all connected devices. Advanced ESB vendors have tools that can aggregate services into complex processes and workflows.



Figure 2.4 Service-Oriented Architecture

SOA is composed of a service provider, service requester, or the service consumer and an optional service directory, which together are leveraged to deliver services using application-to-application messaging for information exchange. The service provider creates a service and publishes descriptive information about the service in the service directory. The service requester queries the descriptive information in the service directory to locate the service and also gathers information about the service and the service provider. The service requester accesses the service hosted in the service provider infrastructure and, in turn, realizes the business value of leveraging the service.

The specific architectural principles for design and service definition focus on specific themes that influence the intrinsic behavior of a system and the style of its design:

- Service encapsulation

Many services are consolidated for use under the SOA. Often such services were not planned to be under SOA.

- Service loose coupling

Services maintain a relationship that minimizes dependencies and only requires that they maintain an awareness of each other.

- Service contract

Services adhere to a communications agreement, as defined collectively by one or more service-description documents.

- Service abstraction

Beyond descriptions in the service contract, services hide logic from the outside world.

- Service reusability

Logic is divided into services with the intention of promoting reuse.

- Service composability

Collections of services can be coordinated and assembled to form composite services.

- Service autonomy

Services have control over the logic they encapsulate.

- Service optimization

All else equal, high-quality services are generally preferable to low-quality ones.

- Service discoverability

Services are designed to be outwardly descriptive so that they can be found and accessed via available discovery mechanisms.

- Service relevance

Functionality is presented at a granularity recognized by the user as a meaningful service.

2.3 Web Service [3],[4]

A Web service is once of remote procedure call (RPC) like technology. RPC is a concept focused on how client object call method on remote object. A Web service is defined by the W3C as "a software system designed to support interoperable machine-to-machine interaction over a network". Web services are frequently just Web application programming interfaces (API) that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.

The most widely accepted and successful type of service is the XML Web service. XML Web service are a Web-based implementation of remote procedure call (RPC). There are two popular of XML Web service are Simple Object Access Protocol (SOAP) and Representational state transfer (REST) Web Services.

REST was chosen for developing web service of this thesis because of its simplicity in development and its suitability in resource oriented architecture design, which requires intensively more resource-oriented then service-oriented.

2.4 Representational State Transfer (REST) [5],[6]

Representational State Transfer (REST) is an architectural style described by Roy Fielding [7]. REST is in many ways a retrospective abstracting of the principles that make the World Wide Web scalable. Roy Fielding argues that unlike transactional systems, in which servers need to maintain complex sessions with clients, the web reduces the server's obligation that arises from a self-contained question to a simple response. In many cases, the response involves sending a pre-computed set of data, the representational state, across the network. The client can essentially navigate across a wide range of existing, pre-computed resources by following links from resource to resource. There are many objects, each identified by a URI, and very few methods (PUT, POST, GET, and DELETE in HTTP 1.0, while HTTP 1.1 allows extensions). In REST, each request sent to an object results in the transfer of a representation of this object (typically in form of an XML document). This document provides the client with the opportunity to change the state of the object by navigating to a different URI which is referenced in the document. The

documents are self-contained and can be transformed through intermediaries, for example proxies.

Transferred to the area of cross-organizational workflow, a REST-style workflow consists of a process factory object, which is referenced by a URI. A request to the process factory results in the creation of a process instance, which in turn can be referenced using a URI. In its purest form, a HTTP GET command sent to either of these URIs results in the return of an XML document. In the case of the process factory, the XML document may contain the properties of the process model represented by the process factory (for example quality of service information), whereas the XML document returned by the process instance may contain information about the current state of the process instance. In essence, every component of a cross-organizational process is identified by an individual URI, that is, each externally accessible process, activity, or operation is identified by a URI.

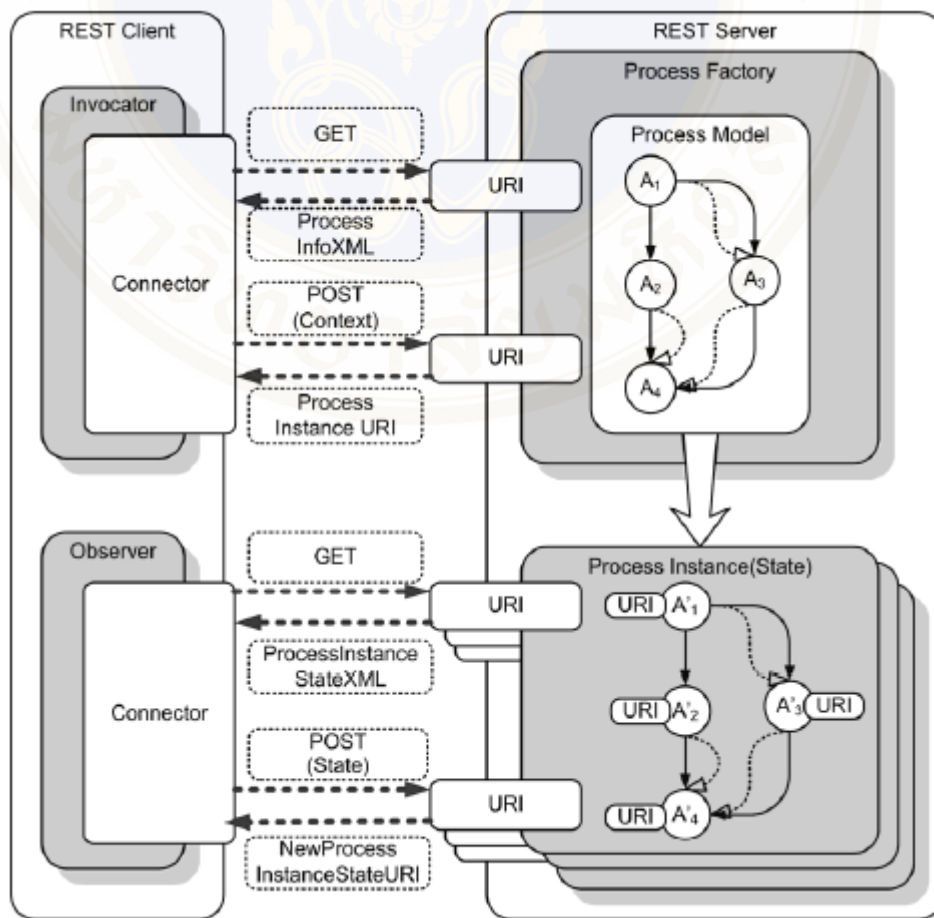


Figure 2.5 REST-style Workflow

Figure 2.5 shows an example for a pure REST workflow application. All the client needs to know about the remote process is the address of this process in form of a URI, which is typically a combination of the machine name, a distinct object type and a process ID. A GET request sent to this URI results in an XML document that gives more information about the process, for example the URI where the process can be invoked. The client then sends a POST operation to the process invocation URI which contains the context data (payload) for the process instance. The process factory creates a process instance and returns the URI of the process instance to the invocator. The subsequent interaction between client and process instance is also characterized through the use of GET and POST commands, but in this case the POST command is used to manipulate the state of the process instance, until it is completed.

REST is a key design idiom that embraces a stateless client-server architecture in which the web services are viewed as resources and can be identified by their URLs. Web service clients that want to use these resources access a particular representation by transferring application content using a small globally defined set of remote methods that describe the action to be performed on the resource. REST is an analytical description of the existing web architecture, and thus the interplay between the style and the underlying HTTP protocol appears seamless.

The HTTP methods such as GET and POST are the verbs that the developer can use to describe the necessary create, read, update, and delete (CRUD) actions to be performed. Some may see an analogy to operations in SQL, which also relies on a few common verbs, as shown in Table. However, the REST style and HTTP protocol are mutually exclusive, and REST does not require HTTP.

Table 2.1 Relationships between SQL and HTTP Verbs

Action	SQL	HTTP
Create	Insert	PUT
Read	Select	GET
Update	Update	POST
Delete	Delete	DELETE

REST refers in the strictest sense to a collection of network architecture principles which outline how resources are defined and addressed. The term is often used more loosely to describe any simple interface which transmits domain-specific data over HTTP without an additional messaging layer such as SOAP or session tracking via HTTP cookies. These two meanings can conflict as well as overlap. It is possible to design a software system in accordance with Fielding's REST architectural style without using HTTP and without interacting with the World Wide Web. It is also possible to design simple XML+HTTP interfaces, which do not conform to REST principles, and instead follow a model of remote procedure call. The difference between the uses of the term "REST" therefore causes some confusion in technical discussions. The following table shows how the HTTP verbs are typically used to implement a web service.

The characteristics of REST are

- Client-Server: a pull-based interaction style: consuming components pull representations.
- Stateless: each request from client to server must contain all the information necessary to understand the request, and cannot take advantage of any stored context on the server.
- Cache: to improve network efficiency responses must be capable of being labeled as cacheable or non-cacheable.
- Uniform interface: all resources are accessed with a generic interface (e.g., HTTP GET, POST, PUT, DELETE).
- Named resources - the system is comprised of resources which are named using a URL.
- Interconnected resource representations - the representations of the resources are interconnected using URLs, thereby enabling a client to progress from one state to another.
- Layered components - intermediaries, such as proxy servers, cache servers, gateways, etc, can be inserted between clients and resources to support performance, security, etc.

2.5 Extensible Markup Language (XML) [8]

XML is an extensible markup language used for the description and delivery of marked-up electronic text over the Web. Important characteristics of XML are its emphasis on descriptive rather than prescriptive (or procedural) markup, its document type concept, its extensibility, and its portability. In XML, the instructions needed to process a document for some specific purpose, e.g., to format it, are sharply distinguished from the descriptive markup, which occurs within the actual XML document. With descriptive instead of procedural markup the same document can readily be processed in many different ways, using only those parts of it that are considered relevant.

An important aspect of XML is its notion of a document type. XML documents are regarded as having types. Its constituent parts and their structure formally define the type of a document. XML Schemas describes the elements and attributes that may be contained in a schema-conforming document and the ways that the elements may be arranged within a document structure. Schemas are more powerful when validating an XML document because of their ability to clarify data types stored within the XML document.

XML can be perceived as a dynamic trading language that enables diverse applications to exchange information flexibly and cost-effectively. XML allows the inclusion of tags pertinent to the contextual meaning of data. These tags make possible precise machine interpretation of data, fine tuning the entire process of information exchange between trading enterprises. In addition the ability of the XML schemas to reuse and refine the data model of other schema architectures enables reuse and extension of components, reduces the development cycle, and promotes interoperability.

As XML can be used to encode complex business information it is ideally suited to support open standards, which are essential to allow rapid establishment of business information exchange and interoperability. For example, XML is well suited to transactional processing in a heterogeneous, asynchronous, open, and distributed architecture that is built upon open standard technologies, such as parsers and interfaces. This applies equally well to enterprise application integration as well as to the e-business style of integration with trading partners.

The ability of XML to model complex data structures combined with the additional ability of XML Schema to reuse and refine the data model of other schema architectures enables reuse and extension of components, reduces the development cycle and promotes interoperability. It is precisely the suit of technologies that are grouped under XML that had profound influence on the development of Web Services technologies and provide the fundamental building blocks for Web Services and service-oriented architectures.

This research use XML and web service as an exchange protocol. The reason was it has most flexibility and high efficiency and suitable to use as fundamental for further development. And XML can use as standard mechanism to promote all institution to use data in form of interoperability independence from sort of data and software or hardware.

A good point for learning XML is the HTML language. HTML is used to describe how information displays on a web browser. However, HTML is a document centric that lack of information quality for information sharing because often, some element contains both text and sub element. XML could solve that problem.

XML could solve the data heterogeneous problem because XML Document is text file and most of operation system can manipulate text file.

Although XML allows to create your own tag, but W3C created the standard rules for create XML document for example:

- XML Document has to consists of two parts are prolog sometime. called XML declaration and root element.
- XML document must have only one root element.
- The root element can contains all other sub element, content, and comment.
- Element must be well nested.

If any XML document follows the XML rules, It can be called “well-formed document” and it easy to manipulation by any application. XML document can be manipulated using tree-based, event-based, or class-based interface. This research used tree-based. The W3C provide a standard tree-base API called Document Object Model (DOM). DOM expresses an XML document using hierarchical tree view. Each branch of tree represents an element in the hierarchy.

DOM is easy to learn but it will be low performance when processing larger sized document.

2.6 Related Researches

Michael zur Muehlen, Jeffrey V. Nickerson and Keith D. Swenson present a case study of development of standards in the area of cross-organizational workflows based on web services on topic “Developing Web Services Choreography Standards – The Case of REST vs. SOAP”[6]. They discuss two opposing types of standards: those based on SOAP, with tightly coupled designs similar to remote procedure calls, and those based on REST, with loosely coupled designs similar to the navigating of web links. We illustrate the standardization process, clarify the technical underpinnings of the conflict, and analyze the interests of stakeholders. The decision criteria for each group of stakeholders are discussed. Finally, they present implications for both the workflow and the wider Internet communities.

Performance Evaluation of RESTful Web Service for Mobile Device (Hatem Haman, 2010) [9] was a study in performance assessment of web service responsibility. This study compared between RESTful and SOAP with size and amount of transmitting data through HTTP protocol. The result of this study showed RESTful had more efficient than SOAP.

A RESTful Approach: Clean UPnP (Universal Plug and Play) without SOAP (Jan Newmarch, 2009) [10] was a study in connection techniques on UPnP protocol by web service. The study compared between RESTful and SOAP with amount of transmitting data through network. The result of this study showed RESTful had rate of response time and quantity of packet less than SOAP many times.

Udomdet Tarahom and Kittima Makabunchakij proposes a Web Service Design for University Services Management[11]. University service department development quality of organization in higher education with responses students and schools need including accommodates requestor accessibility and satisfaction in the services. A system University and government education management. According to link network internal and external in the University. They services are divided into five categories: 1) Education 2) Student 3) School 4) Finance 5) Education Quality

Assurance. This paper proposes education quality assurance in the University (UQA web service) by showing data entity and service methods which is useful for University, Student, School, Higher Education Committee and Education Quality Assurance. The user who is in other organizations can create client application to retrieval web service via internet to request data in XML standard format for processing.

Wisit Wannaphumi and Sipanee Nuchiprasitichai presented the interest topic is “Semantic Web Services Discovery System”[12]. The research was main objective to developing Web Services, which its characteristic was web application, focused searching into meaning the words. Searching Service through key words of ontology technology was compared details of services and definition meaning by OWL language. The application is developed by .Net Technology and Microsoft SQL Server 2005 that is a tool to employ for handling database. The Potege program was employed to design data structure. The evaluation results are investigation satisfaction derived from the experts in Web Services using Likert method that include factors as average and standard deviation. The design structure of data was categorized 5 components as Services (keeping services data of registration), Provider (keeping service data of provider), Business (its operation to classify the kind of business or service types of provider), Publisher (the details of organization or personnel which take registration web services on system), and Quality (Identification quality of services). Summary experiment by users fairly satisfies to system on good level.

Tian Feng studies automatic monitoring, controlling and heterogeneous data integration method in management system for mining safety production on the topic “The Heterogeneous Data Integration Based on XML in Coal Enterprise” [13]. This integration system solve the data heterogeneous problem in production management, integrating data in different operation systems and database systems, making users check data more transparently and conveniently without worrying storage style and location of data in data source. Meanwhile this system can integrate various structuring, half structuring and non structuring data, solving the heterogeneous data integration problem more efficiently, improving safety production and management in coal enterprises.

The study in related literatures showed the development of web service with REST architecture that was more suitable to this research than SOAP protocol. This was because information searching system required good performance with low overhead that can reduce network traffic from more user connection. The examples of REST website composed of Yahoo, Google, Amazon, eBay, etc.



CHAPTER III

MATERIALS AND METHODS

This chapter described the working step of this research and the research tools for developing the system. The steps and tools are as following:

3.1 Step of Research Methodology

The research uses the Waterfall Model which consists of five working steps as depicted in following figure.

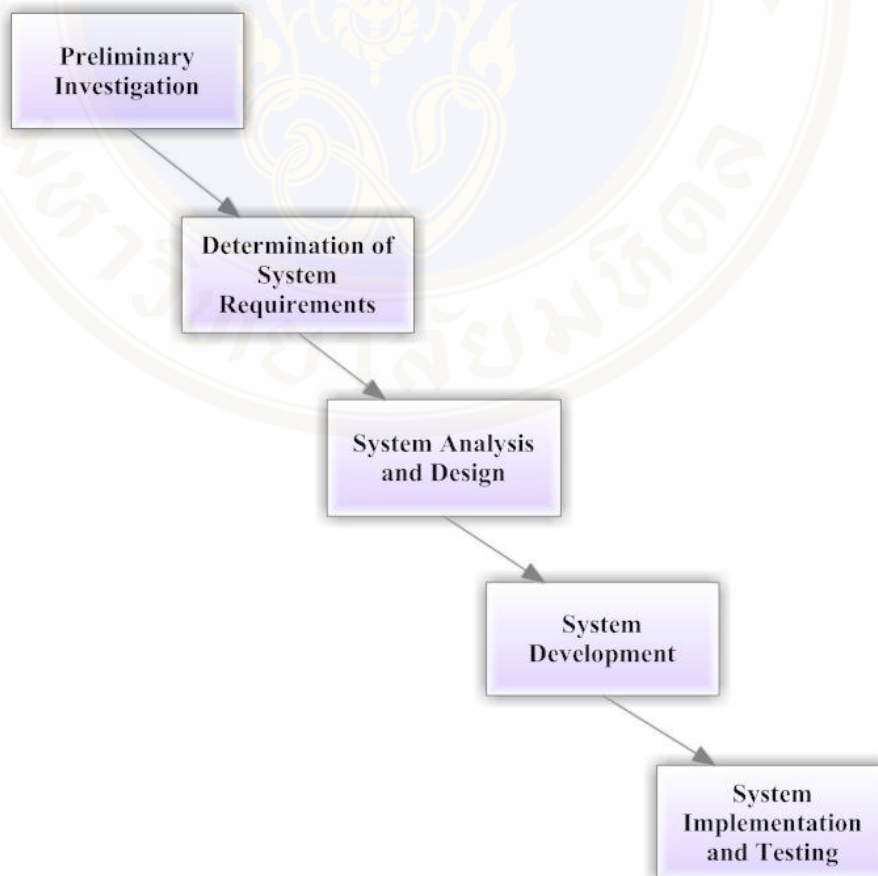


Figure3.1 Step of Waterfall Model

Step 1: Preliminary Investigation: This step is to define problems, investigate problems, identify user requirements, and feasibility study.

Step 2: Determination of System Requirements: This step is to gathering data from interviewing all stakeholders and studying all existing documents.

Step 3: System Analysis and Design: This step is to analyze AS-IS and TO-BE system by studying logical systems and physical systems.

Step 4: System Development: This step is to bring the data received from system analysis and design step to use in system development for implementation of the application of the system.

Step 5: System Implementation and Testing: The system will be installed and tested in order to ensure that the system works properly and accurately as expected. The debugging process will be used if there is error occurred.

3.1.1 Preliminary Investigation

Define problem, investigate problems and user requirements in the system. Gathering data from interviewing officers at Office of the Higher Education Commission who are stakeholders about handling database, administrator, and students who to want further education (Users).

In the present, the information of graduate curriculum was found in individual university website. The Office of the Higher Education Commission is received education data such as student data, curriculum data, financial data, etc, from the university every year. However, this organization has no public website for providing the graduate curriculum information.

3.1.2 Determination of System Requirements

Gathering data from interviewing officers at Office of the Higher Education Commission who are stakeholders about handling database, administrator, and students who to want further education (Users).

Data from all studies will be analyzed in order to obtain user requirements, status of each institution, existing problem and define elements that may effected design of information integration system. Expected results of this working step are including use case and Software requirements specification (SRS).

3.1.2.1 Use Cases:

Use case is a technique for documenting the potential requirements of a new system or software change. Each use case provides one or more scenarios that convey how the system should interact with the end users or other systems in order to achieve specific business goals.

3.1.2.2 Software requirements specification (SRS):

Software requirement specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all of the interactions that the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains nonfunctional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation.

3.1.3 System Analysis and Design

This step is to analyst various problems that found during study the user requirements in each institution especially in the cooperative usage of data. Expected results of this working step include Unified Modeling Language (UML), Entity-Relationship Diagrams (ERDs), Data Dictionary, architecture design, and interface design.

Designing the part of user interface shall be brought as simple as possible. Interface must be easily accessed and understood by all users. Window arrangement of each screen shall orderly and easily for operate and optimal searching desired data efficiently. Therefore, the development shall be based on the following principles: Friendly user interface, Data integrity, optimal search paradigm, Future extendable, Data security, and System stability & performance.

In part of search paradigm, it will be deployed for users to search by keyword and data identifier upon the appropriateness of process of judgment.

The structure system design can categorize performing generally to support to persons 3 groups that are educator, authorized user, and administrator. Educators which can equal as users mean who wanted further education data with Master Degree and Doctor of Philosophy. The system can support debate necessary factors to postgraduate curriculum screening. The authorized user duty is to add and

edit of data into database up-to-date. Administrator responsibilities are to manage system performing smoothly and to solve problems that administrator will handle data flow, limiting authorized policy, and conduct problem from system usage.

System architecture design of system shown as follow:

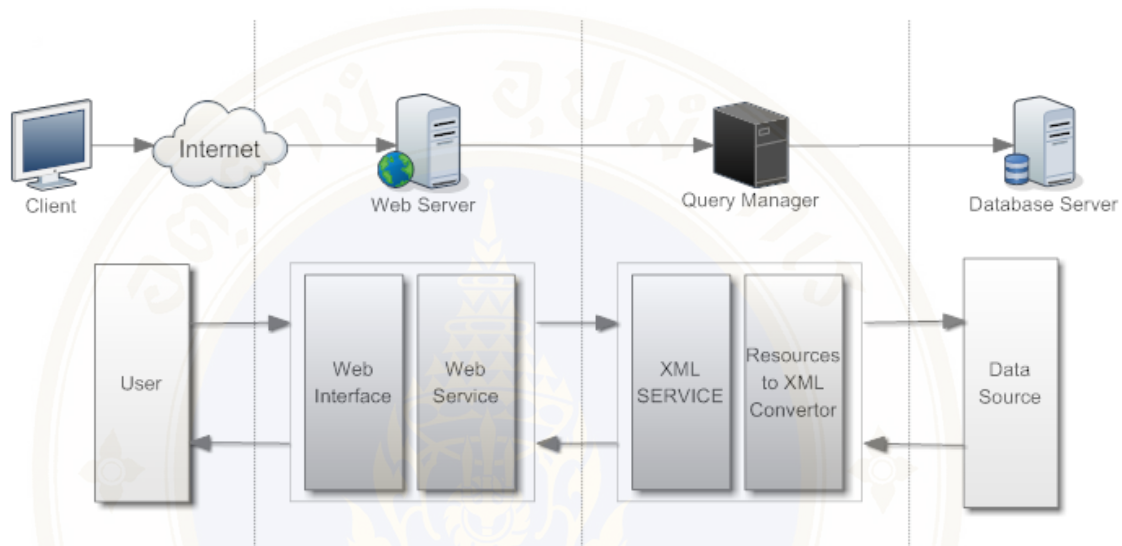


Figure 3.2 System architecture

In system design process, the system was divided into four main components that composed of Client, Web Server, Query Manager, and Database Server. The details of these components were described as follows:

1. Client was used for requesting service from server through internet.
2. Web Server was used for searching and exchanging data center that provided information to user by connection with Query Manager.
3. Query Manager (QM) was used for connection with organization database in order to retrieve data from them. This retrieval data was executed through web service when there was request from user, it could not directly request data through Query Manager. In addition, the retrieved data was in XML language format.
4. Database Server was the database of the Office of the Higher Education Commission.

3.1.4 Development

This step that bringing the data received from analysis and design system forms came to use in system development for implement database and application software for this system.

The system was developed on Windows 7 Enterprise. Apache Web Server was used as web server that run on Appserv and web page was created from PHP, XML, and HTML programming language. MySQL was used for database management and REST protocol was used for connection between web pages and databases as web service. The user interface design was created by Macromedia Dreamweaver CS5.

3.1.5 Testing

The system will be installed and tested in order to ensure that the system work properly and accurately as expected result. The debugging process will be used if there is error.

3.2 Materials

3.2.1 Hardware

- Personal Computer
- CPU : Intel(R) Core(TM)2 Duo processor 2.1 GHz
- RAM : DDR 2 GB
- Hard Disk : 320 GB
- Monitor : WXGA CrystalBrite LCD
- Peripherals : Keyboard, Mouse, Printer, Diskette and CD-ROM Drive

3.2.2 Tool and Software

- Programming Language: PHP
- Database Management System (DBMS): MySQL
- Operating System: Windows 7 (Windows 7 will be installed on personal computer that is used for development program.)

3.3 Research Schedule

The study consumes time as table below

Table 3.1 Research Schedule

Activities	Month						
	1	2	3	4	5	6	7
1. Preliminary Planning	██████████						
2. Collect Related Information	██████████						
3. Requirement Analysis		██████████					
4. Analysis and Design			██████████				
5. Developing of Information System				██████████			
6. Testing and Evaluation					██████████		

CHAPTER IV

SYSTEM IMPLEMENTATION AND RESULTS

This chapter will describe the results of this research. System analysis and design and the details of system database design such as Use Case, Sequence Diagram, ER Diagram and the process description will be presented. User interface and system prototype as well as system test will also be presented.

4.1 Use Case Diagram

Use case is a technique for documenting the potential requirements of a new system or software change. Each Use Case diagram provides one or more scenarios that how the systems should inter act with end users or other systems in order to achieve specific business goals. There are three diagrams of Use Case that are required: Search Curriculum Data use case diagram, Rest API use case diagram and Curriculum Management System use case diagram.

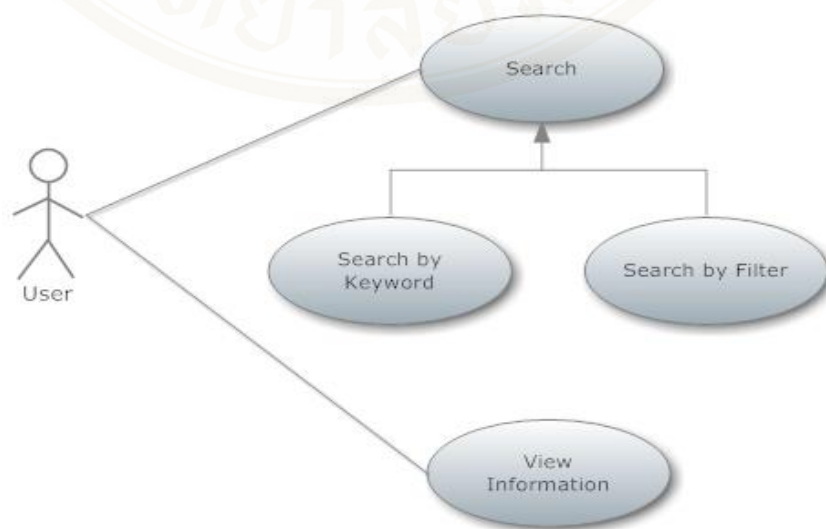


Figure 4.1 Use case diagrams: Search

Figure 4.1 showed Use case diagram of searching system. There was an actor and two use cases of this system. The use cases were composed of search and view information. The description of these use cases were shown in table 4.1 and 4.2.

Table 4.1 Description of Search use case

Use case:	Search
Brief description:	User searches for curriculum data.
Primary actors:	User
Secondary actors:	None
Preconditions:	None
Main flow:	<ol style="list-style-type: none"> 1. User selects keyword search or filter search. <ol style="list-style-type: none"> 1.1 Keyword search: User fills word into form, then press submit button. 1.2 Filter search: User fills word into each filter form then press submit button 2. System presents search result.
Post conditions:	None
Alternative flow:	None

Table 4.2 Description of View information use case

Use case:	View information
Brief description:	User views curriculum data.
Primary actors:	User
Secondary actors:	None
Preconditions:	None
Main flow:	<ol style="list-style-type: none"> 1. The use case starts when user select curriculum data 2. The system displays a web page showing the curriculum details. 3. The top of search result page can be selected to four information views that compose of: <ol style="list-style-type: none"> 3.1 Curriculum information

Table 4.2 (cont.)

Use case:	View information
Main flow:	3.2 University map 3.3 Subject information
Post conditions:	None
Alternative flow:	None

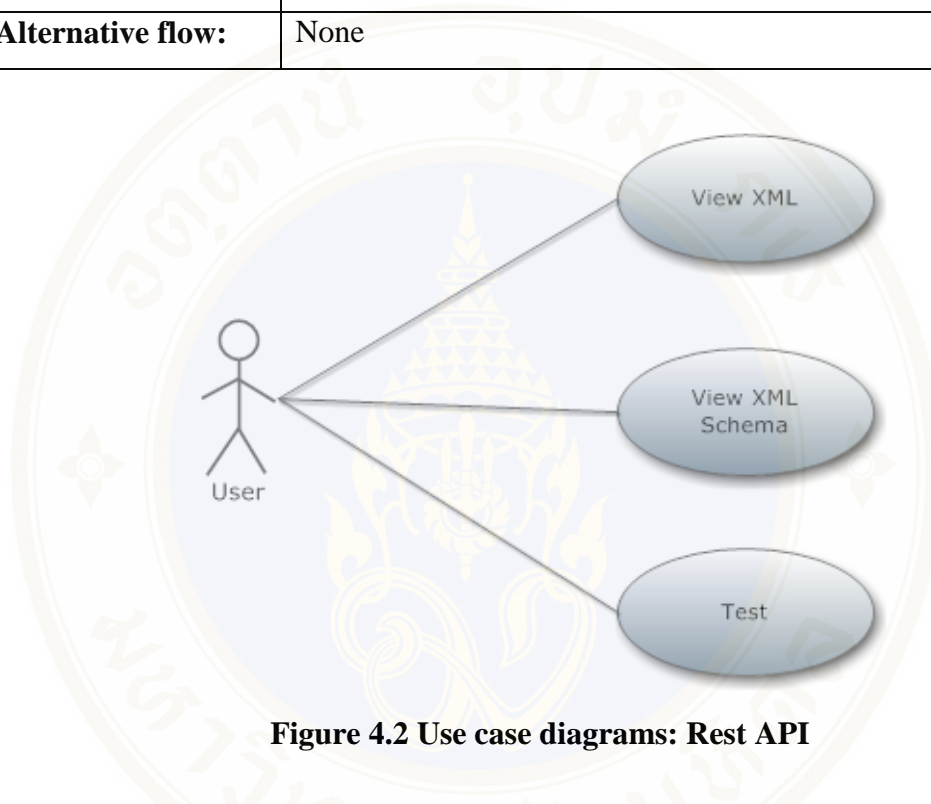


Figure 4.2 Use case diagrams: Rest API

Figure 4.2 showed Use case diagram of REST API. There was an actor and three use cases of this system. The use cases were composed of View XML, View XML Schema and Test. The description of these use cases were shown in table 4.3 to 4.5.

Table 4.3 Description of View XML use case

Use case:	View XML
Brief description:	User views XML of Web service API.
Primary actors:	User
Secondary actors:	None
Preconditions:	None

Table 4.3 (cont.)

Use case:	View XML
Main flow:	<ol style="list-style-type: none"> 1. User selects URL that want to see detail of XML. 2. The System shows detail of XML.
Post conditions:	None
Alternative flow:	None

Table 4.4 Description of View XML Schema use case

Use case:	View XML Schema
Brief description:	User views XML Schema.
Primary actors:	User
Secondary actors:	None
Preconditions:	None
Main flow:	<ol style="list-style-type: none"> 1. User push schema icon that want to see XML Schema. 2. The System shows XML Schema.
Post conditions:	None
Alternative flow:	None

Table 4.5 Description of Test use case

Use case:	Test
Brief description:	User tests Web service API.
Primary actors:	User
Secondary actors:	None
Preconditions:	None
Main flow:	<ol style="list-style-type: none"> 1. User push test icon that want to test. 2. User fills in criteria and limiter. After fills in push apply button. 3. The system shows result of URL Request & XML.
Post conditions:	None
Alternative flow:	None

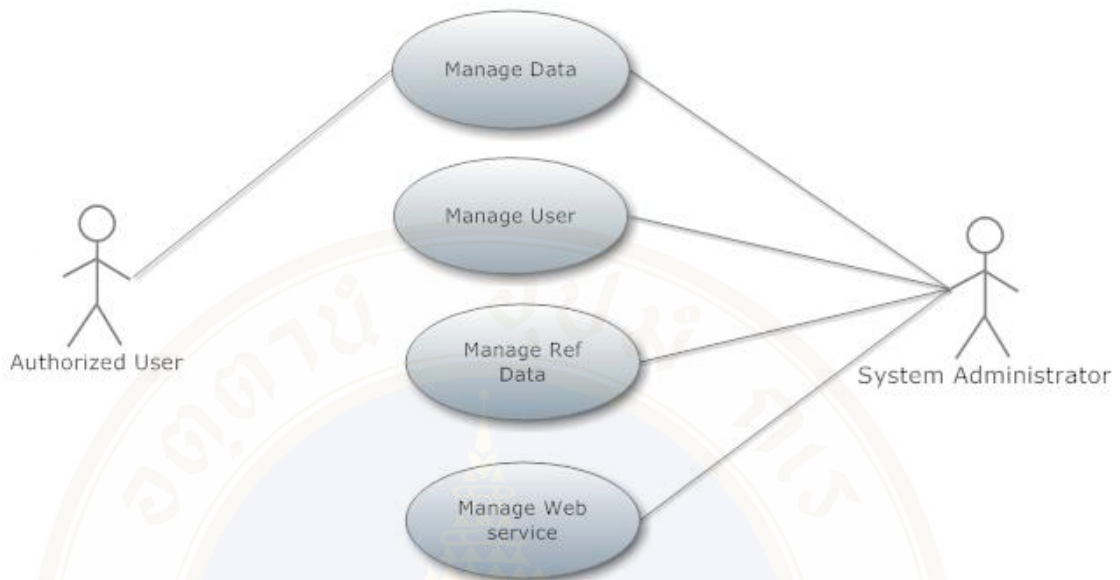


Figure 4.3 Use case diagrams: Curriculum Management System

Figure 4.3 showed Use case diagram of Curriculum Management system. There were two actors and four use cases of this system. The actors were composed of System Administrator and Authorized User. The use cases were composed of manages data, manage user, manage Ref data and manage web service. The description of these use cases were shown in table 4.6 to 4.9.

Table 4.6 Description of Manage Data use case

Use case:	Manage Data
Brief description:	Data management by add, edit and delete action on the data in database
Primary actors:	Authorized User, System Administrator
Secondary actors:	None
Preconditions:	Log in
Main flow:	<ol style="list-style-type: none"> 1. The use case starts when user selects “จัดการข้อมูลหลักสูตร”. 2. If user selects “New” to create new data. <ol style="list-style-type: none"> 2.1 Fills in detail of data and push save button to save data.

Table 4.6 (cont.)

Use case:	Manage Data
Main flow:	<ol style="list-style-type: none"> 3. The use case starts when user selects “จัดการข้อมูลหลักสูตร”. 4. If user selects “New” to create new data. <ol style="list-style-type: none"> 4.1 Fills in detail of data and push save button to save data. 5. If users select curriculum name to edit data. <ol style="list-style-type: none"> 5.1 The system will show detail of data. User can edit data and push save button to save data. 6. If user pushes delete button. <ol style="list-style-type: none"> 6.1 The system will show confirm message. 6.2 If user select Yes, The system will delete data that user selected. If user selects No, The system will cancel delete data.
Post conditions:	None
Alternative flow:	None

Table 4.7 Description of Manage User use case

Use case:	Manage User
Brief description:	User management by add, edit and delete action in the user data, and authentication of data access.
Primary actors:	System Administrator
Secondary actors:	None
Preconditions:	Log in
Main flow:	<ol style="list-style-type: none"> 1. The use case starts when user selects “User Account”. 2. If user selects “New” to create new user. <ol style="list-style-type: none"> 2.1 Fills in detail of user data and push save button to save data. 3. If users select user name to edit data. <ol style="list-style-type: none"> 3.1 The system will show detail of user data. User can edit data and push save button to save data..

Table 4.7 (cont.)

Use case:	Manage User
Main flow:	<p>4. If user push delete button</p> <p>4.1 The system will show confirm message.</p> <p>If user select Yes, The system will delete user data that user selected. If user selects No, The system will cancel delete data.</p>
Post conditions:	None
Alternative flow:	None

Table 4.8 Description of Manage Reference Data use case

Use case:	Manage Reference Data
Brief description:	Reference table management by add, edit and delete action on the reference table.
Primary actors:	System Administrator
Secondary actors:	None
Preconditions:	Log in
Main flow:	<p>1. The use case starts when user selects “Reference data”.</p> <p>2. If user selects “New” to create new reference data.</p> <p>2.1 Fills in detail of data, push save button to save.</p> <p>3. If users select reference name to edit data.</p> <p>3.1 The system will show detail of data. User can edit data and push save button to save data.</p> <p>4. If user pushes delete button.</p> <p>4.1 The system will show confirm message.</p> <p>If user select Yes, The system will delete data that user selected. If user selects No, The system will cancel delete data.</p>
Post conditions:	None
Alternative flow:	None

Table 4.9 Description of Manage Web Service use case

Use case:	Manage Web Service
Brief description:	Web service API management by add, edit and delete action on the API.
Primary actors:	System Administrator
Secondary actors:	None
Preconditions:	Log in
Main flow:	<ol style="list-style-type: none"> 1. The use case starts when user selects “REST API”. 2. If user selects “New” to create new API. <ol style="list-style-type: none"> 2.1 Fills in detail of API information and push save button to save data. 3. If users select API name to edit data. <ol style="list-style-type: none"> 3.1 The system will show detail of API information. User can edit data and push save button to save data. 4. If user pushes delete button <ol style="list-style-type: none"> 4.1 The system will show confirm message. If user select Yes, The system will delete data that user selected. If user selects No, The system will cancel delete data. 5. If user push stage button. <ol style="list-style-type: none"> 5.1 The system will show confirm message. If user select Yes, The system will unpublished API that user selected. If user selects No, The system will cancel unpublished API.
Post conditions:	None
Alternative flow:	None

4.2 Sequence Diagram

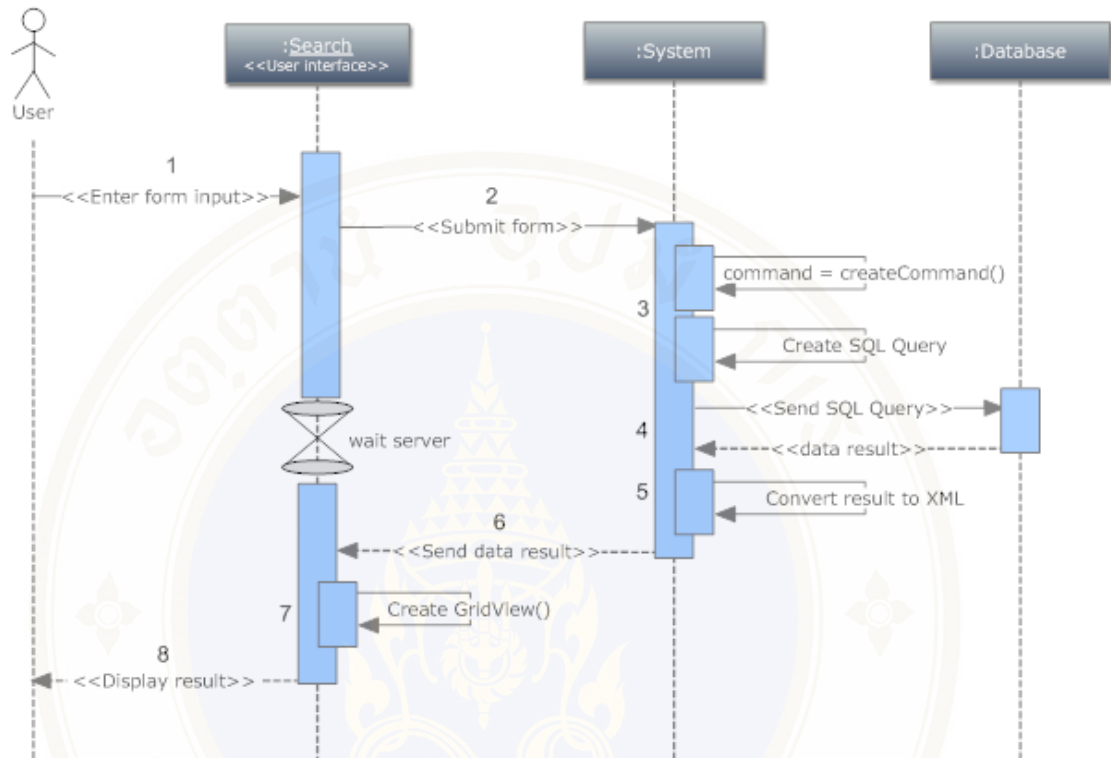


Figure 4.4 UML Sequence Diagram: Search

This diagram shows activity of Local Source Search in QMS module. The system provides GUI for user can search data from local sources. Here is a walkthrough for the activity.

- step 1. User fills values in input form.
- step 2. User submits form to server.
- step 3. When system accepted form request, system must validate the form parameter and create query command.
- step 4. Execute the command and wait for result of server.
- step 5. Convert result to XML
- step 6. Return result to UI.
- step 7. Use JavaScript to create the result in a table form.
- step 8. Return result to user.

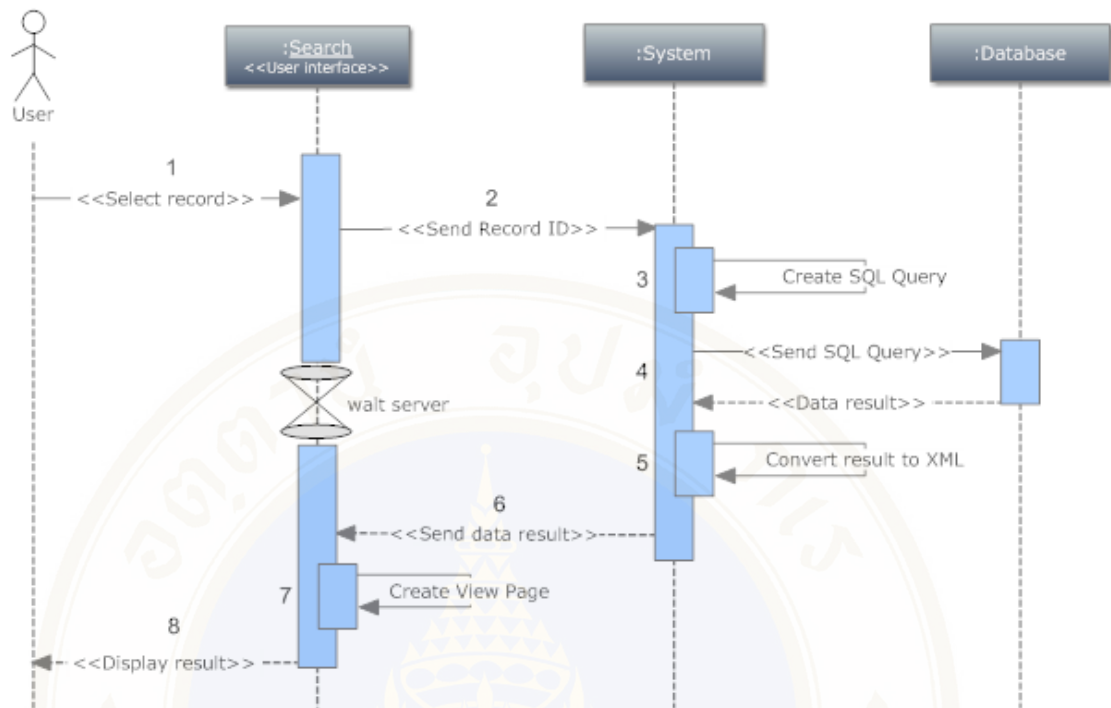


Figure 4.5 UML Sequence Diagram: View

Sequence diagram of view data was shown in figure 4.5. This diagram presented process of selected data viewing that were shown as below:

- step 1. User selects interested curriculum.
- step 2. Search function sends value of selected data ID to system.
- step 3. System converts the value of selected data ID to sql query command.
- step 4. Execute the command and wait for result of server.
- step 5. Convert result to XML
- step 6. Return result to UI.
- step 7. Search information creates selected curriculum information page.
- step 8. Show result to user.

4.3 Database Design

Analysis of designed the system. Use MySQL Server for database storage system. The detail of database design is shown in entity relationship diagram in figure 4.6 to 4.8. Data dictionary is shown in Appendix.

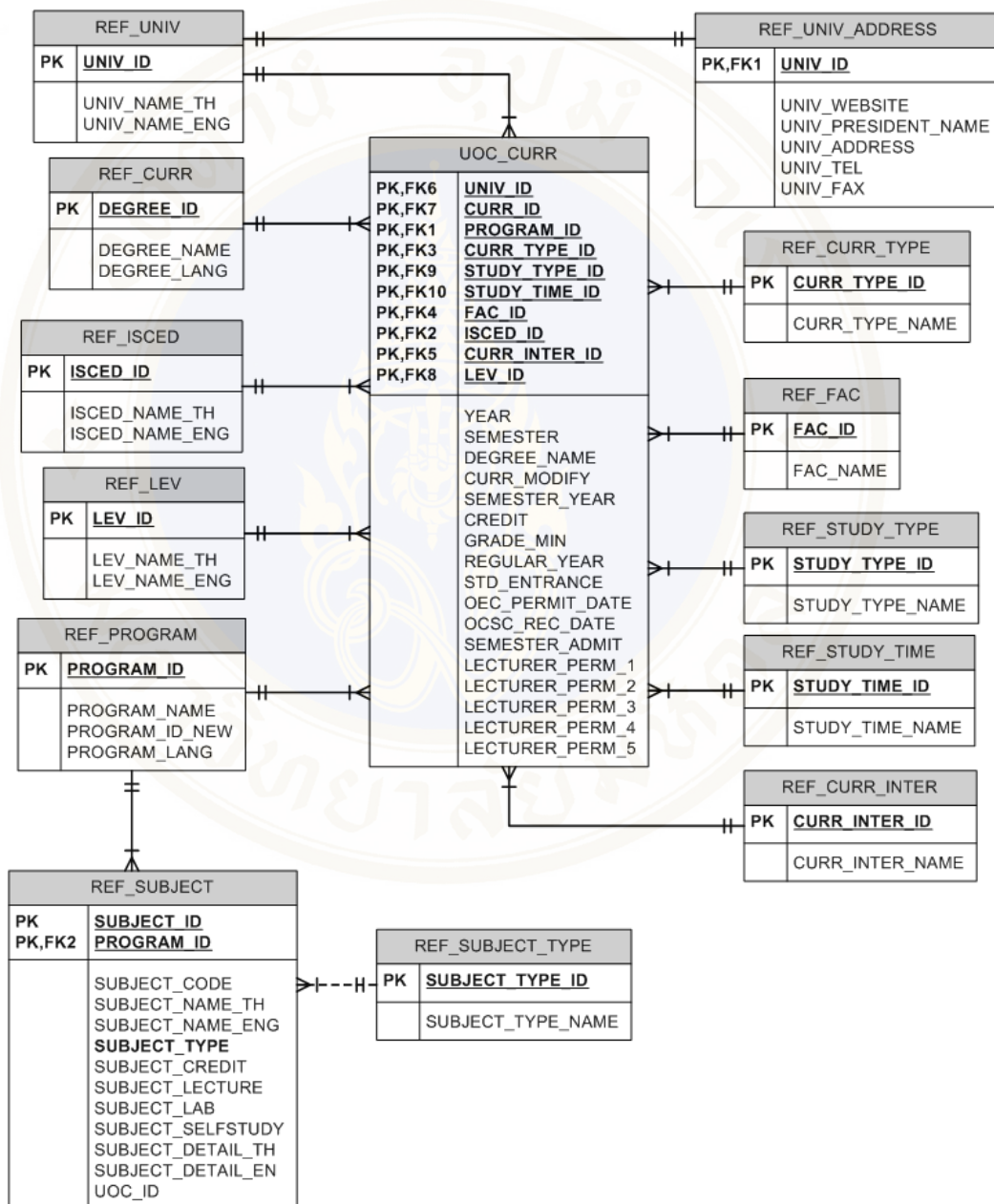


Figure 4.6 ER-Diagram: Curriculum

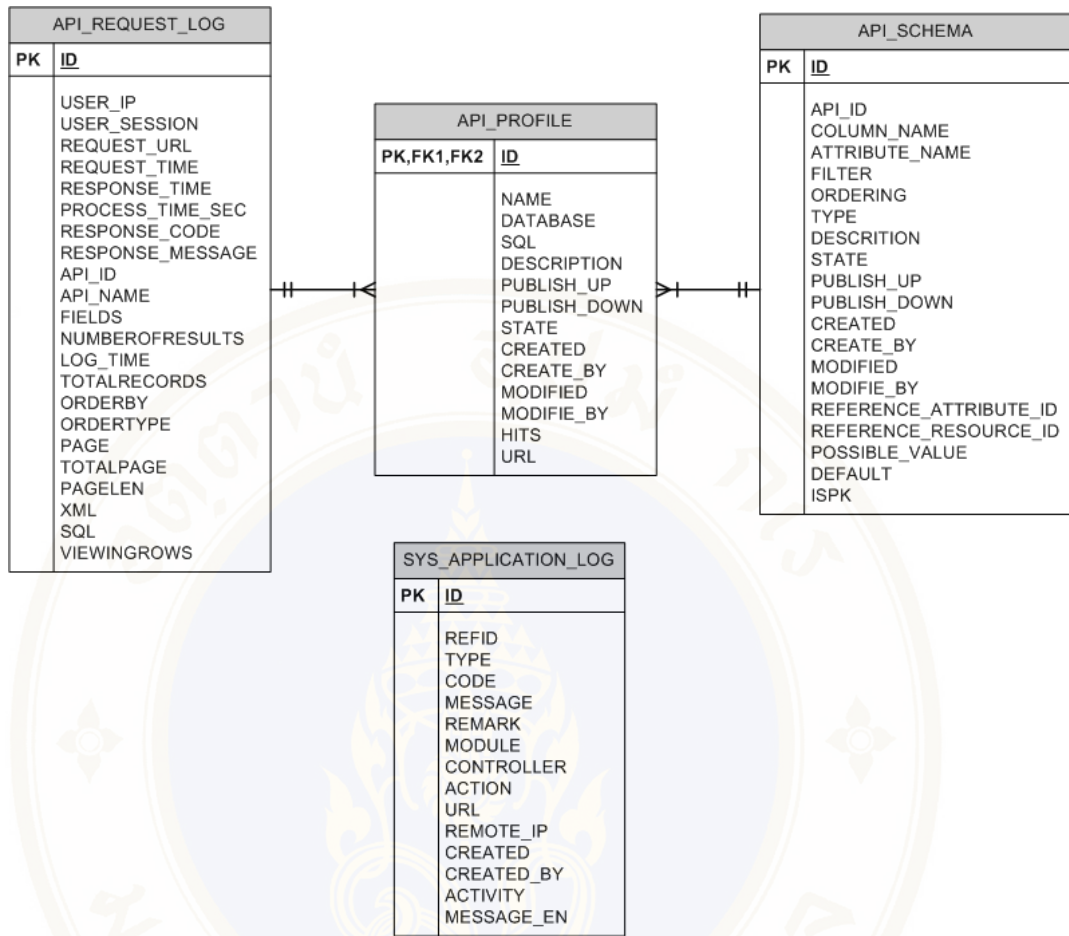


Figure 4.7 ER-Diagram: Web Service API

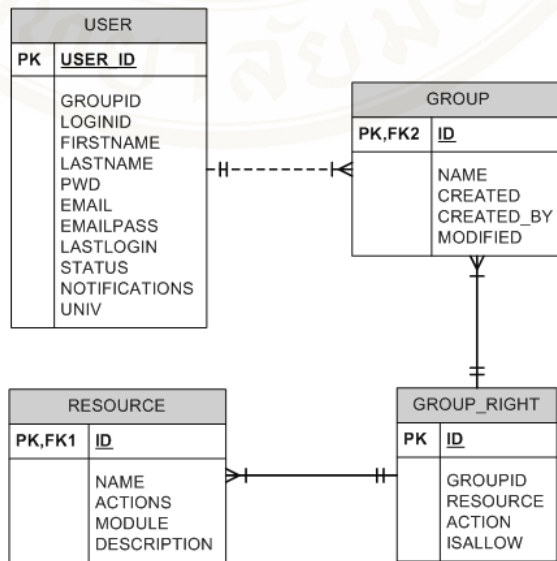


Figure 4.8 ER-Diagram: User

4.4 The Result of System Development

The developed system was divided into three main systems that composed of Curriculum searching, REST API and Service management.

The first main system is curriculum information search, the second one is overall API presentation page and the third one is database management. The details of these main systems were shown as follow:



Figure 4.9 Main Page

4.4.1 Curriculum search system

This main system was used for curriculum information search. It had two search types as below:

4.4.1.1 Keyword search was filled into form by curriculum name that was showed as Figure 4.10

4.4.1.2 Filter search was filled into form by and data that was showed as Figure 4.11. Each form had auto complete for use convenience.

Figure 4.10 Search by Keyword

Figure 4.11 Search by Filters

When words were filled into form and press apply button, the curriculum information result would be shown as Figure 4.12.

ลำดับ	ชื่อหลักสูตร	คณะ	มหาวิทยาลัย
6909	วิทยาศาสตรมหาบัณฑิต (วิทยาศาสตร์และวิศวกรรมวัสดุ) วิทยาศาสตร์และวิศวกรรมวัสดุ	คณะวิทยาศาสตร์	มหาวิทยาลัยมหิดล
6992	วิทยาศาสตรมหาบัณฑิต(อณูพันธุศาสตร์และพันธุวิศวกรรมศาสตร์) อณูพันธุศาสตร์และพันธุวิศวกรรมศาสตร์	สถาบันอณูชีววิทยาและพันธุศาสตร์	มหาวิทยาลัยมหิดล
6924	วิทยาศาสตรมหาบัณฑิต(เทคโนโลยีการจัดการระบบสารสนเทศ) เทคโนโลยีการจัดการระบบสารสนเทศ	คณะวิศวกรรมศาสตร์	มหาวิทยาลัยมหิดล
6920	วิศวกรรมศาสตรมหาบัณฑิต(วิศวกรรมชีวการแพทย์) วิศวกรรมชีวการแพทย์	คณะวิศวกรรมศาสตร์	มหาวิทยาลัยมหิดล
6923	วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมเครื่องกล) วิศวกรรมเครื่องกล	คณะวิศวกรรมศาสตร์	มหาวิทยาลัยมหิดล
6921	วิศวกรรมศาสตรมหาบัณฑิต(วิศวกรรมสิ่งแวดล้อม) วิศวกรรมสิ่งแวดล้อม	คณะวิศวกรรมศาสตร์	มหาวิทยาลัยมหิดล
6922	วิศวกรรมศาสตรมหาบัณฑิต(วิศวกรรมอุตสาหกรรม) วิศวกรรมอุตสาหกรรม	คณะวิศวกรรมศาสตร์	มหาวิทยาลัยมหิดล
6926	วิศวกรรมศาสตรมหาบัณฑิต(วิศวกรรมเคมีบูรณาการ) ลาวศึกษา	คณะวิศวกรรมศาสตร์	มหาวิทยาลัยมหิดล
7097	ปรัชญาดุษฎีบัณฑิต (วิทยาศาสตร์และวิศวกรรมวัสดุ) วิทยาศาสตร์และวิศวกรรมวัสดุ	คณะวิทยาศาสตร์	มหาวิทยาลัยมหิดล
7156	ปรัชญาดุษฎีบัณฑิต(อณูพันธุศาสตร์และพันธุวิศวกรรมศาสตร์) อณูพันธุศาสตร์และพันธุวิศวกรรมศาสตร์	สถาบันอณูชีววิทยาและพันธุศาสตร์	มหาวิทยาลัยมหิดล

Figure 4.12 Search Result

In the searching by keyword, user can input many words and use space to separate each word. The system would use user's keyword to search in Curriculum

name, Faculty name, University name, and ISCED name in database for result display. The process of this searching are showed in Figure 4.13.

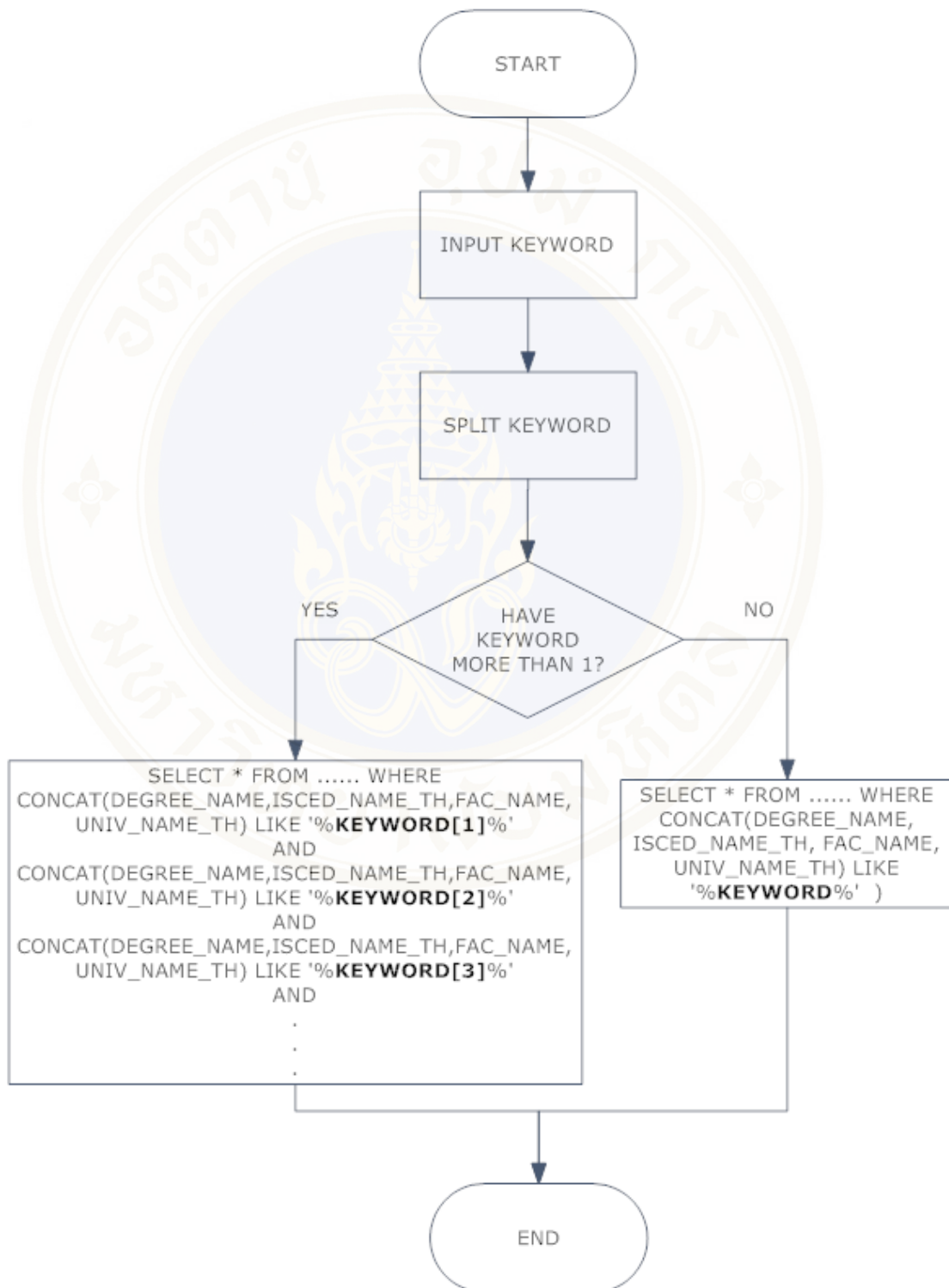


Figure 4.13 Flow Chart Search by Keyword

The curriculum name was pressed for requesting curriculum information page. The top of this page were information selection bar that was composed of:

- Curriculum information was shown as Figure 4.14.
- University map was shown as Figure 4.15.
- Curriculum subject was shown as Figure 4.16.

ข้อมูลหลักสูตร		ข้อมูลรายวิชา	
ชื่อหลักสูตร	เทคโนโลยีการจัดการระบบสารสนเทศ		
ชื่อปริญญา	วิทยาศาสตรมหาบัณฑิต(เทคโนโลยีการจัดการระบบสารสนเทศ)		
ระดับการศึกษา	ปริญญาโท		
คณะ	คณะวิศวกรรมศาสตร์		
มหาวิทยาลัย	มหาวิทยาลัยมหิดล		
ประเภทหลักสูตร	หลักสูตรปกติ		
ความสากลของหลักสูตร	หลักสูตรภาษาไทย		
เวลาเรียน	วันธรรมดากลางวัน		
จำนวนภาคการศึกษาต่อปี	2		
จำนวนปีหลักสูตร	2 ปี		
จำนวนที่รับ	50 คน		
ค่านายกิตติลดลดหลักสูตรภาคและปีการศึกษาที่เริ่มรับนักศึกษา	1/2537		
ที่อยู่	20/25 หมู่ 3 ถ.พุทธมณฑล สาย 4 ต.ศาลายา อ.พุทธมณฑล จ.นครปฐม 73170		
เบอร์โทรศัพท์	0 2849 6302, 0 2849 6010-13		
เบอร์โทรสาร	0 2849 6330		
เว็บไซต์	www.mahidol.ac.th		

Figure 4.14 Detail of curriculum

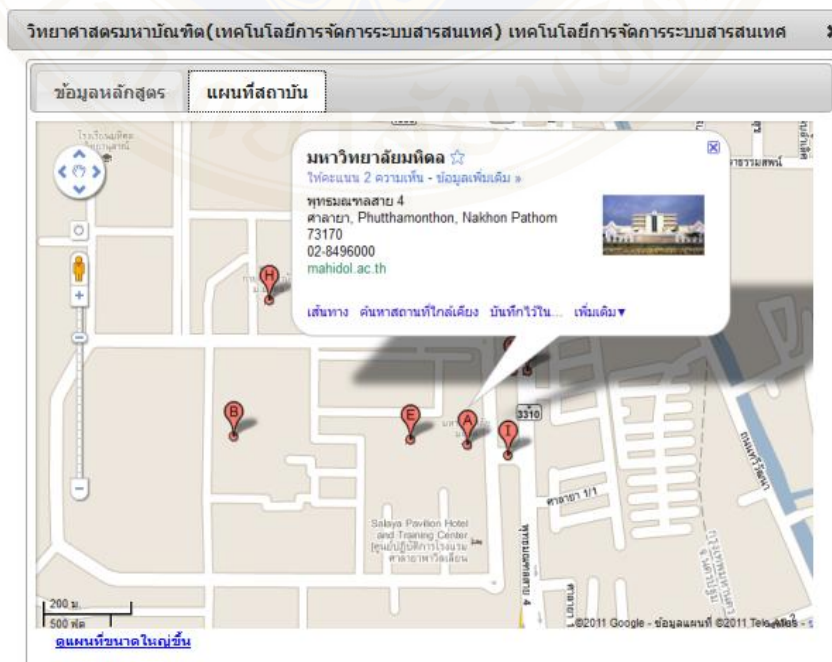


Figure 4.15 University Map


ข้อมูลหลักสูตร		ข้อมูลรายวิชา	
 จำนวนหน่วยกิตรวมตลอดหลักสูตร 13 หน่วยกิต			
หมวดวิชา 2			
322754	วิศวกรรมฐานความรู้	3(3-0-6)	
	Knowledge Engineering		
322755	ระบบสนับสนุนการตัดสินใจ	3(3-0-6)	
	Decision Support Systems		
322772	หัวข้อเรื่องพิเศษทางเทคโนโลยีสารสนเทศ	3(3-0-6)	
	Special topics in Information Technology		

Figure 4.16 Detail of subject

4.4.2 Rest API

This main system was used for overall services presentation of the system that was shown in Figure 4.17. The Rest API composed of:

- URI (position 1) presented the detail of service in XML format that was shown in Figure 4.18
- Info (position 2) presented the detail of service attribute that was shown in Figure 4.19
- Test (position 3) was used for API testing by applying criteria in form. The result of this applying composed of two section; url request and xml, that was shown in Figure 4.20. The url request section presented the url that was created from the criteria. The xml section presented the result in XML format.

XML Data Service				info	test
resource id	URI	description			
program	/service/api/ref/program	ข้อมูลอ้างอิง สาขาวิชา	2		
ref-inter-type	/service/api/ref/inter	ข้อมูลอ้างอิง ความสากล			
curriculum	/service/api/curriculum	ข้อมูลหลักสูตร			
ref-univ	/service/api/ref/univ	ข้อมูลอ้างอิง ชื่อมหาวิทยาลัย			
ref-fac	/service/api/ref/fac	ข้อมูลอ้างอิง คณะ			
ref-lev	/service/api/ref/level	ข้อมูลอ้างอิง ระดับการศึกษา			
ref-curr	/service/api/ref/curr	ข้อมูลอ้างอิง ชื่อปริญญา			
curr-type	/service/api/ref/curr-type	ข้อมูลอ้างอิง ประเภทหลักสูตร			
ref-study-time	/service/api/ref/ref-study-time	ข้อมูลอ้างอิง เวลาเรียน			
ref_subject	/service/api/ref/subject	ข้อมูลรายวิชา			

Figure 4.17 Rest API

```

- <data>
- <header>
  <field>4</field>
  <numberOfResults>10</numberOfResults>
  <totalRecords>6511</totalRecords>
  <orderBy>0</orderBy>
  <orderType>asc</orderType>
  <page>2</page>
  <totalpage>652</totalpage>
  <pagelen>10</pagelen>
  <filter/>
  <viewingRows>11 - 20</viewingRows>
  <fields>programId,programIdNew,programLang,programName</fields>
  <pk>PROGRAM_ID</pk>
  <uri>test=1</uri>
- <sql>
  SELECT * FROM ( SELECT `PROGRAM_ID` , `PROGRAM_ID_NEW` , `PROGRAM_LANG` , `PROGRAM_NAME`
</sql>
</header>
- <items>
- <item id="0010">
  <programId>0010</programId>
  <programIdNew>000010</programIdNew>
  <programLang>th</programLang>
  <programName>กฎหมายเศรษฐกิจ</programName>
</item>
- <item id="0011">
  <programId>0011</programId>
  <programIdNew>000011</programIdNew>
  <programLang>th</programLang>
  <programName>กฎหมายเอกชน</programName>
</item>
- <item id="0012">

```

Figure 4.18 XML

program-info				
field	type	pk	allow filter	
programId	integer	✓	✓	
programIdNew	string		✓	
programLang	string		✓	
programName	string		✓	

Figure 4.19 Service Attribute

filter	type	Operator	value	
criteria:				
programId	integer	=	<input type="text"/>	รหัสสาขาวิชา
programIdNew	string	LIKE	<input type="text"/>	รหัสสาขาวิชา(ใหม่)
programLang	string	LIKE	<input type="text"/>	
programName	string	LIKE	<input type="text"/>	สาขาวิชา
limiter				
sort	integer	=	programName	field id ที่ต้องการเรียงลำดับ
direction	integer	=	desc	ประเภทการเรียงลำดับ
page	integer	=	1	หน้าที่ต้องการ
rows per page	integer	=	10	จำนวนแถวข้อมูลต่อหน้า
<input type="button" value="Apply"/>				
URL Request				
http://curriculum-data.sonaradev.com/service/api/ref/program				
Result				
แท้ม XML นี้ไม่มีสไตลที่เชื่อมโยงกันอยู่ โครงสร้างเอกสารถูกแสดงอยู่ทางด้านล่าง <pre> - <data> - <header> <field>4</field> <numberOfResults>10</numberOfResults> <totalRecords>6511</totalRecords> <orderBy>0</orderBy> <orderType>asc</orderType> </pre>				

Figure 4.20 Test API

4.4.3 Service Management

This main system was used to manage database, web service API, and user account. The access of this main system required log in that the log in interface was showed in Figure 4.21.

ระบบจัดการฐานข้อมูลหลักสูตร
Please enter a valid Login ID and Password to access the Console.
Please use the Chrome.

Login

Login ID:

Password:

Login

Forgot your password?
Enter your Login ID or your E-mail. Password will be send to your E-mail.

Send

Figure 4.21 Log in

The access level of service management was composed of authorized user and administrator. The details of these access levels were shown as follows:

4.4.3.1 Authorized User was the access level of university user. It could only add or edit user's university data that composed of university and curriculum data. Figure 4.22 to 4.24 showed interface of main page, university data, and curriculum data.

ระบบจัดการฐานข้อมูลหลักสูตร

Dashboard | จัดการข้อมูลหลักสูตร

ส่วนงาน : ผู้ดูแลข้อมูล

- ข้อมูลมหาวิทยาลัย 1
- รายชื่อหลักสูตร 2

Dashboard

User
3 account

Figure 4.22 Main Page of Authorized user

จัดการข้อมูลของ จุฬาลงกรณ์มหาวิทยาลัย

รหัสมหาวิทยาลัย*: 00100	เว็บไซต์มหาวิทยาลัย*: www.chula.ac.th	ชื่ออธิการบดี*: ศ. นพ.ภิรมย์ กมลรัตนกุล
เบอร์โทรศัพท์*: 0 2215 0871-3	เบอร์โทรสาร*: 0 2215 4804	
ที่อยู่ ที่อยู่มหาวิทยาลัย*: 254 ถ.พญาไท เขตปทุมวัน กรุงเทพฯ 10330	latitude*: 13.740073	longitude*: 100.530567

Figure 4.23 Edit University Data

โครงสร้างหลักสูตรของ จุฬาลงกรณ์มหาวิทยาลัย

คณะ	สาขาวิชา	ชื่อปริญญา	ระดับการศึกษา
คณะครุศาสตร์			
	การศึกษาปฐมวัย	ครุศาสตรดุษฎีบัณฑิต	ปริญญาเอก
	พัฒนศึกษา	ครุศาสตรดุษฎีบัณฑิต	ปริญญาเอก
	อุดมศึกษา	ครุศาสตรดุษฎีบัณฑิต	ปริญญาเอก
	การศึกษานอกระบบโรงเรียน	ครุศาสตรดุษฎีบัณฑิต	ปริญญาเอก
	หลักสูตรและการสอน	ครุศาสตรดุษฎีบัณฑิต	ปริญญาเอก
	จิตวิทยาการศึกษา	ครุศาสตรดุษฎีบัณฑิต	ปริญญาเอก
	สุขศึกษาและพลศึกษา	ครุศาสตรดุษฎีบัณฑิต	ปริญญาเอก
	บริหารการศึกษา	ครุศาสตรดุษฎีบัณฑิต	ปริญญาเอก
	การวัดและประเมินผลการศึกษา	ครุศาสตรดุษฎีบัณฑิต	ปริญญาเอก
	วิธีวิทยาการวิจัยการศึกษา	ครุศาสตรดุษฎีบัณฑิต	ปริญญาเอก
	เทคโนโลยีและสื่อสารการศึกษา	ครุศาสตรดุษฎีบัณฑิต	ปริญญาเอก
	การศึกษาปฐมวัย	ครุศาสตรบัณฑิต	ปริญญาตรี

Figure 4.24 Edit Curriculum Data

4.4.3.2 Administrator was the access level that could manage overall database, web service API, and user account. The main page of this access level was shown in Figure 4.25

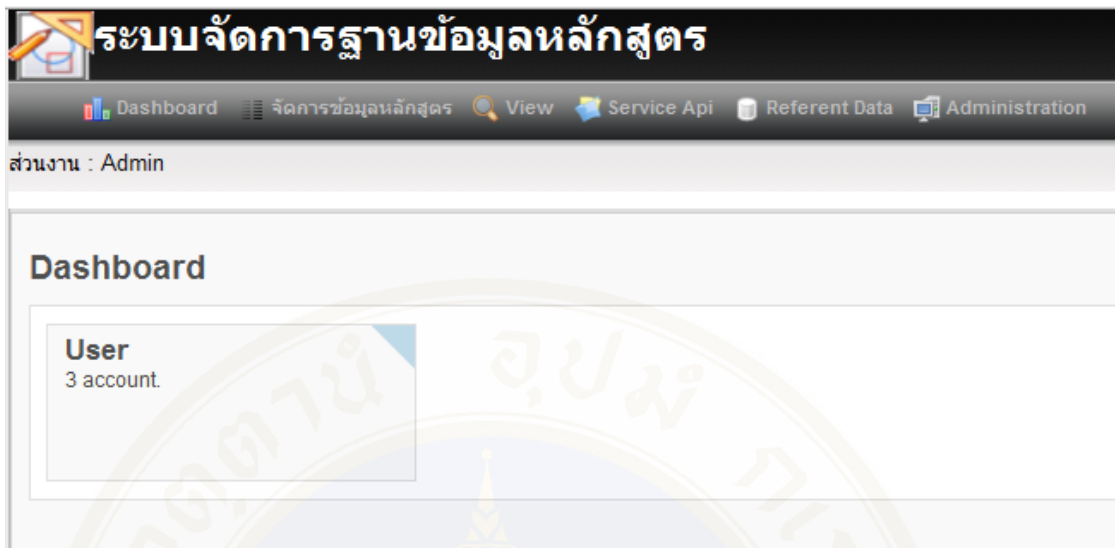


Figure 4.25 Main Page of Administrator

The administrator access level could manage in five sections that composed of Curriculum management, Service API, Reference data, and Administration. The details of these sections were shown as follows:

- Curriculum management section was used to add, edit, or delete university and curriculum data. Their interfaces were shown in Figure 4.26 and 4.27.



Figure 4.26 University management

รหัส	สาขาวิชา	ระดับการศึกษา	ชื่อปริญญา	ชื่อมหาวิทยาลัย	คณะ	ลบ
1	เทคโนโลยีมีตรีเทียม	ปริญญาโท	วิทยาศาสตร์มหาบัณฑิต	จุฬาลงกรณ์มหาวิทยาลัย	วิทยาลัยมีตรีเทียมและมีตรีเทียม	✗
2	เทคโนโลยีมีตรีเทียม	ปริญญาโท	วิทยาศาสตร์มหาบัณฑิต	จุฬาลงกรณ์มหาวิทยาลัย	วิทยาลัยมีตรีเทียมและมีตรีเทียม	✗
3	วิทยาศาสตร์พลีเมอส์	ปริญญาโท	วิทยาศาสตร์มหาบัณฑิต	จุฬาลงกรณ์มหาวิทยาลัย	วิทยาลัยมีตรีเทียมและมีตรีเทียม	✗
4	เทคโนโลยีมีตรีเทียม	ปริญญาเอก	วิทยาศาสตรดุษฎีบัณฑิต	จุฬาลงกรณ์มหาวิทยาลัย	วิทยาลัยมีตรีเทียมและมีตรีเทียม	✗
5	วิทยาศาสตร์พลีเมอส์	ปริญญาเอก	วิทยาศาสตรดุษฎีบัณฑิต	จุฬาลงกรณ์มหาวิทยาลัย	วิทยาลัยมีตรีเทียมและมีตรีเทียม	✗
6	การจัดการธุรกิจระหว่างประเทศ	ปริญญาโท	การจัดการมหาบัณฑิต	จุฬาลงกรณ์มหาวิทยาลัย	คณะพาณิชยศาสตร์และการบัญชี	✗
7	การศึกษารัฐบาล	ปริญญาเอก	ครุศาสตรดุษฎีบัณฑิต	จุฬาลงกรณ์มหาวิทยาลัย	คณะครุศาสตร์	✗
8	พัฒนาศึกษา	ปริญญาเอก	ครุศาสตรดุษฎีบัณฑิต	จุฬาลงกรณ์มหาวิทยาลัย	คณะครุศาสตร์	✗
9	คอมพิวเตอร์	ปริญญาเอก	ครุศาสตรดุษฎีบัณฑิต	จุฬาลงกรณ์มหาวิทยาลัย	คณะครุศาสตร์	✗

Figure 4.27 Curriculum management

- Service API section was used to create, edit, or delete REST API and view Log file of called service. Their interfaces were shown in Figure 4.28 and 4.29.

รหัส	name	database	description	url	hits	state	ลบ
1	program	curr	ข้อมูลอ้างอิง สาขาวิชา	/service/api/ref/program	0	🚩	✗
2	ref-inter-type	curr	ข้อมูลอ้างอิง ความสากล	/service/api/ref/inter	0	🚩	✗
3	curriculum	curr	ข้อมูลหลักสูตร	/service/api/curriculum	0	🚩	✗
4	ref-univ	curr	ข้อมูลอ้างอิง ชื่อมหาวิทยาลัย	/service/api/ref/univ	0	🚩	✗
5	ref-fac	curr	ข้อมูลอ้างอิงคณะ	/service/api/ref/fac	0	🚩	✗
6	ref-lev	curr	ข้อมูลอ้างอิง ระดับการศึกษา	/service/api/ref/level	0	🚩	✗
7	ref-curr	curr	ข้อมูลอ้างอิงชื่อปริญญา	/service/api/ref/curr	0	🚩	✗
8	curr-type	curr	ข้อมูลอ้างอิงประเภทหลักสูตร	/service/api/ref/curr-type	0	🚩	✗
9	ref-study-time	curr	ข้อมูลอ้างอิง เวลาเรียน	/service/api/ref/study-time	0	🚩	✗
10	ref_subject	curr	ข้อมูลรายวิชา	/service/api/ref/subject	0	🚩	✗
11	Finance	curr	sdfsdf	/service/api/ref/dsasd/asdasd/asda/finance	0	🚩	✗

Figure 4.28 Service API

Resource Request Log

รหัส	userIp	userSession	requestUrl	requestTime	responseTime	processTimeSec	responseCode
61	101.108.171.243		/service/api/ref/fac?facid=00001&page=1&limit=10&sort=facid%20desc	01/05/2011	01/05/2011	0.790718078613	202
62	101.108.171.243		/service/api/ref/fac	01/05/2011	01/05/2011	0.69512796402	202
63	101.108.171.243		/service/api/ref/fac	01/05/2011	01/05/2011	0.454684972763	202
64	101.108.171.243		/service/api/ref/fac?facid=0010&page=1&limit=10&sort=facname%20desc	01/05/2011	01/05/2011	0.705565929413	202
65	101.108.171.243		/service/api/ref/fac?facid=00010&page=1&limit=10&sort=facname%20desc	01/05/2011	01/05/2011	0.420017957687	202
			/service/api/ref/fac?facname=%*%e0%b8%aa%e0%b8%b4%e0%b9%88%e0				

Figure 4.29 Request Log

- Reference data section was used to add, edit, or delete reference table. Its interface was shown in Figure 4.30.

รหัสสาขาวิชา	ชื่อ	ภาษา	ลบ
0000	ไม่ระบุ (ในกรณีที่ไม่ได้)	th	✗
0001	กฎหมาย	th	✗
0002	กฎหมายการที่ดินและทรัพยากร	th	✗
0003	กฎหมายทรัพย์สินทางปัญญาธุรกิจการค้าระหว่างประเทศและอนุญาโตตุลาการ	th	✗
0004	กฎหมายธุรกิจ	th	✗
0005	กฎหมายธุรกิจระหว่างประเทศ	th	✗
0006	กฎหมายมหาชน	th	✗
0007	กฎหมายระหว่างประเทศ	th	✗
0008	กฎหมายอาญา	th	✗
0009	กฎหมายอิสลาม	th	✗
0010	กฎหมายเศรษฐกิจ	th	✗
0011	กฎหมายเอกชน	th	✗
0012	กัมการท่องเที่ยว	th	✗
0013	Business	en	✗

Figure 4.30 Manage Reference Data

- Administration section was used to manage user account, set user data access group, and view log file of action in database.

User Account Management was used to create and edit user account data. Its interface was shown in Figure 4.31 to 4.33.

The screenshot shows a web application interface for user management. At the top, there is a navigation menu with options: 'User Account', 'User Group', and 'App log'. Below the menu, the title 'System User' is displayed. A table lists the following user data:

รหัส	loginid	groupid	firstname	lastname	pwd	email	univ	lastlogin	สถานะ	master	ลบ
1002	admin	1	admin	Webmaster	admin	patiwat.w@hotmail.com	00000	2011-05-02 02:28:39	Yes	No	✘
1005	poj	9	poj	poj	1234	pojij349@hotmail.com	00100	2011-05-02 02:28:19	Yes	No	✘
1006	ton	9	ton	ton	123456	ton@hotmail.com	01800	2011-04-22 00:34:30	Yes	No	✘

Figure 4.31 User Account Management

The screenshot shows the 'Create New: user' form. It is divided into several sections for data entry:

- ข้อมูลทั่วไป (General Information):** Includes fields for 'Firstname*', 'Lastname*', and 'Email*'. The 'Univ*' field is a dropdown menu currently set to 'สำนักงานคณะกรรมการการอาชีวศึกษา'.
- ข้อมูล login (Login Information):** Includes fields for 'Groupid*' (dropdown menu set to 'Admin'), 'Loginid*', and 'Pwd*'.
- สถานะ (Status):** Includes a 'Status*' dropdown menu currently set to 'Yes'.

Figure 4.32 Create New User

System User #1002

ข้อมูลทั่วไป

รหัส*: Firstname*: Lastname*:

Email*: Univ*:

ข้อมูล login

Groupid*: Loginid*: Pwd*:

สถานะ

Status*:

Figure 4.33 Edit User

User Group Management was used to add, edit, or delete the group of user and set the data access level of each user group. Its interface was shown in Figure 4.34.

Dashboard | จัดการข้อมูลหลักสูตร | View | Service Api | Referent Data | Administration

ส่วนงาน : Admin

- User Account
- User Group**
- App log

System User Group

รหัส	name	created
1	Admin	
9	ผู้ดูแลข้อมูลในแต่ละมหาวิทยาลัย	2011-04-03 23:32:31
12	test	2011-05-01 21:19:49
13	test2	2011-05-01 21:25:27

Figure 4.34 User Group Management

The data access level could be edited by selecting user group name that was shown in Figure 4.35 and 4.36.

Figure 4.35 Edit User Group

Figure 4.36 Group Permissions

System Application Log was used to view Log file of user action in the system. It interface was shown in Figure 4.37.

System Applog						
วันที่	refId	refType	code	message	message	remark
1236	1002	Sam_Model_User		update user id:1002	userid=1002,loginid=admin,groupid=1,firstname=admin,lastname=Webmaster,pwd=03:15:57,status=Yes,master=No	
1235	1007	Sam_Model_User		update user id:1007	userid=1007,loginid=test,groupid=14,firstname=Test,lastname=Test,pwd=1234,gr03:15:08,status=Yes,master=No	
1234	707	Sam_Model_User_Group_Right		insert user_group_right id:707	groupid=14,resource=view.fac-in-univ,action=delete	
1233	706	Sam_Model_User_Group_Right		insert user_group_right id:706	groupid=14,resource=view.fac-in-univ,action=new	
1232	705	Sam_Model_User_Group_Right		insert user_group_right id:705	groupid=14,resource=view.fac-in-univ,action=edit	

Figure 4.37 System AppLog

4.5 Testing

This section presented the result of system testing of this research. In testing uses SoapUI for testing all Web service API. SoapUI is an open source web service testing tool for service oriented architectures.

Testing Result of SoapUI was shown in Table 4.10 to 4.20 and testing step was shown in Appendix B.

Table 4.10 Test program

Resource Name	program
URI	http://curriculum-data.sonaradev.com/service/api/ref/program
SQL Command	SELECT * FROM (SELECT `PROGRAM_ID` , `PROGRAM_ID_NEW` , `PROGRAM_LANG` , `PROGRAM_NAME` FROM (select * from ref_program) inerTable) AS subTable ORDER BY PROGRAM_ID asc LIMIT 10,10
Result	<pre> <data> <header> <field>4</field> <numberOfResults>10</numberOfResults> <totalRecords>6511</totalRecords> <orderBy>0</orderBy> <orderType>asc</orderType> <page>2</page> <totalpage>652</totalpage> <pagelen>10</pagelen> <filter/> <viewingRows>11 - 20</viewingRows> <fields>programId,programIdNew,programLang,programName</fields> <pk>PROGRAM_ID</pk> <uri>test=1</uri> <sql>SELECT * FROM (SELECT `PROGRAM_ID` , `PROGRAM_ID_NEW` , `PROGRAM_LANG` , `PROGRAM_NAME` <processtime>3.48033905029</processtime> </header> <items> <item id="0010"> <programId>0010</programId> <programIdNew>000010</programIdNew> <programLang>th</programLang> <programName>หลักสูตรเศรษฐศาสตร</programName> </item> <item id="0011"> <programId>0011</programId> <programIdNew>000011</programIdNew> <programLang>th</programLang> <programName>หลักสูตรวิทยาศาสตร</programName> </item> <item id="0012"> <programId>0012</programId> <programIdNew>000012</programIdNew> </pre>
Test Result	Correct

Table 4.11 Test ref-inter-type


Resource Name	ref-inter-type
URI	http://curriculum-data.sonaradev.com/service/api/ref/inter
SQL Command	SELECT * FROM (SELECT `CURR_INTER_ID` , `CURR_INTER_NAME` FROM (select * from ref_curr_inter) inerTable) AS subTable ORDER BY CURR_INTER_ID asc LIMIT 0,10
Result	 <pre> <data> <header> <field>2</field> <numberOfResults>4</numberOfResults> <totalRecords>4</totalRecords> <orderBy>0</orderBy> <orderType>asc</orderType> <page>1</page> <totalpage>1</totalpage> <pagelen>10</pagelen> <filter/> <viewingRows>1 - 10</viewingRows> <fields>currInterId,currInterName</fields> <pk>CURR_INTER_ID</pk> <uri>test=1</uri> <sql>SELECT * FROM (SELECT `CURR_INTER_ID` , `CURR_INTER_NAME` FROM (select * from ref. <processtime>0.250334978104</processtime> </header> <items> <item id="0"> <currInterId>0</currInterId> <currInterName>หลักสูตรภาษาไทย</currInterName> </item> <item id="1"> <currInterId>1</currInterId> <currInterName>หลักสูตรภาษาต่างประเทศ</currInterName> </item> <item id="2"> <currInterId>2</currInterId> <currInterName>หลักสูตรนานาชาติ</currInterName> </item> <item id="3"> <currInterId>3</currInterId> <currInterName>หลักสูตรสองภาษา</currInterName> </item> </items> </pre>
Test Result	Correct

Table 4.12 Test curriculum

Resource Name	curriculum
URI	http://curriculum-data.sonaradev.com/service/api/curriculum
SQL Command	<pre>SELECT * FROM (SELECT `CREDIT` , `CURR_ID` , `CURR_INTER_ID` , `CURR_INTER_NAME` , `CURR_MODIFY` , `CURR_TYPE_ID` , `CURR_TYPE_NAME` , `DEGREE_NAME` , `FAC_ID` , `FAC_NAME` , `GRADE_MIN` , `id` , `ISCED_ID` , `ISCED_NAME_ENG` , `ISCED_NAME_TH` , `LECTURER_PERM_1` , `LECTURER_PERM_2` , `LECTURER_PERM_3` , `LECTURER_PERM_4` , `LECTURER_PERM_5` , `LEV_ID` , `LEV_NAME_ENG` , `LEV_NAME_TH` , `OCSC_REC_DATE` , `OEC_PERMIT_DATE` , `PROGRAM_ID` , `PROGRAM_ID_NEW` , `PROGRAM_LANG` , `PROGRAM_NAME` , `REGULAR_YEAR` , `SEMESTER` , `SEMESTER_ADMIT` , `SEMESTER_YEAR` , `STD_ENTRANCE` , `STUDY_TIME_ID` , `STUDY_TIME_NAME` , `STUDY_TYPE_ID` , `STUDY_TYPE_NAME` , `UNIV_ADDRESS` , `UNIV_FAX` , `UNIV_ID` , `UNIV_NAME_ENG` , `UNIV_NAME_TH` , `UNIV_PRESIDENT_NAME` , `UNIV_TEL` , `UNIV_WEBSITE` , `YEAR` FROM (select * from uoc_curr_view where LEV_ID IN (60,80)) innerTable) AS subTable ORDER BY CREDIT asc LIMIT 10,10</pre>
Result	 <pre>data> <header> <field>47</field> <numberofResults>10</numberofResults> <totalRecords>3756</totalRecords> <orderBy>0</orderBy> <orderType>asc</orderType> <page>2</page> <totalpage>376</totalpage> <pagenum>10</pagenum> <filter/> <viewingRows>11 - 20</viewingRows> <fields>credit,currId,currInterId,currInterName,currModify,currTypeId,currTypeName,degreeName,facId,facName,gradeMin,iscId,iscNameEng,levId,levNameEng,levNameTh,ocscRecDate,oecPermitDate,programId,programIdNew,programLang,programName,regularYear,semester,semesterAdmit,semesterYear,stdEntrance,studyTimeId,studyTimeName,studyTypeId,studyTypeName,univAddress,univFax,univId,univNameEng,univNameTh,univPresidentName,univTel,univWebsite,year</fields> <pk>id</pk> <uri>test=1</uri> <sql>SELECT * FROM (SELECT `CREDIT` , `CURR_ID` , `CURR_INTER_ID` , `CURR_INTER_NAME` , `CURR_MODIFY` , `CURR_TYPE_ID` , `CURR_TYPE_NAME` , `DEGREE_NAME` , `FAC_ID` , `FAC_NAME` , `GRADE_MIN` , `id` , `ISCED_ID` , `ISCED_NAME_ENG` , `ISCED_NAME_TH` , `LECTURER_PERM_1` , `LECTURER_PERM_2` , `LECTURER_PERM_3` , `LECTURER_PERM_4` , `LECTURER_PERM_5` , `LEV_ID` , `LEV_NAME_ENG` , `LEV_NAME_TH` , `OCSC_REC_DATE` , `OEC_PERMIT_DATE` , `PROGRAM_ID` , `PROGRAM_ID_NEW` , `PROGRAM_LANG` , `PROGRAM_NAME` , `REGULAR_YEAR` , `SEMESTER` , `SEMESTER_ADMIT` , `SEMESTER_YEAR` , `STD_ENTRANCE` , `STUDY_TIME_ID` , `STUDY_TIME_NAME` , `STUDY_TYPE_ID` , `STUDY_TYPE_NAME` , `UNIV_ADDRESS` , `UNIV_FAX` , `UNIV_ID` , `UNIV_NAME_ENG` , `UNIV_NAME_TH` , `UNIV_PRESIDENT_NAME` , `UNIV_TEL` , `UNIV_WEBSITE` , `YEAR` FROM (select * from uoc_curr_view where LEV_ID IN (60,80)) innerTable) AS subTable ORDER BY CREDIT asc LIMIT 10,10</sql> <processtime>2.89976406097</processtime> </header> <items> <item id="8470"> <credit>0</credit> <currId>0029</currId> <currInterId>0</currInterId> <currInterName>หลักสูตรนานาชาติ</currInterName> <currModify>01/05/2549</currModify> <currTypeId>1</currTypeId> <currTypeName>หลักสูตรบัณฑิต</currTypeName> <degreeName>ปริญญาตรี</degreeName> <facId>00132</facId> <facName>คณะศึกษาศาสตร์</facName> <gradeMin>3.0</gradeMin> <id>8470</id> <iscId>1000</iscId> <iscNameEng>Education</iscNameEng></pre>
Test Result	Correct

Table 4.13 Test ref-univ


Resource Name	ref-univ
URI	http://curriculum-data.sonaradev.com/service/api/ref/univ
SQL Command	SELECT * FROM (SELECT `UNIV_ID` , `UNIV_NAME_TH` FROM (select * from ref_univ) inertable) AS subTable ORDER BY UNIV_ID asc LIMIT 10,10
Result	 <pre> <data> <header> <field>2</field> <numberOfResults>10</numberOfResults> <totalRecords>297</totalRecords> <orderBy>0</orderBy> <orderType>asc</orderType> <page>2</page> <totalpage>30</totalpage> <pagenum>10</pagenum> <filter/> <viewingRows>11 - 20</viewingRows> <fields>univId,univNameTh</fields> <pk>UNIV_ID</pk> <uri>test=1</uri> <sql>SELECT * FROM (SELECT `UNIV_ID` , `UNIV_NAME_TH` FROM (select * from ref_univ) inertable) . <processtime>0.261344909668</processtime> </header> <items> <item id="00208"> <univId>00208</univId> <univNameTh>มหาวิทยาลัยเกษตรศาสตร์ ราชบุรี</univNameTh> </item> <item id="00209"> <univId>00209</univId> <univNameTh>มหาวิทยาลัยเกษตรศาสตร์ แพร่</univNameTh> </item> <item id="00300"> <univId>00300</univId> <univNameTh>มหาวิทยาลัยขอนแก่น</univNameTh> </item> <item id="00301"> <univId>00301</univId> <univNameTh>มหาวิทยาลัยขอนแก่น วิทยาเขตหนองคาย</univNameTh> </item> </items> </data> </pre>
Test Result	Correct

Table 4.14 Test ref-fac

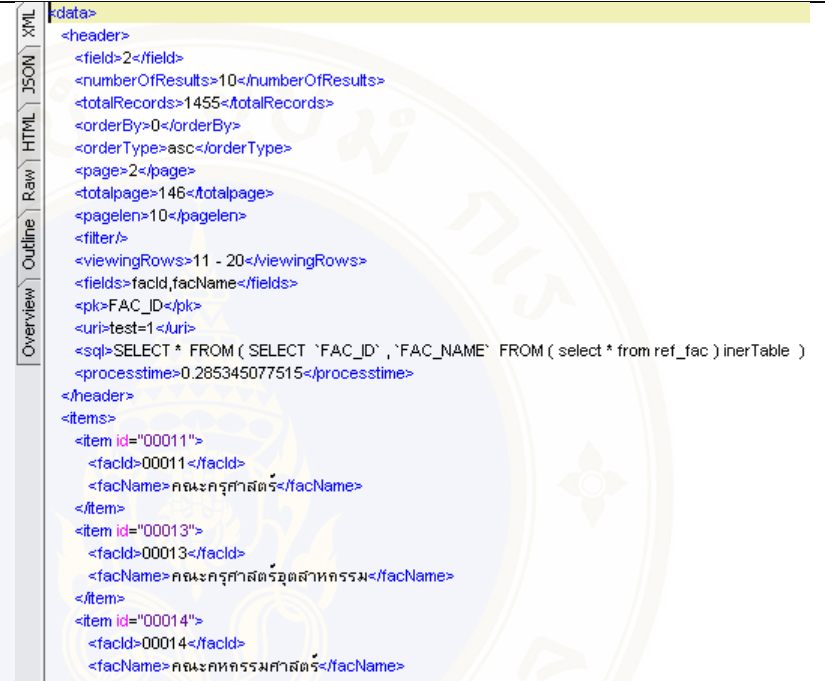
Resource Name	ref-fac
URI	http://curriculum-data.sonaradev.com/service/api/ref/fac
SQL Command	SELECT * FROM (SELECT `FAC_ID` , `FAC_NAME` FROM (select * from ref_fac) inerTable) AS subTable ORDER BY FAC_ID asc LIMIT 10,10
Result	 <pre> <data> <header> <field>2</field> <numberOfResults>10</numberOfResults> <totalRecords>1455</totalRecords> <orderBy>0</orderBy> <orderType>asc</orderType> <page>2</page> <totalpage>146</totalpage> <pagelen>10</pagelen> </header> <filter/> <viewingRows>11 - 20</viewingRows> <fields>facId,facName</fields> <pk>FAC_ID</pk> <uri>test=1</uri> <sql>SELECT * FROM (SELECT `FAC_ID` , `FAC_NAME` FROM (select * from ref_fac) inerTable) <processtime>0.285345077515</processtime> </header> <items> <item id="00011"> <facId>00011</facId> <facName>คณะครุศาสตร์</facName> </item> <item id="00013"> <facId>00013</facId> <facName>คณะครุศาสตร์อุตสาหกรรม</facName> </item> <item id="00014"> <facId>00014</facId> <facName>คณะวิศวกรรมศาสตร์</facName> </item> </items> </pre>
Test Result	Correct

Table 4.15 Test ref-lev

Resource Name	ref-lev
URI	http://curriculum-data.sonaradev.com/service/api/ref/level
SQL Command	SELECT * FROM (SELECT `LEV_ID` , `LEV_NAME_ENG` , `LEV_NAME_TH` FROM (select * from ref_lev where LEV_ID IN (40, 60, 80) innerTable) AS subTable ORDER BY LEV_ID asc LIMIT 0,10
Result	<pre> <data> <header> <field>3</field> <numberOfResults>3</numberOfResults> <totalRecords>3</totalRecords> <orderBy>0</orderBy> <orderType>asc</orderType> <page>1</page> <totalpage>1</totalpage> <pagelen>10</pagelen> <filter /> <viewingRows>1 - 10</viewingRows> <fields>levId,levNameEng,levNameTh</fields> <pk>LEV_ID</pk> <uri>test=1</uri> <sql>SELECT * FROM (SELECT `LEV_ID` , `LEV_NAME_ENG` , `LEV_NAME_TH` FROM (select * f <processtime>0.263208866119</processtime> </header> <items> <item id="40"> <levId>40</levId> <levNameEng>Bachelor Degree</levNameEng> <levNameTh>ปริญญาตรี</levNameTh> </item> <item id="60"> <levId>60</levId> <levNameEng>Master Degree</levNameEng> <levNameTh>ปริญญาโท</levNameTh> </item> </items> </pre>
Test Result	Correct

Table 4.16 Test ref-curr

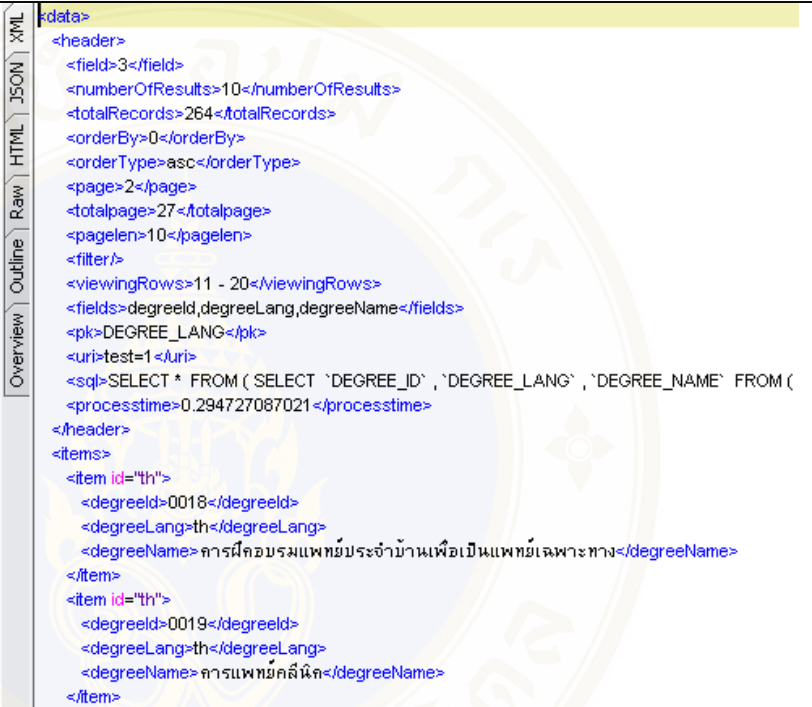
Resource Name	ref-curr
URI	http://curriculum-data.sonaradev.com/service/api/ref/curr
SQL Command	SELECT * FROM (SELECT `DEGREE_ID` , `DEGREE_LANG` , `DEGREE_NAME` FROM (select * from ref_curr) inerTable) AS subTable ORDER BY DEGREE_ID asc LIMIT 10,10
Result	 <pre> <data> <header> <field>3</field> <numberOfResults>10</numberOfResults> <totalRecords>264</totalRecords> <orderBy>0</orderBy> <orderType>asc</orderType> <page>2</page> <totalpage>27</totalpage> <pagelen>10</pagelen> <filter/> <viewingRows>11 - 20</viewingRows> <fields>degreeId,degreeLang,degreeName</fields> <pk>DEGREE_LANG</pk> <uri>test=1</uri> <sql>SELECT * FROM (SELECT `DEGREE_ID` , `DEGREE_LANG` , `DEGREE_NAME` FROM (<processtime>0.294727087021</processtime> </header> <items> <item id="th"> <degreeId>0018</degreeId> <degreeLang>th</degreeLang> <degreeName>การฝึกอบรมแพทย์ประจำบ้านเพื่อเป็นแพทย์เฉพาะทาง</degreeName> </item> <item id="th"> <degreeId>0019</degreeId> <degreeLang>th</degreeLang> <degreeName>การแพทย์คลินิก</degreeName> </item> </items> </pre>
Test Result	Correct

Table 4.17 Test curr-type

Resource Name	curr-type
URI	http://curriculum-data.sonaradev.com/service/api/ref/curr-type
SQL Command	SELECT * FROM (SELECT `CURR_TYPE_ID` , `CURR_TYPE_NAME` FROM (select * from ref_curr_type) inerTable) AS subTable ORDER BY CURR_TYPE_ID asc LIMIT 0,10
Result	<div style="border: 1px solid #ccc; padding: 5px;"> <div style="display: flex; border-bottom: 1px solid #ccc;"> <div style="width: 30px; text-align: center; font-size: 8px; border-right: 1px solid #ccc;">Overview</div> <div style="width: 30px; text-align: center; font-size: 8px; border-right: 1px solid #ccc;">Outline</div> <div style="width: 30px; text-align: center; font-size: 8px; border-right: 1px solid #ccc;">Raw</div> <div style="width: 30px; text-align: center; font-size: 8px; border-right: 1px solid #ccc;">HTML</div> <div style="width: 30px; text-align: center; font-size: 8px; border-right: 1px solid #ccc;">JSON</div> <div style="width: 30px; text-align: center; font-size: 8px;">XML</div> </div> <pre style="margin-top: 5px; font-family: monospace; font-size: 10px;"> <data> <header> <field>2</field> <numberOfResults>2</numberOfResults> <totalRecords>2</totalRecords> <orderBy>0</orderBy> <orderType>asc</orderType> <page>1</page> <totalpage>1</totalpage> <pagelen>10</pagelen> <filter/> <viewingRows>1 - 10</viewingRows> <fields>currTypeId,currTypeName</fields> <pk>CURR_TYPE_ID</pk> <uri>test=1</uri> <sql>SELECT * FROM (SELECT `CURR_TYPE_ID` , `CURR_TYPE_NAME` FROM (select * from r <processtime>0.294479131699</processtime> </header> <items> <item id="1"> <currTypeId>1</currTypeId> <currTypeName>หลักสูตรมกค</currTypeName> </item> <item id="3"> <currTypeId>3</currTypeId> <currTypeName>หลักสูตรต่อเนื่อง</currTypeName> </pre> </div>
Test Result	Correct

Table 4.18 Test ref-study-time


Resource Name	ref-study-time
URI	http://curriculum-data.sonaradev.com/service/api/ref/ref-study-time
SQL Command	SELECT * FROM (SELECT `STUDY_TIME_ID` , `STUDY_TIME_NAME` FROM (select * from ref_study_time) inerTable) AS subTable ORDER BY STUDY_TIME_ID asc LIMIT 0,10
Result	 <pre> <data> <header> <field>2</field> <numberOfResults>4</numberOfResults> <totalRecords>4</totalRecords> <orderBy>0</orderBy> <orderType>asc</orderType> <page>1</page> <totalpage>1</totalpage> <pagen>10</pagen> <filter/> <viewingRows>1 - 10</viewingRows> <fields>studyTimeId,studyTimeName</fields> <pk>STUDY_TIME_ID</pk> <uri>test=1</uri> <sql>SELECT * FROM (SELECT `STUDY_TIME_ID` , `STUDY_TIME_NAME` FROM (select * from ref <processtime>0.263836145401</processtime> </header> <items> <item id="1"> <studyTimeId>1</studyTimeId> <studyTimeName>วันธรรมดากลางวัน</studyTimeName> </item> <item id="2"> <studyTimeId>2</studyTimeId> <studyTimeName>วันธรรมดาคาเย็น</studyTimeName> </item> <item id="3"> <studyTimeId>3</studyTimeId> </item> </items> </pre>
Test Result	Correct

Table 4.19 Test subject



Resource Name	ref_subject
URI	http://curriculum-data.sonaradev.com/service/api/ref/subject
SQL Command	SELECT * FROM (SELECT `subject_code` , `subject_credit` , `subject_detail_en` , `subject_detail_th` , `subject_id` , `subject_lab` , `subject_lecture` , `subject_name_eng` , `subject_name_th` , `subject_selfstudy` , `subject_type` , `uoc_id` FROM (select * from ref_subject) inerTable) AS subTable ORDER BY subject_code asc LIMIT 10,10
Result	 <pre> <data> <header> <field>12</field> <numberOfResults>9</numberOfResults> <totalRecords>19</totalRecords> <orderBy>0</orderBy> <orderType>asc</orderType> <page>2</page> <totalpage>2</totalpage> <pagen>10</pagen> <filter/> <viewingRows>11 - 20</viewingRows> <fields>subjectCode,subjectCredit,subjectDetailEn,subjectDetailTh,subjectId,subjectLab,subjectLecture,subject <pk>subject_id</pk> <uri>test=1</uri> <sql>SELECT * FROM (SELECT `subject_code` , `subject_credit` , `subject_detail_en` , `subject_detail_th` , `s <processtime>0.251999855042</processtime> </header> <items> <item id="5"> <subjectCode>322748</subjectCode> <subjectCredit>3</subjectCredit> <subjectDetailEn/> <subjectDetailTh/> <subjectId>5</subjectId> <subjectLab>0</subjectLab> <subjectLecture>3</subjectLecture> <subjectNameEng>Information Technology Security</subjectNameEng> <subjectNameTh> ความมั่นคงทางเทคโนโลยีสารสนเทศ</subjectNameTh> <subjectSelfstudy>6</subjectSelfstudy> </item> </items> </pre>
Test Result	Correct

Table 4.20 Test ref-iscd

Resource Name	ref-iscd
URI	http://curriculum-data.sonaradev.com/service/api/ref/iscd
SQL Command	SELECT * FROM (SELECT `ISCED_ID` , `ISCED_NAME_ENG` , `ISCED_NAME_TH` FROM (select * from ref_iscd) innerTable) AS subTable ORDER BY ISCED_ID asc LIMIT 10,10
Result	 <pre> <data> <header> <field>3</field> <numberOfResults>10</numberOfResults> <totalRecords>253</totalRecords> <orderBy>0</orderBy> <orderType>asc</orderType> <page>2</page> <totalpage>26</totalpage> <pagelen>10</pagelen> <filter/> <viewingRows>11 - 20</viewingRows> <fields>iscedId,iscedNameEng,iscedNameTh</fields> <pk>ISCED_ID</pk> <uri>test=1</uri> <sql>SELECT * FROM (SELECT `ISCED_ID` , `ISCED_NAME_ENG` , `ISCED_NAME_TH` FROM (select <processtime>0.23296880722</processtime> </header> <items> <item id="1009"> <iscedId>1009</iscedId> <iscedNameEng>curriculum development in non-vocational and vocational subjects</iscedNameEng> <iscedNameTh>การพัฒนากลยุทธ์ระดับวิชาชีพและไม่วิชาชีพ</iscedNameTh> </item> <item id="1010"> <iscedId>1010</iscedId> <iscedNameEng>Educational assessment</iscedNameEng> <iscedNameTh>ประเมินผลการศึกษา</iscedNameTh> </item> <item id="1011"> </pre>
Test Result	Correct

CHAPTER V

DISCUSSIONS

The proposed curriculum information retrieval system was intended to support the person, who would like to search the information for further study. Commission of Higher Education database had been utilized in the study. These were initial information that would be supported the further study decision making.

The study of the comparison between web application and web service in technology and operation showed that they use HTTP protocol or internet for communication, but their objective was different. Web application used HTTP protocol or internet for exchanging files between web servers but web service used them for exchanging services (software components) between information systems through web server. Generally, web application was used in interaction with users by web browser for presenting the information and making the transaction. Web service was used in interaction with web server for exchanging information and operation, or providing services across the systems by web application or application interface to communicate with users. Moreover, web service could be used in many systems but web application could not be directly used in many systems.

The important factor of developing the system by web service instead of web application is that web service used XML language in data communication. The XML language could be used in interaction between multiplatform and multi-programming language system because the XML is standard language that there is the function for using and manage the XML in every programming language.

There were two kinds of popular web service protocols: SOAP and REST. This research developed the searching system that required high performance protocol and low overhead for fast returning result to users. The study in the comparison between these two protocols showed the REST protocol was compatibility in this system because the REST protocol was good performance and low overhead. The cause of these was the REST protocol sent the method and parameter values through URL while the sending and receiving process of the SOAP protocol was more difficult

and complicate from the encoding and decoding of data. In the case of more data, the SOAP protocol was very slow working more than the REST protocol.

The study presented conceptual of implementation on web service that used to develop curriculum information retrieval system. The system be applied the curriculum database that provided from The Office of the Higher Education Commission. This database provided information search resource for person, who would like to educate, especially students. The ability of developed system would be organized as follow:

5.1 The Advantage of System

5.1.1 For User

- User can access the information of postgraduate curriculums from one source.

- User would be received up-to-date information for extending education.

5.1.2 For Developer

- Reduce cost for developing and improvement the system. Because of the data transfer format was XML that would be enhance the ability on continuous improvement of information connection. There improvements were not restrictions on the structure.

- Reduce time for developing application caused no need coding for data query.

- Less time for developed database management.

5.2 Limitation of System

The system needs more time for the large data searching than developed system via web application because step of data format conversion is more.

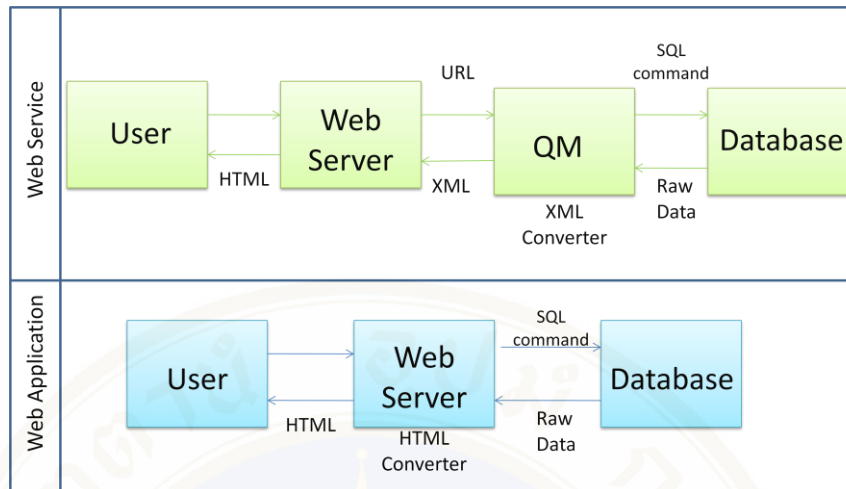


Figure 5.1 Web service vs. Web application

From the figure 5.1 showed steps of information retrieval. There are 5 steps for retrieval of developed system via web service as following;

- 1) Database sent information with raw data to Query Manager.
- 2) Query Manager converted the raw data to XML structure.
- 3) Query Manager sent XML data to Web Server.
- 4) Web server converted XML data again becoming HTML structure.
- 5) The data with HTML structure was presented to users.

In comparison, developed system via web service is more step of conversion than developed via web application that information retrieval steps of web application system as following;

- 1) Database sent information with raw data to web server.
- 2) Web server converted the raw data becoming HTML structure.
- 3) The data with HTML structure was presented to users.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

A requirement of present system development is remote accessibility for more convenient. Web service technology is popular used in system development because it can provide this requirement. An application can use web service by internet protocol and interacting message between users and providers that through XML document. These are causes that web service can be used for cross platform integration.

This research used web service technology in developing information searching system for curriculum database of The Office of the Higher Education Commission. This system could provide curriculum information center for whom wants to study in postgraduate. The user was divided into three groups; User, Authorized user, and Administrator that were described below:

1. User group can use the system to view curriculum information by curriculum name searching.
2. Authorized User group can manage information that belongs to own university.
3. Administrator group can manage overall information in database and web service API.

The objective of this research was to develop prototype system of graduate information and website for searching curriculum information that would become curriculum information center. The system could provide curriculum and department information of master degree and doctorate in universities that belong to The Office of the Higher Education Commission. The curriculum information center composed of three main systems that was shown as follows:

1. Client Demo was website system for searching curriculum information.

2. Rest API was web service system that showed exist service libraries. Developer could use this main system to review web service in system development.

3. Service Management was database management system for Administrator and Authorized User group to update or edit curriculum information of each university. The functions of this main system were composed of:

- 1) Curriculum Management was used to add, edit, or delete curriculum and university data.
- 2) View was used to present curriculum name in university.
- 3) Service API was used to add, close, or delete Service and XML Schema. There was Request Log to present log file of using Service history.
- 4) Reference Data was used to manage reference table.
- 5) Administration was used to manage user data and user access level and present log file of user action history.

The system design in research describes the system development by web service for searching graduate curriculum information from database of The Office of the Higher Education Commission. The structure of this system composed of three layers; Data layer, Application layer, and Client layer.

- Data layer composed of data source and database management. There was Query Manager for connection between data resource and Application layer through REST web service in XML standard format. This research chose REST because of low overhead and response time than the other web service technology. The Query Manager can be used to solve a variety of technology and increase performance and chance of data connection without limitation of system structure or application. These would cause less effect in system extension.

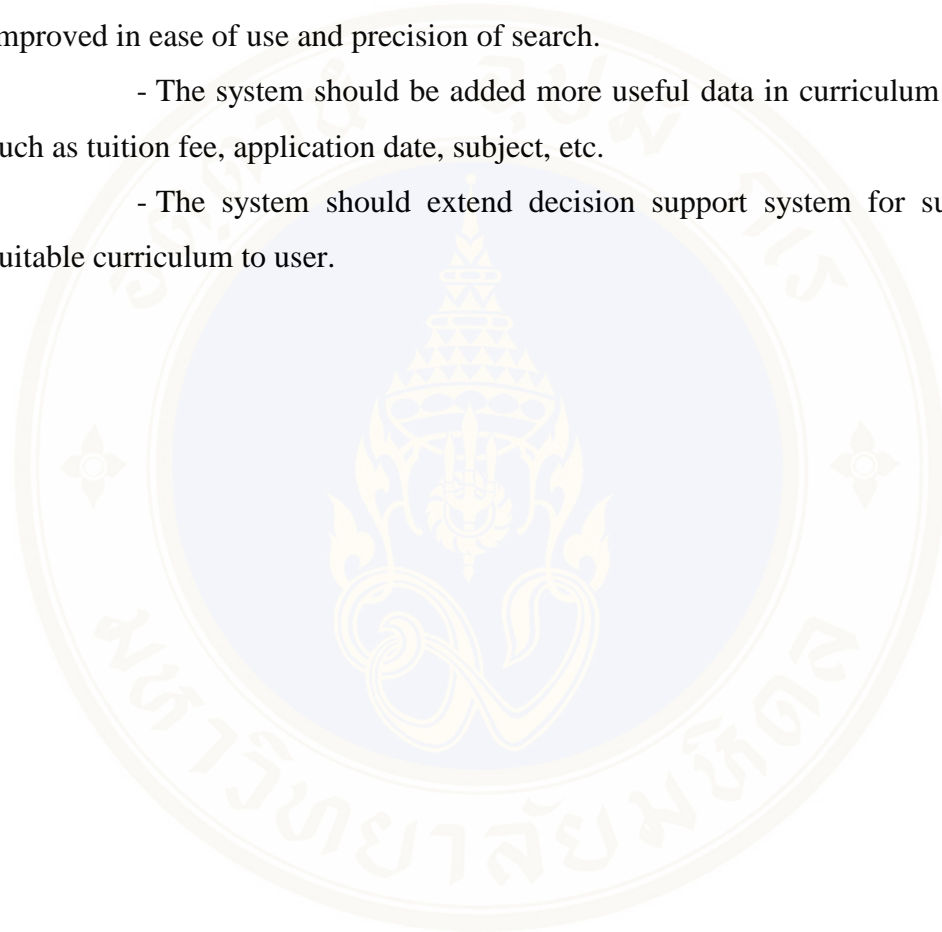
- Application layer was a core of the system. This layer was responsible to query processing from user and send service request to Query Manager. The system was designed to flexible support extended Query Manager and new data in the future.

- Client layer was used for user connection through internet. User could access the system by web browser or application that was developed with web service API.

6.2 Recommendations

Developing of the system could achieve all targeted objectives. However, more development and improvement were still in needs and these can be summarized as follows:

- This searching system of curriculum information should be more improved in ease of use and precision of search.
- The system should be added more useful data in curriculum information such as tuition fee, application date, subject, etc.
- The system should extend decision support system for suggestion of suitable curriculum to user.



REFERENCES

1. Robert R. Korfhage. "Information Storage and Retrieval". USA: Wiley Computer Publishing, 1997.
2. Nickull, D., et al. (2007). *Service Oriented Architecture (SOA) and Specialized Messaging Patterns*. Technical white paper by Adobe, 2-3
3. W3C. "Web Services Architecture", W3C Working Group Note 11 February 2004. (<http://www.w3.org/TR/ws-arch>)
4. Michael Papazoglou. *Web services: Principle and Technology*. Prentice Hall, July 2007.
5. L. Richardson and S. Ruby. "RESTful Web Services". Published by O'Reilly Media, Inc., 2007.
6. Michael zur Muehlen, Jeffrey V. Nickerson and Keit D. Swenson. "Developing Web Services Choreography Standards- The Case of REST vs. SOAP". *Decision Support System* 37(2004), Elsevier, North Holland.
7. Roy Thomas Fielding. "Architectural Styles and the Design of Network-based Software Architectures", PhD Thesis. University of California, Irvine, 2000.
8. Thomas Ert. "Service-Oriented Architecture A Field Guide to Integration XML and Web Service". Prentice Hall RTP, 2004, ISBN 0-13-142898-5
9. Hatem Hamad, Motaz Saad and Ramzi Abeb. "Performance Evaluation of RESTful Web Services for Mobile Devices". paper presented in *International Arab of e-Technology*, Vol.1 No.3, January 2010
10. Jan Newmarch. "A RESTful Approach: Clean UPnP without SOAP". Monash University, 2009
11. Udomdet Tarahom and Kittima Makabunchakij. "A Web Service Design for University Services Management". paper presented in WUNCA (Workshop on UniNet Network and Computer Application), 23 Mar 2010

12. Wisit Wannaphumi and Sipanee Nuchiprasitchai. "Semantic Web Services Discovery System". Paper presented in The 5th National Conference on Computing and Information Technology, 22 May 2009
13. Tian Feng. "The Heterogeneous Data Integration Based on XML in Coal Enterprise". Computer Science and Computational Technology, 2008. ISCSCT '08. International Symposium on 20-22 Dec. 2008, Volume: 1, pages 438-441, 2008.





APPENDIX A

DATA DICTIONARY

Table A.1 Table UOC_CURR

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	YEAR	Text	4	ปีการศึกษา (ปีการศึกษาที่สกอ.กำหนดให้ส่ง)
2	SEMESTER	Text	1	ภาคการศึกษา (ภาคการศึกษาที่สกอ.กำหนดให้ส่ง)
3	UNIV_ID	Text	5	รหัสสถานศึกษา
4	CURR_ID	Text	4	รหัสชื่อหลักสูตร (ตามมาตรฐานกลาง)
5	PROGRAM_ID	Text	6	รหัสสาขาวิชา
6	DEGREE_NAME	Text	150	ชื่อปริญญา (ภาษาไทยหรือภาษาอังกฤษ)
7	CURR_TYPE_ID	Text	1	รหัสประเภทหลักสูตร
8	STUDY_TYPE_ID	Text	1	รหัสประเภทการเรียน
9	STUDY_TIME_ID	Text	1	รหัสประเภทเวลาเรียน
10	FAC_ID	Text	5	รหัสคณะที่เปิดหลักสูตร
11	ISCED_ID	Text	4	รหัสกลุ่มสาขาวิชา ISCED
12	LEV_ID	Text	2	รหัสระดับการศึกษา
13	CURR_MODIFY	Text	10	วัน เดือน ปีที่ปรับปรุงหลักสูตรครั้งสุดท้าย (พ.ศ.)
14	SEMESTER_YEAR	Text	1	จำนวนภาคการศึกษาของหลักสูตรต่อปี
15	CREDIT	Text	3	จำนวนหน่วยกิตตามหลักสูตร
16	GRADE_MIN	Text	4	ระดับเกรดต่ำสุดที่สามารถจบได้ของหลักสูตร
17	REGULAR_YEAR	Text	3	จำนวนปีปกติของหลักสูตร
18	CURR_INTER_ID	Text	1	รหัสความสากลของหลักสูตร
19	STD_ENTRANCE	Text	7	จำนวนนักศึกษาที่รับเข้าโดยประมาณการแต่ละปี
20	OEC_PERMIT_DATE	Text	10	วันที่สภานุมัติหลักสูตรครั้งแรก (พ.ศ.)
21	OCSC_REC_DATE	Text	10	วันที่กพ. รับรองหลักสูตรครั้งแรก (พ.ศ.)
22	SEMESTER_ADMIT	Text	7	ภาค / ปี ที่เริ่มรับนักศึกษา

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
23	LECTURE_PERM_1	Text	13	อาจารย์ประจำหลักสูตร คนที่ 1
24	LECTURE_PERM_2	Text	13	อาจารย์ประจำหลักสูตร คนที่ 2
25	LECTURE_PERM_3	Text	13	อาจารย์ประจำหลักสูตร คนที่ 3
26	LECTURE_PERM_4	Text	13	อาจารย์ประจำหลักสูตร คนที่ 4
27	LECTURE_PERM_5	Text	13	อาจารย์ประจำหลักสูตร คนที่ 5

Table A.2 Table REF_UNIV

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	UNIV_ID	Text	5	รหัสมหาวิทยาลัย
2	UNIV_NAME_THAI	Text	150	ชื่อมหาวิทยาลัย (ไทย)
3	UNIV_NAME_ENG	Text	150	ชื่อมหาวิทยาลัย (อังกฤษ)

Table A.3 Table REF_UNIV_ADDRESS

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	UNIV_ID	Text	5	รหัสมหาวิทยาลัย
2	UNIV_WEBSITE	Text	150	เว็บไซต์มหาวิทยาลัย
3	UNIV_PRESIDENT_NAME	Text	150	ชื่ออธิการบดีมหาวิทยาลัย
4	UNIV_ADDRESS	Text	200	ที่อยู่มหาวิทยาลัย
5	UNIV_TEL	Text	50	เบอร์โทรศัพท์มหาวิทยาลัย
6	UNIV_FAX	Text	50	เบอร์โทรสารมหาวิทยาลัย
7	UNIV_COOR1	Text	20	พิกัดละติจูด
8	UNIV_COOR2	Text	20	พิกัดลองจิจูด

Table A.4 Table REF_CURR

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	DEGREE_ID	Text	4	รหัสหลักสูตร
2	DEGREE_NAME	Text	150	ชื่อหลักสูตร (ไทย)
3	DEGREE_LANG	Text	150	ชื่อหลักสูตร (อังกฤษ)

Table A.5 Table REF_PROGRAM

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	PROGRAM_ID	Text	4	รหัสสาขาวิชา
2	PROGRAM_NAME	Text	150	ชื่อสาขาวิชา
3	PROGRAM_ID_NEW	Text	6	รหัสสาขาวิชา (ใหม่)
4	PROGRAM_LANG	Text	10	ภาษาสาขาวิชา

Table A.6 Table REF_ISCED

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	ISCED_ID	Text	4	รหัสกลุ่มสาขาวิชา
2	ISCED_NAME_TH	Text	150	ชื่อกลุ่มสาขาวิชา (ไทย)
3	ISCED_NAME_ENG	Text	150	ชื่อกลุ่มสาขาวิชา (อังกฤษ)

Table A.7 Table REF_LEV

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	LEV_ID	Text	2	รหัสระดับการศึกษา
2	LEV_NAME_TH	Text	150	ชื่อระดับการศึกษา (ไทย)
3	LEV_NAME_ENG	Text	150	ชื่อระดับการศึกษา (อังกฤษ)

Table A.8 Table REF_CURR_TYPE

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	CURR_TYPE_ID	Text	1	รหัสประเภทหลักสูตร
2	CURR_TYPE_NAME	Text	150	ชื่อประเภทหลักสูตร

Table A.9 Table REF_FAC

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	FAC_ID	Text	5	รหัสคณะ
2	FAC_NAME	Text	150	ชื่อคณะ

Table A.10 Table REF_STUDY_TYPE

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	STUDY_TYPE_ID	Text	1	รหัสประเภทการเรียน
2	STUDY_TYPE_NAME	Text	150	ชื่อประเภทการเรียน

Table A.11 Table REF_STUDY_TIME

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	STUDY_TIME_ID	Text	1	รหัสประเภทเวลาเรียน
2	STUDY_TIME_NAME	Text	150	ชื่อประเภทเวลาเรียน

Table A.12 Table REF_CURR_INTER

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	CURR_INTER_ID	Text	1	รหัสประเภทการเรียน
2	CURR_INTER_NAME	Text	150	ชื่อประเภทการเรียน

Table A.13 Table REF_SUBJECT

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	SUBJECT_ID	Text	5	ลำดับวิชา
2	SUBJECT_CODE	Text	6	รหัสวิชา
3	SUBJECT_NAME_TH	Text	150	ชื่อวิชา ไทย
4	SUBJECT_NAME_ENG	Text	150	ชื่อวิชา อังกฤษ
5	SUBJECT_TYPE	Text	1	รหัสประเภทวิชา
6	SUBJECT_CREDIT	Text	2	หน่วยกิต
7	SUBJECT_LECTURE	Text	2	เวลาเรียนทฤษฎี (ชั่วโมง/สัปดาห์)
8	SUBJECT_LAB	Text	2	เวลาเรียนปฏิบัติ (ชั่วโมง/สัปดาห์)
9	SUBJECT_SELFSTUDY	Text	2	เวลาเรียนศึกษาด้วยตนเอง (ชั่วโมง/สัปดาห์)
10	UOC_ID	Text	6	รหัสหลักสูตร
11	SUBJECT_DETAIL_TH	Text	150	รายละเอียดวิชา ไทย
12	SUBJECT_DETAIL_EN	Text	150	รายละเอียดวิชา อังกฤษ

Table A.14 Table REF_SUBJECT_TYPE

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	SUBJECT_TYPE_ID	Text	1	รหัสประเภทวิชา
2	SUBJECT_TYPE_NAME	Text	150	ชื่อประเภทวิชา

Table A.15 Table API_REQUEST_LOG

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	ID	Int	11	ลำดับ Log
2	USER_IP	Text	50	หมายเลข IP ADDRESS
3	USER_SESSION	Text	50	
4	REQUEST_URL	Text	255	URL ที่ถูกเรียกใช้
5	REQUEST_TIME	Datetime		เวลาที่ถูกเรียกใช้
6	RESPONSE_TIME	Datetime		เวลาที่ตอบสนองหลังถูกเรียกใช้

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
7	PROCESS_TIME_SEC	Text	50	เวลาที่ระบบประมวลผล
8	RESPONSE_CODE	Text	10	รหัสการตอบสนองของระบบ
9	RESPONSE_MESSAGE	Text	100	ข้อความตอบสนองของระบบ
10	API_ID	Int	11	รหัส API
11	API_NAME	Text	200	ชื่อ API
12	FIELDS	Text	255	ชื่อฟิลด์
13	NUMBEROFRESULTS	Int	11	จำนวนของผลลัพธ์
14	LOG_TIME	Datetime		เวลาของ log ที่บันทึก
15	TOTALRECORDS	Int	11	จำนวนrecord ที่พบ
16	ORDERBY	Int	11	การเรียงลำดับ
17	ORDERTYPE	Int	11	การเรียกตามชนิด
18	PAGE	Int	11	หน้า
19	TOTALPAGE	Int	11	จำนวนหน้าทั้งหมด
20	PAGELEN	Int	11	จำนวนข้อมูลต่อหน้า
21	XML	Text	255	รายละเอียด XML
22	SQL	Text	255	รายละเอียดคำสั่ง SQL
23	VIEWINGROWS	Text	10	

Table A.16 Table API_PROFILE

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	ID	Int	11	รหัส API
2	NAME	Text	200	ชื่อ API
3	DATABASE	Text	50	ชื่อฐานข้อมูล
4	SQL	Text	255	คำสั่ง SQL
5	DESCRIPTION	Text	255	คำอธิบาย
6	PUBLISH_UP	Date		วันเริ่มใช้งาน
7	PUBLISH_DOWN	Date		วันสิ้นสุด
8	STATE	Int	1	สถานะ

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
9	CREATED	Datetime		วันที่สร้าง
10	CREATE_BY	Text	255	ผู้ที่สร้าง
11	MODIFIED	Datetime		วันที่แก้ไข
12	MODIFIE_BY	Text	255	ผู้ที่แก้ไข
13	HITS	Int	11	จำนวนครั้งที่เรียกดู
14	URL	Text	255	ชื่อ URL
15	XMLSCHEMA	Text	255	XML SCHEMA

Table A.17 Table API_SCHEMA

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	ID	Int	11	ลำดับ API
2	API_ID	Int	11	ลำดับตาราง API PROFILE
3	COLUMN_NAME	Text	200	ชื่อ Column ในฐานข้อมูล
4	ATTRIBUTE_NAME	Text	200	ชื่อ Attribute
5	FILTER	int	1	การเป็น Filter
6	ORDERING	Int	11	ลำดับ
7	TYPE	Text	255	ประเภทข้อมูล
8	DESCRIPTION	Text	255	คำอธิบาย
9	STATE	Int	1	0 = แสดงในXML , 1=ไม่แสดง
10	PUBLISH_UP	Datetime		บันทึกวันที่ใช้งาน
11	PUBLISH_DOWN	Datetime		บันทึกวันที่ยกเลิก
12	CREATED	Datetime		วันที่สร้าง
13	CREATE_BY	Text	255	ผู้สร้าง
14	MODIFIED	Datetime		วันที่แก้ไข
15	MODIFIE_BY	Text	255	ผู้แก้ไข
16	REFERENCE_ATTRIBUTE_ID	Int	11	ID ของตาราง API
17	REFERENCE_RESOURCE_ID	Int	11	ID ของตาราง Resource
18	POSSIBLE_VALUE	Text	255	ค่าที่เป็นไปได้

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
19	DEFAULT	Text	255	ค่าปริยายในกรณีที่ข้อมูลเป็น NULL
20	ISPK	Int	1	ค่าความเป็น Primary Key

Table A.18 Table SYS_APPLICATION_LOG

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	ID	Int	11	ลำดับ
2	REFID	Int	11	หมายเลขRef
3	TYPE	Text	100	ชนิดข้อมูล
4	CODE	Text	100	คำสั่งที่ใช้
5	MESSAGE	Text	255	ข้อความแสดง action
6	REMARK	Text	255	ID ผู้แก้ไข
7	MODULE	Text	100	ส่วนที่
8	CONTROLLER	Text	100	Controller
9	ACTION	Text	100	Action
10	URL	Text	255	URL ที่แก้ไข
11	REMOTE_IP	Text	50	หมายเลข IP ผู้แก้ไข
12	CREATED	Datetime		วันที่สร้าง log
13	CREATE_BY	Text	100	ผู้ที่สร้าง log
14	ACTIVITY	Text	100	คำสั่ง
15	MESSAGE_EN	Text	255	ข้อความ

Table A.19 Table USER

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	USERID	Int	10	รหัสผู้ใช้งาน
2	GROUPID	Text	20	รหัสกลุ่มผู้ใช้งาน
3	LOGINID	Text	75	ID ผู้ใช้งาน
4	FIRSTNAME	Text	75	ชื่อผู้ใช้งาน

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
5	LASTNAME	Text	75	นามสกุลผู้ใช้งาน
6	PWD	Text	50	Password
7	EMAIL	Text	255	E mail
8	EMAILPASS	Text	20	E mail password
9	LASTLOGIN	Datetime		วันที่เข้าใช้ล่าสุด
10	STATUS	Text	10	สถานะผู้ใช้งาน
11	MASTER	Text	10	
12	NOTIFICATIONS	Text	100	การแจ้งเตือน e mail
13	UNIV	Text	200	มหาวิทยาลัยของผู้ใช้งาน

Table A.20 Table GROUP

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	ID	Int	11	รหัสกลุ่มผู้ใช้งาน
2	NAME	Text	255	ชื่อกลุ่มผู้ใช้งาน
3	CREATED	Datetime		วันที่สร้าง
4	CREATED_BY	Int	11	ผู้ที่สร้าง
5	MODIFIED	Datetime		วันที่แก้ไข

Table A.21 Table RESOURCE

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	ID	Int	11	ลำดับ
2	NAME	Text	100	ชื่อของ Resource
3	ACTIONS	Text	255	ชื่อของ Action
4	MODULE	Text	255	ชื่อของ Module
5	DESCRIPTION	Text	255	คำอธิบาย

Table A.22 Table GROUP_RIGHT

ลำดับ	ชื่อรายการข้อมูล	ประเภท	ขนาด	คำอธิบาย
1	ID	Int	11	
2	GROUPID	Int	11	
3	RESOURCE	Text	100	
4	ACTION	Text	100	
5	ISALLOW	Text	5	

APPENDIX B

TESTING WEB SERVICE API

วิธีการทดสอบ REST API

โปรแกรมที่ใช้ทดสอบ SoapUI 3.61 Pro

1. เริ่มจากการสร้าง REST Service ขึ้นมาก่อนโดยการเลือก File แล้วเลือก New soapUI Project ดังรูปที่ B.1

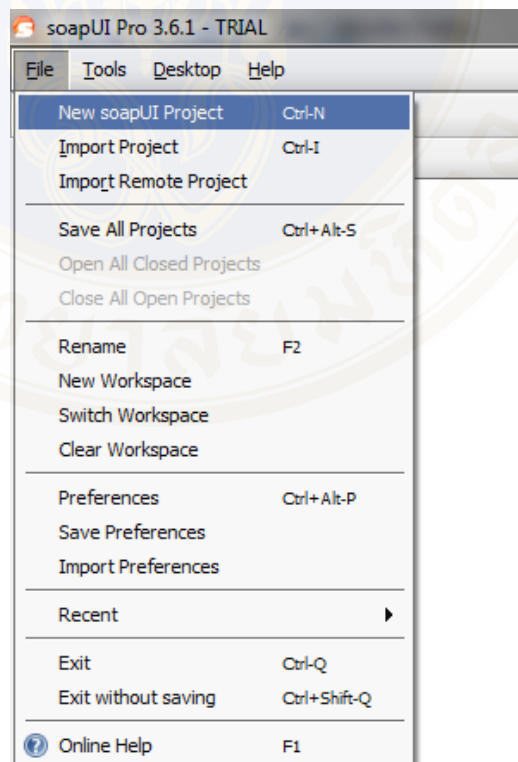


Figure B.1 Menu SoapUI

- โปรแกรมจะแสดงหน้าต่าง New soapUI Project ขึ้นมา ให้ตั้งชื่อ Project ในช่องProject name: และ เลือก Opens dialog to create REST service แล้วกด OK ดังรูปที่ B.2

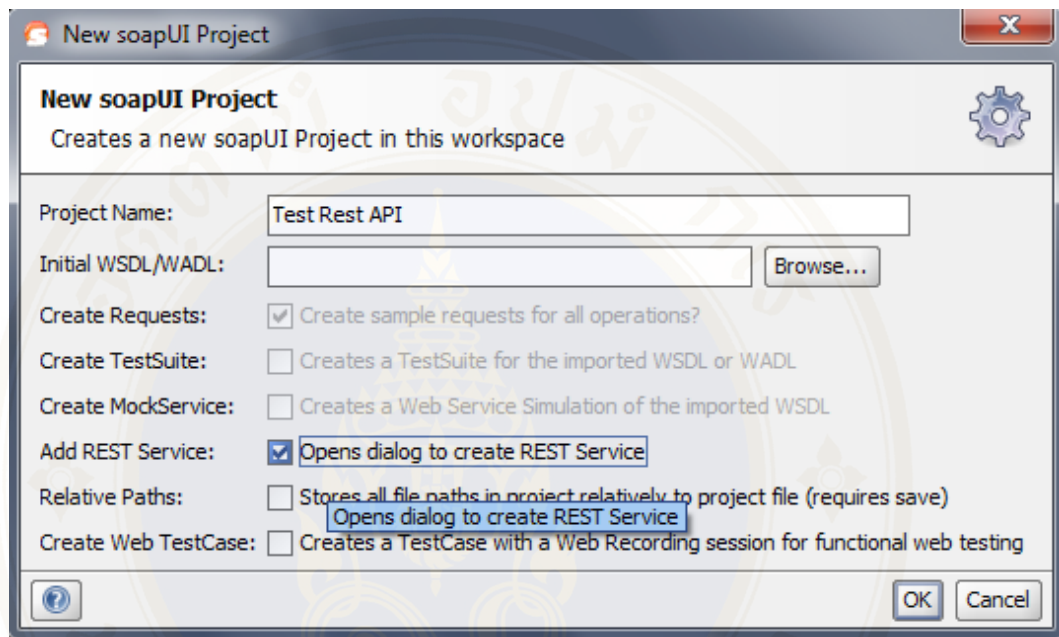


Figure B.2 New soapUI Project

- โปรแกรมจะแสดงหน้าต่าง New REST Service ขึ้นมา ให้ใส่ URL ที่ต้องการทดสอบลงในช่อง Service Endpoint: และ กดเลือก Extract Resource and Method from specified Endpoint แล้วกด OK ดังรูปที่ B.3

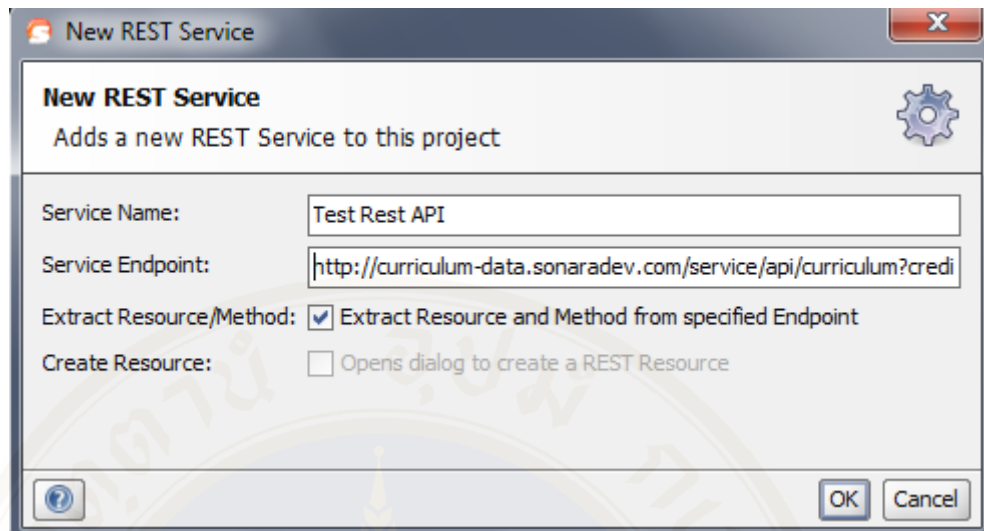


Figure B.3 New REST Service

4. โปรแกรมจะแสดงหน้าต่าง New REST Resource ขึ้นมา ให้ใส่ Parameters ที่ต้องการทดสอบ เมื่อใส่แล้วกด OK ดังรูปที่ B.4

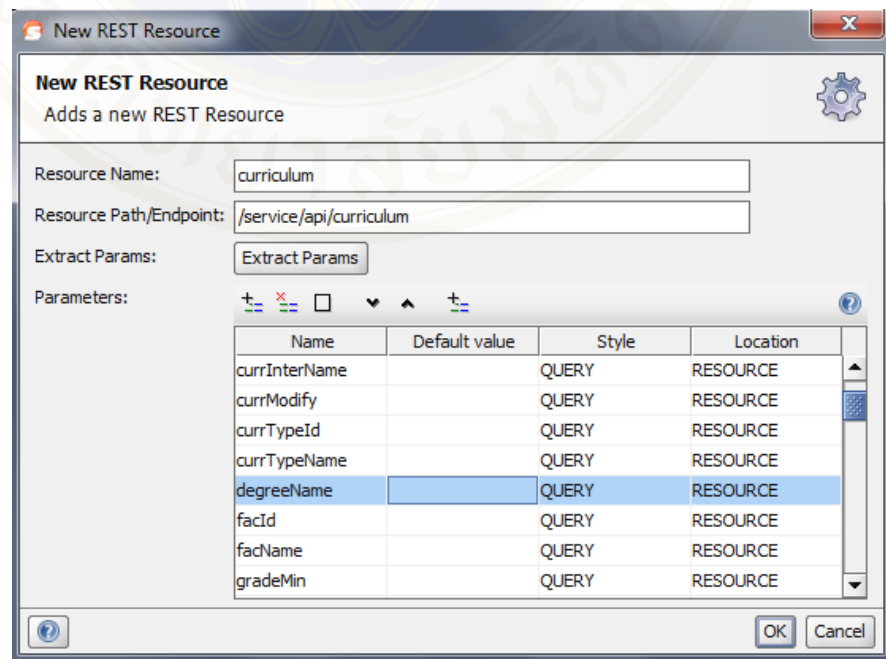
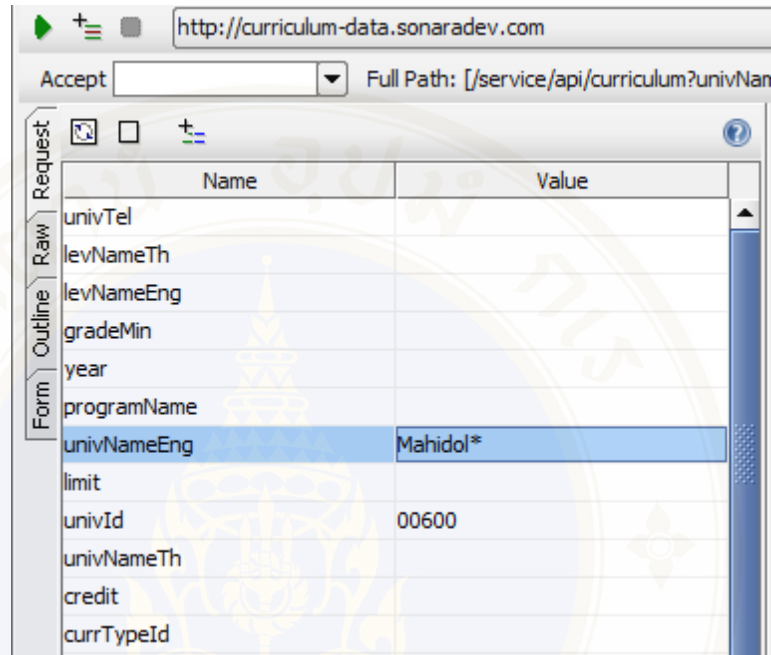



Figure B.4 New REST Resource

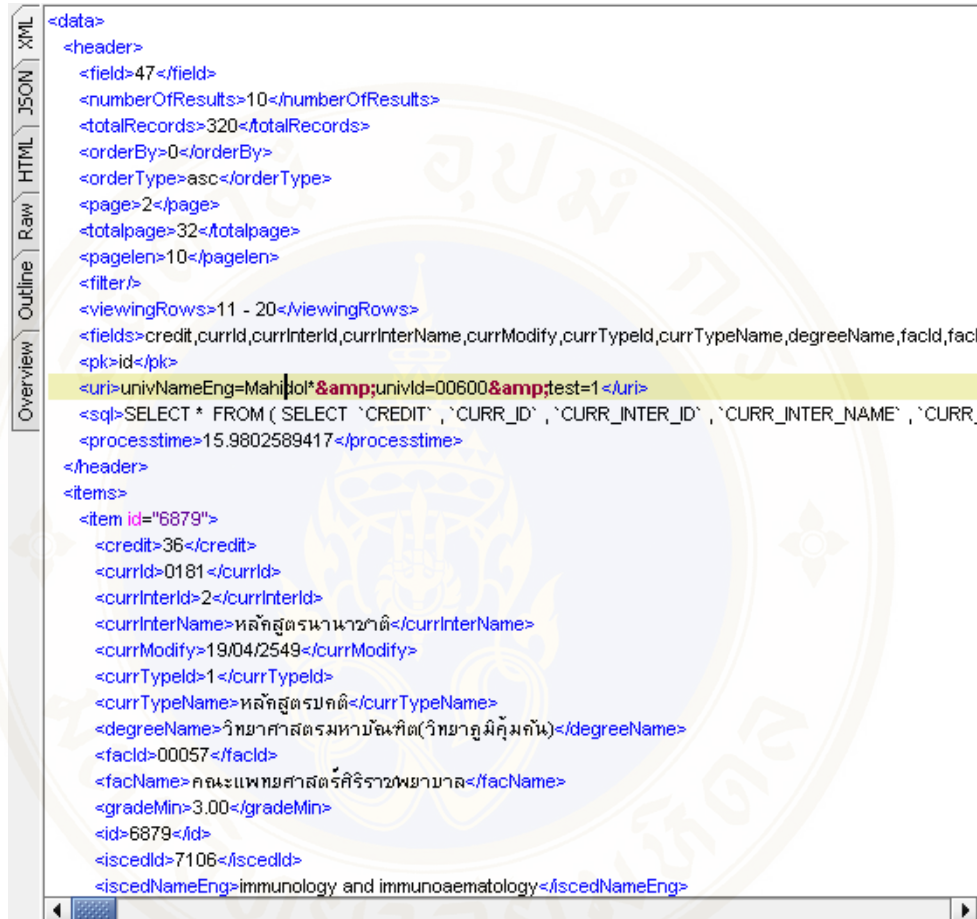
5. โปรแกรมจะแสดงหน้าจอหลักขึ้นมา ให้ใส่ค่า Parameters ที่ต้องการค้นหา จากนั้นกดปุ่ม 



Name	Value
univTel	
levNameTh	
levNameEng	
gradeMin	
year	
programName	
univNameEng	Mahidol*
limit	
univId	00600
univNameTh	
credit	
currTypeId	

Figure B.5 Input Parameters

6. หลังจากกดปุ่ม  หน้าจอทางขวาจะแสดงผลที่ออกมาในรูปแบบ XML ดังรูปที่ B.6



```

<data>
  <header>
    <field>47</field>
    <numberOfResults>10</numberOfResults>
    <totalRecords>320</totalRecords>
    <orderBy>0</orderBy>
    <orderType>asc</orderType>
    <page>2</page>
    <totalpage>32</totalpage>
    <pagelen>10</pagelen>
    <filter/>
    <viewingRows>11 - 20</viewingRows>
    <fields>credit,currId,currInterId,currInterName,currModify,currTypeId,currTypeName,degreeName,facId,fac
    <pk>id</pk>
    <uri>univNameEng=Mahidol*&univId=00600&test=1</uri>
    <sql>SELECT * FROM ( SELECT `CREDIT` , `CURR_ID` , `CURR_INTER_ID` , `CURR_INTER_NAME` , `CURR
    <processtime>15.9802589417</processtime>
  </header>
  <items>
    <item id="6879">
      <credit>36</credit>
      <currId>0181</currId>
      <currInterId>2</currInterId>
      <currInterName>หลักสูตรนานาชาติ</currInterName>
      <currModify>19/04/2549</currModify>
      <currTypeId>1</currTypeId>
      <currTypeName>หลักสูตรมกค</currTypeName>
      <degreeName>วิทยาศาสตรมหาบัณฑิต(วิทยาศาสรมัคมน)</degreeName>
      <facId>00057</facId>
      <facName>คณะแพทยศาสตร์ศิริราชพยาบาล</facName>
      <gradeMin>3.00</gradeMin>
      <id>6879</id>
      <iscedId>7106</iscedId>
      <iscedNameEng>immunology and immunoematology</iscedNameEng>
    </item>
  </items>
</data>
  
```

Figure B.6 Results

BIOGRAPHY

NAME	Mr. Poj Thammathiwat
DATE OF BIRTH	9 January 1977
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Rangsit University, 1999 – 2004: Bachelor of Science (Computer Science) Mahidol University, 2006 – 2011: Master of Science (Technology of Information System Management)
HOME ADDRESS	108/586 Kritsadanakorn10, Rattanathibet Rd. Bangrakpattana, Bangbuethong, Nonthaburi, 11110 Tel. 081-344-4449 E-mail : pojy349@hotmail.com