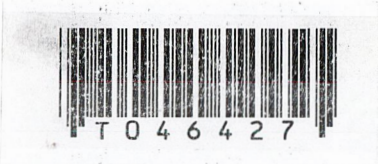


เครื่องมือสร้างเว็บฟอร์มพร้อมระบบตรวจสอบความถูกต้องของข้อมูล

A WEB FORM WITH VALIDATION GENERATOR



โดย นางสาวกนวรรณ อุปคำ
นางสาวนิภาพรรณ สมนใจ

อาจารย์ผู้ควบคุมปริญญาโท
อาจารย์ภูซังค์ หงษ์สุวรรณ

ปริญญาโทนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาโทวิศวกรรมศาสตรบัณฑิต

ณ
ก.ย. ๒๕๖๑
๒๕๖๑

สาขาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เลขที่.....

เลขทะเบียน 46427

วัน, เดือน, ปี - 1 เม.ย. 2546

b.....
i.....

รับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ เครื่องมือสร้างเว็บฟอร์มพร้อมระบบตรวจสอบความถูกต้องของข้อมูล
TITLE A WEB FORM WITH VALIDATION GENERATOR
โดย นางสาวกนกวรรณ อุปคำ รหัสประจำตัว 41014002
นางสาวนิภาพรรณ สนใจ รหัสประจำตัว 41014225
อาจารย์ที่ปรึกษา อาจารย์ฤษงค์ หงษ์สุวรรณ
ภาควิชา วิศวกรรมสารสนเทศ
ปีการศึกษา 2544

ปริญญานิพนธ์ฉบับนี้ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบัง



.....
(อาจารย์ฤษงค์ หงษ์สุวรรณ)
อาจารย์ผู้ควบคุมปริญญานิพนธ์

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เครื่องมือสร้างเว็บฟอร์มพร้อมระบบตรวจสอบความถูกต้องของข้อมูล	
	A WEB FORM WITH VALIDATION GENERATOR	
นักศึกษา	นางสาวกนกวรรณ อุปคำ	รหัสประจำตัว 41014002
	นางสาวนิภาพรณ สนใจ	รหัสประจำตัว 41014225
อาจารย์ที่ปรึกษา	อาจารย์ภูซงค์ หงษ์สุวรรณ	
ระดับการศึกษา	ปริญญาวิศวกรรมศาสตรบัณฑิต	
ภาควิชา	วิศวกรรมสารสนเทศ	
ปีการศึกษา	2544	

บทคัดย่อ

กระบวนการตรวจสอบความถูกต้องของข้อมูลจากแบบฟอร์ม มักกระทำที่ฝั่งเซิร์ฟเวอร์ หากข้อมูลที่ถูกกรอกลงไปไม่ถูกต้อง เซิร์ฟเวอร์ต้องส่งฟอร์มนั้นกลับมายังผู้ใช้และแจ้งความผิดพลาดที่เกิดขึ้น ทำให้เสียเวลาในการส่งแบบฟอร์มเป็นอย่างมาก โดยเฉพาะอย่างยิ่งในเวลาที่ปริมาณการใช้งานอินเทอร์เน็ตสูง วิธีแก้ปัญหานี้ทำได้โดยการสร้างฟอร์มที่สามารถตรวจสอบความถูกต้องของข้อมูลได้อย่างอัตโนมัติ โดยการใช้จาวาสคริปต์ แต่ยิ่งสร้างให้ฟอร์มมีความสามารถมากขึ้นเท่าใด ความยุ่งยากในการเขียนโปรแกรมก็เพิ่มมากขึ้นด้วย โครงการนี้จึงได้ทำการศึกษาทฤษฎีของจาวาสคริปต์แล้วนำมาประยุกต์ใช้เพื่อพัฒนาเครื่องมือที่ช่วยผู้ใช้ออกแบบและสร้างเว็บฟอร์ม พร้อมสามารถกำหนดให้มีระบบการตรวจสอบความถูกต้องของข้อมูลได้ตามความต้องการ แล้วสร้างโค้ดของฟอร์มออกมา ซึ่งสามารถนำไปใช้งานได้จริงบนเว็บเบราว์เซอร์ เพื่อให้ความสะดวกแก่ผู้ที่ต้องการสร้างเว็บฟอร์มแต่ขาดความรู้ด้านการเขียนโปรแกรม

THESIS TITLE	A WEB FORM WITH VALIDATION GENERATOR	
NAME	Miss Kanokwan Uppakum	No. 41014002
	Miss Nipapan Sonjai	No. 41014225
ADVISOR	Mr. Puchong Hongsuwan	
COURSE	Bachelor Degree of Information Engineering	
DEPARTMENT	Information Engineering	
YEAR	2001	

ABSTRACT

In the process of form submission via internet, the valid data are required. Data validation is generally operated at server. The form will be sent back to a user to correct data and try submission again, if invalid data are detected. It therefore will take very long time when traffic is crowded. As this problem, to creating a form included with an automatically validating system becomes a nice solution, which is related to a JavaScript Programming. But the more efficiently a form can validate, the more complex a programming algorithm is. The purpose of this project is to study the theory of JavaScript and be able to apply a computer programming to develop a tool which is able to generate efficiently HTML and JavaScript code of a form. As this result, it can be useful for users, who have no a programming background to easily create form with validation.

กิตติกรรมประกาศ

ขอขอบพระคุณอาจารย์ภูษงค์ หงษ์สุวรรณ อาจารย์ผู้ควบคุมปริญญาานิพนธ์ เป็นอย่างสูง ที่เสียสละเวลาอันมีค่ามาดูแลและให้คำปรึกษาในการทำงานครั้งนี้แทบทุกขั้นตอน และขอขอบคุณ คุณประภาส มีาเทศ ที่ร่วมแบ่งปันประสบการณ์อันเป็นประโยชน์ตลอดจนคำแนะนำต่าง ๆ จนสามารถทำให้ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ตามความคาดหมาย

และสิ่งสำคัญที่ผลักดันให้เรามุ่งมั่นทำงานครั้งนี้จนประสบความสำเร็จคือกำลังใจจากบิดา มารดา ที่เคารพรัก จึงขอขอบพระคุณมา ณ ที่นี้เป็นอย่างสูง และขอบคุณเพื่อน ๆ ในภาควิชาวิศวกรรมสารสนเทศทุกคน ตลอดเวลา 3 ปีที่ได้รู้จักกัน มิตรภาพและน้ำใจที่ได้รับ เป็นอีกส่วนหนึ่งที่ผลักดันให้เกิดความสำเร็จในชีวิตครั้งนี้ได้

น.ส.กนกวรรณ อุปคำ
 น.ส.นิภาพรณ สนใจ
 ผู้จัดทำ

สารบัญ

บทที่	เรื่อง	หน้า
	บทคัดย่อภาษาไทย	ก
	บทคัดย่อภาษาอังกฤษ	ข
	กิตติกรรมประกาศ	ค
1	บทนำ	
	1.1 ที่มาและแนวคิดของ โครงการงาน	1
	1.2 วัตถุประสงค์ของ โครงการงาน	2
	1.3 ขอบเขตของ โครงการงาน	2
	1.4 วิธีการศึกษาและพัฒนา	3
2	พื้นฐานการสร้างฟอร์มใน HTML และพื้นฐานจาวาสคริปต์	
	2.1 โครงสร้างพื้นฐานของ HTML	5
	2.2 วิธีการเขียนภาษา HTML	6
	2.3 รูปแบบการทำงานของฟอร์มและ CGI	6
	2.4 การสร้างแบบฟอร์ม	7
	2.5 อุปกรณ์ในการรับข้อมูล	8
	2.5.1 การสร้าง TEXT BOX	8
	2.5.2 การสร้างช่องรับข้อมูลแบบ PASSWORD	9
	2.5.3 การสร้าง TEXT AREA	10
	2.5.4 การสร้าง CHECKBOX	12
	2.5.5 การสร้าง RADIO BUTTON	13
	2.5.6 การสร้าง COMBO BOX	14
	2.5.7 การสร้างปุ่ม SUBMIT	16
	2.5.8 การสร้างปุ่ม RESET	17
	2.5.9 การสร้างปุ่ม BUTTON	17
	2.6 จาวาสคริปต์ (JavaScript)	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ เรื่อง	หน้า
2.7 พื้นฐานของออบเจกต์ (Object) ,เมธอด (Method) และอีเวนต์ (Event)	18
2.7.1 ออบเจกต์	19
2.7.2 เมธอด	19
2.7.3 อีเวนต์และลำดับการทำงานของโปรแกรม	19
2.8 พื้นฐานการเขียนสคริปต์	20
2.8.1 การกำหนดภาษาจาวาสคริปต์	21
2.8.2 ตำแหน่งของสคริปต์ในเอกสาร HTML	22
2.9 กลไกในจาวาสคริปต์	22
2.9.1 การอ้างอิงชื่อในจาวาสคริปต์	22
2.9.2 การตั้งชื่อตัวแปรและฟังก์ชัน	23
2.9.3 คำสงวน	24
2.9.4 ตัวดำเนินการและตัวแปร	24
2.10 การสร้างออบเจกต์ใหม่	25
2.11 ฟังก์ชันและเมธอด	25
2.12 การใช้งานอาร์เรย์	26
2.13 การโปรแกรมแบบใช้เงื่อนไขและการทำซ้ำ	26
2.13.1 คำสั่ง if	26
2.13.2 คำสั่ง if...else	27
2.13.3 คำสั่ง switch	27
2.13.4 คำสั่ง for	28
2.13.5 คำสั่ง while	28
2.13.6 คำสั่ง break	29
2.13.7 คำสั่ง continue	29
2.13.8 คำสั่ง with	30
2.14 การใช้ฟอร์มร่วมกับจาวาสคริปต์	30
2.14.1 การสร้างแบบฟอร์ม	31
2.14.2 ฟอร์มออบเจกต์	32
2.14.3 ออบเจกต์ต่าง ๆ ในฟอร์ม	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่	เรื่อง	หน้า
	2.14.3.1 แท็กช้อบเจ็ทต์	33
	2.14.3.2 ซับมีตออบเจ็ทต์	33
	2.14.3.3 รีเซ็ทออบเจ็ทต์	34
	2.14.3.4 เซ็ทบ็อกซ์ออบเจ็ทต์	34
	2.14.3.5 เรดิโอออบเจ็ทต์	34
	2.14.3.6 ซีล็กต์ออบเจ็ทต์	34
	2.15 UML (Unified Modeling Language)	35
3	วิธีการออกแบบ	
	3.1 การออกแบบทำงานของระบบ	38
	3.2 การออกแบบด้วยวิธีการ UML	44
	3.2 การออกแบบฐานข้อมูลรหัสไปรษณีย์ของแต่ละจังหวัดในประเทศไทย	57
4	ผลการทดลอง	
	4.1 ผลการทดลองในส่วนการสร้างฟอร์ม	58
	4.2 ผลการทดลองในส่วนสร้างโค้ด	61
	4.3 ผลการทดลองในส่วนแสดงฟอร์มบนเบราว์เซอร์	63
	4.3.1 การแสดงฟอร์มธรรมดาบนเบราว์เซอร์	63
	4.3.2 การตรวจสอบความถูกต้องของฟอร์ม	63
5	สรุปผลการทดลอง	
	5.1 วิจารณ์และสรุปผลการทดลอง	68
	5.2 แนวทางในการพัฒนาต่อ	69
	บรรณานุกรม	70

สารบัญรูป

รูปที่	ชื่อรูป	หน้า
1.1	แสดงการทำงานของโปรแกรมสร้างเว็บฟอร์มพร้อมระบบตรวจสอบข้อมูล	4
2.1	ตัวอย่าง Textbox	9
2.2	ตัวอย่างช่องรับข้อมูลแบบ Password	10
2.3	ตัวอย่างของ Text Area	11
2.4	Text Area ในกรณีของ Virtual หรือ Physical	12
2.5	Text Area ในกรณีของ Off	12
2.6	ตัวอย่าง Checkbox	12
2.7	ตัวอย่างของ Radio Button	14
2.8	ตัวอย่างของ Combo Box แบบ Menu	15
2.9	ตัวอย่างของ Combo Box แบบ List	16
2.10	ตัวอย่างปุ่ม Submit	16
2.11	ตัวอย่างปุ่ม Reset	17
2.12	ตัวอย่างการสร้างปุ่ม Button	17
2.13	ลำดับการสืบทอดออบเจกต์ (Object Hierarchy)	20
2.14	เปรียบเทียบรูปแบบการวนซ้ำและลำดับการทำงานตามอีเวนต์ในจาวสคริปต์	21
3.1	Work Flow การทำงานของระบบของโปรแกรมสร้างเว็บฟอร์ม พร้อมระบบตรวจสอบความถูกต้องของข้อมูล	39
3.2	Work Flow การทำงานของส่วน Create Form	40
3.3	Work Flow การทำงานของส่วน Open Form	41
3.4	Work Flow การทำงานของส่วน Generate HTML and JavaScript Code	42
3.5	Work Flow การทำงานของส่วน Show Form on Browser with Code	43
3.6	Work Flow การทำงานของส่วน Save	43
3.7	แสดง Class Diagram ของระบบ	44
3.8	แสดง High-Level Use Case ของระบบ	45
3.9	แสดง Use Cases Diagram ของระบบ	46
3.10	แสดง Activity Diagram ส่วน Add Control	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ชื่อรูป	หน้า
3.11 แสดง Activity Diagram ส่วน Generate HTML without JavaScript	48
3.12 แสดง Activity Diagram ส่วน Generate HTML with JavaScript	49
3.13 การออกแบบฐานข้อมูล NIAM	50
3.14 รูปตารางฐานข้อมูลใน Relational Database	51
3.15 การออกแบบฐานข้อมูลของรหัสไปรษณีย์ในประเทศไทย	57
3.16 รูปตารางของข้อมูลรหัสไปรษณีย์ใน Relational Database	57
4.1 หน้าจอหลักเมื่อเริ่มเปิดใช้งาน โปรแกรม	58
4.2 การวางอุปกรณ์รับข้อมูลลงบนฟอร์ม	59
4.3 การสร้างฟอร์มพร้อมการตรวจสอบความถูกต้องของข้อมูล	60
4.4 การบันทึกฟอร์มลงฐานข้อมูล	61
4.5 HTML CODE ของฟอร์ม	62
4.6 HTML และ JAVASCRIPT CODE ของฟอร์ม	62
4.7 ตัวอย่างการแสดงผลฟอร์มบนเบราว์เซอร์	63
4.8 การตรวจสอบจำนวนอักษรในช่อง Password	64
4.9 การตรวจสอบข้อมูลในช่อง Confirm Password	64
4.10 การตรวจสอบความถูกต้องของฟอร์มทั้งหมด	65
4.11 การส่งแบบฟอร์มสำเร็จ	65
4.12 แสดงรายการของจังหวัด, อำเภอ และรหัสไปรษณีย์โดยที่ยังไม่มีความสัมพันธ์กัน	66
4.13 ตัวอย่างการแสดงผลข้อมูลอำเภอที่สัมพันธ์กับจังหวัด	67

สารบัญตาราง

ตารางที่	ชื่อตาราง	หน้า
3.1	Data Dictionary ของ Entity Table	52
3.2	Data Dictionary ของ Form Table	52
3.3	Data Dictionary ของ Label Table	52
3.4	Data Dictionary ของ Textbox Table	53
3.5	Data Dictionary ของ Radio Table	53
3.6	Data Dictionary ของ Checkbox Table	54
3.7	Data Dictionary ของ Combobox Table	55
3.8	Data Dictionary ของ ComboboxOption Table	55
3.9	Data Dictionary ของ Button Table	56
3.10	Data Dictionary ของ Textarea Table	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ที่มาและแนวคิดของโครงการ

ปกติการสร้างเว็บฟอร์มหรือแบบฟอร์มบนเว็บเพิ่งสามารถสร้างได้ด้วยการเขียนภาษา HTML บนเทกซ์อีดิเตอร์แต่ในปัจจุบันมีหลายโปรแกรมที่ช่วยให้ผู้ใช้สามารถสร้างเว็บฟอร์มในลักษณะ WYSIWYG (What You See Is What You Get.) ได้อย่างง่ายดาย เช่น NetObject Fusion, Dreamweaver หรือ Microsoft FrontPage แต่โปรแกรมเหล่านี้ยังไม่มีความสามารถในการสร้างฟอร์มที่พร้อมด้วยระบบตรวจสอบความถูกต้องของข้อมูลในตัวได้ ซึ่งโดยปกติแล้วข้อมูลทั้งหมดที่กรอกในฟอร์มจะถูกส่งไปตรวจสอบที่เครื่องเซิร์ฟเวอร์ หากเกิดความผิดพลาดขึ้นฟอร์มนั้นจะถูกส่งกลับมา เพื่อให้ผู้ใช้กรอกอีกครั้ง เป็นผลทำให้การดำเนินการเกิดความล่าช้า เช่นในกรณีที่กำหนดให้เติมตัวเลข 5 ตัวลงในช่องกรอกข้อมูล แต่ผู้ใช้กลับเติมตัวอักษรลงไป เมื่อแบบฟอร์มนั้นถูกส่งไปยังเซิร์ฟเวอร์ จะมีการเตือนกลับมาว่าข้อมูลไม่ถูกต้อง แสดงว่าการส่งฟอร์มนั้นไม่สำเร็จ การเพิ่มความสามารถของฟอร์มให้มีการตรวจสอบความถูกต้องของข้อมูลได้นั้น จะช่วยลดการทำงานของเซิร์ฟเวอร์ อีกทั้งเพิ่มความรวดเร็วในการดำเนินการเกี่ยวกับฟอร์ม แต่จะมีความยุ่งยากตรงที่ต้องแทรกโปรแกรมภาษาสคริปต์ซึ่งเป็นภาษาที่ทำงานบนระบบอินเทอร์เน็ตเข้าไปในเอกสาร HTML ทำให้เกิดความไม่สะดวกแก่ผู้สร้างฟอร์มที่ขาดความรู้และประสบการณ์ด้านนี้เป็นอย่างมาก

ดังนั้นหากมีโปรแกรมที่สามารถสร้างฟอร์มพร้อมทั้งมีระบบตรวจสอบความถูกต้องของข้อมูลในตัว โดยสร้างโค้ดให้อย่างอัตโนมัติ ที่พร้อมนำไปใช้งานได้จริง จะเป็นการดียิ่งยิ่งสำหรับผู้ที่มีขาดความรู้ด้านการเขียนโปรแกรมสคริปต์ ด้วยเหตุผลดังกล่าวจึงเป็นที่มาของโครงการซึ่งโครงการนี้ได้เลือกใช้ภาษาจาวาสคริปต์เพื่อตรวจสอบความถูกต้อง เนื่องจากเป็นภาษาที่คนส่วนใหญ่นิยมใช้ และเรียนรู้ง่าย

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาการเขียนโปรแกรมโดยใช้ภาษาจาวาสคริปต์และ HTML
2. เพื่อศึกษาวิธีการ UML
3. เพื่อศึกษาการเขียนโปรแกรมโดยใช้ภาษา Visual Basic
4. สามารถนำเทคนิคการเขียนโปรแกรมที่ศึกษามาประยุกต์สร้างเครื่องมือสำหรับสร้างเว็บฟอร์ม พร้อมมีระบบตรวจสอบความถูกต้องของข้อมูลได้ โดยสร้างในรูปแบบของ HTML และจาวาสคริปต์

1.3 ขอบเขตของโครงการ

โครงการนี้เป็นการนำความรู้ด้านการเขียนโปรแกรมมาประยุกต์สร้างเครื่องมือในลักษณะของโปรแกรม GUI (Graphic User Interface) และ WYSIWYG ที่ช่วยในการออกแบบเว็บฟอร์มได้ตามความต้องการ พร้อมสามารถกำหนดให้มีการตรวจสอบความถูกต้องของข้อมูลได้ และสร้างโค้ด HTML และจาวาสคริปต์ของฟอร์มให้ทันที โดยให้ผลออกมาเป็นไฟล์ .txt และ .html ซึ่งอุปกรณ์รับข้อมูลบนฟอร์ม หรือตัว Control มีให้เลือกใช้ได้หลายชนิด ได้แก่ Label, Textbox, Combo Box, Button, Submit, Reset, Radio Button, Text Area และ Checkbox

Control แต่ละตัวสามารถให้ผู้ใช้กำหนดพารามิเตอร์ต่าง ๆ ตามคำสั่งในภาษา HTML เพื่อนำข้อมูลนั้นไปสร้างเป็นโค้ด เช่น Textbox มีพารามิเตอร์ type, name, value, maxlength, size หรือ button มีพารามิเตอร์ type, name, value เป็นต้น

ในการตรวจสอบความถูกต้องของข้อมูลส่วนใหญ่จะให้ความสำคัญกับ Textbox มากกว่า Control ชนิดอื่น ๆ ดังนี้

- สามารถตรวจสอบรูปแบบของอีเมลแอดเดรสได้ทั้ง ID Name และ Domain Name แบบ 2 ระดับ เช่น xxx@hotmail.com และ 3 ระดับ เช่น xxx@kmitl.ac.th
- สามารถกำหนดขนาดข้อมูลและจัดรูปแบบของหมายเลขโทรศัพท์ (ภายในประเทศ) ให้ถูกต้อง
- สามารถตรวจสอบได้เมื่อกำหนดให้รับเฉพาะข้อมูลตัวเลขอย่างเดียว
- สามารถตรวจสอบได้เมื่อกำหนดให้รับเฉพาะข้อมูลตัวอักษรอย่างเดียว
- สามารถตรวจสอบความตรงกันของข้อมูลได้ ในกรณีที่เป็นการ Confirm Password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถกำหนดจำนวน digit สูงสุดของข้อมูลที่ได้รับได้
- สามารถตรวจสอบได้เมื่อฟิลด์ที่สำคัญในฟอร์มไม่ได้ถูกรอกข้อมูล

นอกจากการตรวจสอบความถูกต้องของข้อมูลแล้ว โปรแกรมยังเพิ่มประสิทธิภาพให้กับฟอร์มได้ในกรณีของ Combo Box นั่นคือสามารถเรียกชุดข้อมูลที่สัมพันธ์กันมาไว้ในลิสต์เพื่อเพิ่มความสะดวกแก่ผู้ใช้ในขณะที่กรอกแบบฟอร์ม ชุดข้อมูลดังกล่าวมีดังนี้

- ชุดข้อมูล อำเภอ จังหวัด และรหัสไปรษณีย์ของประเทศไทย
- ชุดข้อมูลวัน เดือน ปี

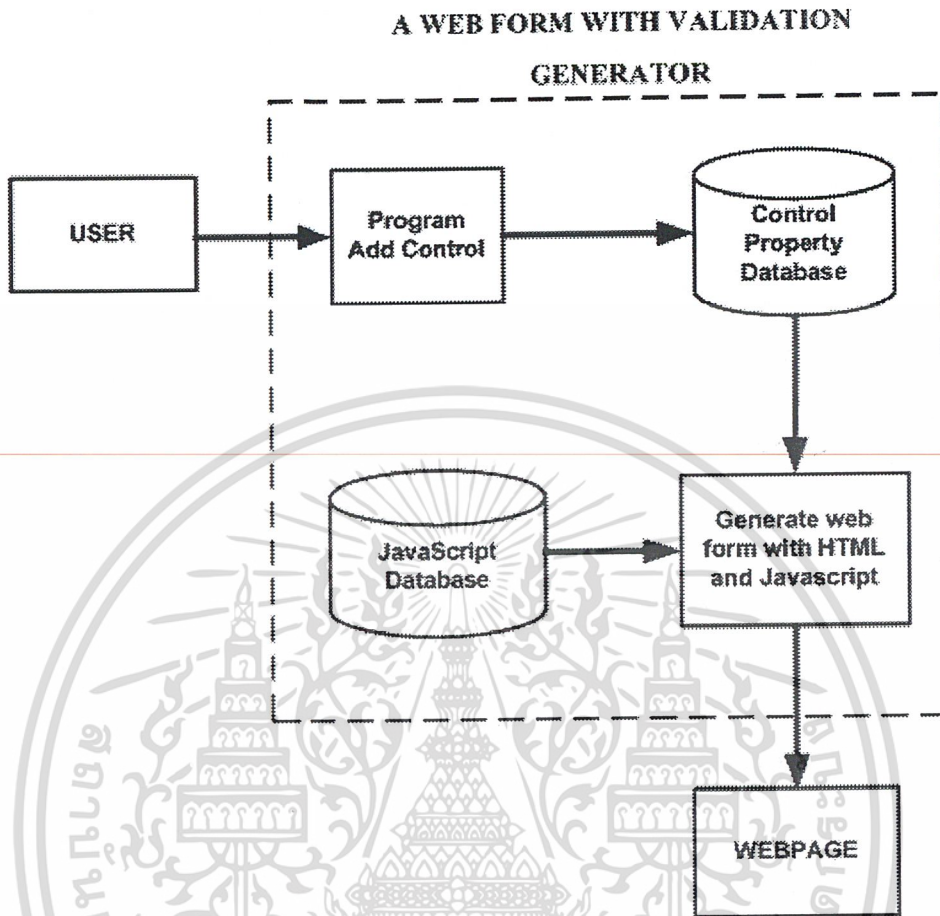
1.4 วิธีการศึกษาและพัฒนา

สำหรับโครงการนี้ได้มีการศึกษาทฤษฎีการเขียนโปรแกรมไปพร้อมกับการพัฒนาซึ่งตัวโปรแกรมสร้างเว็บฟอร์มนั้นพัฒนาจากภาษา Visual Basic ส่วนการศึกษาจาวาสคริปต์ เพื่อนำมาประยุกต์ใช้เป็นเอาต์พุตของโปรแกรม ในการพัฒนาโครงการได้แบ่งการทำงานหลัก ๆ ออกเป็น 3 ส่วน คือ

1. การออกแบบโปรแกรมสร้างเว็บฟอร์ม คือการออกแบบส่วนอินเทอร์เฟซของโปรแกรม, เมนูการทำงาน, ส่วนในการกำหนดพารามิเตอร์ของ Control แต่ละตัว และส่วนสำหรับกำหนดความถูกต้องของข้อมูล เป็นต้น

2. การรวบรวมโค้ดจาวาสคริปต์ แล้วดัดแปลงให้เหมาะสมกับการนำไปใช้งาน

3. การออกแบบวิธีการสร้างโค้ด HTML และ จาวาสคริปต์ ส่วนนี้เป็นส่วนที่สำคัญที่สุดของโครงการ โดยส่วนใหญ่จะสัมพันธ์กับการออกแบบฐานข้อมูลเพื่อใช้เก็บโค้ดจาวาสคริปต์และค่าต่าง ๆ ที่ได้จากการสร้างฟอร์ม รวมถึงต้องใช้เทคนิคการเขียนโปรแกรมเพื่อนำข้อมูลเหล่านี้มารวมกันแล้วได้สำเร็จออกมาเป็นโค้ดที่พร้อมนำไปใช้งานได้จริง ซึ่งในรูปแบบที่ 1.1 ได้แสดงภาพรวมของการทำงานของระบบเพื่อให้สามารถเข้าใจได้ง่ายขึ้น



รูปที่ 1.1 แสดงการทำงานของโปรแกรมสร้างเว็บฟอร์มพร้อมระบบตรวจสอบข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

พื้นฐานการสร้างฟอร์มใน HTML, พื้นฐานจาวาสคริปต์ และ UML

HTML ย่อมาจากคำว่า Hypertext Markup Language เป็นภาษาหลักที่ใช้ในการแสดงผลบนเว็บเบราว์เซอร์ในอินเทอร์เน็ต โดยเราสามารถนำเสนอข้อมูลที่มีทั้งตัวอักษร ภาพ เสียง ภาพยนตร์ และสามารถเชื่อมโยงกับเอกสารอื่น ๆ ได้ง่าย ภาษา HTML ได้รับการพัฒนาตลอดเวลา ปัจจุบันเป็นเวอร์ชัน 4.0 ภาษา HTML นั้นมีข้อดีในแง่ของการเขียนที่เราสามารถเรียนรู้ได้ง่ายและรวดเร็ว

ภาษา HTML แบ่งออก 2 ส่วนคือ ส่วนที่เป็นข้อความทั่ว ๆ ไป และส่วนที่เป็นคำสั่งที่ใช้ในการกำหนดรูปแบบของข้อความที่แสดง ซึ่งเรียกว่า แท็ก (Tag) โดยคำสั่งของ HTML จะอยู่ในเครื่องหมาย <> มีหลักการเขียนดังนี้

- รูปแบบคำสั่งจะแยกเป็น 2 ส่วน โดยจะต้องมีส่วนเริ่มต้นของคำสั่ง และส่วนท้ายจะต้องมีเครื่องหมาย Slash (/) เช่น <TITLE>ข้อความ</TITLE>

สำหรับในบางกรณีที่ต้องการใช้คำสั่งซ้อนกันมากกว่า 1 คำสั่ง จะต้องใช้คำสั่งปิดส่วนที่อยู่สิ้นสุดก่อน แล้วจึงปิดลำดับในชั้นถัด ๆ มาจนถึงนอกสุด เช่น <H2><I> ข้อความ </I></H2>

- บางแท็กอาจจะไม่ต้องมีแท็กปิดก็สามารถใช้งานได้ เช่น

- ใน HTML เวอร์ชัน 4 ขึ้นไป สามารถพิมพ์ด้วยอักษรพิมพ์เล็กหรือพิมพ์ใหญ่ก็ได้ ซึ่งจะมีความหมายเหมือนกัน เช่น <HTML>, <html>, <Html>
- บางแท็กจะมีตัวกำหนดคุณสมบัติ เรียกว่า แอททริบิวต์ (Attribute) โดยจะเขียนไว้หลังคำสั่ง เช่น <HR WIDTH=200 SIZE=6>

2.1 โครงสร้างพื้นฐานของ HTML

โครงสร้างของภาษา HTML แบ่งออกได้เป็น 4 ส่วนหลัก ดังนี้

<HTML>

<HEAD>

<TITLE> หัวข้อเว็บเพจ </TITLE>

</HEAD>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<BODY>

ส่วนที่มีเนื้อหา, ข้อมูลภาพ หรือส่วนเชื่อมโยง ที่ปรากฏในเว็บเพจ

</BODY>

</HTML>

<HTML>...</HTML> เป็นคำสั่งแรกที่ต้องมีในภาษา HTML ซึ่งเป็นเครื่องแสดงว่านี่คือภาษา HTML โดยอยู่ที่จุดเริ่มต้นของเอกสารและท้ายเอกสารในแต่ละแฟ้ม

<HEAD>...</HEAD> เป็นส่วนกำหนดรายละเอียดหัวข้อเรื่อง ภายในจะมีคำสั่งย่อยอีกหนึ่งคำสั่ง คือ <TITLE>...</TITLE> โดยข้อความภายในคำสั่งนี้มีไว้เพื่อเป็นหัวเรื่องของเว็บเพจที่ต้องการ ซึ่งจะต้องมีความยาวไม่เกิน 64 ตัวอักษร หัวเรื่องนี้จะไปปรากฏที่ Title bar ของเบราว์เซอร์ที่เราใช้งานอยู่ และปรากฏอยู่บน Bookmark ของโปรแกรมเบราว์เซอร์ด้วย

<BODY>...</BODY> เป็นส่วนของเนื้อหาเว็บเพจ ไม่ว่าจะเป็นข้อความ รูปภาพ ตาราง เสียง และการเชื่อมโยงต่อไปยังเอกสารอื่น ๆ ที่จะไปแสดงผลบนเว็บเบราว์เซอร์

2.2 วิธีการเขียนภาษา HTML

ในการสร้างเว็บเพจด้วย HTML นั้น จำเป็นต้องมีเครื่องมือที่ใช้ในการเขียนและแก้ไขโค้ด เรียกว่าเท็กซ์อีดิเตอร์ (Text Editor) ซึ่งปัจจุบันมีอยู่มากมายเช่น Notepad, EditPlus, Hotdog และ Homesite เป็นต้น

หลังจากที่เราพิมพ์เอกสารตามโครงสร้างของภาษา HTML เสร็จเรียบร้อยแล้ว จะต้องบันทึกเป็นเอกสาร HTML โดยบันทึกเป็นนามสกุล .html หรือ .htm เมื่อทำการบันทึกเรียบร้อยแล้ว สามารถเปิดเบราว์เซอร์เพื่อเรียกดูหน้าตาของเว็บเพจที่เราสร้างขึ้นได้ และหากต้องการแก้ไขเว็บเพจนั้น ให้กลับไปโปรแกรม เท็กซ์อีดิเตอร์แล้วทำการแก้ไขไฟล์ตามต้องการ และต้องบันทึกทุกครั้งที่ทำกรแก้ไข แล้วให้เบราว์เซอร์โหลดเว็บเพจที่ได้แก้ไขขึ้นมาดู

คำสั่งของภาษา HTML มีมากมาย แต่ในบทนี้จะขอล่าวถึงเฉพาะคำสั่งที่เกี่ยวข้องกับการสร้างแบบฟอร์มเท่านั้น

2.3 รูปแบบการทำงานของฟอร์ม และ CGI

เว็บเพจเป็นสื่อที่แตกต่างจากเอกสารธรรมดา เพราะเป็นสื่อสองทางที่ผู้ใช้สามารถสื่อสารกับเจ้าของเว็บเพจได้ มีการใช้แบบฟอร์มเพื่อเก็บข้อมูล ตั้งแต่รายละเอียดเกี่ยวกับผู้ที่มาเยี่ยมชมเอกสารนั้นเป็นเอกสารที่ส่งวันไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติหน้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชมเว็บเพจ การตั้งชื่อสินค้า ไปจนถึงการรวบรวมความคิดเห็นเกี่ยวกับเรื่องต่าง ๆ ที่กำลังอยู่ในความสนใจ เช่น การเลือกตั้ง การลงคะแนนความคิดเห็น เป็นต้น

ข้อมูลที่ได้รับจากแบบฟอร์ม จะต้องมีการนำข้อมูลที่ได้ไปประมวลผลต่อ วิธีการนำข้อมูลไปประมวลผลนี้มีชื่อว่า CGI (Common Gateway Interface) และโปรแกรมที่ใช้ในการประมวลผลข้อมูลที่ได้รับจากแบบสอบถาม มีชื่อว่า CGI Script

CGI Script ถูกเขียนขึ้นมาจากภาษาคอมพิวเตอร์ เช่น Perl หรือ VBScript และจะเก็บอยู่ที่เครื่องเซิร์ฟเวอร์ สำหรับลักษณะการใช้ CGI ในเครื่องเซิร์ฟเวอร์แต่ละระบบจะมีแนวทางที่คล้ายกัน แต่รายละเอียดเพิ่มเติมขึ้นอยู่กับบริษัทที่ให้บริการติดตั้งเครื่องเซิร์ฟเวอร์ หรือเว็บไซต์ที่นำเว็บเพจ ไปเก็บ เมื่อต้องการนำฟอร์มมาใช้กับเว็บไซต์จริง ๆ

2.4 การสร้างแบบฟอร์ม

ในการสร้างฟอร์ม เราใช้คำสั่ง FORM วางในส่วน BODY ของ HTML ดังตัวอย่างต่อไปนี้

```
<BODY>
<FORM METHOD="POST" ACTION="/cgi-bin/search.pl" NAME="TESTFORM">
...
</FORM>
</BODY>
```

METHOD คือ วิธีการรับ-ส่งข้อมูลจากแบบฟอร์มไปยังเครื่องเซิร์ฟเวอร์ ผ่านทาง CGI มีให้เลือก 2 แบบคือ

1. POST คือ การส่งข้อมูลภายในฟอร์ม ไปยัง CGI ที่ละบรรทัด รวมทั้งไม่ให้ปรากฏค่า http และรายละเอียดที่ช่อง URL แอดเดรส โดยมากการประมวลผลฟอร์มด้วย CGI จะเป็นการ Post
2. GET คือ การถ่ายข้อมูลทั้งหมดจากฟอร์มไปยัง CGI ในครั้งเดียว แต่ต้องไม่เกิน 8192 ตัวอักษรเนื่องจากเป็นข้อจำกัดของ URLs รวมถึงจะให้ปรากฏค่า http และรายละเอียดที่ช่อง URL แอดเดรส

ACTION คือ การกำหนดตำแหน่งของโปรแกรม CGI บนเครื่องเซิร์ฟเวอร์ที่ต้องการให้ประมวลผล โดยจะเขียนอยู่ในรูป URL เช่น `http://www.thaicom.com/cgi-bin/search.pl` หรือ `/cgi-bin/search.pl` เป็นต้น

NAME คือ การกำหนดชื่อให้แก่ฟอร์ม เพื่อให้เป็นค่าที่โปรแกรมภาษาสคริปต์ เช่น JavaScript หรือ VBScript สามารถเรียกใช้ได้จากพรอเพอร์ตี้ Name

2.5 อุปกรณ์ในการรับข้อมูล

อุปกรณ์รับข้อมูลที่ใช้ในฟอร์มมีหลายชนิด ได้แก่ Textbox, Checkbox, Radio Button, Button, Text Area และ Combo box (Selection box) เป็นต้น

การสร้างอุปกรณ์ในการรับข้อมูลทุกชนิดจะเขียนอยู่ภายในแท็ก `<FORM>...</FORM>` ส่วนใหญ่จะใช้คำสั่ง `<INPUT>` ซึ่งทั่วไปแล้วจะมีรูปแบบการเขียนดังนี้ `<INPUT TYPE = "ชนิดของอุปกรณ์รับข้อมูล" NAME = "ชื่อเรียก".....>`

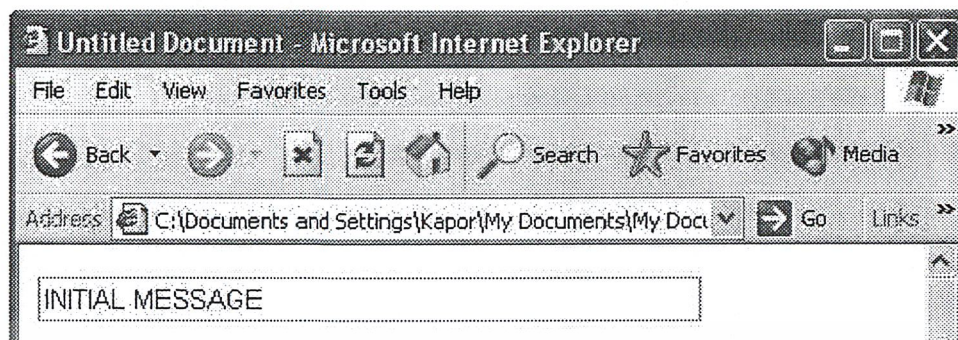
แต่อุปกรณ์รับข้อมูลบางชนิดเช่น Text Area และ Combo Box จะมีรูปแบบคำสั่งที่แตกต่างออกไป ซึ่งจะแสดงรายละเอียดไว้ในวิธีการสร้างอุปกรณ์ต่อไป

2.5.1 การสร้าง TEXTBOX

ทำหน้าที่เป็นช่องรับข้อมูลที่มีลักษณะเป็นข้อความบรรทัดเดียวให้เราพิมพ์ป้อนเข้าไป ซึ่งรูปแบบการสร้าง Textbox มีลักษณะดังตัวอย่างต่อไปนี้

```
<INPUT TYPE="TEXT" NAME="TEXT FIELD" SIZE="50" VALUE="INITIAL MESSAGE"
MAXLENGTH="100">
```

จากตัวอย่างนี้จะได้ผลปรากฏบนเบราว์เซอร์ดังรูปที่ 2.1



รูปที่ 2.1 ตัวอย่าง Textbox

INPUT TYPE คือ การกำหนดชนิดของอุปกรณ์รับข้อมูล สำหรับ Textbox จะกำหนดด้วย "TEXT"

NAME คือ การกำหนดชื่อสำหรับ Textbox เพื่อนำไปอ้างอิงในสคริปต์ เช่น จาวาสคริปต์ ในที่นี้เรียกว่า Text field และแต่ละ Textbox จะมีชื่อไม่ซ้ำกัน

SIZE คือ การกำหนดขนาดความยาวของ Textbox ตามจำนวนอักขระที่กำหนด เช่น SIZE=50 ช่อง Textbox ที่ปรากฏบนเบราว์เซอร์ ก็จะมีขนาดความยาวสำหรับ 50 ตัวอักษร

MAXLENGTH คือ การกำหนดความยาวสูงสุดของข้อมูลที่ได้รับเข้ามาได้ ตามตัวอย่าง MAXLENGTH = 100 แสดงว่ารับอักขระเข้ามาได้ 100 ตัว แม้ว่าจะกำหนดขนาดของตัว Textbox ไว้ที่ 50 ตัวอักษรก็ตาม

VALUE คือ การกำหนดข้อความล่วงหน้าเอาไว้ ใน Textbox ถ้าไม่มีการกำหนด จะไม่มีคำสั่งนี้ หรืออาจเขียนเป็น VALUE="" ก็ได้

2.5.2 การสร้างช่องรับข้อมูลแบบ PASSWORD

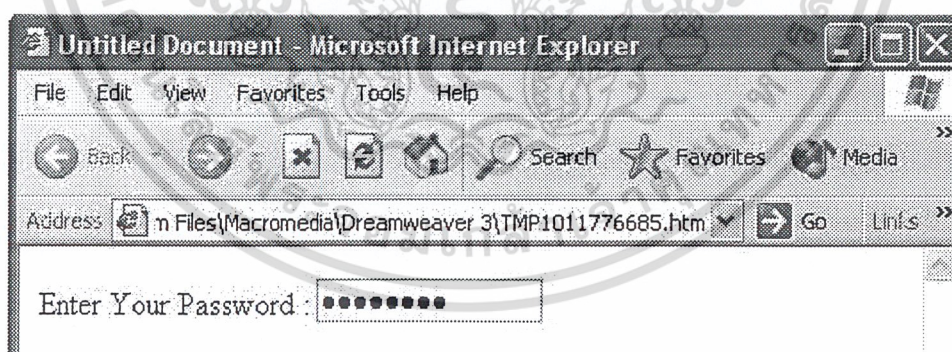
ทำหน้าที่เช่นเดียวกับช่องรับข้อมูล หรือ Textbox ธรรมดา แต่จะแทนที่สิ่งที่เราพิมพ์ด้วยเครื่องหมาย "*" ใช้สำหรับพิมพ์รหัสผ่าน โดยไม่สามารถทำการคัดลอกสิ่งที่ถูกพิมพ์เข้าไปใน Textbox ลง Clipboard ได้ เพราะคุณสมบัติของช่องรับ Password จะปกป้องข้อมูลที่เป็นรหัสผ่านเอาไว้ การสร้างช่อง Password ใช้คำสั่งดังนี้

```
<INPUT TYPE ="PASSWORD" NAME="GETPASSWORD" SIZE ="8" MAXLENGT ="8">
```

ความหมายของพรอเพอร์ตี้ต่าง ๆ ของช่องกรอกข้อมูลแบบ Password เหมือนกับ Textbox ทุกประการ แต่จะแตกต่างกันตรง TYPE ซึ่งเราต้องกำหนดให้เป็น "PASSWORD" ตัวอย่างการสร้างช่องรับข้อมูลแบบ Password มีดังนี้

```
<HTML>
<HEAD>
<TITLE>PASSWORD</TITLE>
</HEAD>
<BODY BGCOLOR= "#FFFFFF">
<FORM METHOD ="POST" ACTION ="CGISCRIP">
Enter Your Password :
<INPUT TYPE ="password" NAME = "Getpassword" SIZE ="15" MAXLENGTH ="8">
</FORM>
</BODY>
</HTML>
```

ผลที่ได้จากตัวอย่างข้างต้นเมื่อแสดงบนเบราว์เซอร์จะเป็นดังรูปที่ 2.2



รูปที่ 2.2 ตัวอย่างช่องรับข้อมูลแบบ Password

2.5.3 การสร้าง TEXT AREA

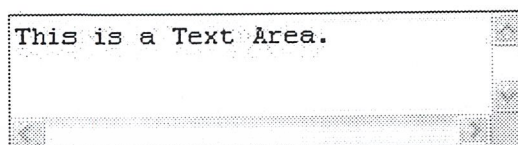
Text Area เป็นอุปกรณ์ที่ใช้ในกรณีที่มีข้อมูลที่ต้องป้อนในฟอร์มมากกว่า 1 บรรทัด โดยคำสั่งที่ใช้จะแตกต่างจาก Textbox ดังตัวอย่างต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<TEXTAREA NAME = "TextAreaField" COLS = "10" ROWS = "3" WRAP = "OFF">
```

```
This is a Text Area.</TEXTAREA>
```

ผลที่ได้จากตัวอย่างจะปรากฏบนเบราว์เซอร์ดังรูปที่ 2.3



รูปที่ 2.3 ตัวอย่างของ Text Area

<TEXTAREA....>...</TEXTAREA> คือ คำสั่งที่กำหนดฟอร์มเป็น Text Area และภายในแท็กส่วนแรก จะมีคำสั่งกำหนดพารามิเตอร์ของ Text Area ดังนี้

NAME คือ การกำหนดชื่อของ Text Area หนึ่ง ๆ ในฟอร์ม เพื่อนำไปอ้างอิงในสคริปต์

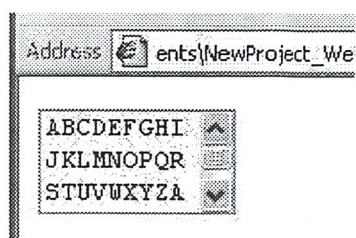
COLS คือ การกำหนดจำนวนคอลัมน์หรือความยาวตามจำนวนอักขระของ Text Area

ROWS คือ การกำหนดจำนวนแถวหรือบรรทัดของ Text Area

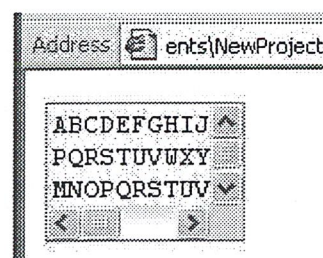
WRAP คือ การกำหนดรูปแบบของ Scroll Bar ใน Text Area เมื่อมีการป้อนข้อมูลเข้าไป โดยจะมีการกำหนดหลายแบบได้แก่

- Virtual และ Physical มีหน้าที่คล้ายกันคือ เมื่อป้อนอักขระจนครบตามที่กำหนดไว้ใน COLS จะทำการขึ้นบรรทัดใหม่ให้โดยไม่ต้องกดคีย์ Enter และจะปรากฏ Scroll Bar ขึ้นที่ด้านขวามือ เมื่อจำนวนบรรทัดเกินจากที่กำหนดไว้ใน ROWS แสดงไว้ในรูปที่ 2.4 (ถ้าไม่มีคำสั่ง WRAP ภาษา HTML จะเข้าใจว่าเป็นกรณีนี้) แต่ความแตกต่างของทั้งสองจะเป็นมุมมองระหว่างขณะสร้างในโปรแกรมจำพวก Dreamweaver กับมุมมองที่ปรากฏบนเบราว์เซอร์

- OFF คือ การให้มี Scroll Bar เกิดขึ้นเมื่อมีการเพิ่มจำนวนบรรทัด และป้อนอักขระเกินกว่าจำนวนคอลัมน์ที่กำหนดไว้ ซึ่งจะปรากฏ Scroll Bar ขึ้นที่ด้านขวา และด้านล่างของ Text Box แสดงไว้ในรูปที่ 2.5



รูปที่ 2.4 Text Area ในกรณีของ Virtual หรือ Physical



รูปที่ 2.5 Text Area ในกรณีของ Off

2.5.4 การสร้าง CHECKBOX

Checkbox มีไว้สำหรับให้ผู้ใช้เลือกหรือยกเลิกสิ่งที่เลือกได้มากกว่า 1 ครั้งโดยการคลิกเมาส์ มีลักษณะเป็นช่องสี่เหลี่ยมเล็ก ๆ รูปแบบของคำสั่งเป็นดังนี้

```
<INPUT TYPE = "checkbox" NAME = "newcheckbox" VALUE="checkbox" CHECKED>
```

ผลที่แสดงบนเบราว์เซอร์ เมื่อโหลดฟอร์มขึ้นมาครั้งแรกจะมีการเลือกที่ Checkbox อยู่ก่อนแล้ว ดังรูปที่ 2.6



รูปที่ 2.6 ตัวอย่าง Checkbox

TYPE คือ การกำหนดชนิดของอุปกรณ์รับข้อมูล สำหรับ Checkbox จะกำหนดเป็น “checkbox”

NAME คือ การกำหนดชื่อของ Checkbox แต่ละตัว เพื่อใช้อ้างอิงในสคริปต์ ในตัวอย่างนี้ให้ชื่อว่า “newcheckbox” และ Checkbox แต่ละตัวบนฟอร์มต้องมีชื่อไม่ซ้ำกัน

VALUE คือ การกำหนดค่าของข้อมูลสำหรับ Checkbox ว่าเมื่อถูกเลือกแล้ว และผู้ใช้ได้กดส่งฟอร์มไป ค่านี้จะถูกส่งไปยัง โปรแกรม CGI ทางฝั่งเซิร์ฟเวอร์

CHECKED คือ การกำหนดให้ตัว Checkbox ถูกเลือกไว้ล่วงหน้าเมื่อฟอร์มถูกโหลดขึ้นมา และไม่ต้องเติมคำสั่งใดเพิ่มในกรณีที่ไม่ต้องการให้ Checkbox ถูกเลือกไว้ก่อน

2.5.5 การสร้าง RADIO BUTTON

การใช้ Radio button เป็นการทำให้ผู้ใช้สามารถคลิกเมาส์ เลือกตัวเลือกที่ถูกกำหนดไว้ก่อน โดยจะคลิกเมาส์เลือกได้ที่ละตัวเลือกเท่านั้น และ Radio Button จะต้องรวมกันเป็นกลุ่มตั้งแต่ 2 ตัวขึ้นไป สิ่งสำคัญคือ จะต้องกำหนดชื่อ NAME ของตัวเลือกแต่ละตัวที่อยู่ในกลุ่มเดียวกันให้เป็นชื่อเดียวกัน รูปแบบมีดังนี้

```
<INPUT TYPE = "radio" NAME = "newradio" VALUE = "radiobutton" CHECKED>
```

TYPE คือ การกำหนดชนิดอุปกรณ์รับข้อมูล สำหรับ Radio Button จะกำหนดเป็น "radio"

NAME คือ การกำหนดชื่อให้ Radio Button ซึ่งในกลุ่มเดียวกันจะต้องกำหนดชื่อเหมือนกัน เพื่อใช้อ้างอิงในสคริปต์

VALUE คือ การกำหนดค่าของ Radio Button ว่าเมื่อถูกเลือก ผู้ใช้ได้ทำการส่งฟอร์มจะมีการส่งค่านี้ไปยัง CGI และในกลุ่มหนึ่ง ๆ ค่า Value ของ Radio Button แต่ละตัวจะต้องไม่เหมือนกัน เพื่อใช้อ้างอิงในสคริปต์

CHECKED คือ การกำหนดค่าเริ่มต้นให้ถูกเลือกไว้ก่อน ถ้าไม่ต้องการกำหนดค่าเริ่มต้นใด ๆ ไม่ต้องใส่พารามิเตอร์นี้ในแท็กคำสั่ง

ตัวอย่างการสร้าง Radio Button

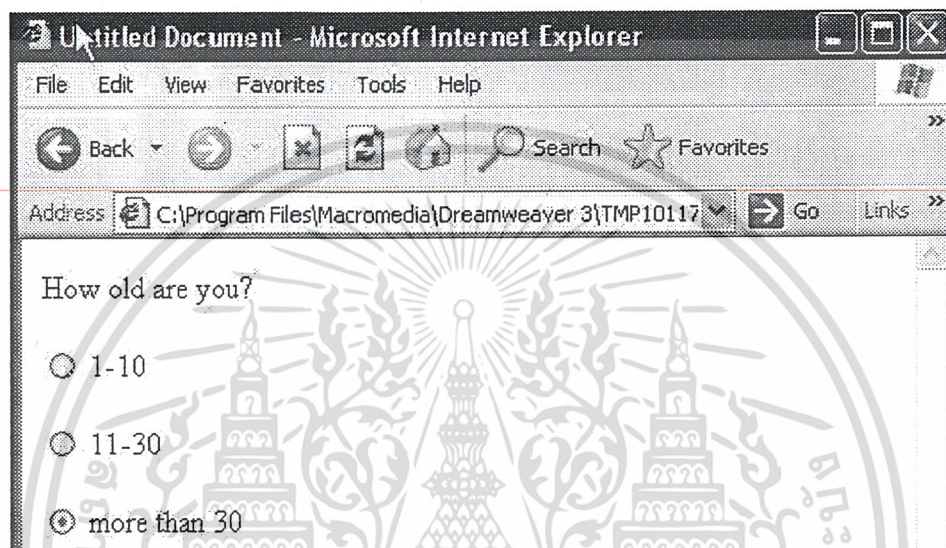
```
<HTML>
<HEAD>
<TITLE>Creating Radio Button</TITLE>
</HEAD>
<FORM METHOD = "post" action = "">
  <P>How old are you?</P>
  <P><INPUT TYPE="radio" NAME="radiobutton" VALUE="radiobutton1">1-10</P>
  <P> <INPUT TYPE = "radio" NAME = "radiobutton" VALUE = "radiobutton2">11-30</P>
  <P> <INPUT TYPE = "radio" NAME = "radiobutton" VALUE = "radiobutton3" CHECKED>
  more than 30</P>
</FORM>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

</BODY>

</HTML>

ผลที่ได้เมื่อแสดงบนเบราว์เซอร์จะเป็นดังรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างของ Radio Button

2.5.6 การสร้าง COMBO BOX

Combo Box หรือ Selection Box เป็นอุปกรณ์ที่สามารถใส่รายการจำนวนมากให้ผู้ใช้เลือกได้ และใช้พื้นที่ไม่มากเมื่อเทียบกับ Checkbox หรือ Radio Button

ชนิดของ Combo Box มี 2 ประเภทคือ

1. Menu เป็น Combo Box ที่ไม่สามารถกำหนดความสูง หรือจำนวนบรรทัดได้ และผู้ใช้สามารถเลือกได้เพียง 1 ตัวเลือกเท่านั้น สำหรับขนาดความยาวของตัว Combo Box ภาษา HTML จะปรับให้โดยอัตโนมัติให้พอดีกับความยาวของข้อความในรายการ ตัวอย่างการสร้างมีดังนี้

```
<SELECT NAME = "choose color">
```

```
<OPTION VALUE = "ColorNo.1" SELECTED>Red</ OPTION >
```

```
<OPTION VALUE = "ColorNo.2">Blue</ OPTION >
```

```
< OPTION VALUE = "ColorNo.3">Green</ OPTION >
```

```
< OPTION VALUE = "ColorNo.4">Yellow</ OPTION >
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
< OPTION VALUE = "ColorNo.5">White</ OPTION >
```

```
</SELECT>
```

ผลที่ได้จะปรากฏบนเบราว์เซอร์ ดังรูปที่ 2.8



รูปที่ 2.8 ตัวอย่างของ Combo Box แบบ Menu

`<SELECT ...></SELECT>` คือ การกำหนด Combo Box หรือ Selection Box และภายในคำสั่งส่วนแรกจะมีพรอเพอร์ตี้ดังนี้

NAME คือ การกำหนดชื่อของ Combo Box เพื่อใช้อ้างอิงในจาวาสคริปต์ และแต่ละตัวบนฟอร์มต้องมีชื่อไม่เหมือนกัน จากตัวอย่าง ให้ชื่อ Combo Box นี้ว่า “choose color”

OPTION คือ การกำหนด Item หรือตัวเลือกแต่ละตัวในลิสต์ของ Combo Box

VALUE คือ การกำหนดค่าที่ส่งคืนไปยังเซิร์ฟเวอร์เมื่อตัวเลือกนั้นถูกเลือก และค่าแต่ละตัวเลือกจะไม่ซ้ำกัน

SELECT คือ การกำหนดค่าเริ่มต้นให้ตัวเลือกนี้เป็นตัวที่ถูกเลือก

2. List เป็น Combo Box ที่ผู้ใช้สามารถกำหนดความสูง หรือจำนวนบรรทัดได้ และสามารถกำหนดให้ทำการเลือกได้มากกว่า 1 ตัวเลือก ตัวอย่างการสร้างมีดังนี้

```
<SELECT NAME= "select" MULTIPLE SIZE="3">
```

```
<OPTION VALUE="ColorNo.1">Red</ OPTION >
```

```
<OPTION VALUE = "ColorNo.2">Blue</ OPTION >
```

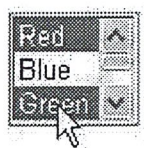
```
<OPTION VALUE = "ColorNo.3">Green</ OPTION >
```

```
<OPTION VALUE = "ColorNo.4">Yellow</ OPTION >
```

```
<OPTION VALUE = "ColorNo.5">White</ OPTION >
```

```
</SELECT>
```

ผลที่ได้จากตัวอย่างเมื่อแสดงบนเบราว์เซอร์จะเป็นดังรูปที่ 2.9



รูปที่ 2.9 ตัวอย่างของ Combo Box แบบ List

พรอเพอร์ตี้ที่มีใน Combo Box แบบ List จะเหมือนกับแบบ Menu แต่มีบางชนิดที่เพิ่มขึ้นมาได้แก่

SIZE คือ การกำหนดความสูงของ Combo Box จากตัวอย่าง size="3" จะแสดงรายการให้เห็น 3 บรรทัด

MUTIPLE คือ การกำหนดให้ผู้ใช้เลือกได้ครั้งละหลายๆ ตัวเลือก แต่ถ้าต้องการให้ผู้ใช้เลือกได้เพียง 1 ตัวเลือกเท่านั้น ไม่ต้องพิมพ์คำสั่งนี้

2.5.7 การสร้างปุ่ม SUBMIT

ปุ่ม Submit Button จัดเป็น Button ชนิดหนึ่ง ซึ่ง Button ถูกแบ่งออกเป็น 3 ชนิด คือ Submit, Reset และ Button

ในการ Submit สามารถทำได้ 2 วิธีคือ

1. กดคีย์ <Enter> เพื่อ Submit กรณีนี้จะใช้เมื่อในฟอร์มมี Textbox เพียงอันเดียว

2. คลิกเมาส์ที่ปุ่ม Submit กรณีนี้จะใช้เมื่อในฟอร์มมีอุปกรณ์รับข้อมูลหลายตัว จึงต้องสร้างปุ่ม Submit ขึ้นมา โดยปกติจะนำมาใช้เพื่อยืนยันความถูกต้องของข้อมูล โดยใช้คำสั่ง

```
<INPUT TYPE="submit" NAME="Submit" VALUE="Submit">
```

เมื่อดูผลจากเบราว์เซอร์จะเป็นดังรูปที่ 2.10



รูปที่ 2.10 ตัวอย่างปุ่ม Submit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPE คือ การกำหนดให้เป็นปุ่ม Submit สำหรับการส่งข้อมูลจากฟอร์มไปยังเครื่องเซิร์ฟเวอร์ หรือกำหนดรูปแบบของข้อมูลทั้งหมดในฟอร์มให้เป็นลักษณะของการยืนยันข้อมูล สำหรับปุ่ม Submit จะกำหนดเป็น "submit"

NAME คือ การกำหนดชื่อปุ่ม Submit เพื่อนำไปอ้างอิงในสคริปต์

VALUE คือ การกำหนดให้โปรแกรมเบราเซอร์สร้างปุ่มตามข้อความที่กำหนด

2.5.8 การสร้างปุ่ม RESET

ปุ่ม Reset หรือ Clear มีไว้เพื่อให้ผู้ใช้ยกเลิกข้อมูลที่ป้อนเข้าไปในฟอร์ม หรือกลับไปสู่สถานะเริ่มต้น การสร้างปุ่ม Reset มีรูปแบบดังนี้

```
<INPUT TYPE = "reset" NAME="Reset" VALUE = "Reset">
```

เมื่อแสดงผลบนเบราเซอร์จะเป็นดังรูปที่ 2.11



รูปที่ 2.11 ตัวอย่างปุ่ม Reset

จะสังเกตว่า การกำหนด TYPE สำหรับปุ่ม Reset จะเป็น “reset” ส่วนพรอเพอร์ตี้ตัวอื่น ๆ จะมีความหมายเช่นเดียวกับปุ่ม Submit

2.5.9 การสร้างปุ่ม BUTTON

กรณีที่เราต้องการไม่ต้องการให้ปุ่มในฟอร์มทำหน้าที่ Submit หรือ Reset สามารถกำหนดได้ตามตัวอย่างของคำสั่งต่อไปนี้

```
<INPUT TYPE = "button" NAME="Button" VALUE= "OK">
```

ผลที่ปรากฏบนเบราเซอร์จะเป็นดังรูปที่ 2.12



รูปที่ 2.12 ตัวอย่างการสร้างปุ่ม Button

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPE สำหรับปุ่ม Button จะกำหนดเป็น "button" ส่วนพรอเพอร์ตี้อื่น ๆ นั้นมีความหมายเช่นเดียวกับปุ่ม Submit และ Reset

2.6 จาวาสคริปต์ (JavaScript)

จาวาสคริปต์เป็นภาษาสคริปต์เชิงวัตถุ พัฒนาโดยบริษัท เน็ตสเคป คอมมิวนิเคชัน (Netscape Communications Corporation) ซึ่ง ถูกออกแบบมาสำหรับการเขียนโปรแกรมบนอินเทอร์เน็ตโดยเพิ่มเข้าไปในเว็บเพจเพื่อใช้ประโยชน์สำหรับงานด้านการคำนวณ การแสดงผล การรับ-ส่งข้อมูล และที่สำคัญคือสามารถโต้ตอบกับผู้ใช้ได้ทันที จาวาสคริปต์สามารถทำงานได้ทั้งฝั่งไคลเอนต์ ซึ่งสคริปต์จะทำงานหลังจากถูกโหลดลงในหน่วยความจำในเครื่องของผู้ใช้แต่ละคน อีกทั้งยังทำงานได้บนฝั่งเซิร์ฟเวอร์ ซึ่งกรณีนี้สคริปต์จะถูกแปลทางฝั่งเซิร์ฟเวอร์ แต่ใช้ได้เฉพาะ LiveWire ของเน็ตสเคปโดยตรง นอกจากนี้ยังสามารถทำงานข้ามแพลตฟอร์มได้ ทั้งยังมีความสามารถด้านอื่น ๆ อีกหลายประการที่ช่วยสร้างความน่าสนใจให้กับเว็บเพจได้เป็นอย่างมาก

จาวาสคริปต์นั้นมีความคล้ายคลึงกับภาษาจาวาในด้านรูปแบบโครงสร้างภาษา และการเขียนโปรแกรมเชิงวัตถุ แต่จาวาจะเป็นภาษาที่มีความซับซ้อนและใช้งานได้ยากกว่า โดยจาวาสคริปต์นั้นไม่ต้องกำหนดค่าตัวแปร ,ไม่ต้องคอมไพล์ก่อนถูกแปลโดยไคลเอนต์ อีกทั้งไม่จำเป็นต้องรู้จักคลาสและการสืบทอดเหมือนจาวา และจาวาสคริปต์ยังใช้เทคนิคการตรวจสอบออบเจกต์อ้างอิงที่รันไทม์ (Dynamic binding) แต่จาวานั้นมีการเรียกใช้งานออบเจกต์อ้างอิงขณะคอมไพล์ (Static binding)

2.7 พื้นฐานของออบเจกต์ (Object), เมธอด (Method) และอีเวนต์ (Event)

การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) เป็นวิธีการเขียนโปรแกรมสมัยใหม่ มุ่งเน้นประสิทธิภาพและช่วยให้การออกแบบและเขียนโปรแกรม เป็นธรรมชาติมากขึ้นโดยจะมองทุกอย่างภายในโปรแกรม และสภาพแวดล้อมเป็นเสมือนออบเจกต์ต่าง ๆ

2.7.1 ออบเจกต์

ออบเจกต์ (Object) ในจาวาสคริปต์ คือชุดของพรอเพอร์ตี้และเมธอดที่ประกอบกันขึ้นเป็นชุดของการกำหนดลักษณะที่สามารถดู แก้ไขและโต้ตอบได้ พรอเพอร์ตี้ดีเป็นตัวอย่างที่ดี เพราะมองเห็นง่าย และด้วยการเขียนโปรแกรมเชิงวัตถุเราสามารถกำหนดพรอเพอร์ตี้ได้ เช่น ความเร็วในการเคลื่อนไหวของภาพ ลักษณะของฟอนต์บนเพจ หรือขนาดของรูปหลายเหลี่ยมที่แสดงบนเพจพรอเพอร์ตี้ที่ใช้กับ สีเหมือนกับที่ใช้ในเอกสาร HTML ควบคุมได้โดยการอ้างอิงถึงออบเจกต์และพรอเพอร์ตี้ด้วยชื่อ จากนั้นจึงกำหนดค่า เช่น สีพื้นหลังของเพจสามารถเปลี่ยนได้ด้วยคำสั่ง `document.bgcolor = "blue"`

2.7.2 เมธอด

เมธอด (Method) คือเทคนิคที่ใช้เพื่อทำงานที่เกี่ยวข้องกับออบเจกต์และพรอเพอร์ตี้ของออบเจกต์ นอกจากจาวาสคริปต์สามารถควบคุมการทำงานของออบเจกต์ที่ถูกกำหนดไว้ในตัวภาษาเองแล้วยังสามารถควบคุมออบเจกต์ของเบราว์เซอร์ ออบเจกต์ของ HTML และออบเจกต์ที่สร้างในฟอร์มของจาวาแอปเพล็ต โดยไม่จำเป็นต้องรู้ถึงรายละเอียดของแอปเพล็ตนั้น เช่น `document.write("Hi!")` เป็นการประยุกต์ใช้เมธอด `write` กับออบเจกต์ `document` ให้แสดงข้อความที่ระบุบนจอภาพ

เมธอดจะเป็นตัวกำหนดขอบเขตการโต้ตอบกับออบเจกต์ ในจาวาสคริปต์ `window` ซึ่งเป็นออบเจกต์ชนิดหนึ่ง สามารถเปิดหรือปิด ได้โดยใช้เมธอด `open` และ `close` ตามลำดับ

2.7.3 อีเวนต์และลำดับการทำงานของโปรแกรม

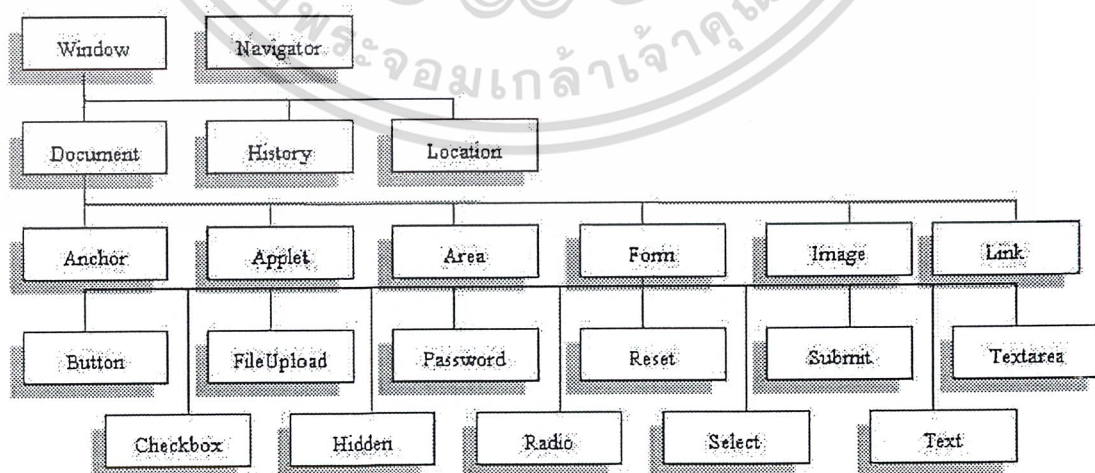
ออบเจกต์มีการทำงานเป็นลำดับขั้น ดังรูปที่ 2.13 จาวาสคริปต์ที่ฝังในเอกสาร HTML จะถูกแปล (interpret) บรรทัดต่อบรรทัด คำสั่งใดเขียนก่อนจะถูกแปลไปใช้งานก่อน แต่บางคำสั่งจะทำงานได้ก็ต่อเมื่อมีการระบุออบเจกต์ที่แน่นอนและมีอีเวนต์เกิดขึ้นก่อน ซึ่งอีเวนต์เหล่านี้จะควบคุมลำดับการทำงานภายในโปรแกรมสคริปต์ เช่นต้องคลิกปุ่มก่อนจึงจะทำงานต่อไปได้

จาวาสคริปต์แตกต่างจากภาษาอื่น ๆ เช่น C, Pascal หรือ Basic โดยไม่ขึ้นกับการวนรอบของโปรแกรมหลัก แต่การทำงานจะขึ้นอยู่กับลำดับของโปรแกรมที่สร้างไว้ข้างในซึ่งเกิดเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการที่ เบราเซอร์อ่านเอกสารแล้วตอบสนองต่อออบเจกต์ที่พบ การตอบสนองต่ออีเวนต์ที่เกิดขึ้นจากการที่ผู้ใช้ได้กระทำลงไปคือ อีเวนต์แฮนด์เลอร์ (eventhandler) จากรูปที่ 2.14 แสดงให้เห็นลำดับการทำงานที่ขึ้นกับอีเวนต์ของจาวาสคริปต์ การโหลดเอกสาร HTML ขึ้นมาจะเป็นการเรียกอีเวนต์แฮนด์เลอร์ onLoad ซึ่งจะทำให้จาวาสคริปต์ทำงาน หรือถ้าผู้ใช้กรอกฟอร์มและคลิกปุ่ม Submit อีเวนต์แฮนด์เลอร์ onSubmit จะถูกเรียกซึ่งอาจจะมีการตรวจสอบความถูกต้องของข้อมูลในฟอร์มผ่านจาวาสคริปต์ต่อไป ส่วนการเลื่อนพอยน์เตอร์ของเมาส์ผ่านการเชื่อมโยงของ HTML ถือเป็นอีเวนต์ที่ทำให้จาวาสคริปต์ทำงานหากมีการกำหนดสคริปต์ที่สอดคล้องกับการเชื่อมโยง และการยกเลิกการโหลดเอกสารคืออีเวนต์สุดท้ายในลำดับการทำงานของโปรแกรม สังเกตได้ว่าลำดับการทำงานของโปรแกรมทั้งหมดไม่ได้ขึ้นอยู่กับการวนซ้ำ แต่จะสัมพันธ์กับลำดับการกระทำของผู้ใช้

2.8 พื้นฐานการเขียนสคริปต์

ภาษาจาวาสคริปต์เป็นภาษาที่ง่ายต่อการเรียนรู้ มีฟังก์ชันการทำงานในระดับสูงที่สามารถทำงานด้วยการใช้คำสั่งเพียง 1- 2 คำสั่ง ถ้าเทียบกับภาษาจาวาหรือ C การทำงานในลักษณะเดียวกันอาจจำเป็นต้องเขียนโปรแกรมหลายบรรทัด และสำหรับผู้ที่ไม่มีความชำนาญด้านการเขียนโปรแกรม สามารถดาวน์โหลดสเน็ปเล็ต (snippets) จากอินเทอร์เน็ต ซึ่งเป็นโปรแกรมสคริปต์สั้น ๆ สามารถใช้ได้ทันที โดยนำมาวางไว้บนเอกสาร HTML มีประโยชน์เพื่อเพิ่มความสามารถต่าง ๆ ให้กับเว็บเพจนอกเหนือจากคำสั่ง HTML



รูปที่ 2.13 ลำดับการสืบทอดออบเจกต์ (Object Hierarchy)

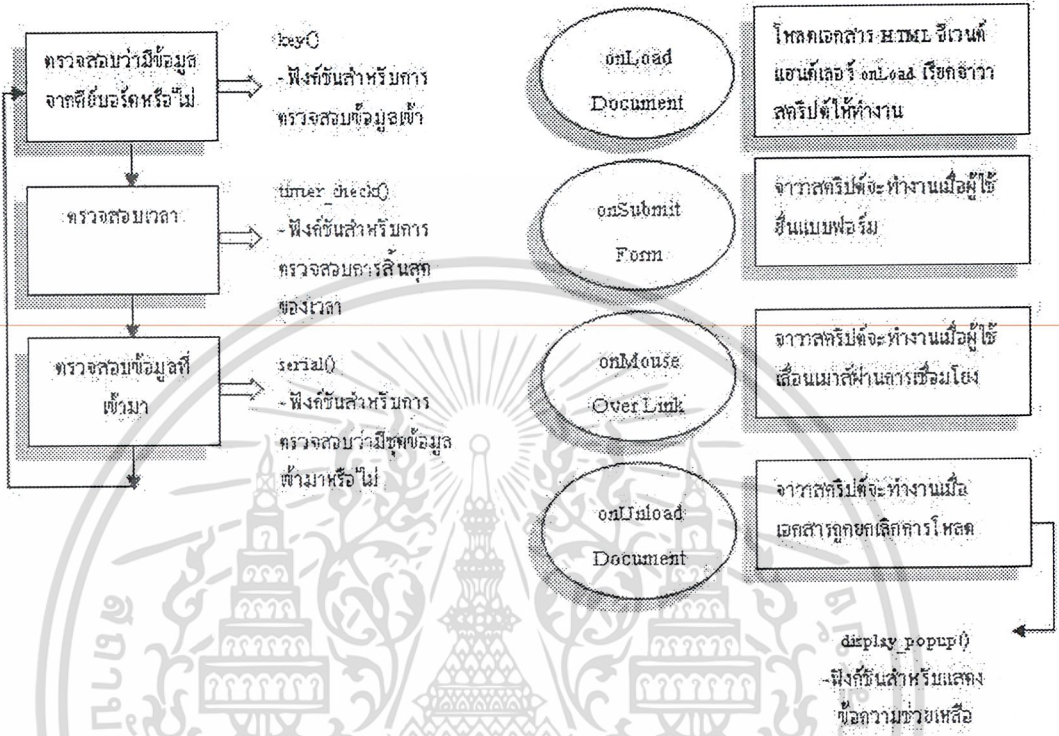
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการวนซ้ำในโปรแกรมหลัก

ลำดับการทำงานบนเดสทอปอ็อบเจกต์ของจาวาสคริปต์

Main()

อีเวนต์แอสเสอร์



รูปที่ 2.14 เปรียบเทียบรูปแบบการวนซ้ำและลำดับการทำงานตามอีเวนต์ในจาวาสคริปต์

2.8.1 การกำหนดภาษาจาวาสคริปต์

การกำหนดชุดคำสั่งของจาวาสคริปต์ในเอกสาร HTML สามารถทำได้โดยใช้คำสั่ง `<SCRIPT LANGUAGE = "JavaScript">` เป็นคำสั่งเริ่มต้น แล้วปิดท้ายด้วย คำสั่ง `</SCRIPT>`

ถ้าสำหรับเบราว์เซอร์รุ่นเก่าที่ไม่สนับสนุนจาวาสคริปต์ จะเกิดความผิดพลาดขึ้นเมื่อแสดงผลเว็บเพจทางหน้าจอ ดังนั้นจึงต้องเพิ่มเครื่องหมาย `<!---->` ระหว่างคำสั่งกำหนดภาษาสคริปต์ดังนี้

```
<SCRIPT LANGUAGE = "JavaScript">
<!-- JavaScript Statement
..... -->
</SCRIPT>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเพิ่มเครื่องหมายนี้จะช่วยให้เบราว์เซอร์ที่ไม่สนับสนุนจาวาสคริปต์ตีความข้อความภายในนี้เป็นหมายเหตุ (comment) ตามรูปแบบคำสั่งกำหนดหมายเหตุของ HTML (`<!--.....-->`) และควรใช้คำสั่งนี้ทุกครั้งที่เขียนโปรแกรม เพราะจะไม่มีผลใด ๆ ทั้งสิ้นกับการทำงานโดยตรงของจาวาสคริปต์

2.8.2 ตำแหน่งของสคริปต์ในเอกสาร HTML

คำสั่ง `<SCRIPT>...</SCRIPT>` เป็นการกำหนดบริเวณพื้นที่และขอบเขตของการสั่งงานด้วยชุดคำสั่งจาวาสคริปต์ หากเขียนชุดคำสั่งนอกเหนือจากบริเวณนี้ HTML จะตีความคำสั่งสคริปต์เป็นเหมือนข้อความธรรมดาบนเว็บเพจ

การเขียนชุดคำสั่งของจาวาสคริปต์สามารถเขียนได้ทั้งในส่วน HEAD และ ส่วน BODY ของเอกสาร HTML แต่ผู้ใช้ส่วนใหญ่นิยมเขียนไว้ในส่วน HEAD มากกว่า เพราะมีข้อดีคือคำสั่งและฟังก์ชันต่าง ๆ ของจาวาสคริปต์ จะถูกโหลดเก็บไว้ในหน่วยความจำก่อนการโหลดส่วนอื่น ๆ ของเอกสาร HTML และทำให้สคริปต์ส่วนนี้สามารถใช้งานได้ทันทีเมื่อถูกโหลด อีกทั้งเป็นการทำให้แน่ใจว่าลำดับของอีเวนต์ต่าง ๆ ที่ถูกกำหนดโดยอีเวนต์แฮนด์เลอร์ จะดำเนินไปอย่างถูกต้อง

สำหรับการเขียนสคริปต์ไว้ในส่วน BODY นิยมใช้ร่วมกับฟอร์ม `<FORM>....</FORM>` โดยกำหนดอีเวนต์แฮนด์เลอร์ไว้กับส่วนประกอบต่าง ๆ ของฟอร์ม

นอกจากการเขียนสคริปต์ลงในเอกสาร HTML แล้ว สามารถทำเป็นไฟล์อ้างอิงได้โดยใช้คำสั่ง `<SCRIPT SRC = "ชื่อไฟล์.JS"> </SCRIPT>` ซึ่งตำแหน่งของไฟล์ที่ระบุด้วยคำสั่ง SRC จะอยู่ที่ฝั่งเซิร์ฟเวอร์มากกว่าที่จะเป็นฝั่งไคลเอนต์

2.9 กลไกในจาวาสคริปต์

2.9.1 การอ้างอิงชื่อในจาวาสคริปต์

การอ้างอิงออบเจกต์ต่าง ๆ จะใช้โครงสร้างการแบ่งชื่อออบเจกต์, พรอเพอร์ตี้และเมธอดด้วยจุดทำให้เกิดความกระชับในการเขียนโปรแกรม ออบเจกต์ในจาวาสคริปต์มีหลายชนิด เช่น anchor, frame, document เป็นต้น ในการระบุชื่อต้องเรียงตามระดับของออบเจกต์เริ่มจากระดับสูงแล้วจึงเป็นออบเจกต์ที่ต้องการเฉพาะเจาะจงสุดท้ายจึงจะเป็นเมธอดเช่น `document.form(1).submit` เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีนี้ document คือการระบุถึงออบเจกต์ซึ่งเป็นเอกสารที่โหลดในปัจจุบัน form(1) คือการระบุถึงฟอร์มตัวที่สองในเอกสาร (ลักษณะนี้จะเป็นอาร์เรย์ ซึ่งฟอร์มแรกจะเป็น form(0) จากนั้นจะเป็นการประยุกต์ใช้เมธอด submit() กับออบเจกต์ เพื่อให้ในการประมวลผลมีการยอมรับฟอร์มเกิดขึ้น

ซึ่งยังสามารถอ้างอิงพรอปเพอร์ตี้ของออบเจกต์ได้ เช่น window.status เป็นการอ้างอิงถึงข้อความสถานะที่ปรากฏทางส่วนล่างซ้ายของวินโดว์ของเบราเซอร์ ซึ่งออบเจกต์แต่ละตัวจะมีพรอปเพอร์ตี้แตกต่างกัน เช่นออบเจกต์ document จะมีพรอปเพอร์ตี้ document.bgColor ที่อ้างอิงสีพื้นหลังของเอกสาร document.title อ้างอิงชื่อของเอกสาร HTML ที่กำหนดด้วยแท็ก <TITLE> และ document.image(3) อ้างอิงภาพบนเอกสารจากลำดับที่ 4 ของอาร์เรย์ เป็นต้น

เมื่อมีการระบุชื่อของออบเจกต์ถูกต้องแล้ว สามารถเขียนคำสั่งเพื่อกำหนดค่าของพรอปเพอร์ตี้ได้ เช่น เมื่อต้องการระบุค่า Check box ชื่อ document.inquiry.check1 ถูกเลือก คำสั่งจะเป็นดังนี้

```
document.inquiry.check1.checked = true
```

จากคำสั่งนี้ฟอร์มไม่ได้กำหนดด้วยการใช้อาร์เรย์ แต่เป็นการกำหนดชื่อฟอร์มโดยระบุไว้ระหว่างแท็ก <FORM>...<FORM> ให้ชื่อฟอร์มถูกตั้งเป็น NAME = "inquiry"

2.9.2 การตั้งชื่อตัวแปรและฟังก์ชัน

ตัวแปร เป็นพื้นที่ในหน่วยความจำของระบบซึ่งทำหน้าที่รองรับค่าที่ใช้ในสคริปต์ สามารถเก็บตัวเลข กลุ่มตัวอักษร และการอ้างอิงทางตรรกศาสตร์ และมีการเข้าถึงโดยการระบุชื่อตัวแปรนั้น ๆ

ฟังก์ชันใช้สำหรับการสร้างและเก็บส่วนโมดูลของโปรแกรมที่ใช้ในการทำงาน เช่น การคำนวณค่าและทำหน้าที่เป็นตัวจัดการอีเวนต์ (eventhandler) ภายในจาวาสคริปต์

การตั้งชื่อตัวแปรและฟังก์ชันนั้นเหมือนกันโดยมีหลักการอยู่ว่าต้องไม่ซ้ำกับคำสงวน (reserved word) ในจาวาสคริปต์ ต้องเริ่มต้นด้วยตัวอักษรในภาษาอังกฤษ และเครื่องหมาย _ หรือ \$ หลังจากนั้นจะตามด้วยตัวเลข 0-9 หรือตัวอักษรใด ๆ ก็ได้ แต่ภายในชื่อต้องไม่มีช่องว่าง หรือสัญลักษณ์พิเศษ เช่น *, #, @

ตัวอักษรภาษาอังกฤษที่ใช้ จะเป็นตัวพิมพ์เล็กหรือพิมพ์ใหญ่ก็ได้ ซึ่งจาวาสคริปต์จะจำแนกความแตกต่างของตัวแปรนั้น ๆ ว่าเป็นตัวแปรคนละตัวกัน

2.9.3 คำสงวน

คำสงวน (reserved word) หมายถึงคำที่จาวาสคริปต์สงวนไว้ไม่ให้นำมาตั้งชื่อในโปรแกรม เพราะคำเหล่านี้มีความหมายเฉพาะตัวอยู่แล้ว คำสงวนมีอยู่หลายคำ บางคำก็นำมาใช้ได้เลย แต่บางคำต้องระบุพารามิเตอร์เพิ่มเข้าไป บางคำต้องใช้ร่วมกับออบเจกต์ หรือฟังก์ชัน เช่น abstract, boolean, catch, else, for, if, null, this, with, void, int หรือ import ฯลฯ

2.9.4 ตัวดำเนินการและตัวแปร

ตัวดำเนินการ (operator) เป็นส่วนหนึ่งของโครงสร้างทางตรรกศาสตร์ภายในจาวาสคริปต์ ซึ่งเป็นเครื่องหมายกำหนดกรรมวิธีทางคณิตศาสตร์, พีชคณิต, บูลีน, การเปรียบเทียบ ฯลฯ ระหว่างข้อมูล 2 ตัวที่เรียกว่า โอเปแรเอนด์ (operand) โดยอาจมีค่าเป็นตัวเลข ข้อความ ค่าคงที่ หรือตัวแปรต่าง ๆ เช่น บวก ลบ คูณ หาร ฯลฯ

ตัวดำเนินการในจาวาสคริปต์มีหลายชนิด เช่น assignment, comparison, arithmetic, logical, string, bitwise และ special operator

การผสมตัวดำเนินการและตัวแปรเข้าด้วยกันเรียกว่า นิพจน์ อาจประกอบไปด้วยตัวเลข ทั้งทศนิยมและจำนวนเต็ม ค่าทางตรรกศาสตร์ หรือกลุ่มตัวอักษรเฉพาะ นิพจน์จะถูกประเมินด้วยตัวดำเนินการต่างๆ เพื่อให้ได้ผลลัพธ์ออกมา เช่น $(11 \leq 9) \ \&\& \ (16 > 5)$

ในการประกาศตัวแปร สามารถทำได้โดยใช้ตัวดำเนินการ assignment (=) ซึ่งจะเก็บค่าที่ได้รับจากนิพจน์กำหนดค่าทางขวา และใช้คำสั่ง var หรือไม่จำเป็นต้องใช้ก็ได้ดังนี้

StateBird = "Robin" หรือ var StateBird = "Robin"

จากตัวอย่างเป็นการกำหนดกลุ่มตัวอักษร Robin ให้กับตัวแปร StateBird

ตัวแปรจะมีขอบเขตเป็นโลคอล หรือ โกลบอล ขึ้นอยู่กับตำแหน่งที่ถูกกำหนด ซึ่งตัวแปรแบบโกลบอลจะถูกสร้างภายนอกฟังก์ชันโดยใช้คีย์เวิร์ด var โดยส่วนต่าง ๆ ของโปรแกรมสามารถเข้าถึงได้ แต่ตัวแปรแบบโลคอล จะถูกสร้างไว้ในขอบเขตของฟังก์ชัน โดยส่วนอื่น ๆ ของโปรแกรมนอกเหนือจากในฟังก์ชันไม่สามารถเข้าถึงได้

2.10 การสร้างออบเจ็กต์ใหม่

ในการสร้างออบเจ็กต์ใหม่ขั้นตอนแรกต้องกำหนดชนิดของออบเจ็กต์นั้นก่อน โดยการใช้ฟังก์ชัน และภายในฟังก์ชันต้องใช้คำสั่งกำหนดพรอเพอร์ตี้และเมธอด โดยการใช้คีย์เวิร์ด `this` ซึ่งช่วยในการอ้างถึงออบเจ็กต์ที่อยู่ในปัจจุบัน จากตัวอย่างเป็นการสร้างออบเจ็กต์เพื่ออธิบายชนิดของรถยนต์ชื่อ `car` มีพรอเพอร์ตี้ `make`, `model`, `year` นำมาสร้างเป็นฟังก์ชันได้ดังนี้

```
function car(make, model, year) {
  this.make = make
  this.model = model
  this.year = year
}
```

หลังจากกำหนดชนิดของออบเจ็กต์เรียบร้อยแล้ว ต่อไปต้องสร้างอินสแตนซ์ออบเจ็กต์ขึ้นมาด้วยคีย์เวิร์ด `new` ตัวอย่างนี้จะเป็นการสร้างออบเจ็กต์ใหม่ ที่เรียกว่า `mycar` ขึ้นมา

```
mycar = new car("Eagle", "Talon Tsi", 1993)
```

ดังนั้น พรอเพอร์ตี้ของออบเจ็กต์ `mycar` นั่นคือ `mycar.make` มีค่าเป็น `Eagle`, `mycar.model` มีค่าเป็น `Talon Tsi` และ `mycar.year` จะมีค่าเป็น `1993`

2.11 ฟังก์ชันและเมธอด

เมธอดจัดว่าเป็นฟังก์ชันชนิดหนึ่ง ที่จาวาสคริปต์กำหนดให้นำไปใช้งานได้ที่ เมธอดมีอยู่หลายตัวเช่น `submit`, `exp`, `write`, `anchor`, `go`, `link`, `focus`, `click`, `select` และ `clear` เป็นต้น และ ฟังก์ชันอีกประเภทหนึ่งเป็นฟังก์ชันที่ผู้ใช้กำหนดขึ้นมาเอง

ในการเรียกใช้ฟังก์ชันหรือเมธอดนั้น จะใช้ในลักษณะของคำสั่ง และผลของการเรียกใช้ฟังก์ชันจะมีการส่งค่ากลับคืน ไปยังโปรแกรมส่วนที่เรียกด้วย เช่น `CovertChar()`

การเขียนฟังก์ชัน เริ่มต้นด้วยคีย์เวิร์ด `function` ดังตัวอย่างต่อไปนี้

```
function RepeatFunction() { document.write("Welcome to JavaScript world.<BR>"); }
```

ในการเรียกใช้ฟังก์ชันนี้เพียงแต่เขียน `RepeatFunction()` ก็จะมีปรากฏข้อความ `Welcome to JavaScript world.` บนหน้าจอของเบราว์เซอร์

2.12 การใช้งานอาร์เรย์

อาร์เรย์ คือที่เก็บข้อมูลชนิดต่าง ๆ เช่น กลุ่มตัวอักษร เลขจำนวนเต็ม และทศนิยม หรือ ออบเจกต์แต่ละตัว ในการสร้างอาร์เรย์สามารถกำหนดคุณสมบัติและความยาวได้ เช่น ต้องการ สร้างอาร์เรย์ชื่อ students มีความยาว 35 ตัวอักษร ที่ใช้ในการเก็บชื่อของนักเรียนแต่ละชั้น สามารถ ป้อนคำสั่งต่อไปนี้ลงไปในสคริปต์

```
students = new Array(35)
```

ตำแหน่งดัชนีแรกในอาร์เรย์คือ 0 และส่วนประกอบของอาร์เรย์จะถูกปิดด้วยวงเล็บ ตัวอย่างต่อไปนี้เป็นการป้อนชื่อนักเรียน 2 คนแรกลงในอาร์เรย์

```
students[0] = "John Brown"
```

```
students[1] = "Walter Abrego"
```

2.13 การโปรแกรมแบบใช้เงื่อนไขและการทำซ้ำ

คำสั่งเงื่อนไขในภาษาคอมพิวเตอร์จะถูกใช้สำหรับควบคุมลำดับการทำงานของ โปรแกรม แบริช (branch) คือตำแหน่งที่สคริปต์หนึ่ง ๆ สามารถแยกการทำงานออกไปใน สองทิศทางหรือมากกว่า โดยขึ้นอยู่กับเงื่อนไข ซึ่งคำสั่งที่นิยมใช้กันมากในการโปรแกรมจาวาสคริปต์ ได้แก่ if...else

นอกจากการใช้เงื่อนไขแล้ว ในการสร้างโปรแกรมคอมพิวเตอร์ไม่ว่าภาษาใดก็ตาม มักมีการให้โปรแกรมทำงานตามฟังก์ชันหรือคำสั่งซ้ำ ๆ กัน สำหรับจาวาสคริปต์มักใช้ คีย์เวิร์ด for และ while เพื่อเริ่มต้นวงรอบการทำงาน ซึ่งสามารถรองรับการทำงานชนิดต่าง ๆ ได้อย่างมีประสิทธิภาพ

2.13.1 คำสั่ง if

if เป็นคำสั่งสำหรับกำหนดให้โปรแกรมทำงานอย่างมีเงื่อนไข โดยทำการทดสอบค่า นิพจน์ ซึ่งผลที่ได้จะเป็นจริงหรือเท็จค่าใดค่าหนึ่ง ถ้าเป็นจริงก็จะดำเนินการตามคำสั่งที่กำหนดไว้ในวงเล็บ {} แต่ถ้าผลเป็นเท็จ จะข้ามการทำงานนั้นไป ตัวอย่างการเขียนคำสั่ง if มีดังนี้

```
if (form.weight.value > form.height.value-110) { alert ("You are very fat.") }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่าง จะมีการรับข้อมูลน้ำหนักตัว และส่วนสูงของผู้ใช้จากฟอร์ม หากน้ำหนักตัวมากกว่าส่วนสูงลบด้วย 110 ตามที่นิพจน์เงื่อนไขกำหนด จะปรากฏกล่องข้อความที่เขียนว่า You are very fat. ขึ้นมาบนหน้าจอของเบราเซอร์

2.13.2 คำสั่ง if...else

if...else เป็นคำสั่งสำหรับการทดสอบเงื่อนไขที่ทำงานได้เหมือนกับการใช้คำสั่ง if เพียงแต่สามารถรองรับผลการทดสอบที่ให้ค่าเป็นเท็จโดยใช้ตัววิธ else ได้ด้วย ช่วยให้โปรแกรมมีขีดความสามารถมากกว่าเดิม ดังตัวอย่างต่อไปนี้

```
if (form.weight.value > form.height.value-110) { alert ("You are very fat.") }
else {
    if (form.weight.value == form.height.value-110) { alert ("You are perfect.") }
    else { alert ("You are quite skinny.") }
```

2.13.3 คำสั่ง switch

ในกรณีที่ต้องการทดสอบเงื่อนไขให้ได้มากกว่า 2 ผลลัพธ์ การใช้คำสั่ง switch จะสะดวกกว่าการใช้คำสั่ง if...else ซ้อนกันหลาย ๆ ชุด มีรูปแบบการเขียนดังนี้

```
switch (expression){
    case label :
        statement ;
        break ;
    case label :
        statement;
        break ;
    .....
    default : statement ;
}
```

เมื่อผลการทดสอบนิพจน์ในวงเล็บหลังคำสั่ง switch ให้ผลลัพธ์ออกมามาตรงกับ label (เป็นไปได้อีกทั้งตัวเลขและสตริง) โค้ดโปรแกรมก็จะทำงานตามคำสั่งที่อยู่ใน label นั้น แต่ถ้าผลการตรวจสอบไม่ตรงกับ label ใดเลย โปรแกรมจะทำงานตามคำสั่งที่อยู่ใน default

ในทุก ๆ case แต่ละชุด ต้องมีคำสั่ง break ปิดท้ายด้วยเสมอ เมื่อมีความผิดพลาดที่เกิดขึ้น คำสั่งนี้จะบังคับให้โปรแกรมเลิกการทำงานในบล็อกคำสั่ง switch ทันที

2.13.4 คำสั่ง for

for เป็นคำสั่งที่ใช้สำหรับสั่งให้โปรแกรมทำงานซ้ำ ๆ กันภายในวงรอบ โดยมีข้อแม้ว่าเงื่อนไขที่กำหนดต้องมีค่าเป็นจริงเสมอ วิธีการใช้งานคำสั่ง for สามารถเขียนได้ตามตัวอย่างต่อไปนี้

```
for (count = 1; count <= 10; count++) { document.write (“แฉะตัวที่ : “+ count +”<BR>”); }
```

เริ่มแรกเรากำหนดค่าตัวแปร count ให้มีค่าเริ่มต้นเป็น 1 จากนั้นจึงนำตัวแปร count มาทดสอบว่ามีค่าน้อยกว่าหรือเท่ากับ 10 หรือไม่ หากนิพจน์นี้มีค่าเป็นจริง จึงทำการเพิ่มค่าตัวแปร count ขึ้นทีละ 1 จากนั้นจึงทำงานตามคำสั่งภายในบล็อกทั้งหมด แล้ววนกลับมาทดสอบนิพจน์เงื่อนไขอีกครั้ง ทำไปเรื่อย ๆ จนกว่าตัวแปร count จะถูกเพิ่มค่าไปมากกว่า 10 เป็นผลทำให้นิพจน์มีค่าเป็นเท็จแล้วจึงจบการทำงาน ซึ่งผลลัพธ์ที่ได้บนเบราว์เซอร์จะเป็นดังนี้

แฉะตัวที่ : 1

แฉะตัวที่ : 2

.....

แฉะตัวที่ : 10

2.13.5 คำสั่ง while

คำสั่ง while เป็นคำสั่งที่ใช้สำหรับสร้างวงรอบการทำงานเช่นเดียวกับ for แต่คำสั่ง while จะทำการทดสอบเฉพาะนิพจน์เงื่อนไขที่กำหนดให้เป็นจริงเท่านั้น ไม่มีการดำเนินการใด ๆ กับตัวแปรหรือส่วนประกอบในนิพจน์ที่นำมาทดสอบเหมือนที่มีในคำสั่ง for โดยให้เป็นหน้าที่ของโปรแกรมส่วนอื่นที่ต้องดำเนินการเอง ทำให้เกิดความยืดหยุ่นในการใช้งานมากขึ้น แต่ต้องอาศัยการเขียนโปรแกรมเพิ่มมากขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเดียวกันกับคำสั่ง for สามารถเขียนโปรแกรมโดยใช้คำสั่ง while ได้ดังนี้

```
count = 1
while (count <= 10) {
    document.write("แคะตัวที่ : "+count+"<BR>")
    count++;
}
```

คำสั่ง for นั้นมีการทำงานวนรอบเป็นจำนวนรอบที่แน่นอนตายตัว แต่สำหรับ while การวนรอบต้องขึ้นอยู่กับผลการทดสอบเงื่อนไขที่เป็นจริงทุกครั้ง ดังนั้นการวนลูปจึงไม่แน่นอน

2.13.6 คำสั่ง break

จาวาสคริปต์เตรียมคำสั่ง break ไว้สำหรับขัดจังหวะการทำงานซ้ำโดยคำสั่ง for และ while เพื่อให้จบการทำงานในรอบนั้น ๆ ได้ตามความต้องการทันที โดยไม่ต้องรอให้โปรแกรมทำงานจนครบรอบก่อน

```
for (count = 1; count <= 10; count++) {
    if (count == 6) break;
    document.write ("แคะตัวที่ : "+ count + "<BR>"); }
document.write ("<HR>ขณะนี้แคะออกจากลูปแล้ว เมื่อมีการกระโดดถึงตัวที่ : "+count+"<BR>");
```

ตัวอย่างนี้มีการทำงานในวงรอบทั้งสิ้น 10 รอบ ตามคำสั่ง for แต่ได้มีการกำหนดเงื่อนไขไว้ว่าถ้า count มีค่าเท่ากับ 6 เมื่อใดให้หยุดการทำงานโดยคำสั่ง break แล้วข้ามไปทำงานในชุดคำสั่งที่อยู่ถัดจากบล็อก

2.13.7 คำสั่ง continue

คำสั่ง continue ทำหน้าที่ขัดจังหวะการทำงานภายในวงรอบการทำงานซ้ำคล้ายคำสั่ง break แต่คำสั่งนี้จะสั่งให้โปรแกรมกลับไปเริ่มต้นทำงานใหม่จากจุดเริ่มต้นวงรอบทันทีโดยไม่ต้องรอให้โปรแกรมทำงานไปจนครบรอบปกติก่อน ปกติแล้วคำสั่ง continue สามารถนำมาใช้ภายในวงรอบของคำสั่ง while และ for ได้ แต่มีความแตกต่างกันดังนี้

ในคำสั่ง while เมื่อพบคำสั่ง continue โปรแกรมจะกลับไปทดสอบนิพจน์เงื่อนไขที่ ส่วนหัวของวงรอบใหม่ทันที

ส่วนในคำสั่ง for เมื่อพบคำสั่ง continue โปรแกรมจะกลับไปเพิ่มค่าให้กับตัวแปร ของนิพจน์เงื่อนไขในส่วนหัวของวงรอบใหม่ทันที

2.13.8 คำสั่ง with

การนำคำสั่ง with มาช่วยเรียกใช้งานเมธอดและพรอเพอร์ตี้เพื่อระบุว่าเป็นของ ออบเจกต์ใดออบเจกต์หนึ่งทีแน่นอนนั้น ช่วยให้เรลดเวลาและขนาดความยาวของการเขียน โปรแกรมลงได้มาก เพราะไม่ต้องมาคอยระบุชื่อออบเจกต์ซ้ำ ๆ อยู่ทุกเมธอดหรือทุก ๆ พรอเพอร์ตี้ ตัวอย่างการใช้งานมีดังนี้

```
function test() {
  with (document.forms[0].textbox1) {
    alert(name);
    alert(value);
  }
}
```

ฟังก์ชันนี้เป็นการแสดงข้อมูลของออบเจกต์ document.forms[0].textbox ผ่านทาง โดอะล็อกบ็อกซ์ การใช้ชื่อและค่าของพรอเพอร์ตี้ภายในบล็อกคำสั่ง with ทำให้โปรแกรมเข้าใจ ว่ามีการอ้างถึงออบเจกต์ตามที่กำหนดในบรรทัดคำสั่ง with

2.14 การใช้ฟอร์มร่วมกับจาวาสคริปต์

การใช้งานจาวาสคริปต์ที่เห็นได้ชัดเจนอย่างหนึ่งคือการเพิ่มความสามารถของแท็ก <FORM> ของ HTML และสิ่งที่เกี่ยวข้องกับฟิลด์ ปุ่ม และเมนู ซึ่งจะช่วยให้เว็บเพจมีการโต้ตอบกับ ผู้ใช้ อีกทั้งยังช่วยให้ผู้ใช้กรอกข้อมูลบนฟอร์มได้อย่างถูกต้อง

2.14.1 การสร้างแบบฟอร์ม

การสร้างแบบฟอร์มจาก HTML โดยตรงนั้นมีข้อแตกต่างกับการสร้างจากจาวาสคริปต์ โดยมีหลักสำคัญของการสร้างแบบฟอร์มเพื่อใช้ทำงานร่วมกับจาวาสคริปต์นั้นจะขึ้นอยู่กับอีเวนต์แฮนด์เลอร์ หรือเครื่องมือที่นำมาใช้ในอีเวนต์ต่าง ๆ ของฟอร์มมากกว่า เช่น onClick หรือ onSubmit ที่อ้างอิงด้วยพรอเพอร์ตี้ Action เมื่อผู้ใช้กระทำการบางอย่างในแบบฟอร์ม เช่น เมื่อผู้ใช้คลิกปุ่มกด เครื่องมือที่กำหนดไว้ตามอีเวนต์แฮนด์เลอร์ onClick จะเริ่มทำงานทันที ดังตัวอย่าง

```
<FORM NAME = "myform" ACTION = "" METHOD = "GET">
  Enter something in the box : <BR>
  <INPUT TYPE = "button" NAME = "button" VALUE = "Click"
  onClick = "testResults(this.form)">
</FORM>
```

NAME คือ การกำหนดชื่อให้กับฟอร์มออบเจกต์ ซึ่งเป็นค่าที่เรียกใช้ผ่านพรอเพอร์ตี้ NAME

ACTION คือ การกำหนดค่า URL ที่จะใช้ส่งข้อมูลในแบบฟอร์มนั้นออกไป โดยอาจกำหนดเป็น URL ของโปรแกรม CGI หรือ LiveWire ที่อยู่บนเครื่องเซิร์ฟเวอร์

METHOD คือ การกำหนดวิธีรับส่งข้อมูลจากฟอร์มไปยังเซิร์ฟเวอร์ ตาม URL ที่กำหนด

- GET เว็บเบราว์เซอร์จะนำข้อมูลในแบบฟอร์มส่งต่อท้ายไปกับ URL โดยโปรแกรม CGI จะสามารถตรวจสอบข้อมูลนี้ได้จากตัวแปร QUERY_STRING บนเครื่องเซิร์ฟเวอร์

- POST ส่งข้อมูลในฟอร์มผ่านเซิร์ฟเวอร์ทาง stdin โดยโปรแกรม CGI สามารถตรวจสอบขนาดข้อมูลโดยใช้ตัวแปร CONTENT_LENGTH บนเครื่องเซิร์ฟเวอร์ ในระหว่างการประมวลผลสามารถติดต่อกับฐานข้อมูลเพื่อแก้ไขข้อมูลได้ และได้ผลลัพธ์ออกมาทาง stdout และส่งผลคำตอบกลับคืนไปยังเซิร์ฟเวอร์เพื่อส่งกลับไปยังผู้ที่ร้องขอในรูปแบบต่าง ๆ เช่น ในรูปของเว็บเพจ หรือ e-mail

INPUT TYPE คือ การกำหนดชนิดของฟิลด์รับข้อมูล มีหลายชนิดเช่น text, button, select ฯลฯ

VALUE คือ การกำหนดค่าเริ่มต้นให้กับฟิลด์รับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SIZE คือ การกำหนดขนาดของฟิลด์รับข้อมูล (มีใช้บางชนิด)

onClick = “testResults(this.form)” คือ การกำหนดอีเวนต์แฮนด์เลอร์ เมื่อคลิกปุ่มกด จะทำการเรียกฟังก์ชัน testResults ในทุก ๆ ที่บนเว็บเพจ และส่งผ่าน ไปยังฟอร์มออบเจกต์

2.14.2 ฟอร์มออบเจกต์

การอ้างถึงเอลิเมนต์ในฟอร์ม โดยจาวาสคริปต์ สามารถทำได้ด้วยการใช้ชื่อเอลิเมนต์ หรือด้วยอาร์เรย์ของเอลิเมนต์ภายในฟอร์ม

นอกเหนือจากแอตทริบิวต์ที่กำหนดด้วย HTML มาตรฐานแล้ว เมื่อนำมาใช้ร่วมกับจาวาสคริปต์จะมีการเพิ่มอีเวนต์แฮนด์เลอร์ของฟอร์มเข้ามาอีก 2 ตัว คือ onReset (ฟังก์ชันที่กำหนดไว้ จะถูกเรียกขึ้นมาเมื่อปุ่ม reset ถูกคลิก) และ onSubmit (ฟังก์ชันที่กำหนดไว้จะถูกเรียกขึ้นมาเมื่อปุ่ม submit ถูกคลิก)

การอ้างถึงฟอร์มในจาวาสคริปต์ยังสามารถใช้ฟอร์มอาร์เรย์ ที่สามารถอ้างถึงฟอร์มออบเจกต์ทุก ๆ ตัว เช่นถ้าบนเว็บเพจมีฟอร์มอยู่ 3 ฟอร์ม จะอ้างถึงอาร์เรย์ได้เป็น document.form[0], document.form[1], document.form[2]

เราสามารถเขียนพรอเพอร์ตี้และเมธอดของฟอร์มออบเจกต์ ได้ตามรูปแบบต่อไปนี้

อ้างถึงโดยใช้ชื่อฟอร์ม

formname.methodName(parameters)

formname.propertyName

อ้างถึงโดยใช้หมายเลขลำดับในฟอร์มอาร์เรย์

form[index].methodName(parameters)

form[index].propertyName

พรอเพอร์ตี้ของฟอร์มออบเจกต์ ได้แก่ action, element[index], encoding, length และ name เมธอดของฟอร์มออบเจกต์ ได้แก่ submit , reset

2.14.3 ออบเจกต์ต่าง ๆ ในฟอร์ม

ฟิลด์ต่าง ๆ ที่ทำหน้าที่รับข้อมูลในฟอร์ม จาวาสคริปต์จะถือว่าเป็นออบเจกต์ชนิดหนึ่ง ซึ่งมีรูปแบบการเขียนเช่นเดียวกับ HTML เพียงแต่จะมีการกำหนดอีเวนต์แฮนด์เลอร์ต่าง ๆ เพิ่มเข้ามาทำงานด้วย ซึ่งแต่ละออบเจกต์ต่างก็มีพรอปอร์ตี้, เมธอด และอีเวนต์แฮนด์เลอร์แตกต่างกัน

2.14.3.1 เท็กซ์ออบเจกต์

เท็กซ์ออบเจกต์ (text object) ที่เป็นช่องสำหรับกรอกข้อมูล มีการกำหนดอีเวนต์แฮนด์เลอร์หลายชนิดเพิ่มเข้ามาจากรูปแบบการเขียน HTML ปกติ ได้แก่ onBlur (ทำงานเมื่อข้อความในฟิลด์สูญเสียบโฟกัส), onChange (แปลงข้อมูลในฟิลด์ให้เป็นไปตามฟังก์ชันที่กำหนดไว้), onFocus (ใช้เลือกข้อความทั้งหมดในฟิลด์เมื่อผู้ใช้เลื่อนแท็บหรือคลิกเมาส์เข้าไป) และ onSelect (ใช้เลือกข้อความบางส่วนในฟิลด์เมื่อผู้ใช้เลื่อนแท็บหรือคลิกเมาส์เข้าไป)

เท็กซ์ออบเจกต์ที่ใช้ในจาวาสคริปต์มีอยู่ 3 แบบ คือ

- ฮิดเดนออบเจกต์ (INPUT TYPE = "hidden") เป็นออบเจกต์ที่ถูกซ่อนเอาไว้ไม่ให้มองเห็นได้จากหน้าเว็บ โดยจะทำหน้าที่ส่งผ่านข้อมูลในลักษณะ NAME/VALUE ไปยังเซิร์ฟเวอร์
- พาสเวิร์ดออบเจกต์ (INPUT TYPE = "password") ทำหน้าที่ติดต่อกับเครื่องเซิร์ฟเวอร์โดยผู้ที่จะเข้าสู่ระบบต้องแสดงรหัสผ่านก่อน อักษรที่ผู้ใช้ป้อนเข้าไปจะแสดงเป็นเครื่องหมาย *
- เท็กซ์แอเรียออบเจกต์ (<TEXTAREA>) ทำหน้าที่รับการป้อนข้อมูลจากผู้ใช้ได้ครั้งละหลาย ๆ บรรทัด

2.14.3.2 ซับมิตออบเจกต์

ซับมิตออบเจกต์ (INPUT TYPE = "submit") เมื่อผู้ใช้คลิกที่ปุ่มกดจะเป็นการส่งข้อมูลในฟอร์มไปยังเครื่องเซิร์ฟเวอร์ นั่นคือทำให้เกิดอีเวนต์แฮนด์เลอร์ onSubmit ขึ้น แม้ว่าซับมิตออบเจกต์จะมีลักษณะเป็นปุ่มกดเช่นเดียวกับบัตตันออบเจกต์ เราก็ไม่สามารถนำอีเวนต์แฮนด์เลอร์ onClick มาใช้แทน onSubmit สำหรับฟอร์ม หรือจากซับมิตเมธอดได้

2.14.3.3 รีเซ็ตออบเจกต์

รีเซ็ตออบเจกต์ (INPUT TYPE = “reset”) เมื่อผู้ใช้คลิกปุ่มกด จะเป็นผลให้ทุก ๆ เอลิเมนต์ในฟอร์มนั้นมีค่ากลับไปอยู่ในสถานะเริ่มต้น เช่นเดียวกับที่ใช้ใน HTML เพียงแต่เพิ่มอีเวนต์แอสต์เลอร์ onClick เข้ามา

2.14.3.4 เช็กรูปออบเจกต์

เช็กรูปออบเจกต์ เป็นเอลิเมนต์ที่ทำงานได้โดยอิสระ ไม่มีผลต่อเอลิเมนต์ที่เป็นชุดเดียวกันเหมือนเรดิโอออบเจกต์ เราสามารถทดสอบการทำเครื่องหมายในช่องทำเครื่องหมายของเช็กรูปออบเจกต์โดยผู้ใช้ได้จากพรอเพอร์ตี้ checked เมื่อฟอร์มถูกโหลดครั้งแรกจะมีเครื่องหมายเช็กรูปปรากฏ

เช็กรูปออบเจกต์มีเมธอดที่ดำเนินการได้เพียงเมธอดเดียวเท่านั้นคือ click() และมีอีเวนต์แอสต์เลอร์คือ onClick

2.14.3.5 เรดิโอออบเจกต์

เรดิโอออบเจกต์เป็นรายการแบบเลือกตอบ โดยจะแสดงหัวข้อต่าง ๆ ให้เลือกหลายรายการ โดยให้ผู้ใช้เลือกตอบได้เพียงรายการเดียวเท่านั้น และเมื่อเลือกตอบรายการใดรายการหนึ่งไปแล้วจะไม่สามารถยกเลิกการเลือกได้ แต่ยังสามารถเปลี่ยนไปเลือกรายการอื่น ๆ ได้เช่นเดิม

เรดิโอออบเจกต์ทำหน้าที่สนองตอบต่อเซตของเรดิโอบัตตันของฟอร์มในจาวาสคริปต์ ซึ่งการเข้าถึงเรดิโอออบเจกต์นี้สามารถเลือกได้เพียงรายการเดียวเท่านั้น

เรดิโอออบเจกต์มีอีเวนต์แอสต์เลอร์ที่ดำเนินการได้เพียงตัวเดียวเท่านั้นคือ onClick และมีเมธอดเดียว คือ click()

2.14.3.6 ซีล็กต์ออบเจกต์

ซีล็กต์ออบเจกต์ เป็นรายการแบบเลือกตอบอีกชนิดหนึ่งที่แสดงตัวเลือกมีลักษณะ

เป็นรายการแบบดรอปดาวน์ (drop-down list box) ซึ่งจะยอมให้ผู้ใช้สามารถทำการเลือกตอบได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพียงตัวเลือกเดียว หรือแบบแถบเลื่อน (list box) ซึ่งจะยอมให้ผู้ใช้เลือกครั้งละหนึ่งหรือหลายตัวได้จากรายการ แล้วแต่จะกำหนด

ซีเล็กต์ออบเจกต์เป็นรายการเลือกซึ่งยอมให้ผู้ใช้เลือกหนึ่งข้อความหรือออบเจกต์ที่อยู่ด้านขวา ซึ่งผู้ใช้สามารถเลือกข้อความได้หนึ่งรายการหรือมากกว่านั้น ข้อมูลที่อยู่ภายในฟอร์มจะมีคำสั่ง <OPTION> เป็นตัวแสดงข้อมูลหนึ่งหรือหลายบรรทัดได้ และต้องเขียนอยู่ภายในคำสั่ง <SELECT>...</SELECT> อีเวนต์แฮนด์เลอร์ที่ใช้กับซีเล็กต์ออบเจกต์มีหลายตัวได้แก่ onBlur, onChange และ onFocus

ในการอ้างถึงตัวเลือกในซีเล็กต์ชันลิสต์ จะใช้อ้างเป็นสมาชิกของออบชันอาร์เรย์ (option array) โดยใช้หมายเลขสำหรับการอ้างเรียงตามลำดับ เช่น ซีเล็กต์ชันลิสต์ musicStyle มีตัวเลือก 3 ตัวเลือก การอ้างถึงทั้งสามทำได้คือ musicStyle.options[0], musicStyle.options[1] และ musicStyle.options[2] เป็นต้น

2.15 UML (Unified Modeling Language)

UML คือ รูปแบบหนึ่งของภาษาหรือมาตรฐานที่ถูกกำหนดขึ้นเพื่อใช้ในการอธิบายถึงโครงสร้าง การทำงาน หรือความสัมพันธ์ของระบบที่มีความซับซ้อน เป็นผลงานที่คิดค้นโดย Grady Booch, James Rumbaugh และ Ivar Jacobson ซึ่งเป็นผู้เชี่ยวชาญระดับแนวหน้าของโลกในด้านการเขียนโปรแกรมแบบอ้างอิงวัตถุ โดยการเข้าร่วมงานใน Rational Software Cooperation ในปี ค.ศ. 1994

UML เป็นภาษาสำหรับการออกแบบซอฟต์แวร์ในรูปแบบภาพวาด ประกอบกันเป็นไดอะแกรม (Diagram) โดยมีกฎในการประกอบกันของส่วนต่าง (Element) ไดอะแกรมจะแสดงถึงมุมมองต่าง ๆ (Multiple view) ของระบบ ซึ่งจะรวมเรียกว่าโมเดล (Model) โมเดล UML จะบ่งบอกถึงรายละเอียดของระบบ แต่จะไม่ระบุถึงวิธีการในการพัฒนาระบบ (Implementation) ซึ่งการออกแบบนั้นจะประกอบไปด้วยการออกแบบสิ่งต่าง ๆ ดังต่อไปนี้

2.15.1 Class Diagram

Class Diagram เป็นการแสดงถึงแนวความคิดของขอบเขตของปัญหาใน UML หรือแสดงถึงกลุ่มของโครงสร้างของไดอะแกรม ซึ่งจะไม่มีการอธิบายถึงการกระทำต่าง ๆ แต่จะมีการแสดงความสัมพันธ์กัน ออบเจกต์ต่าง ๆ ในระบบจะประกอบด้วยคุณสมบัติ และวิธีการ หรือคุณเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะ ออบเจกต์ที่มีคุณสมบัติและวิธีการเดียวกันจะรวมเรียกว่าคลาส หรือกล่าวในอีกนัยหนึ่งคือ คลาสจะเป็นต้นแบบหรือพิมพ์เขียวของออบเจกต์ และในทางกลับกันออบเจกต์ก็ คือ อินสแตนซ์ (Instance) ของคลาส

ขั้นตอนการเขียนคลาสไคอะแกรมเบื้องต้นมีดังต่อไปนี้

1. กำหนดคลาสต่าง ๆ ที่ควรมีในระบบ
2. กำหนดคุณสมบัติและวิธีการการทำงานของแต่ละคลาสที่คลาสนั้น ๆ สามารถทำได้
3. ทำการวาดคลาสไคอะแกรมตามที่กำหนด
4. เชื่อมโยงความสัมพันธ์ (Associate) ระหว่างคลาสต่าง ๆ

2.15.2 Use Case Diagram

Use Case Diagram จะอธิบายถึงกิจกรรมของระบบจากมุมมองของผู้ใช้ ในแง่ของนักพัฒนาระบบ เป็นการอธิบายขั้นตอนการทำงานของระบบโดยใช้ภาพประกอบถึงการทำงานของระบบ (Use Case), ผู้กระทำ และความสัมพันธ์ระหว่าง Use Case กับผู้กระทำ ซึ่ง Use Case จะแสดงด้วยวงรี ส่วนผู้กระทำ (Actor) จะแสดงโดยใช้รูปคน และเส้นตรงจะเป็นส่วนติดต่อระหว่างผู้กระทำกับ Use Case Diagram นี้มีความสำคัญมากในจุดเริ่มต้นในการกำหนดความต้องการของระบบ โดยพิจารณาจากมุมมองของผู้ใช้ ซึ่งจะทำให้ได้ระบบที่ตรงกับความต้องการและสามารถนำไปใช้งานได้จริง

ขั้นตอนในการเขียน Use Case มีดังต่อไปนี้

1. กำหนดหน้าที่การทำงานของระบบ , กำหนดขอบเขต, กำหนดผู้กระทำ และ Use Case และลำดับขั้นตอนการทำงานของ Use Case
2. เขียนผู้กระทำ (Actor)
3. เขียน Use Case
4. เชื่อมโยงความสัมพันธ์ระหว่างผู้กระทำ กับ Use Case
5. หาก Use Case ใดที่มีขั้นตอนรายละเอียดใน Use Case นั้นมาก ก็สามารถเขียน Use Case นั้นในระดับที่ต่ำลงไปได้อีก โดยจะต้องแสดงความสัมพันธ์ของ Use Case ใหม่ที่เกิดขึ้นกับ Use Case เก่าด้วย

2.15.3 Activity Diagram

กิจกรรมที่เกิดขึ้นตาม Use Case หรือเกิดจากพฤติกรรมของออบเจกต์เองตามปกติ เป็นลำดับของกิจกรรม สามารถเขียนอยู่ในรูปของ Activity Diagram ซึ่งจะอธิบายการทำงานโดยละเอียดของ Use Case นั้น ๆ ว่าจะต้องทำการส่งค่าอะไร ไปที่ขั้นตอนไหนก่อนหลังเป็นลำดับ ซึ่งโคอะแกรมนี้จะมีส่วนช่วยในการออกแบบหน้าจอที่ไว้ติดต่อกับผู้ใช้ และรูปแบบฐานข้อมูลแบบคร่าว ๆ หรืออาจช่วยให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้

ขั้นตอนในการเขียน Activity Diagram มีดังนี้

1. นำ Use Case มาวิเคราะห์หาขั้นตอนในการใช้ ออกแบบวิธีการส่ง และวิธีการเก็บข้อมูล
2. แยกส่วนของระบบเป็นส่วน ๆ เช่น ส่วนติดต่อกับผู้ใช้ ส่วนฐานข้อมูล ส่วนเซิร์ฟเวอร์ หรือโคไลเอ็นท์ ขึ้นอยู่กับความเหมาะสม
3. เขียนขั้นตอนต่าง ๆ ทั้งหมด
4. เขียนความสัมพันธ์แบบลำดับขั้น
5. เขียนกำกับสิ่งที่ต้องส่งค่าให้

บทที่ 3

วิธีการออกแบบ

3.1 การออกแบบการทำงานของระบบ

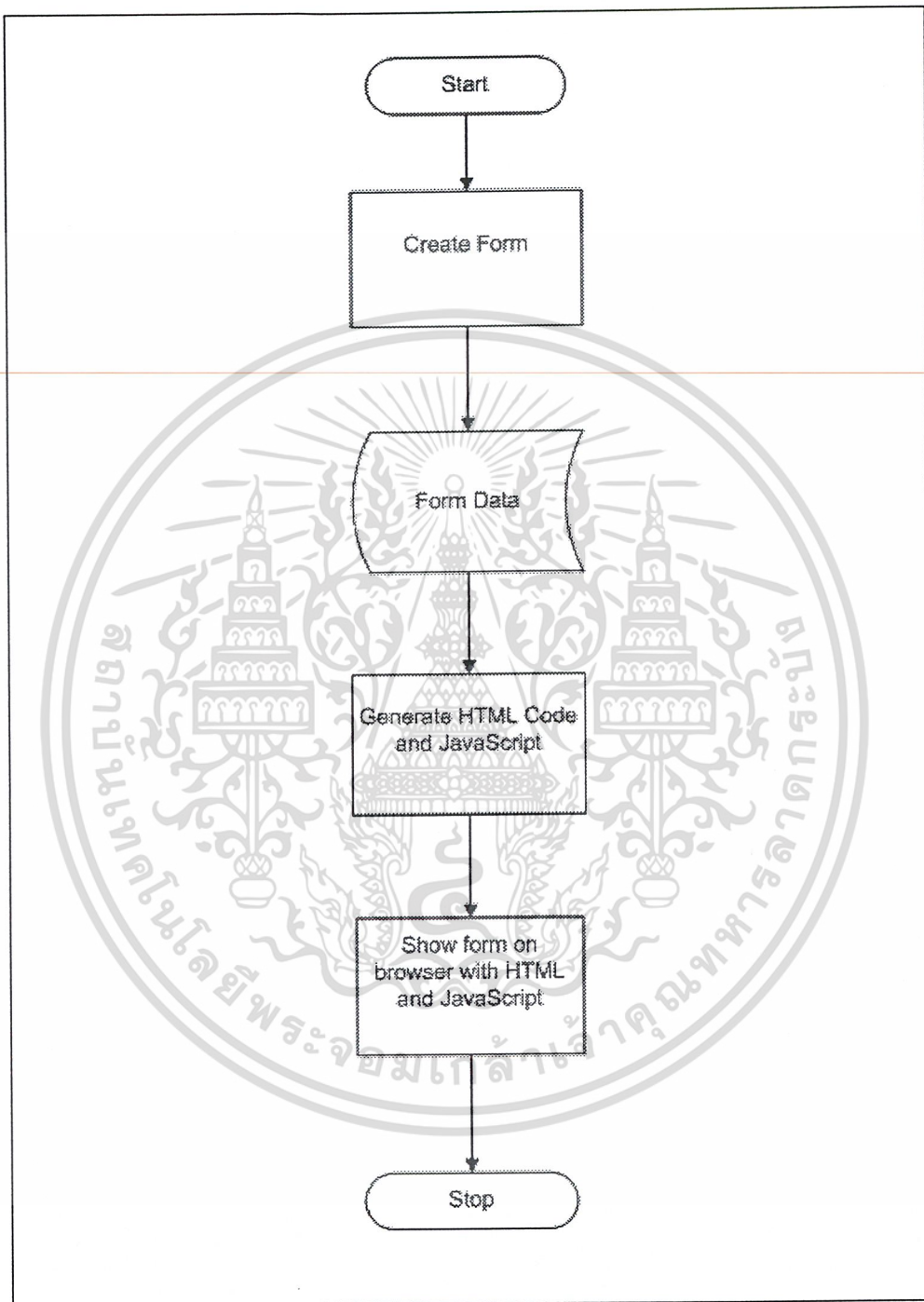
โปรแกรมสร้างเว็บฟอร์มหรือระบบตรวจสอบข้อมูล มีระบบการทำงานหลัก ๆ ตามรูปที่ 3.1 และแต่ละส่วนมีความหมายดังนี้

1. Create Form ให้ผู้ใช้ออกแบบลักษณะของฟอร์ม โดยทำการเลือกและจัดวางอุปกรณ์รับข้อมูล หรือตัว Control ได้ตามความต้องการ พร้อมกับกำหนดค่าพารามิเตอร์ต่าง ๆ ของอุปกรณ์รับข้อมูลตามคำศัพท์ในภาษา html เช่น ความกว้างของ Textbox, ชนิดของข้อมูลที่ต้องการรับเข้ามาเป็นตัวเลข หรือตัวอักษร, ชื่อของตัวอุปกรณ์ เป็นต้น และสามารถให้ผู้ใช้เลือกระบบตรวจสอบความถูกต้องของข้อมูลแบบต่าง ๆ ให้แก่ฟอร์มนั้นได้ เช่น การตรวจสอบรูปแบบของหมายเลขโทรศัพท์ หรือการตรวจสอบรูปแบบของอีเมลแอดเดรส เป็นต้น ซึ่งเทคนิคสำคัญในการสร้างฟอร์ม จะตั้งจัดวางอุปกรณ์รับข้อมูลให้ในรูปแบบของตาราง (row และ column) เพื่อช่วยให้ง่ายแก่การ Generate Source Code โดยการดำเนินงานปลื้กย่อยของส่วน Create Form ดูได้จากรูป 3.2

2. Generate HTML and JavaScript Code หลังจากส่วน Create Form จะมีการนำค่าต่าง ๆ มาป้ไว้ในฐานข้อมูล แล้วโปรแกรมจะนำข้อมูลที่สัมพันธ์กับภาษา html และจาวาสคริปต์มาต่อกันเป็นโค้ด รายละเอียดของส่วนนี้ดูได้จากรูป 3.4

3. Show Form on Browser with Code ส่วนนี้จะนำฟอร์มที่ผู้ใช้สร้างไว้ออกไปแสดงเป็นตัวอย่างให้เห็นบนเบราว์เซอร์ และสามารถเรียกดูโค้ดของฟอร์มนั้นได้ รายละเอียดของส่วนนี้ดูได้จากรูป 3.5

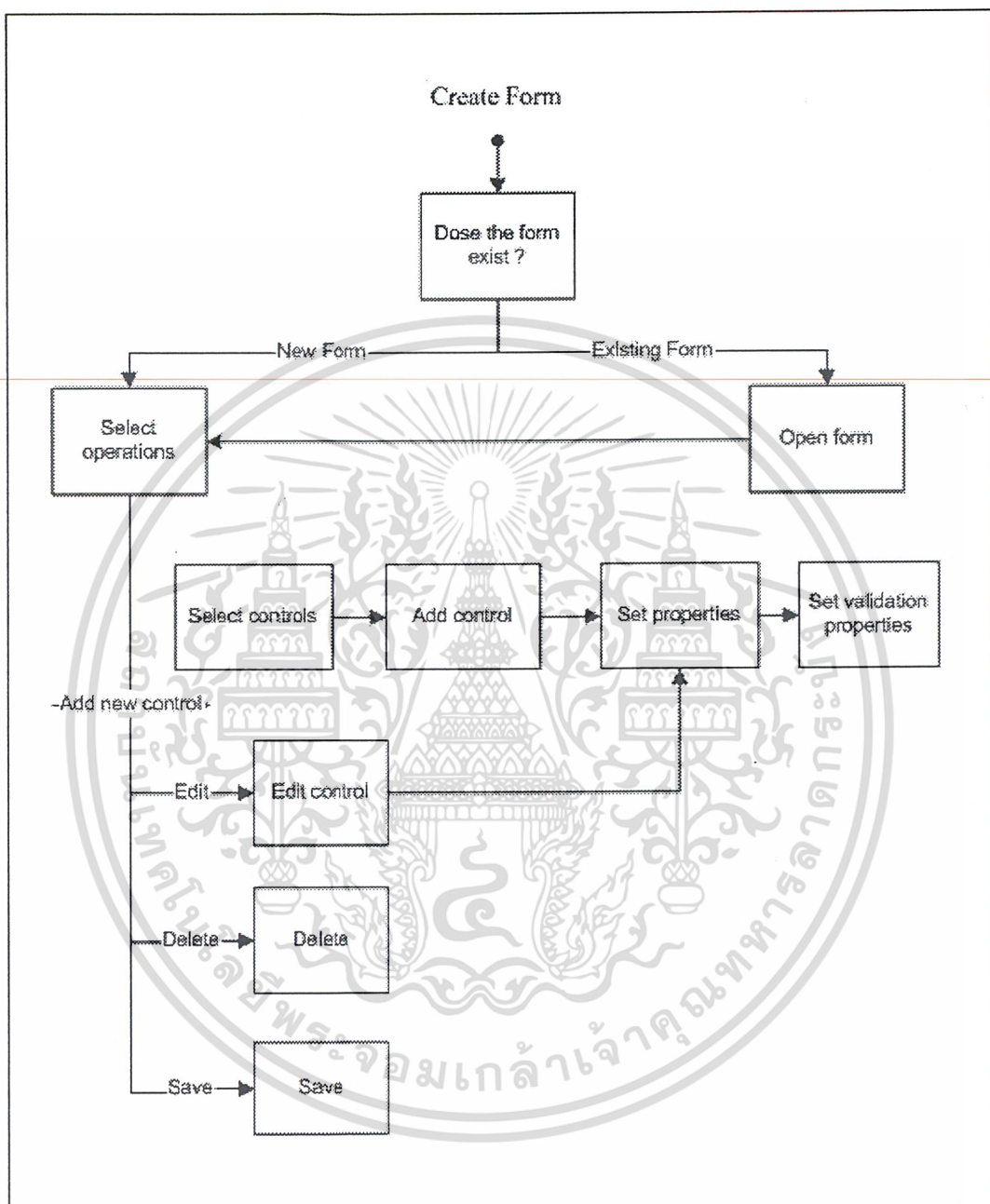
Work Flow of Web Form with Validation Generator



รูปที่ 3.1 Work Flow การทำงานของระบบของโปรแกรมสร้างเว็บฟอร์มพร้อมระบบตรวจสอบความถูกต้องของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

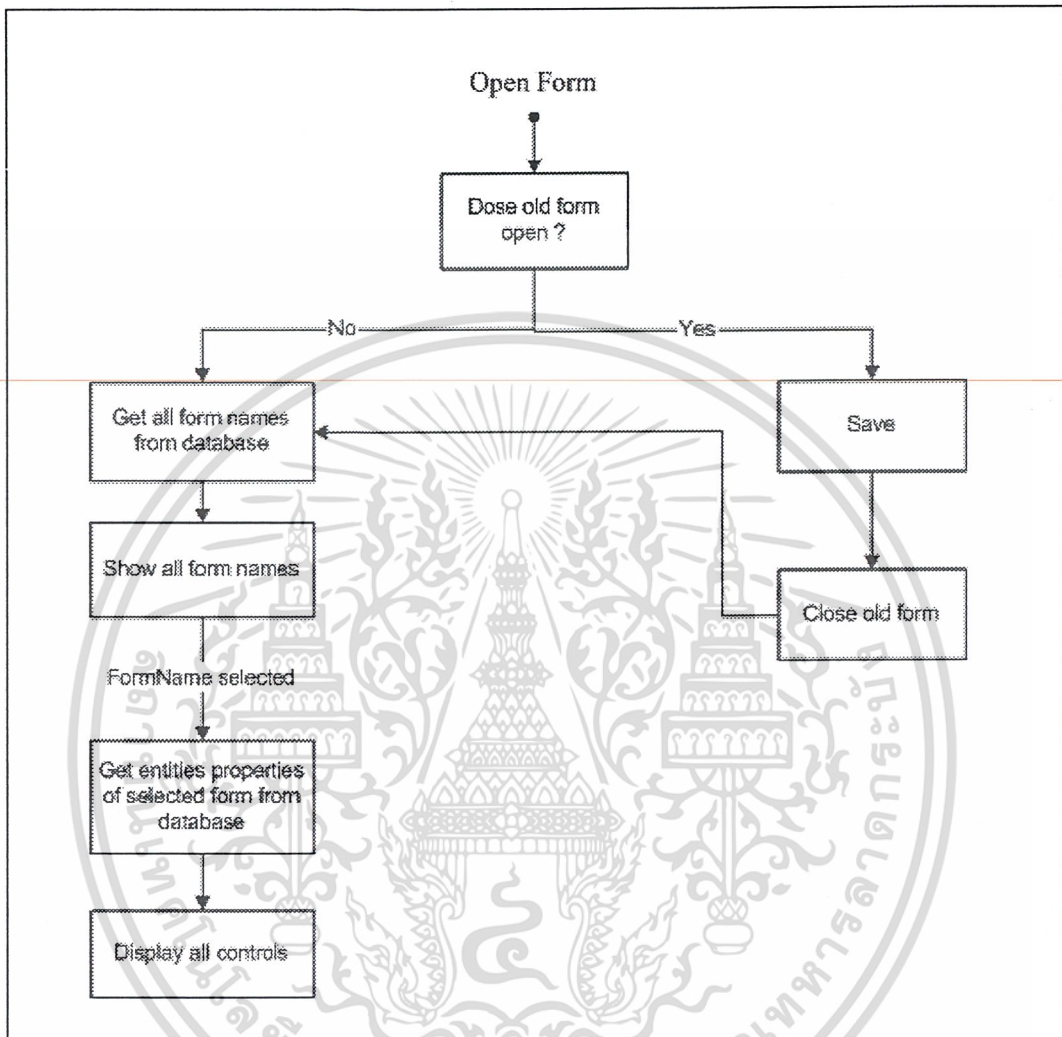
Work Flow of Create Form



รูปที่ 3.2 Work Flow การทำงานของส่วน Create Form

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

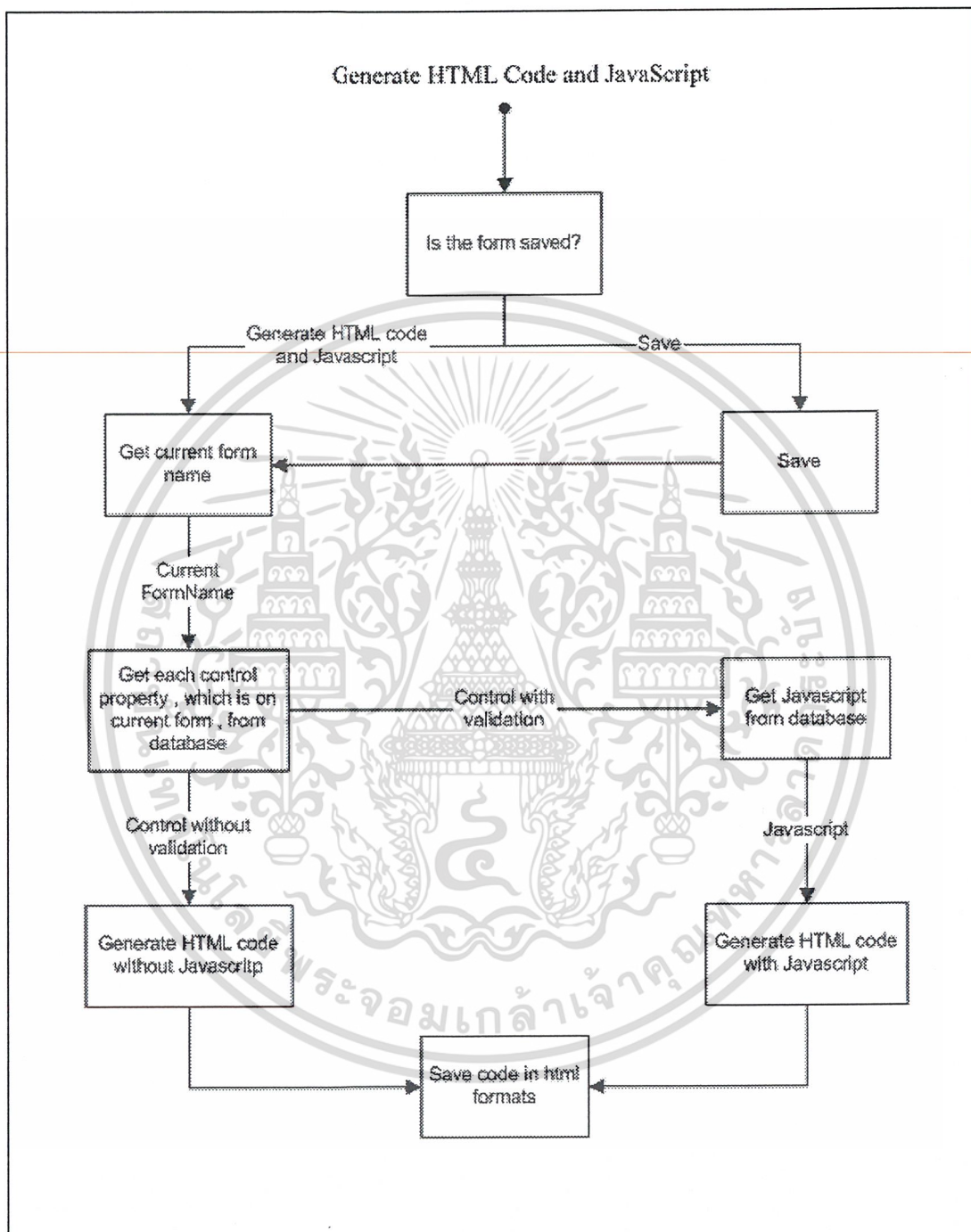
Work Flow of Open Form



รูปที่ 3.3 Work Flow การทำงานของส่วน Open Form

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

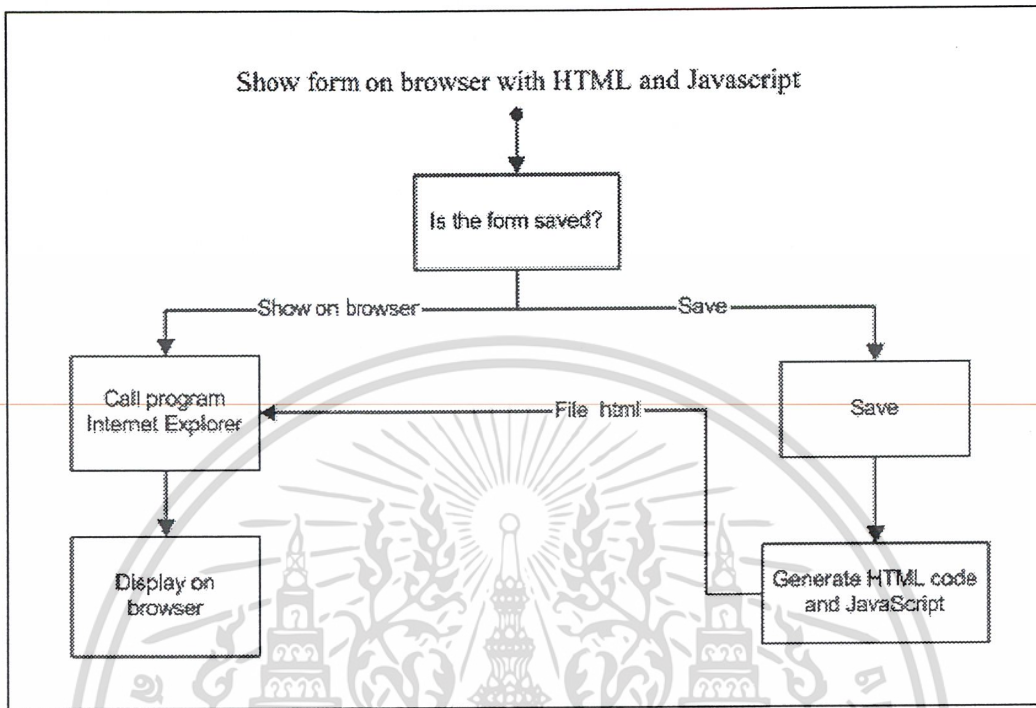
Work Flow of Generate HTML Code and JavaScript



รูปที่ 3.4 Work Flow การทำงานของส่วน Generate HTML and JavaScript Code

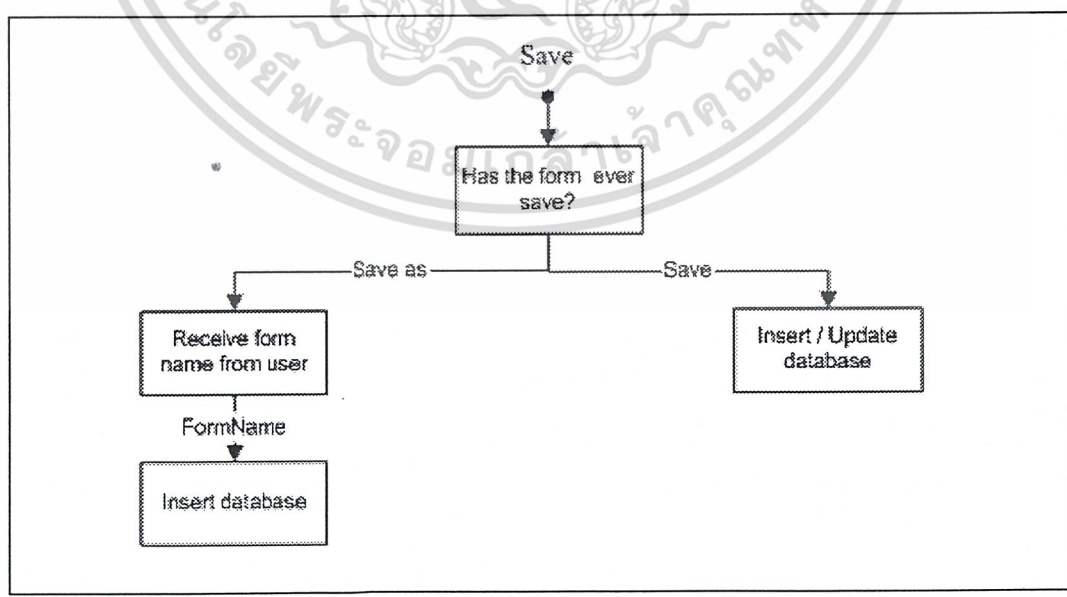
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Work Flow of Show Form on Browser with Code



รูปที่ 3.5 Work Flow การทำงานของส่วน Show Form on Browser with Code

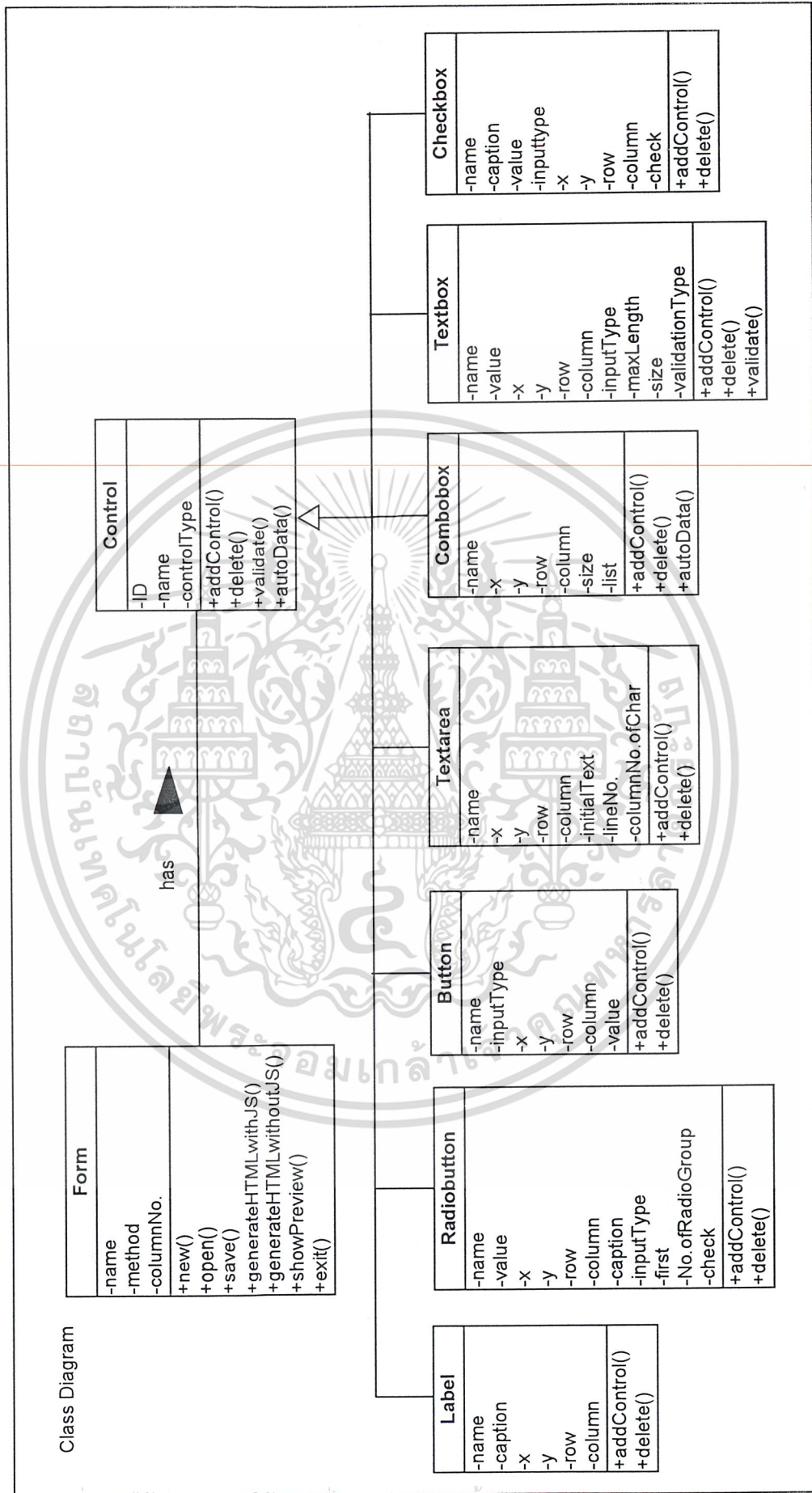
Work Flow of Save



รูปที่ 3.6 Work Flow การทำงานของส่วน Save

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

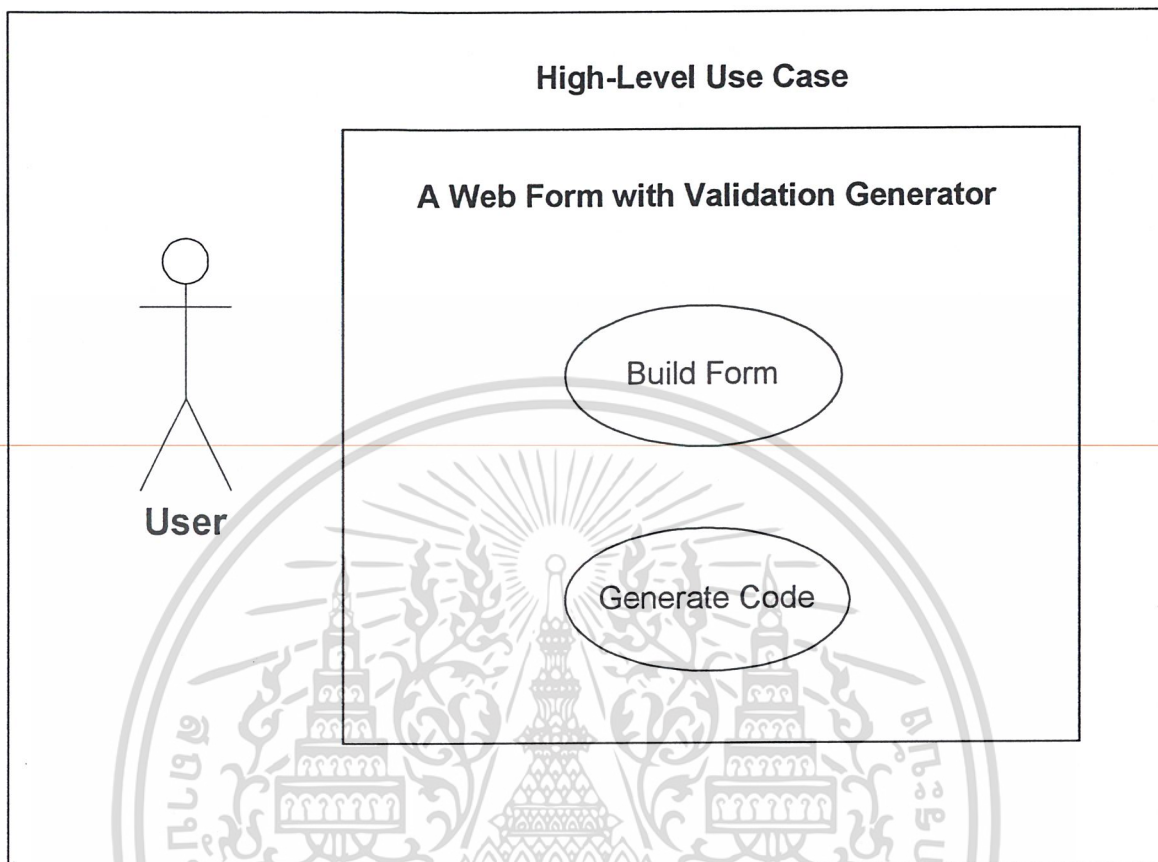
3.2 การออกแบบด้วยวิธีการ UML



รูปที่ 3.7 แสดง Class Diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

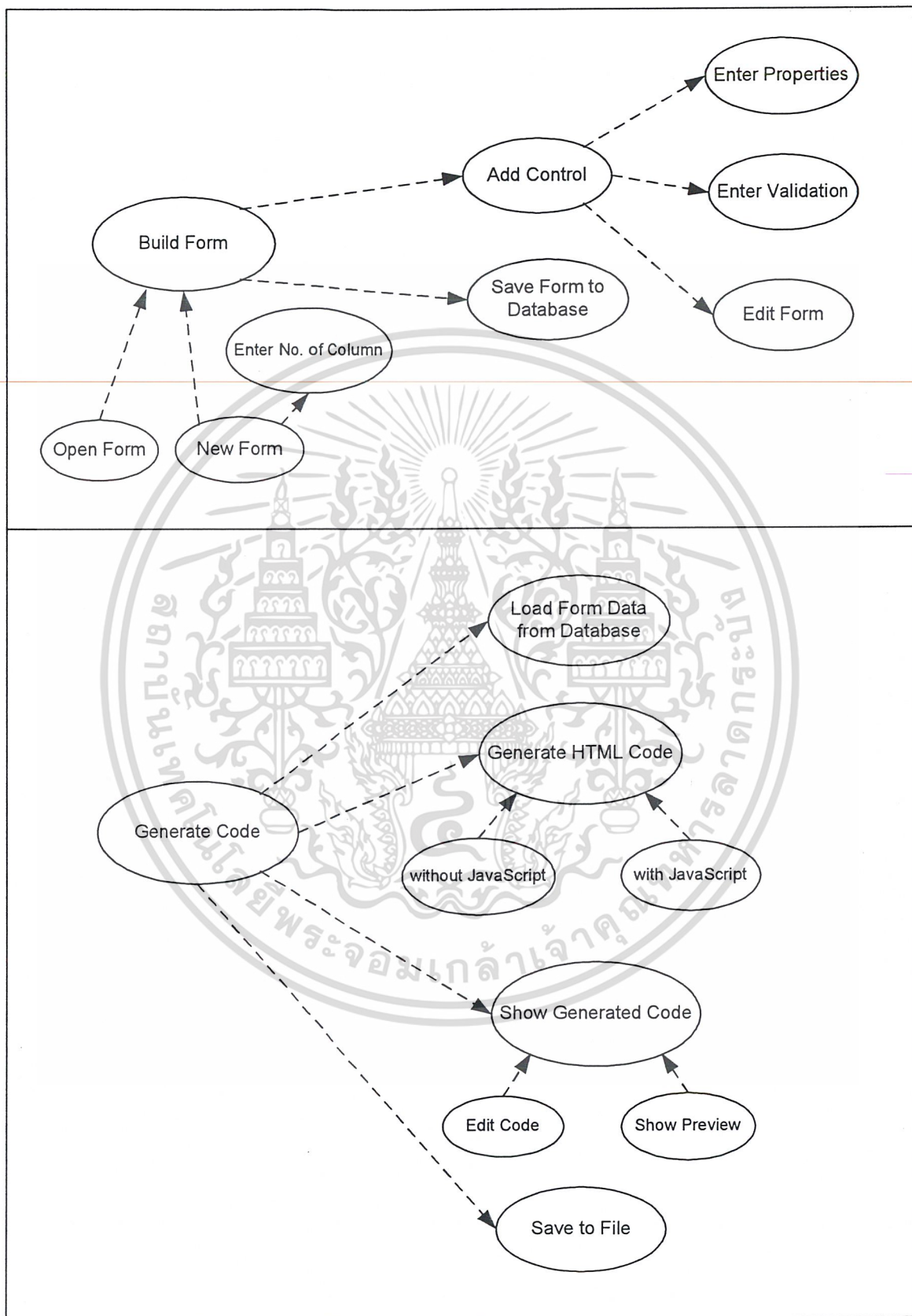
Use Case Diagram



รูปที่ 3.8 แสดง High - Level Use Case ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

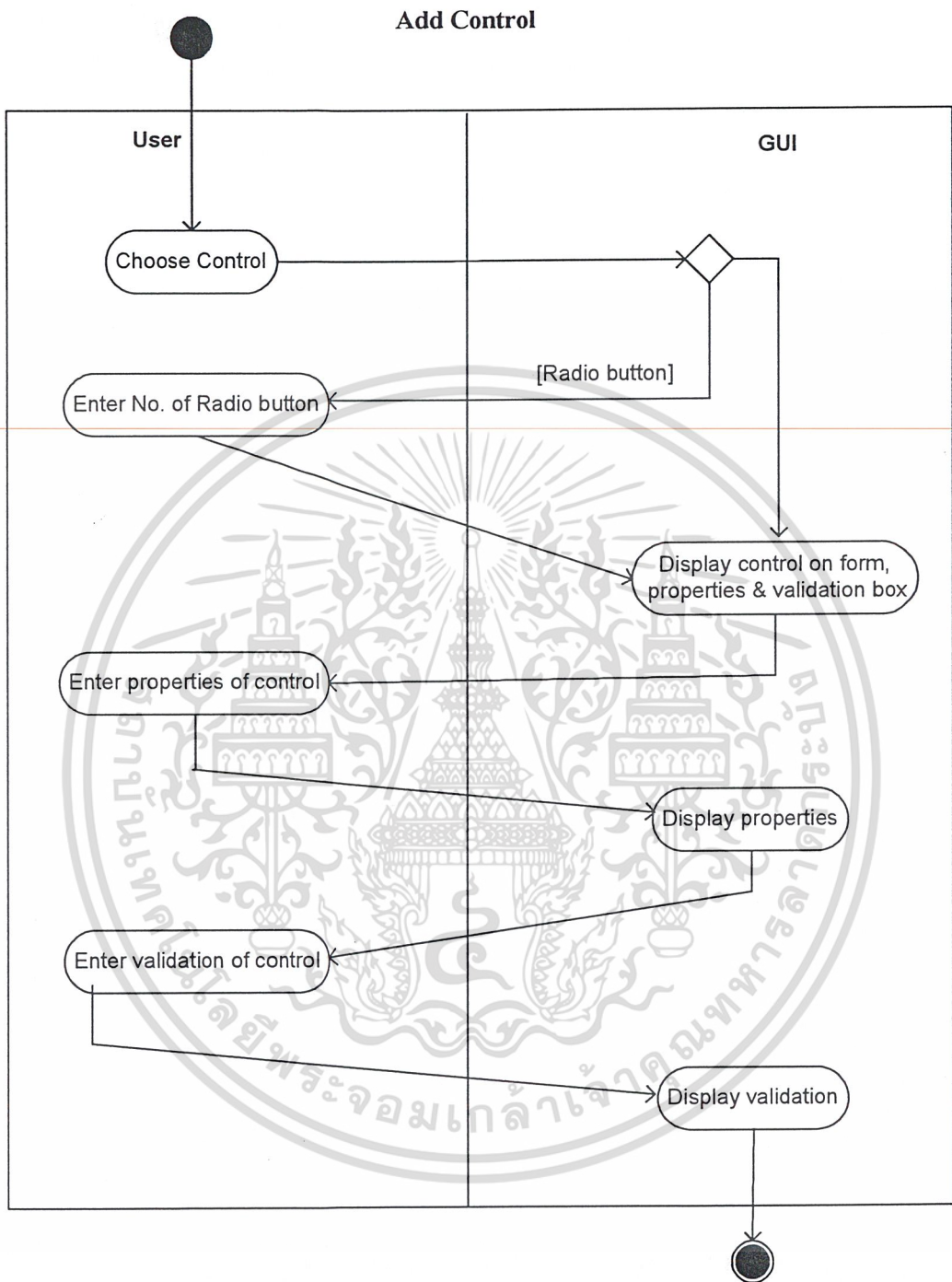
Use Cases



รูปที่ 3.9 แสดง Use Cases Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

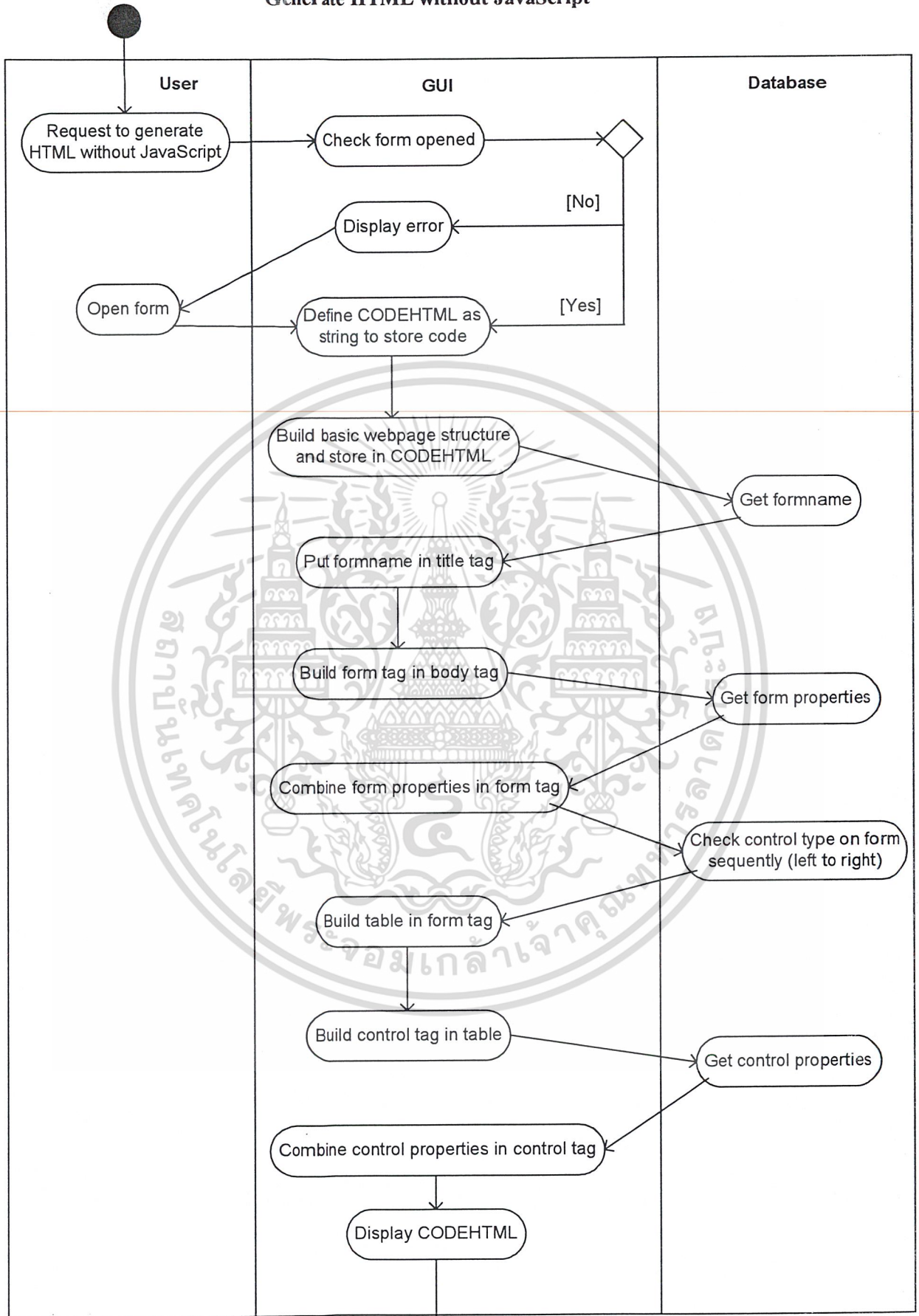
Activity Diagram



รูปที่ 3.10 แสดง Activity Diagram ของส่วน Add Control

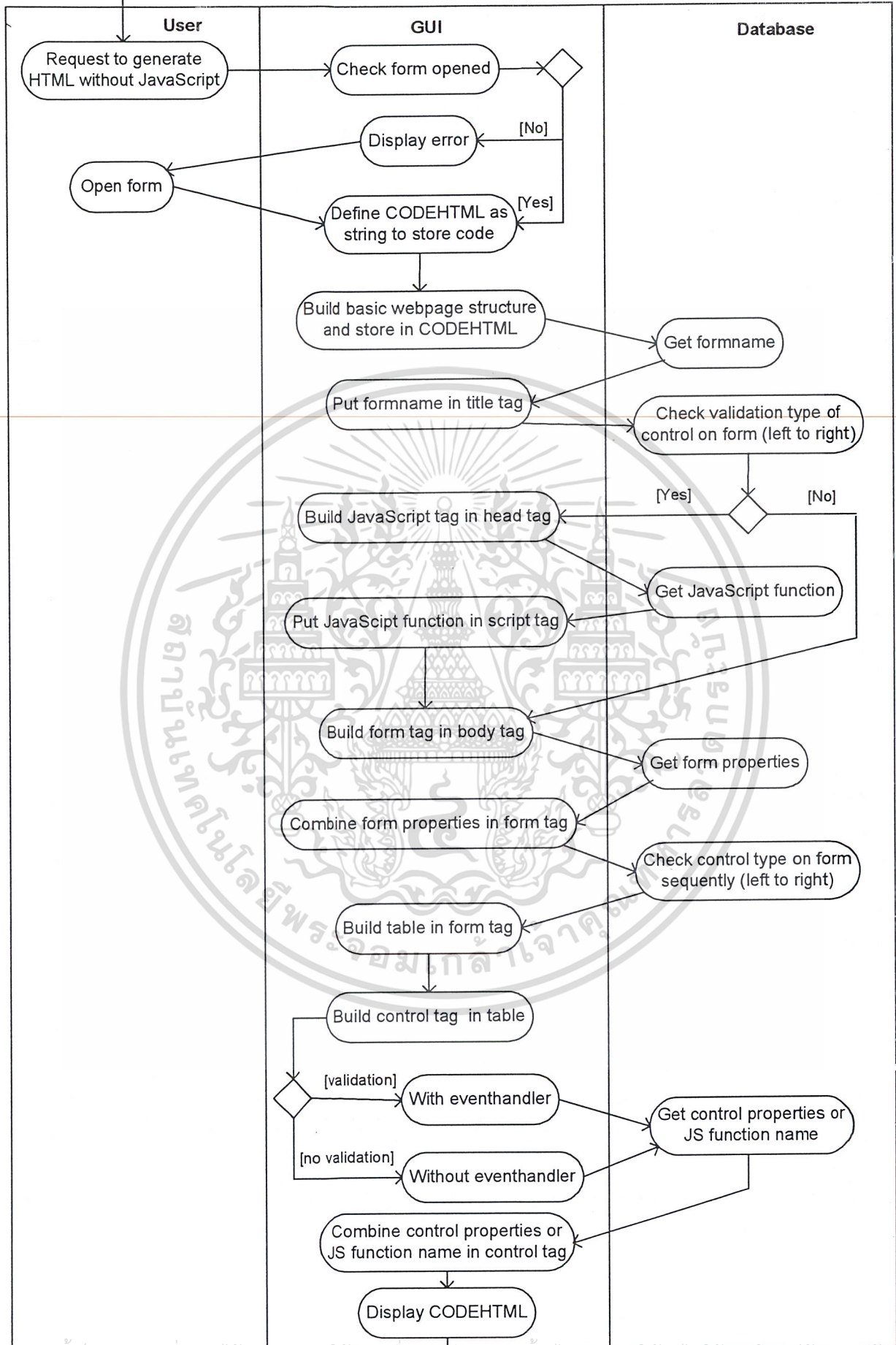
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Generate HTML without JavaScript



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.11 Activity Diagram ของส่วน Generate HTML without JavaScript
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Generate HTML with JavaScript



รูปที่ 3.12 Activity Diagram ของส่วน Generate HTML with JavaScript

Data Dictionary

ตารางที่ 3.1 Entity Table

Attributes	Data Type	Meaning
FormName (FK)	Char(50)	ชื่อที่ใช้อ้างอิงฟอร์มหนึ่ง ๆ
EntityID (PK)	Integer	หมายเลขที่ใช้อ้างอิง Control แต่ละตัวบนฟอร์มหนึ่ง ๆ
Type	Char(50)	ชนิดของ Control แต่ละตัวบนฟอร์ม เช่น Label, Textbox

ตารางที่ 3.2 Form Table

Attributes	Data Type	Meaning
FormName (PK)	Char(50)	ชื่อที่ใช้อ้างอิงฟอร์มหนึ่ง ๆ
Method	Char(50)	วิธีการรับ-ส่งข้อมูลของฟอร์ม ไปยังเซิร์ฟเวอร์ มี 2 แบบคือ Post และ Get
AmountCol	Integer	จำนวนคอลัมน์สำหรับวาง Control บนฟอร์ม

ตารางที่ 3.3 Label Table

Attributes	Data Type	Meaning
EntityID (PK)	Integer	หมายเลขที่ใช้อ้างอิง Label แต่ละตัวบนฟอร์มหนึ่ง ๆ
LabelName	Char(50)	ชื่ออ้างอิงของ Label แต่ละตัวบนฟอร์ม
LabelCaption	Char(50)	ข้อความที่ปรากฏบน Label ของฟอร์ม
LabelTop	Integer	ตำแหน่งของ Label บน Work Area ตามแนวแกน X ของหน้าจอ
LabelLeft	Integer	ตำแหน่งของ Label บน Work Area ตามแนวแกน Y ของหน้าจอ
LabelRow	Integer	ตำแหน่งของ Label ที่แสดงบนเบรเซอร์ ตามแนวแกน Y
LabelColumn	Integer	ตำแหน่งของ Label ที่แสดงบนเบรเซอร์ ตามแนวแกน X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 Textbox Table

Attributes	Data Type	Meaning
EntityID (PK)	Integer	หมายเลขที่ใช้อ้างอิง Textbox แต่ละตัวบนฟอร์มหนึ่งๆ
TextboxName	Char(50)	ชื่ออ้างอิงของ Textbox แต่ละตัวบนฟอร์ม
TextboxSize	Integer	ขนาดความยาวของ Textbox ที่แสดงบนเบราว์เซอร์
TextboxValue	Char(50)	ข้อความเริ่มต้นที่ปรากฏบน Textbox ของฟอร์ม
TextboxInputtype	Char(50)	ค่าที่ใช้กำหนดชนิดของ Textbox ตามคำสั่งในภาษา HTML มี 2 ชนิดคือ Text และ Password
TextboxMaxlength	Integer	จำนวนอักขระสูงสุดที่ Textbox รับได้
TextboxTop	Integer	ตำแหน่งของ Textbox บน Work Area ตามแนวแกน X
TextboxLeft	Integer	ตำแหน่งของ Textbox บน Work Area ตามแนวแกน Y
TextboxRow	Integer	ตำแหน่งของ Textbox ที่แสดงบนเบราว์เซอร์ในแกน Y
TextboxColumn	Integer	ตำแหน่งของ Textbox ที่แสดงบนเบราว์เซอร์ในแกน X
TextboxValidationtype	Char(50)	ชนิดของการตรวจสอบที่ต้องการกำหนดให้ Textbox
TextboxConfirmPair	Char(50)	ชื่อของ Textbox ที่ใช้เป็นช่องสำหรับรับการ Confirm Password

ตารางที่ 3.5 Radio Table

Attributes	Data Type	Meaning
EntityID (PK)	Integer	หมายเลขที่ใช้อ้างอิง Radio Button แต่ละตัวบนฟอร์ม
RadioName	Char(50)	ชื่อที่ใช้อ้างอิง Radio Button แต่ละตัวบนฟอร์ม (Radio Button กลุ่มเดียวกันจะมีชื่อเหมือนกัน แต่ต่างกันตรง EntityID)
RadioCheck	Boolean	เป็นค่าเริ่มต้นที่ใช้กำหนดว่าจะให้ Radio Button ถูก Check ไว้ก่อนหรือไม่ ถ้าเป็น Yes คือถูก Check, No จะไม่ถูก Check
RadioCheckIndex	Integer	หมายเลขตามลำดับของ Radio Button แต่ละตัวในกลุ่มหนึ่ง ๆ ที่ถูกเลือกให้มีการ Check ไว้เป็นค่าเริ่มต้น
RadioCaption	Char(50)	ข้อความประจำ Radio Button แต่ละตัวบนฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Attributes	Data Type	Meaning
RadioValue	Char(50)	ค่าของ Radio Button แต่ละตัว ที่ใช้อ้างอิงในสคริปต์ หรือ ค่าที่ใช้ส่งไปยังเซิร์ฟเวอร์
RadioInputtype	Char(50)	ค่าที่ใช้กำหนดความเป็น Radio Button ในคำสั่งภาษา HTML โดยใช้ "Radio" เป็น Input Type
RadioAmount	Integer	จำนวน Radio Button ทั้งหมดที่อยู่ในกลุ่มเดียวกัน
RadioFirst	Boolean	เป็นการระบุว่า Radio Button ตัวนั้นเป็นตัวแรกของกลุ่ม ซึ่งกำหนดโดย Yes หรือ No
RadioTop	Integer	ตำแหน่งของ Radio บน Work Area ตามแนวแกน X
RadioLeft	Integer	ตำแหน่งของ Radio บน Work Area ตามแนวแกน Y
RadioRow	Integer	ตำแหน่งของ Radio ที่แสดงบนเบรเซอร์ในแกน Y
RadioColumn	Integer	ตำแหน่งของ Radio ที่แสดงบนเบรเซอร์ในแกน X

ตารางที่ 3.6 Checkbox Table

Attributes	Data Type	Meaning
EntityID (PK)	Integer	หมายเลขที่ใช้อ้างอิง Checkbox แต่ละตัวบนฟอร์ม
CheckboxName	Char(50)	ชื่ออ้างอิงของ Checkbox แต่ละตัวบนฟอร์ม
CheckboxCaption	Char(50)	ข้อความประจำ Checkbox แต่ละตัวบนฟอร์ม
CheckboxCheck	Boolean	เป็นค่าเริ่มต้นที่ใช้กำหนดจะให้ Checkbox ถูก Check ไว้ก่อนหรือไม่ ถ้าเป็น Yes คือถูก Check, No จะไม่ถูก Check
CheckboxInputtype	Char(50)	ค่าที่ใช้กำหนดความเป็น Checkbox ในคำสั่งภาษา HTML โดยใช้ "Checkbox" เป็น Inputtype
CheckboxValue	Char(50)	ค่าของ Checkbox แต่ละตัว ที่ใช้อ้างอิงในสคริปต์ หรือ ค่าที่ใช้ส่งไปยังเซิร์ฟเวอร์
CheckboxTop	Integer	ตำแหน่งของ Checkbox บน Work Area ตามแนวแกน X
CheckboxLeft	Integer	ตำแหน่งของ Checkbox บน Work Area ตามแนวแกน Y
CheckboxRow	Integer	ตำแหน่งของ Checkbox ที่แสดงบนเบรเซอร์ในแกน Y
CheckboxColumn	Integer	ตำแหน่งของ Checkbox ที่แสดงบนเบรเซอร์ในแกน X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 Combobox Table

Attributes	Data Type	Meaning
EntityID (PK)	Integer	หมายเลขที่ใช้อ้างอิง Combobox แต่ละตัวบนฟอร์ม
ComboboxName	Char(50)	ชื่ออ้างอิงของ Combobox แต่ละตัวบนฟอร์ม
ComboboxSize	Integer	จำนวนบรรทัดที่ใช้แสดงลิสต์ของ Combobox
ComboboxTop	Integer	ตำแหน่งของ Combobox บน Work Area ตามแนวแกน X ของหน้าจอ
ComboboxLeft	Integer	ตำแหน่งของ Combobox บน Work Area ตามแนวแกน Y ของหน้าจอ
ComboboxRow	Integer	ตำแหน่งของ Combobox ที่แสดงบนเบราว์เซอร์ตามแนวแกน Y ของหน้าจอ
ComboboxColum	Integer	ตำแหน่งของ Combobox ที่แสดงบนเบราว์เซอร์ตามแนวแกน X ของหน้าจอ
ComboboxAutoData	Char(50)	ชื่อของ Combobox ที่ใช้เป็นลิสต์แสดงข้อมูลอัตโนมัติ
ComboboxDatePairOf	Char(50)	ชื่อของ Combobox ที่ใช้เป็นลิสต์แสดงวันที่อัตโนมัติ
ComboboxYearPairOf	Char(50)	ชื่อของ Combobox ที่ใช้เป็นลิสต์แสดงปีอัตโนมัติ
ComboboxFirstPair	Char(50)	ชื่อของ Combobox ที่ใช้เป็นลิสต์แสดงอำเภออัตโนมัติ เมื่อจังหวัดถูกเลือก
ComboboxSecondPair	Char(50)	ชื่อของ Combobox ที่ใช้เป็นลิสต์แสดงรหัสไปรษณีย์อัตโนมัติ เมื่ออำเภอถูกเลือก

ตารางที่ 3.8 ComboboxOption Table

Attributes	Data Type	Meaning
EntityID (PK)	Integer	หมายเลขที่ใช้อ้างอิง Combobox บนฟอร์มหนึ่งๆ
ComboboxOption	Char(50)	รายการที่มีให้เลือกในลิสต์ของ Combobox เรียงตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 Button Table

Attributes	Data Type	Meaning
EntityID (PK)	Integer	หมายเลขที่ใช้อ้างอิง Button แต่ละตัวบนฟอร์มหนึ่งๆ
ButtonName	Char(50)	ชื่อสำหรับอ้างอิงของ Button แต่ละตัวบนฟอร์ม
ButtonInputtype	Char(50)	ค่าที่ใช้กำหนดว่าเป็น Button ชนิดใด ในคำสั่งภาษา HTML มี 3 แบบคือ Submit, Reset, และ Button
ButtonTop	Integer	ตำแหน่งของ Button บนWork Area ตามแนวแกนX
ButtonLeft	Integer	ตำแหน่งของ Button บนWork Area ตามแนวแกนY
ButtonRow	Integer	ตำแหน่งของ Button ที่แสดงบนเบรเซอร์ในแกนY
ButtonColumn	Integer	ตำแหน่งของ Button ที่แสดงบนเบรเซอร์ในแกนX

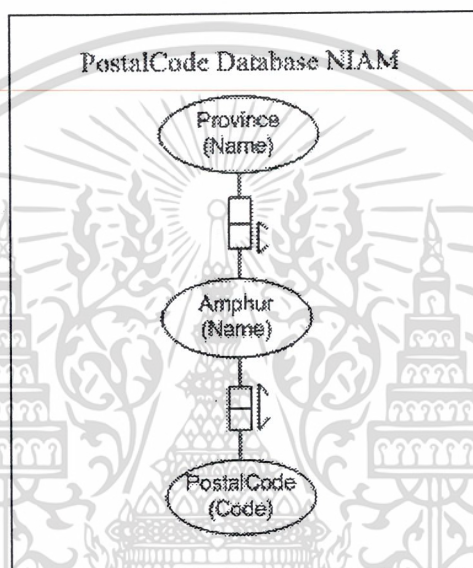
ตารางที่ 3.10 Textarea Table

Attributes	Data Type	Meaning
EntityID (PK)	Integer	หมายเลขที่ใช้อ้างอิงTextarea แต่ละตัวบนฟอร์มหนึ่งๆ
TextareaName	Char(50)	ชื่อสำหรับอ้างอิงของ Textarea แต่ละตัวบนฟอร์ม
TextareaRowLine	Integer	จำนวนบรรทัดหรือขนาดความกว้างของ Textarea ที่ให้แสดงบนเบรเซอร์
TextareaColChar	Integer	จำนวนช่องอักขระหรือขนาดความยาวของ Textarea ที่ให้แสดงบนเบรเซอร์
TextareaInitialText	Char(50)	ข้อความเริ่มต้นที่ให้ปรากฏใน Textarea เมื่อเพจถูกโหลดขึ้นมา
TextareaTop	Integer	ตำแหน่งของTextarea บนWork Area ตามแนวแกนX
TextareaLeft	Integer	ตำแหน่งของ Textarea บนWork Area ตามแนวแกนY
TextareaRow	Integer	ตำแหน่งของ Textarea ที่แสดงบนเบรเซอร์ในแกนY
TextareaColumn	Integer	ตำแหน่งของ Textarea ที่แสดงบนเบรเซอร์ในแกนX

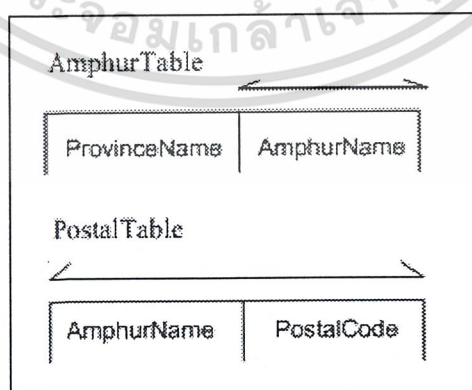
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบฐานข้อมูลรหัสไปรษณีย์ของแต่ละจังหวัดในประเทศไทย

ความสามารถอีกอย่างหนึ่งของฟอร์มนอกจากตรวจสอบความถูกต้องของข้อมูลแล้ว ยังสามารถเรียกชุดข้อมูลที่สัมพันธ์กันออกมาแสดงในลิสต์ของ Combo Box หรือ Selection Box เพื่อให้ความสะดวกแก่ผู้กรอกแบบฟอร์ม ในกรณีนี้เมื่อผู้กรอกใส่ชื่ออำเภอ หรือ จังหวัดลงไป ฟอร์มจะแสดงลิสต์ของจังหวัด หรือรหัสไปรษณีย์ที่สัมพันธ์กัน ให้ผู้กรอกได้เลือก ซึ่งเป็นการตรวจสอบความถูกต้องของข้อมูลอีกรูปแบบหนึ่ง



รูปที่ 3.15 การออกแบบฐานข้อมูลของรหัสไปรษณีย์ในประเทศไทย (สำหรับอำเภอที่ซ้ำกัน จะแก้ไขโดยเพิ่มชื่อของจังหวัดนั้นลงไป)



รูปที่ 3.16 ตารางของข้อมูลรหัสไปรษณีย์ใน Relational Database

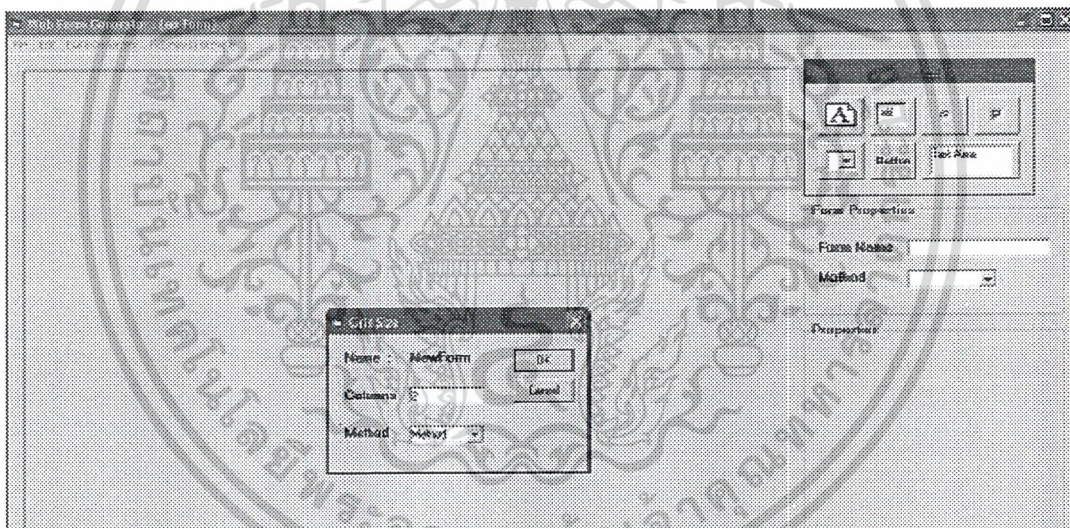
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ผลการทดลองในส่วนการสร้างฟอร์ม

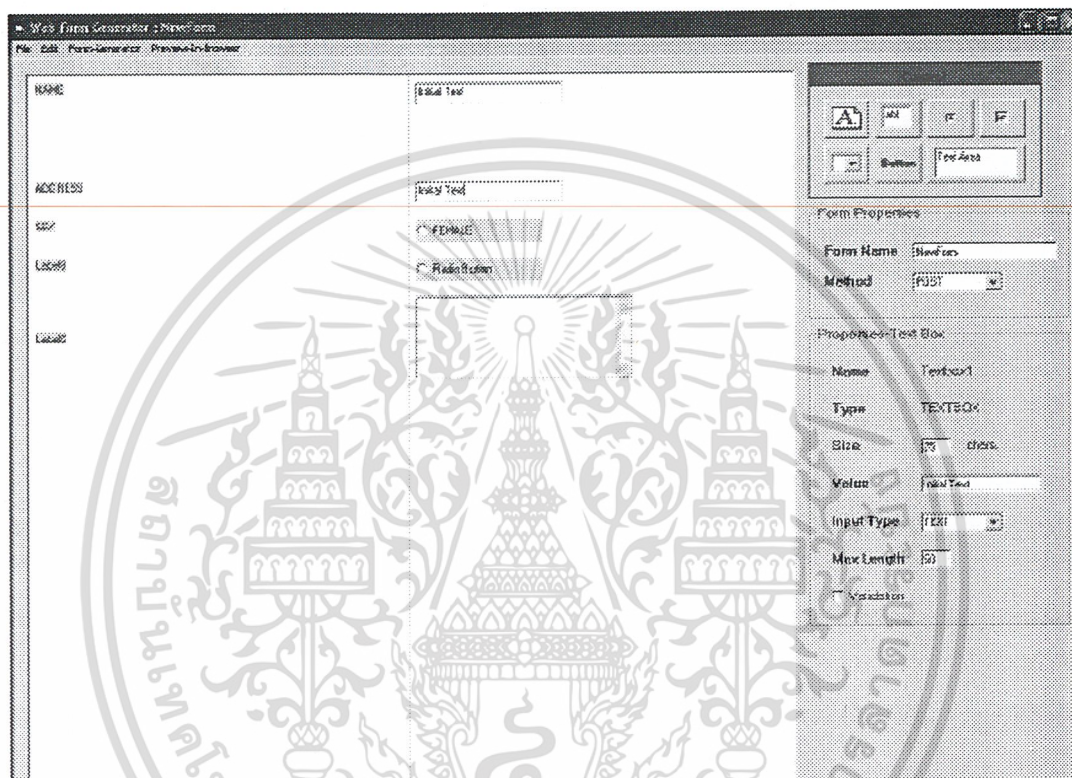
เมื่อเริ่มเปิดใช้งาน โปรแกรมสร้างเว็บฟอร์มพร้อมระบบตรวจสอบข้อมูล จะมี ไดอะล็อกบ็อกซ์ปรากฏขึ้นมาโดยอัตโนมัติ เพื่อให้ผู้ใช้กำหนดคอตัมน์ในฟอร์มใหม่ที่ต้องการสร้าง และกำหนด METHOD ซึ่งเป็นวิธีการรับ-ส่งฟอร์มไปยังเครื่องเซิร์ฟเวอร์ (GET/POST) หากไม่ต้องการสร้างฟอร์มใหม่สามารถกดปุ่ม Cancel ได้ดังรูปที่ 4.1



รูปที่ 4.1 หน้าจอหลักเมื่อเริ่มเปิดใช้งานโปรแกรม

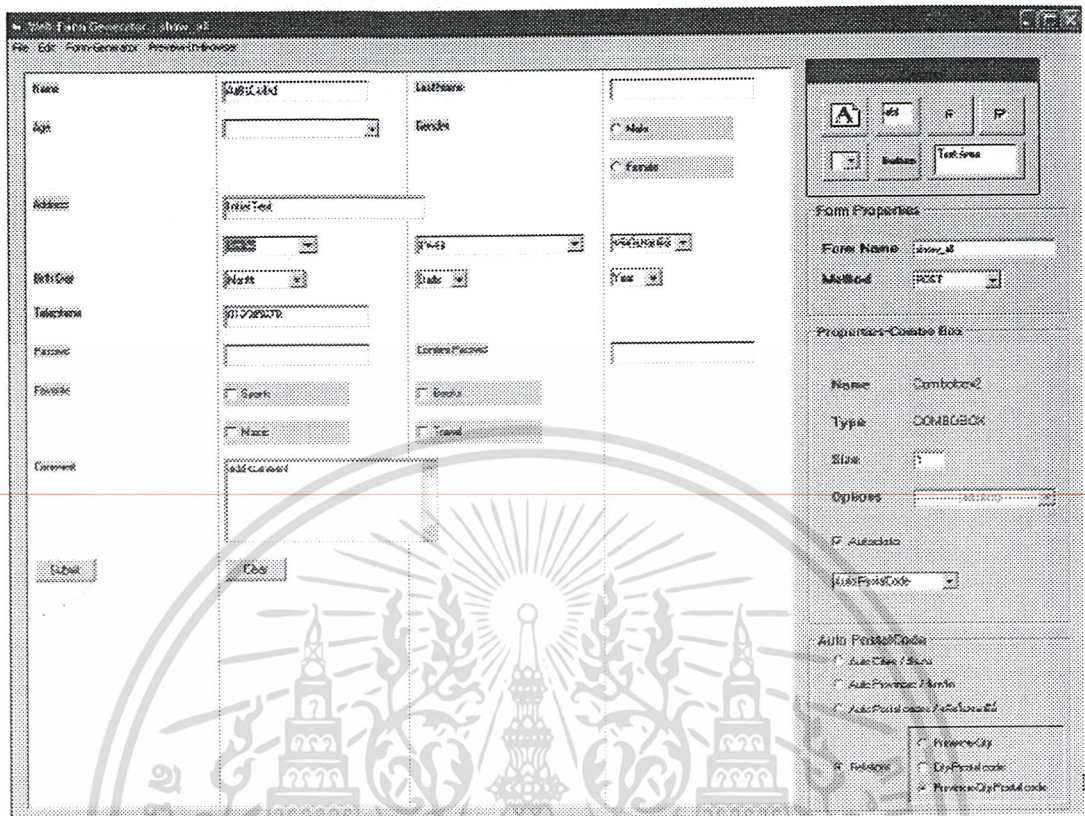
หลังจากที่ผู้ใช้กำหนดคอตัมน์และเมธอดของฟอร์มเรียบร้อยแล้ว ต่อไปจะเป็น ขั้นตอนการออกแบบรูปร่างหน้าตาของฟอร์ม โดยผู้ใช้สามารถคลิกแล้วลากตัว Control หรือ อุปกรณ์รับข้อมูลทุกชนิดยกเว้น Radio Button จาก Control Box ทางขวามือ แล้วนำมาวางไว้บน พื้นที่สำหรับสร้างฟอร์มทางซ้ายมือได้ตามความต้องการ

เมื่อคลิกที่ Control ตัวใดบนฟอร์ม ให้มีสภาพ Active จะปรากฏ Properties Box ของ Control นั้น ๆ อยู่ทางด้านขวา เพื่อให้ผู้ใช้กำหนดค่าพารามิเตอร์ต่าง ๆ สำหรับนำไปสร้าง ดังตัวอย่างในรูปที่ 4.2



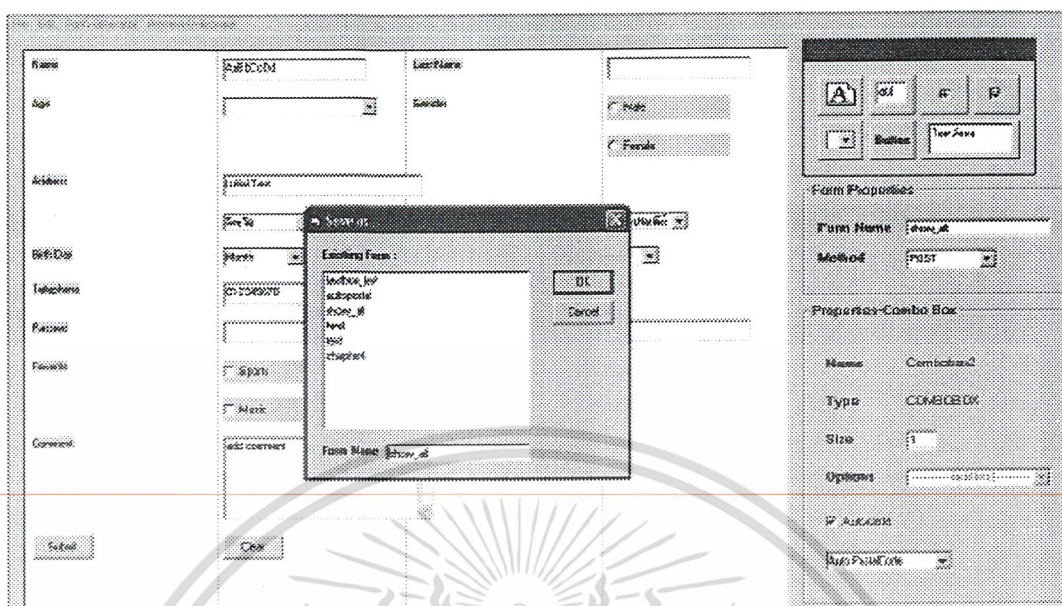
รูปที่ 4.2 การวางอุปกรณ์รับข้อมูลลงบนฟอร์ม

สำหรับ Control ที่เป็น Text Box จะสามารถทำการ Validate ข้อมูลได้ เพียงคลิก Check Box ที่ Properties Box ของ Text Box แล้วจะปรากฏ Validation Box ขึ้นมาให้ผู้ใช้เลือกจะทำให้ Text Box สามารถ Validate ข้อมูลแบบใดได้บ้าง ดังรูปที่ 4.3



รูปที่ 4.3 การสร้างฟอร์มพร้อมการ ตรวจสอบความถูกต้องของข้อมูล

ทุกครั้งที่ผู้ใช้ต้องการบันทึกฟอร์มที่สร้างไว้ สามารถคลิกที่คำสั่ง save หรือ save as บนเมนู File ซึ่งในโปรแกรมจะปรากฏไดอะล็อกบ็อกซ์ แสดงรายชื่อของฟอร์มที่เคยบันทึกไว้ แล้วมีช่องไว้สำหรับเขียนชื่อฟอร์มใหม่นั้นลงไป ดังในรูปที่ 4.4 ในคำสั่ง save นี้เป็นการบันทึกข้อมูล พรอเพอร์ตี้ต่าง ๆ ของฟอร์มลงในฐานข้อมูลเท่านั้น ยังไม่ใช้การบันทึกให้อยู่ในรูปแบบของไฟล์ใด ๆ



รูปที่ 4.4 การบันทึกฟอร์มลงฐานข้อมูล

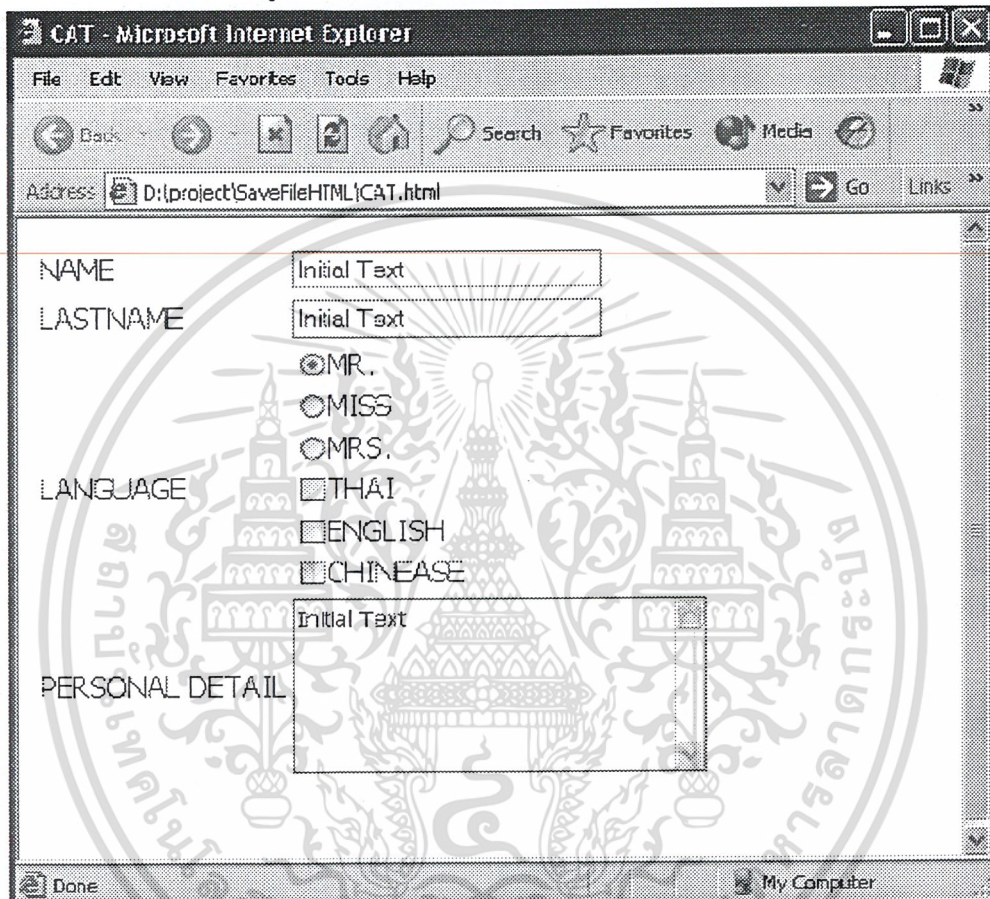
4.2 ผลการทดลองในส่วนสร้างโค้ด

หลังจากการ save ฟอร์มเรียบร้อยแล้ว หากผู้ใช้ต้องการดูโค้ดของฟอร์มนั้น ๆ ก็ สามารถเลือกคำสั่ง Generate HTML หรือ Generate with JavaScript (กรณีที่ฟอร์มนั้นสามารถ ตรวจสอบความถูกต้องของข้อมูลได้) จากเมนู Form-Generator จากนั้นโปรแกรมจะแสดงโค้ด ออกมา แล้วมีปุ่ม save ให้ผู้ใช้สามารถทำการบันทึกเป็นไฟล์ .html หรือ .txt ได้ตามความต้องการ จากรูปที่ 4.5เป็นการสร้างโค้ด HTML ล้วน ๆ เนื่องจากไม่มีการทำ Validate ซึ่งเป็นผลมาจากการ สร้างฟอร์มในรูป 4.2 ส่วนในรูปที่ 4.6 เป็นโค้ดที่เป็นผลมาจากการสร้างฟอร์มพร้อมกับการ Validate ข้อมูลในรูป 4.3

4.3 ผลการทดลองในส่วนแสดงฟอร์มบนเบราว์เซอร์

4.3.1 การแสดงฟอร์มธรรมดาบนเบราว์เซอร์

ตัวอย่างในรูปที่ 4.7 เป็นผลบนเบราว์เซอร์จากการสร้างฟอร์มลักษณะ 2 คอลัมน์

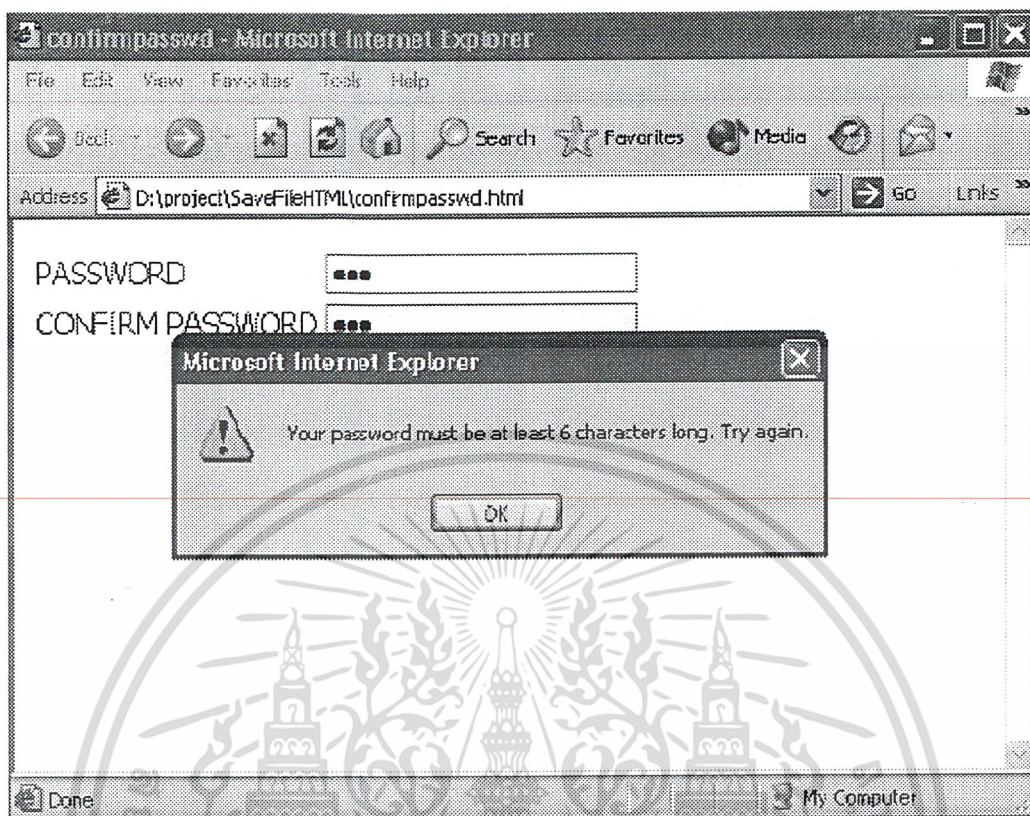


รูปที่ 4.7 ตัวอย่างการแสดงฟอร์มบนเบราว์เซอร์

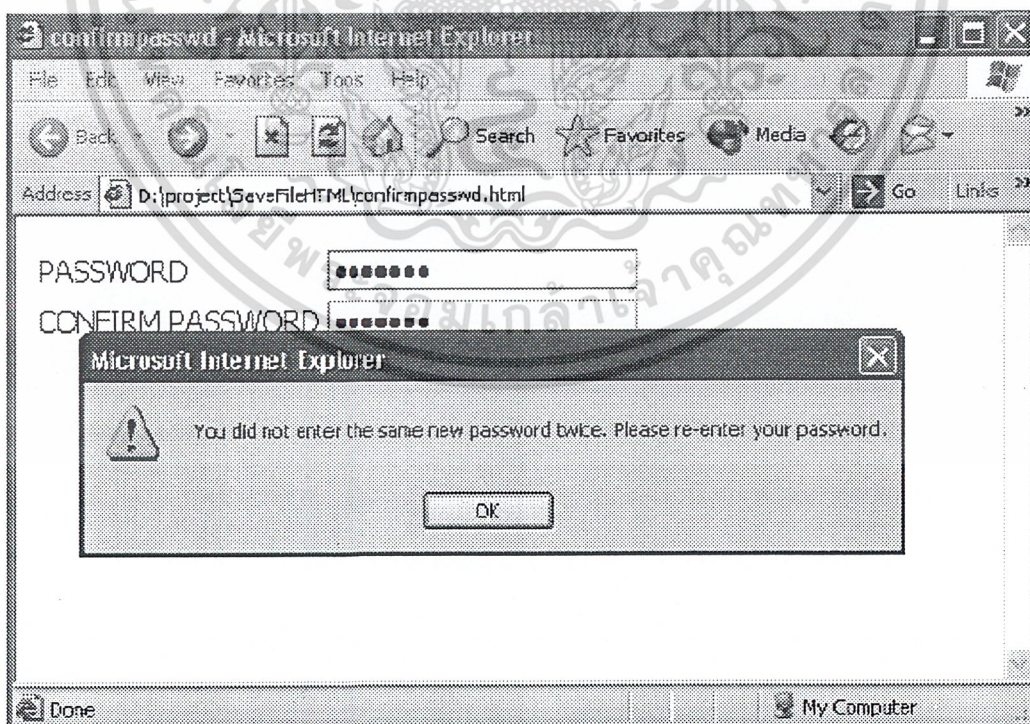
4.3.2 การตรวจสอบความถูกต้องของฟอร์ม

ตัวอย่างในรูปที่ 4.8 เป็นการแสดงการตรวจสอบความถูกต้องของพาสเวิร์ด ในกรณีที่กำหนดให้จำนวนอักษรที่รับเข้ามาต้องมีอย่างน้อย 6 ตัวอักษร เมื่อผู้ใช้ป้อนข้อมูลที่น้อยกว่าที่กำหนดไว้ จะปรากฏคำเตือนขึ้นมา เพื่อให้ทำการป้อนใหม่

ในรูปที่ 4.9 เป็นการตรวจสอบความถูกต้องของช่องพาสเวิร์ดเช่นเดียวกัน แต่เป็นกรณีที่การยืนยันพาสเวิร์ดไม่ตรงกัน จะมีการแจ้งความผิดพลาดขึ้นอย่างชัดเจน



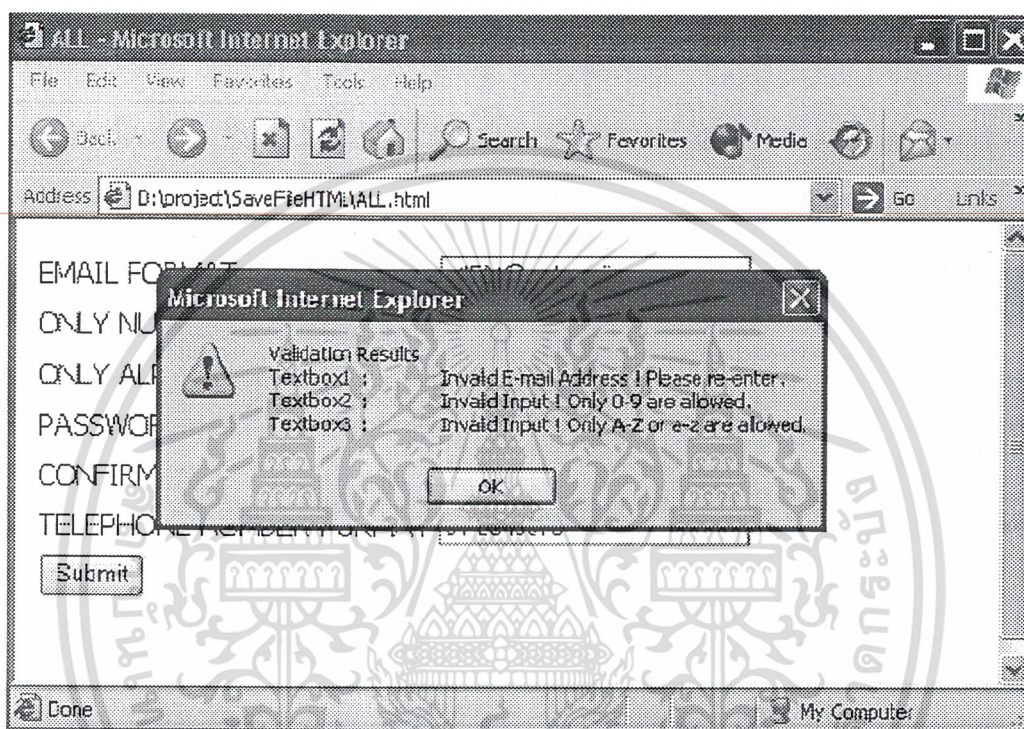
รูปที่ 4.8 การตรวจสอบจำนวนอักษรในช่อง Password



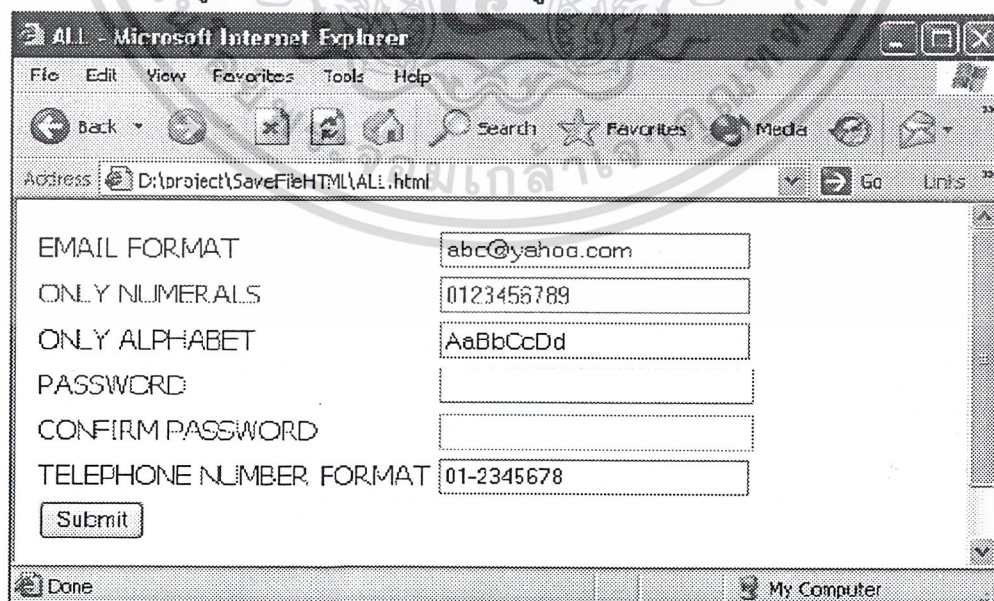
รูปที่ 4.9 การตรวจสอบข้อมูลในช่อง Confirm Password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่รูปที่ 4.10 เป็นความสามารถตรวจสอบความถูกต้องของข้อมูลของฟอร์ม ที่สามารถตรวจพบข้อมูลที่ผิดพลาดทั้งหมดที่เกิดขึ้นจากการกรอกแบบฟอร์มของผู้ใช้ แล้วทำการแจ้งให้ทราบทันทีหลังจากที่ฟอร์มถูกคลิก Submit สำหรับรูปที่ 4.11 เป็นตัวอย่างกรณีที่มีการกรอกข้อมูลถูกต้อง จะไม่ปรากฏไดอะล็อกบ็อกซ์ขึ้นมาเตือนความผิดพลาดใดๆ



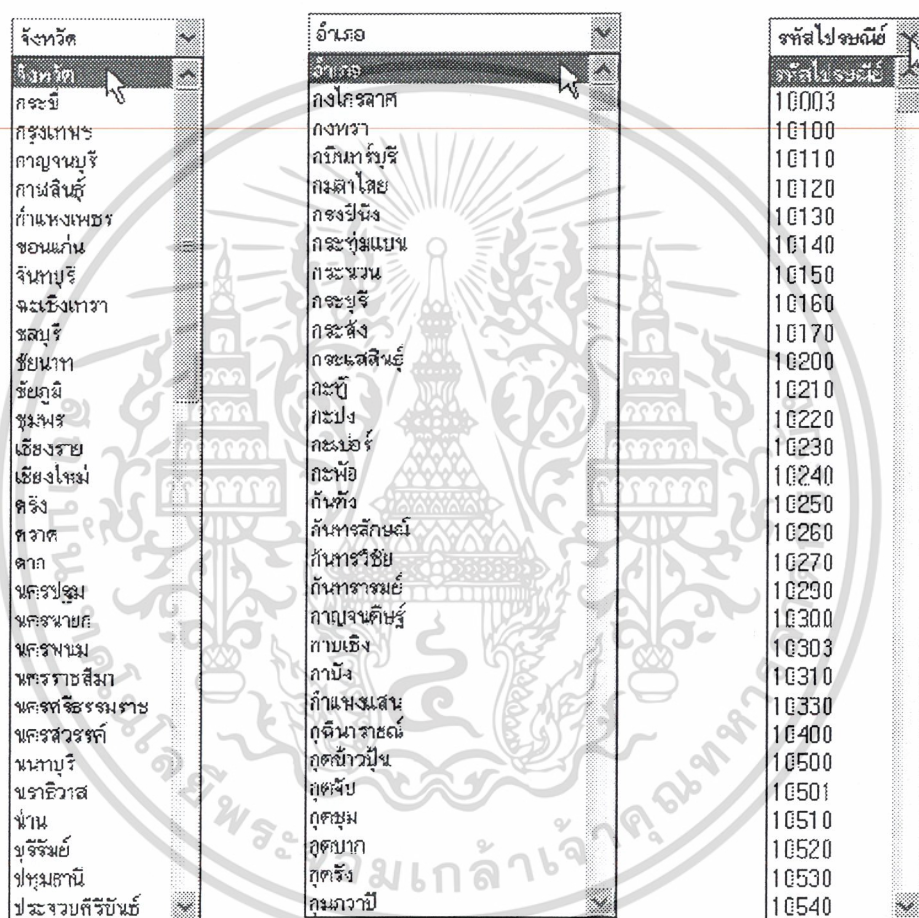
รูปที่ 4.10 การตรวจสอบความถูกต้องของฟอร์มทั้งหมด



รูปที่ 4.11 การส่งแบบฟอร์มสำเร็จ

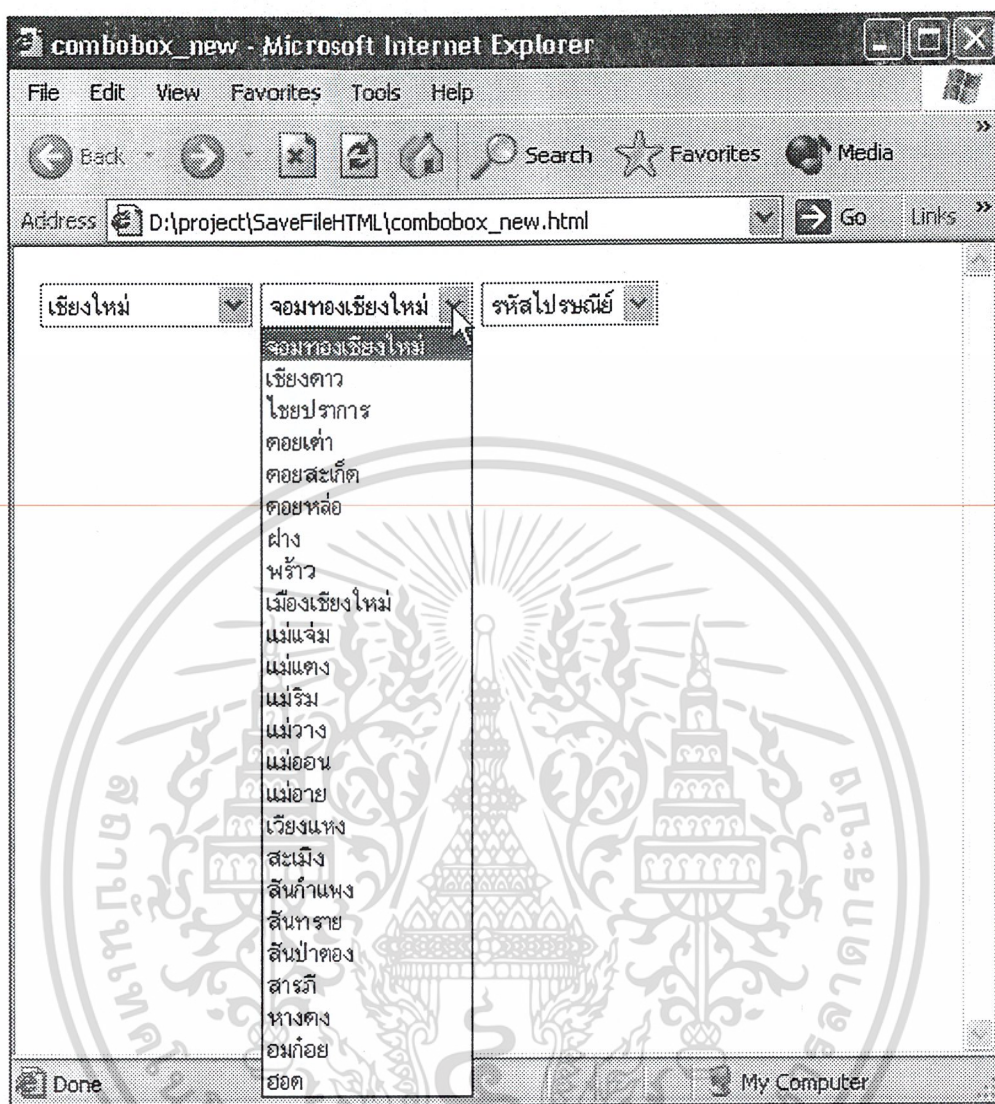
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปตัวอย่างต่อไปเป็นรูปของ Combo box ที่ประกอบด้วยชุดข้อมูลอัตโนมัติของ จังหวัด, อำเภอ และรหัสไปรษณีย์ ซึ่งเมื่อผู้ใช้ทำการเลือกจังหวัดใดจังหวัดหนึ่งในลิสต์ ช่อง Combo box ต่อไปจะปรากฏรายชื่ออำเภอทั้งหมดที่มีอยู่ในจังหวัดนั้น และเมื่อเลือกอำเภอ อีกช่อง จะมีรหัสไปรษณีย์ของอำเภอนั้น ๆ ให้ผู้ใช้เลือกตามลำดับ ซึ่งวิธีนี้เป็นการช่วยลดความผิดพลาด ในการกรอกข้อมูลของผู้ใช้ลงส่วนหนึ่ง



รูปที่ 4.12 แสดงรายการของจังหวัด, อำเภอ และรหัสไปรษณีย์โดยที่ยังไม่มีความสัมพันธ์กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ตัวอย่างการแสดงผลข้อมูลอำเภอที่สัมพันธ์กับจังหวัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

5.1 วิจัยและสรุปผลการทดลอง

จากการออกแบบและพัฒนาโปรแกรมซึ่งเป็นเครื่องมือสร้างเว็บฟอร์มพร้อมมีระบบตรวจสอบความถูกต้องของข้อมูล ตามวัตถุประสงค์ของโครงการ พบว่าโปรแกรมมีข้อจำกัดดังนี้

1. ในกระบวนการ Add Form หรือวิธีจัดวางอุปกรณ์รับข้อมูลลงในฟอร์ม ต้องให้อยู่ในรูปของตาราง เพื่อประโยชน์ในการสร้างโค้ด แต่ขณะนี้โปรแกรมสามารถสร้างตารางได้สูงสุด 20 Row และ 5 Column

2. การตรวจสอบความถูกต้องของข้อมูลในเบื้องต้นนี้ใช้ได้เฉพาะกรณีของ Text Box ซึ่งข้อมูลที่สามารถตรวจสอบได้ขณะนี้คือ รูปแบบของอีเมลแอดเดรส, พาสเวิร์ดและการคอนเฟิร์มพาสเวิร์ด, หมายเลขโทรศัพท์ พร้อมกับการจัดให้อยู่ในรูปแบบที่ถูกต้องโดยอัตโนมัติ แต่มีข้อจำกัดว่าต้องเป็นหมายเลขโทรศัพท์ภายในประเทศไทยเท่านั้น และข้อมูลที่เป็นตัวเลข หรือตัวอักษรเพียงอย่างเดียว

3. Auto Combo Box หรือการนำชุดข้อมูลที่สัมพันธ์กันมาใส่ในลิสต์ของ Combo Box เพื่อให้ความสะดวกแก่ผู้กรอกแบบฟอร์ม ในขณะนี้ทำได้เฉพาะชุดข้อมูลของรหัสไปรษณีย์ในประเทศไทย และวัน เดือน ปี

สังเกตได้ว่าข้อจำกัดที่สำคัญของโปรแกรม คือ มีระบบตรวจสอบความถูกต้องของข้อมูลให้ผู้ใช้เลือกได้เพียงไม่กี่แบบ และยังไม่สามารถทำได้ในอุปกรณ์อื่น ๆ นอกจาก Text Box อีกทั้งยังขาดความยืดหยุ่นที่ให้ผู้ใช้งานสามารถปรับเปลี่ยนลักษณะการ validate ได้ตามความต้องการ

ปัญหาที่เกิดขึ้นนี้ คาดว่าเป็นเพราะการออกแบบระบบในบางส่วน ยังขาดความสมบูรณ์ อีกทั้งฟอร์มที่สามารถทำการ validate ได้ นั้น มักมีอัลกอริทึมในการเขียนโปรแกรม จาวาสคริปต์ที่ซับซ้อน ทำให้เกิดความยุ่งยากสำหรับนำมาประยุกต์ใช้ในโปรแกรม

5.2 แนวทางในการพัฒนาต่อ

โปรแกรมต้นแบบที่เป็นเครื่องมือสร้างเว็บฟอร์มพร้อมระบบตรวจสอบความถูกต้องของข้อมูลนี้ จัดได้ว่ามีประสิทธิภาพในระดับหนึ่ง รูปแบบการตรวจสอบข้อมูลที่มีให้เลือกในโปรแกรมนั้น เป็นรูปแบบหลัก ๆ ที่มีอยู่ในเว็บฟอร์มทั่วไป ผู้ใช้จึงสามารถนำฟอร์มที่สร้างขึ้นมาไปใช้งานได้จริง โดยไม่จำเป็นต้องมีความรู้ด้านภาษา HTML และจาวาสคริปต์

แนวทางในการพัฒนาโปรแกรมต่อไป คือการเพิ่มชุดข้อมูลที่มีความสัมพันธ์กันให้มากยิ่งขึ้น เพื่อเพิ่มความสะดวกในการสร้างฟอร์ม เช่น รหัสโทรศัพท์ทางไกลระหว่างประเทศ หรือรายชื่อประเทศในแต่ละทวีป รวมทั้งควรมีชุดข้อมูลอื่น ๆ ที่มักใช้บ่อยครั้งในการสร้างฟอร์มพร้อมให้ผู้เลือกเติมในลิสต์ได้หลากหลาย โดยที่ผู้ใช้ไม่จำเป็นต้องเสียเวลาเพิ่มรายการบางอย่างที่มีจำนวนมาก เช่น ชุดของข้อมูลสี, อาชีพ หรือ รายชื่อผลิตภัณฑ์ เป็นต้น และจะเป็นประโยชน์อย่างมากหากพัฒนาโปรแกรมให้ผู้ใช้สามารถสร้างลิสต์ของ Combo box ได้ด้วยตัวเอง และสามารถบันทึกเก็บไว้ใช้ได้ในครั้งต่อไป อีกทั้งสามารถแก้ไขเพิ่มเติมลิสต์ได้ตามความต้องการ

นอกจากนี้การแก้ไขข้อจำกัดที่มีอยู่ในขณะนี้ เป็นอีกแนวทางหนึ่งควรพัฒนาให้โปรแกรมมีความสมบูรณ์ในการใช้งานให้มากยิ่งขึ้น

บรรณานุกรม

- กิตติ กักดีวัฒนะกุล และ จำลอง ทรูอุตสาหะ, ม.ป.ป., VISUAL BASIC 6 ฉบับโปรแกรมเมอร์. พิมพ์ครั้งที่ 9. กรุงเทพฯ : ไทยเจริญการพิมพ์.
- จิตเกษม พัฒนาศิริ, 2541, เสริมแต่งโฮมเพจให้มีชีวิตชีวาด้วยจาวาสคริปต์. พิมพ์ครั้งที่ 1. กรุงเทพฯ : สำนักพิมพ์วิดิทัศน์ กรู๊ป.
- ชัยณุกพงศ์ ชาญฤกษ์ และ พันจันทร์ ธนวัฒน์เสถียร, ม.ป.ป., สร้างเว็บเพจมือแบบมืออาชีพ HTML. พิมพ์ครั้งที่ 1. กรุงเทพฯ : ซัคเซส มีเดีย.
- ชาริน สัทธรรมชารี และ สุรสิทธิ์ คิวประสพศักดิ์, ม.ป.ป., VISUAL BASIC 6.0 ฉบับเพื่อการประยุกต์ใช้งาน. พิมพ์ครั้งที่ 3. กรุงเทพฯ : ซัคเซส มีเดีย.
- ศุภชัย สมพานิช, 2544, สร้างระบบงานฐานข้อมูลด้วย VISUAL BASIC ฉบับโปรแกรมเมอร์. พิมพ์ครั้งที่ 1. กรุงเทพฯ : อินโฟเพรส.
- ทรงศักดิ์ บรรจงมณี, 2541, เริ่มต้นเรียนรู้ JAVASCRIPT. พิมพ์ครั้งที่ 1. กรุงเทพฯ : ซีเอ็ดดูเซ็น.

เว็บไซต์อ้างอิง

- <http://www.cat.co.th>
- <http://javascript.internet.com/master-list/f-master-list.html>
- <http://www.devguru.com>
- <http://tech.irt.org/articles/js042/#1>
- <http://www.geocities.com/Eureka/Pard/3490/>