

การสร้างเอกสารการพิมพ์ที่สวยงามจากเว็บแอปพลิเคชัน

GENERATE PRETTY-PRINTING FORMAT FROM WEB APPLICATION



H003451



โดย

สุทธิศักดิ์ เหมาะะประไพพันธุ์

SUTISAK MOHPRAPAIAN

อาจารย์ที่ปรึกษา

ผศ.ดร.พรฤดี เนติโสภาคกุล

6/18/9880
1/19/26/94

วัน เดือน ปี.....	04. 8. 2550
เลขทะเบียน.....	H003451
เลขเรียกหนังสือ.อพ. ๑๗๗ ก	2549
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจส."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
 หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
 คณะเทคโนโลยีสารสนเทศ
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ภาคเรียนที่ 2 ปีการศึกษา 2549

**GENERATE PRETTY-PRINTING FORMAT FROM WEB
APPLICATION**



**A SYSTEM DEVELOPMENT PROJECT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2007

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	การสร้างเอกสารการพิมพ์ที่สวยงามจากเว็บแอปพลิเคชัน
นักศึกษา	นายสุทธิศักดิ์ เหมาะะประไพพันธุ์
รหัสนักศึกษา	45061635
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2549
อาจารย์ที่ปรึกษา	ผศ.ดร.พรฤดี เนติโสภากุล

บทคัดย่อ

ปัจจุบันเว็บแอปพลิเคชันมีความสามารถโดดเด่นมากมาย แต่ปัญหาหนึ่งซึ่งเป็นข้อจำกัดในปัจจุบัน คือ การพิมพ์รายงานผ่านเว็บแอปพลิเคชันนั้น ยังไม่มีมาตรฐานเพื่อการพิมพ์เอกสารในการพิมพ์เอกสารผ่านเว็บแอปพลิเคชัน เป็นผลทำให้ผลงานการพิมพ์เอกสารจากเว็บแอปพลิเคชัน จะแตกต่างกันออกไป ขึ้นอยู่กับ โปรแกรมบราวเซอร์ที่ใช้ และจะพิมพ์ออกมาไม่เหมือนกันเมื่อใช้เครื่องพิมพ์ต่างรุ่นต่างยี่ห้อกัน โครงการพัฒนาระบบงานนี้จึงได้พัฒนาชุดโปรแกรมที่จะสามารถทำให้สามารถพิมพ์เอกสารออกจากเครื่องพิมพ์ได้อย่างสวยงามลงตามตำแหน่งในหน้ากระดาษตามรูปแบบที่ผู้พัฒนาเว็บแอปพลิเคชันกำหนดจากเว็บแอปพลิเคชัน

Title	Generate Pretty-Printing Format from Web Application
Student	Mr. Sutisak Mohprapaipan
Student ID.	45061635
Degree	Master of Science
Programme	Information Science
Academic Year	2006
Advisor	Asst. Prof. Dr. Ponrudee Netisopakul

ABSTRACT

At present, the web application has many features but there are some limitations. One of the numbers of problems is report printing. There is no standard for printing report from any web application that could output same results in a form of hardcopy. Usually the web server or web browser who hosts the web application must handle this aspect of printing the report. This project paper is developing a program to print aspect report from web application.

กิตติกรรมประกาศ

โครงการนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก ผศ.ดร.พรฤดี เนติโสภาคกุล ซึ่งเป็นอาจารย์ที่ปรึกษา ข้าพเจ้ารู้สึกทราบบ้างในความอนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอกราบพระคุณคณาจารย์แขนงวิทยาการสารสนเทศ สาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

ขอขอบคุณบัณฑิตศึกษาและบัณฑิตวิทยาลัย คณะเทคโนโลยีสารสนเทศที่ให้ความช่วยเหลือ ในเรื่องต่างๆ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุกท่าน

สุทธิศักดิ์ เหมะประไพพันธุ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 แผนการดำเนินการ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.6 ขั้นตอนของการศึกษา.....	3
บทที่ 2 หลักการที่ใช้ในการพัฒนาโปรแกรม.....	4
2.1 พิมพ์จากคำสั่งพิมพ์ขอบเว็บเบราว์เซอร์.....	4
2.2 พิมพ์จากไฟล์ Portable Document Format (PDF).....	6
2.3 พิมพ์จากโปรแกรม Crystal Report.....	9
บทที่ 3 การวิเคราะห์และออกแบบโปรแกรม.....	11
3.1 ความต้องการของโปรแกรม.....	11
3.2 แบบจำลองเชิงแนวคิดในการใช้งาน โปรแกรม.....	12
3.3 ขอบเขตของโปรแกรม.....	13
3.4 แบบจำลองทางกายภาพของการพัฒนา โปรแกรม.....	13

สารบัญ (ต่อ)

	หน้า
บทที่ 4 การสร้างโปรแกรม.....	15
4.1 ศึกษาการทำงานของ Library HTML2FPDF.....	15
4.2 ดัดแปลง Library HTML2FPDF เพื่อให้สามารถทำงานร่วมกับภาษาไทยได้...	18
4.3 ศึกษาการทำงานของ Library FPDF.....	19
4.4 ดัดแปลง Library FPDF เพื่อให้สามารถทำงานร่วมกับภาษาไทยได้.....	27
4.5 สร้างโปรแกรมเพื่อประมวลผลจาก HTML ไปเป็น PDF โดยอัตโนมัติ.....	30
บทที่ 5 การใช้งานโปรแกรม.....	32
5.1 การติดตั้งโปรแกรม.....	32
5.2 การใช้งาน โปรแกรมสำหรับผู้ใช้งานเว็บแอปพลิเคชัน.....	35
5.3 การใช้งาน โปรแกรมสำหรับผู้สร้างเว็บแอปพลิเคชัน.....	36
บทที่ 6 บทสรุป.....	41
6.1 สรุปผลการพัฒนาโปรแกรม.....	42
6.2 ข้อจำกัดของ โปรแกรมที่พัฒนาขึ้น.....	42
6.3 ข้อเสนอแนะในการพัฒนาต่อ.....	43
บรรณานุกรม.....	44
ภาคผนวก.....	45
ประวัติผู้เขียน.....	56

สารบัญตาราง

ตารางที่	หน้า
4.1 ฟังก์ชันภายใน Library HTML2FPDF.....	5
4.2 ฟังก์ชันภายใน Library FPDF.....	23



สารบัญรูป

รูปที่	หน้า
2.1 การใช้งานคำสั่งพิมพ์ของเว็บเบราว์เซอร์.....	5
2.2 รายงานก่อนพิมพ์ของเว็บเบราว์เซอร์.....	5
2.3 หน้าตาของโปรแกรม Acrobat Reader.....	7
2.4 รายงานที่สามารถพิมพ์ออกจาก Acrobat Reader.....	8
2.5 หน้าจอที่ทำการรอรับค่าก่อนประมวลผล.....	9
2.6 รายงานที่พร้อมสำหรับการพิมพ์ผ่านเว็บ จากโปรแกรม Crystal Report.....	10
3.1 ตัวอย่างแบบฟอร์มที่ต้องพิมพ์ลงตำแหน่งที่ถูกต้อง.....	11
3.2 แบบจำลองเชิงแนวคิดของโปรแกรม.....	12
3.3 แบบจำลองทางกายภาพของการพัฒนาโปรแกรม.....	14
4.1 แผนภาพฟังก์ชันของ Library HTML2FPDF.....	15
4.2 แผนภาพรหัส Encoding ISO-8859-1.....	18
4.3 แผนภาพรหัส Encoding ISO-8859-11.....	19
4.4 แผนภาพฟังก์ชันของ Library FPDF.....	19
4.5 แผนภาพขั้นตอนการสร้าง Font.....	27
4.6 แผนภาพขั้นตอนการทำงานของโปรแกรม.....	30
4.7 แผนภาพเมื่อโปรแกรมกำลังใช้งานบนเว็บเซิร์ฟเวอร์.....	31
5.1 หน้าจอการทำงานของโปรแกรม.....	34
5.2 ปุ่มสั่งให้โปรแกรมเริ่มทำงาน.....	35
5.3 ผลลัพธ์หน้ารายงานผ่าน โปรแกรม.....	35
5.4 เมื่อไม่ใช่คำสั่งขึ้นหน้าใหม่.....	37
5.5 ผลลัพธ์ก่อนพิมพ์เมื่อไม่ใช่คำสั่งขึ้นหน้าใหม่.....	37
5.6 เมื่อใช้คำสั่งขึ้นหน้าใหม่.....	38
5.7 ผลลัพธ์ก่อนพิมพ์เมื่อใช้คำสั่งขึ้นหน้าใหม่.....	38
5.8 เมื่อไม่กำหนดความกว้าง.....	39
5.9 ผลลัพธ์ก่อนพิมพ์เมื่อไม่กำหนดความกว้าง.....	39
5.10 เมื่อกำหนดความกว้าง.....	40
5.11 ผลลัพธ์เมื่อกำหนดความกว้างตาราง.....	40

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การเผยแพร่ข้อมูลข่าวสารบน Internet มีใช้กันอย่างแพร่หลาย เนื่องด้วยจากการพัฒนาด้าน IP (Internet Protocol), HTML (Hyper Text Markup Language) และ HTTP (Hyper Text Transport Protocol) ซึ่งเป็นการปฏิวัติและสร้างมิติใหม่ในการกระจายข้อมูลและสารสนเทศ (Information) เพื่อการนำเสนอ การค้นหา ตลอดจนการโต้ตอบทางข้อมูล (วัชชัย สุริยะทองธรรม, 2545) โดยผู้ใช้งานสามารถใช้สารสนเทศได้ง่ายผ่านเว็บเบราว์เซอร์ ซึ่งมีอยู่มากมายหลายตัวในปัจจุบัน แต่ละเว็บเบราว์เซอร์ จะดึงและแปลงข้อมูลจากเว็บเซิร์ฟเวอร์ต่างๆ ที่อยู่ในเครือข่ายอินเทอร์เน็ต โดยเว็บเซิร์ฟเวอร์นี้ก็มีอยู่มากมายหลายตัวในปัจจุบันเช่นกัน ทำให้เกิดมาตรฐานการแสดงผลลัพธ์ขึ้น เพื่อกำหนดให้เว็บเซิร์ฟเวอร์ และ เว็บเบราว์เซอร์แสดงผลลัพธ์ออกมาให้เหมือนกันในทุกสถานการณ์ เพื่อให้ผู้ใช้งานสามารถเรียกดูข้อมูลได้เหมือนกันโดยไม่ยึดติดกับเว็บเซิร์ฟเวอร์ หรือเว็บเบราว์เซอร์ที่เป็นมาตรฐานมากที่สุดเพื่อการแสดงผลผ่านหน้าจอในปัจจุบันนี้คือ มาตรฐาน XHTML แต่มาตรฐานสำหรับการพิมพ์รายงานผ่านอินเทอร์เน็ตบนเว็บแอปพลิเคชันนั้น ยังไม่มีมาตรฐานใดๆเลยที่จะสามารถทำให้พิมพ์รายงานได้อย่างสวยงามผ่านเว็บแอปพลิเคชัน จึงเป็นที่มาที่ต้องการจะพัฒนาโปรแกรมเพื่อจะทำได้ทำให้สามารถที่จะพิมพ์รายงานผ่านเว็บแอปพลิเคชันเป็นภาษาไทย และภาษาอังกฤษ รวมถึงรูปภาพได้อย่างสวยงาม

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

การศึกษาและพัฒนาโปรแกรมในครั้งนี้มีวัตถุประสงค์เพื่อนำไปใช้ประโยชน์ในการสร้างรายงานเพื่อการพิมพ์รายงานออกเครื่องพิมพ์ผ่านเว็บแอปพลิเคชันได้จริง ซึ่งการพิมพ์รายงานผ่านเว็บแอปพลิเคชันนั้นยังเป็นปัญหาหนึ่งในการออกรายงานเพื่อการพิมพ์ในปัจจุบัน ดังนั้นจึงได้ทำการพัฒนาโปรแกรมเพื่อช่วยให้มีประสิทธิภาพในการทำการพิมพ์รายงานผ่านเว็บแอปพลิเคชันได้มากยิ่งขึ้น โดยมีวัตถุประสงค์ดังต่อไปนี้

- 1.2.1 รวบรวมและศึกษาความต้องการต่อการพิมพ์รายงานผ่านเว็บแอปพลิเคชันในปัจจุบัน เพื่อศึกษาค้นวิธีที่ดีที่สุดสำหรับผู้ใช้งาน ในการพิมพ์รายงานผ่านเว็บแอปพลิเคชัน
- 1.2.2 ศึกษากระบวนการประมวลผลที่เหมาะสมที่สุดสำหรับผู้พัฒนาโปรแกรม เพื่อนำมาใช้ในการออกรายงานให้เหมาะสมกับการพิมพ์รายงานผ่านเว็บแอปพลิเคชัน
- 1.2.3 พัฒนาโปรแกรมที่จะสามารถพิมพ์รายงานผ่านเว็บแอปพลิเคชันได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.2.4 นำโปรแกรมที่พัฒนาได้ไปใช้งานจริง เพื่อช่วยให้สามารถทำการพิมพ์รายงานผ่านเว็บแอปพลิเคชันได้อย่างมีประสิทธิภาพมากขึ้น

1.3 ขอบเขตของโครงการ

โปรแกรมที่ทำการพัฒนาขึ้นนี้ จะเป็นการพัฒนาโปรแกรมขึ้นมาเพื่อให้สามารถใช้ออกรายงานเพื่อการพิมพ์ผ่านเว็บแอปพลิเคชัน โดยมีขอบเขตของโครงการดังต่อไปนี้

- 1.3.1 สามารถออกรายงานแบบสวยงามผ่านเว็บแอปพลิเคชันเพื่อใช้ในการพิมพ์ได้ทุกที่ทุกเวลาผ่านเครื่องคอมพิวเตอร์ได้ทุกเครื่อง และสามารถผ่านเครื่องพิมพ์ได้ทุกเครื่อง
- 1.3.2 สามารถออกรายงานเป็นภาษาไทยและภาษาอังกฤษได้
- 1.3.3 สามารถออกรายงานที่ประกอบไปด้วยรูปภาพได้

1.4 แผนการดำเนินการ

ขั้นตอนในการการพัฒนาโปรแกรมจะประกอบด้วยขั้นตอนดังต่อไปนี้

- 1.4.1 ศึกษาความเป็นไปได้ของเทคโนโลยีที่มีอยู่แล้วในการพิมพ์รายงานผ่านเว็บแอปพลิเคชันในปัจจุบัน
- 1.4.2 คิดค้นหรือคัดแปลง วิธีที่เหมาะสมที่สุดในการพิมพ์รายงานผ่านเว็บแอปพลิเคชัน
- 1.4.3 พัฒนาโปรแกรมเพื่อการพิมพ์รายงานผ่านเว็บแอปพลิเคชันเพื่อการใช้งานจริง
- 1.4.4 ทดสอบการใช้งานและปรับปรุงแก้ไขโปรแกรม
- 1.4.5 สรุปผลการทดสอบจากการใช้งานที่เกิดขึ้น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ประโยชน์ต่อผู้ทำการพัฒนาโปรแกรม

- เป็นการนำความรู้ที่ได้จากการศึกษามาประยุกต์ใช้ในพัฒนาโปรแกรม เพื่อใช้งานจริง
- เป็นการศึกษาเรียนรู้และประยุกต์ใช้เทคโนโลยีที่มีอยู่ในปัจจุบันมาใช้ในการสร้างโปรแกรมใหม่ที่มีประสิทธิภาพ
- รู้จักวิธีการวางแผนการพัฒนา การแก้ปัญหาที่เกิดขึ้นในการพัฒนาโปรแกรม

1.5.2 ประโยชน์ต่อผู้นำโปรแกรมไปใช้งานจริง

- ผู้สร้างเว็บแอปพลิเคชันสามารถสร้างรายงานเพื่อการพิมพ์ผ่านเว็บแอปพลิเคชันได้อย่างสวยงาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ผู้สร้างเว็บแอปพลิเคชันสามารถสร้างรายงานเพื่อการพิมพ์ผ่านเว็บแอปพลิเคชันได้
อย่างสะดวกสบาย
- ผู้อ่านรายงานสามารถพิมพ์รายงานออกเครื่องพิมพ์ได้อย่างสวยงาม

1.6 ขั้นตอนของการศึกษา

เอกสารฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ ขอบเขตของการวิจัย ประโยชน์ที่คาดว่าจะได้รับ และขั้นตอนการศึกษา

บทที่ 2 กล่าวถึงหลักการที่ใช้ในการพัฒนาโปรแกรม ด้วยการวิเคราะห์ขั้นตอนการทำงานของเทคโนโลยีต่างๆ ที่สามารถที่จะสั่งพิมพ์รายงานผ่านเว็บแอปพลิเคชันในปัจจุบัน

บทที่ 3 กล่าวถึงการวิเคราะห์และออกแบบโปรแกรม

บทที่ 4 กล่าวถึงการสร้างโปรแกรม ซึ่งรวมถึงการตัดแปลง Library ที่มีอยู่ให้สามารถใช้งานเป็นภาษาไทยได้

บทที่ 5 กล่าวถึงการใช้งานโปรแกรม รวมทั้งการติดตั้งโปรแกรม

บทที่ 6 เป็นบทสรุปและข้อเสนอแนะ

บทที่ 2

หลักการที่ใช้ในการพัฒนาโปรแกรม

ในบทนี้จะกล่าวถึงทฤษฎีและหลักการต่างๆ ที่ใช้ในการพัฒนาโปรแกรม ซึ่งในการพัฒนาโปรแกรมนี้จะเริ่มต้นจากการศึกษาเทคโนโลยีเพื่อการพิมพ์รายงานที่มีอยู่แล้วในปัจจุบัน ซึ่งจากการวิเคราะห์โปรแกรมถึงขั้นตอนการทำงานต่างๆที่สามารถที่จะส่งพิมพ์รายงานผ่านเว็บแอปพลิเคชันในปัจจุบันนั้น มีความเป็นไปได้ในการพิมพ์รายงานผ่านเว็บแอปพลิเคชันอยู่ ทั้งหมด 3 วิธี คือ พิมพ์จากคำสั่งพิมพ์ของเว็บเบราว์เซอร์ หรือ พิมพ์จากไฟล์ PDF ที่ประมวลผลจากเว็บแอปพลิเคชัน หรือ พิมพ์จากโปรแกรม Crystal Report ซึ่งสามารถแสดงรายละเอียดเป็นข้อๆได้ดังต่อไปนี้

2.1 พิมพ์จากคำสั่งพิมพ์ของเว็บเบราว์เซอร์

คำสั่งพิมพ์ของเว็บเบราว์เซอร์มีอยู่แล้วในโปรแกรมประเภทเว็บเบราว์เซอร์ทุกค่ายเช่น Internet Explorer, Netscape, Opera ,... etc โดยมีขั้นตอนการทำงานเพื่อการพิมพ์รายงานคือ

- นักพัฒนาโปรแกรมประเภทเว็บแอปพลิเคชัน ทำหน้าจอบแสดงผลรายงานเป็นรูปแบบที่เว็บเบราว์เซอร์สามารถที่จะ Render ผลลัพธ์ออกมาได้ เช่น HTML หรือ JPG เป็นต้น
- ผู้ที่ต้องการพิมพ์รายงานผ่านเว็บแอปพลิเคชัน กดคำสั่งพิมพ์ของโปรแกรมเว็บเบราว์เซอร์ เพื่อสั่งให้เครื่องพิมพ์ทำการพิมพ์รายงานที่แสดงผ่านเว็บเบราว์เซอร์ ดังรูปที่ 2.1 ซึ่งมีข้อดีคือ ทำงานได้หลากหลายภาษา สะดวกในการพัฒนาเว็บแอปพลิเคชัน สามารถสั่งพิมพ์ได้จากทุกเว็บเบราว์เซอร์ สามารถใช้ร่วมกันได้กับทุกเว็บแอปพลิเคชัน สามารถใช้ร่วมกันได้กับทุกเว็บเซิร์ฟเวอร์ และ ไม่ต้องลงโปรแกรมพิเศษเพิ่มเติม แต่ก็มีข้อเสียคือไม่สามารถได้มาซึ่งความแน่นอน ของการตั้งค่าน้ำกระดาษดังรูปที่ 2.2 ทำให้ไม่สามารถที่จะได้งานพิมพ์ที่ต้องการให้พิมพ์ลงตามตำแหน่งต่างๆได้ทั้งหมด

Beyond eXtreme :: Computer Hardware PriceList :: 27/08/2003 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Refresh Home Search Favorites Media History Mail Print Edit Discuss FlashGet ICQ Pro

com/PriceList/price.cgi

[.. hdd ..]

:: HDD ::

		Size	Interface	More	Price	Update	Chg	Pos
		16M	USB	?	610	31/08/2003	=	=
		32M	USB	?	720 (700)	31/08/2003	▼	=
		64M	USB	?	930 (890)	31/08/2003	▼	=
Apacer	HandyDrive	128M	USB	?	1,550 (1,490)	31/08/2003	▼	=
Apacer	HandyDrive	256M	USB	?	2,550 (2,530)	31/08/2003	▼	=
Apacer	HandyDrive	512M	USB	?	5,650 (5,290)	31/08/2003	▼	=
Apacer	HandyDrive	1.0 G	USB	?	13,730	31/08/2003	=	=
Apacer	HandyDrive	32M	USB 2.0	?	1,290	10/08/2003	=	=
Apacer	HandyDrive	64M	USB 2.0	?	1,050	31/08/2003	=	=
Apacer	HandyDrive	128M	USB 2.0	?	1,700	31/08/2003	▼	=

Prints this page.

Start WEBAPP_AlterMidterm.D... Beyond eXtreme :: Co... 15:47

รูปที่ 2.1 การใช้งานคำสั่งพิมพ์ของเว็บเบราว์เซอร์

Print Preview

Print... Page 1 of 5 75% Help Close

Beyond eXtreme :: Computer Hardware PriceList :: 27/08/2003 Page 1 of 5

[.. hdd ..]

:: HDD ::

Brand	Model	Size	Interface	More	Price	Update	Ch
Apacer	HandyDrive	16M	USB	?	610	31/08/2003	=
Apacer	HandyDrive	32M	USB	?	720 (700)	31/08/2003	▼
Apacer	HandyDrive	64M	USB	?	930 (890)	31/08/2003	▼
Apacer	HandyDrive	128M	USB	?	1,550 (1,490)	31/08/2003	▼
Apacer	HandyDrive	256M	USB	?	2,550 (2,530)	31/08/2003	▼
Apacer	HandyDrive	512M	USB	?	5,650 (5,290)	31/08/2003	▼
Apacer	HandyDrive	1.0 G	USB	?	13,730	31/08/2003	=
Apacer	HandyDrive	32M	USB 2.0	?	1,290	10/08/2003	=
Apacer	HandyDrive	64M	USB 2.0	?	1,050	31/08/2003	=
Apacer	HandyDrive	128M	USB 2.0	?	1,700	31/08/2003	▼
Apacer	HandyDrive	256M	USB 2.0	?	2,700	31/08/2003	▼
Apacer	HandyDrive	512M	USB 2.0	?	8,390	31/08/2003	=
Apacer	HandyDrive	64M	USB	Bootable	990	31/08/2003	=
Apacer	HandyDrive	128M	USB	Bootable	1,790	31/08/2003	=
Apacer	HandyDrive	256M	USB	Bootable	2,810	31/08/2003	=
Billington	USB Drive	32M	USB	?	700	31/08/2003	=
Billington	USB Drive	64M	USB	?	1,700	31/08/2003	=

Start WEBAPP_AlterMidterm.D... Beyond eXtreme :: Co... 15:50

รูปที่ 2.2 รายงานก่อนพิมพ์ของเว็บเบราว์เซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 พิมพ์จากไฟล์ Portable Document Format (PDF)

ในทางคอมพิวเตอร์ PDF (มาจากคำเต็มว่า Portable Document Format) คือรูปแบบไฟล์ลักษณะหนึ่งพัฒนาโดยบริษัทอโดบีซิสเต็มส์จำกัด สำหรับการแสดงเอกสาร ที่สามารถใช้งานได้ ในทุกระบบปฏิบัติการ และยังคงลักษณะเอกสารเหมือนต้นฉบับ เอกสารในรูปแบบ PDF สามารถจัดเก็บ ตัวอักษร รูปภาพ รูปลายเส้น ในลักษณะเป็นหน้าหนังสือ ตั้งแต่ หนึ่งหน้า หรือหลายพันหน้าได้ในไฟล์เดียวกัน PDF เป็นมาตรฐานที่เปิดให้คนอื่นสามารถเขียน โปรแกรมมาทำงานร่วมกับ PDF ได้

การใช้งานไฟล์แบบ PDF เหมาะสมสำหรับงานที่การแสดงผลให้มีลักษณะเดียวกันกับต้นฉบับ ซึ่งแตกต่างกับการใช้งานรูปแบบอื่น เช่น HTML โดยการแสดงผลของ HTML จะแตกต่างกันออกไป ขึ้นอยู่กับโปรแกรมเบราว์เซอร์ที่ใช้ และจะแสดงผลต่างกัน ถ้าใช้คอมพิวเตอร์ต่างกัน โดย PDF มีข้อดีดังต่อไปนี้ (Portable Document Format. 2007)

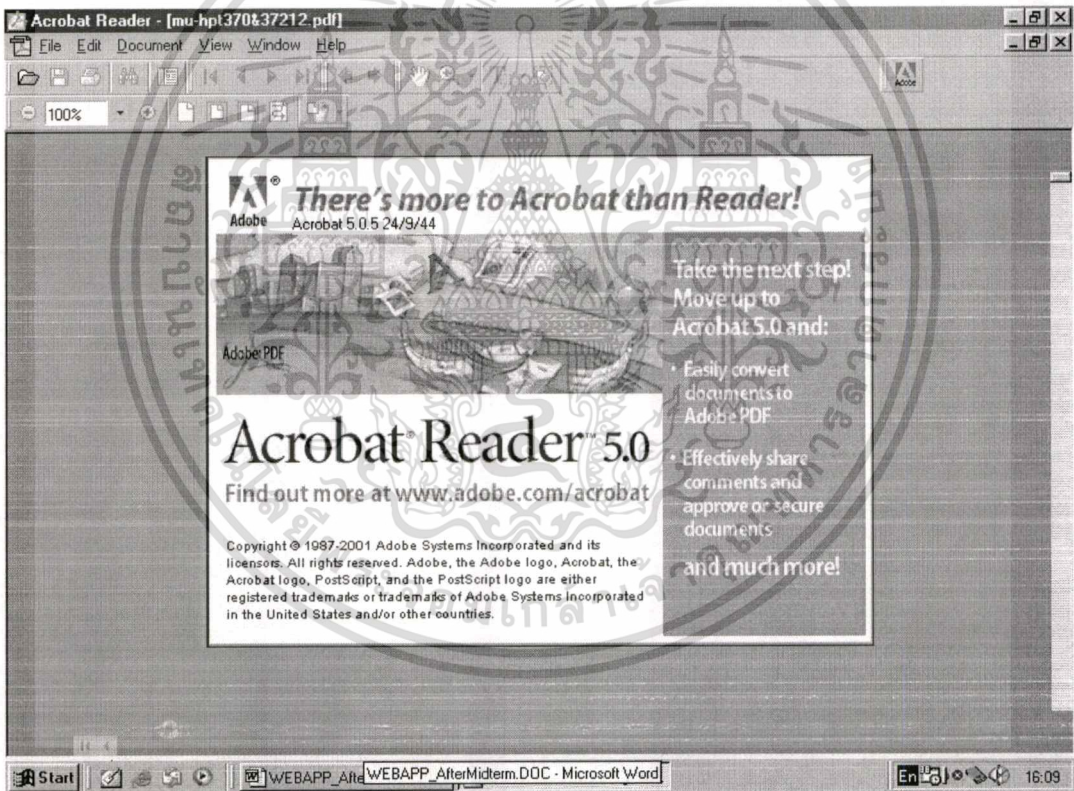
- ใช้ไฟล์ในการเก็บข้อมูลเพียงไฟล์เดียว (Single File) ไม่ว่าเอกสารจะประกอบไปด้วยข้อมูลหลายๆข้อมูล เช่น รูปภาพ ตัวอักษร แต่สุดท้ายก็จะเก็บข้อมูลรวมกันเป็นไฟล์เดียว ทำให้ลดความซับซ้อนในการใช้งาน และจัดเก็บไฟล์
- พร้อมใช้งาน (Already interpreted) เนื่องจากได้แปลงไฟล์ที่ได้จากโพสต์สคริปท์ออกมาแล้ว จึงทำให้ไม่ต้องแปลงอีก ทำให้เกิดความรวดเร็วในการใช้งาน
- เก็บวัตถุร่วมกับเอกสาร (Object Transparency) ทำให้สามารถจัดเก็บวัตถุ เช่น ไฟล์อื่นๆ เพื่อแนบไปกับเอกสาร PDF ได้
- มีฟอนต์สำรอง (Font substitution) เพื่อให้สามารถเปิดอ่านได้ทุกกรณี แม้แต่เครื่องของผู้ใช้ไม่มีฟอนต์ที่ตัวอักษรต้องการใช้ในการแสดงผล แต่ PDF ก็มี Embedding Font เพื่อให้แสดงผลได้อย่างถูกต้อง
- หน้าแต่ละหน้าเป็นอิสระต่อกัน (Independent Pages) ทำให้การแสดงผลของเอกสารในแต่ละหน้าไม่กระทบต่อการแสดงผลต่อหน้าถัดไป

ซึ่งลักษณะของไฟล์ PDF ไฟล์ PDF ประกอบด้วย 3 เทคโนโลยีหลัก เพื่อใช้ในการเก็บข้อมูลทั้งหมด ให้เป็นไฟล์ในรูปแบบ PDF ได้แก่

- เทคโนโลยีของ โพสต์สคริปท์ (PostScript) เฉพาะในการจัดวางรูปหน้า ตัวอักษร และรูปภาพ ซึ่งเป็นรูปแบบข้อมูลที่ใช้อ่านเท่านั้น ไม่จำเป็นต้องมีการประมวลผลที่ซับซ้อนอีกจึงไม่เปลืองทรัพยากรของเครื่องคอมพิวเตอร์ในระหว่างการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เทคโนโลยีการเก็บฟอนต์ (Font) ภายในเอกสาร เพื่อว่าในกรณีที่เครื่องที่ต้องการเปิดอ่านไม่มีฟอนต์ในการแสดงผล ก็ยังสามารถที่จะแสดงผลออกมาอย่างถูกต้องตามต้นฉบับที่แท้จริงได้
- เทคโนโลยีการบีบอัดข้อมูล (Compression) ทำให้ใช้เนื้อที่ในการจัดเก็บน้อยลง ซึ่งการบีบอัดข้อมูล มีความสำคัญในระบบการสื่อสารและจัดเก็บข้อมูล เนื่องจากทำให้เก็บหรือรับส่งข้อมูลได้มากขึ้น โดยใช้เนื้อที่เท่าเดิม (คำว่าเนื้อที่นี้ อาจจะเป็นเนื้อที่จัดเก็บข้อมูล หรือเนื้อที่ในช่องสัญญาณก็ได้) โดยการบีบอัดข้อมูลแบ่งได้เป็นสองประเภทใหญ่ ๆ ตามคุณภาพของข้อมูลที่ถูกบีบอัดแล้ว คือ การบีบอัดข้อมูลแบบไม่สูญเสียคุณภาพ (Lossless compression) และ การบีบอัดข้อมูลแบบสูญเสียคุณภาพ (Lossy compression)



รูปที่ 2.3 หน้าตาของโปรแกรม Acrobat Reader

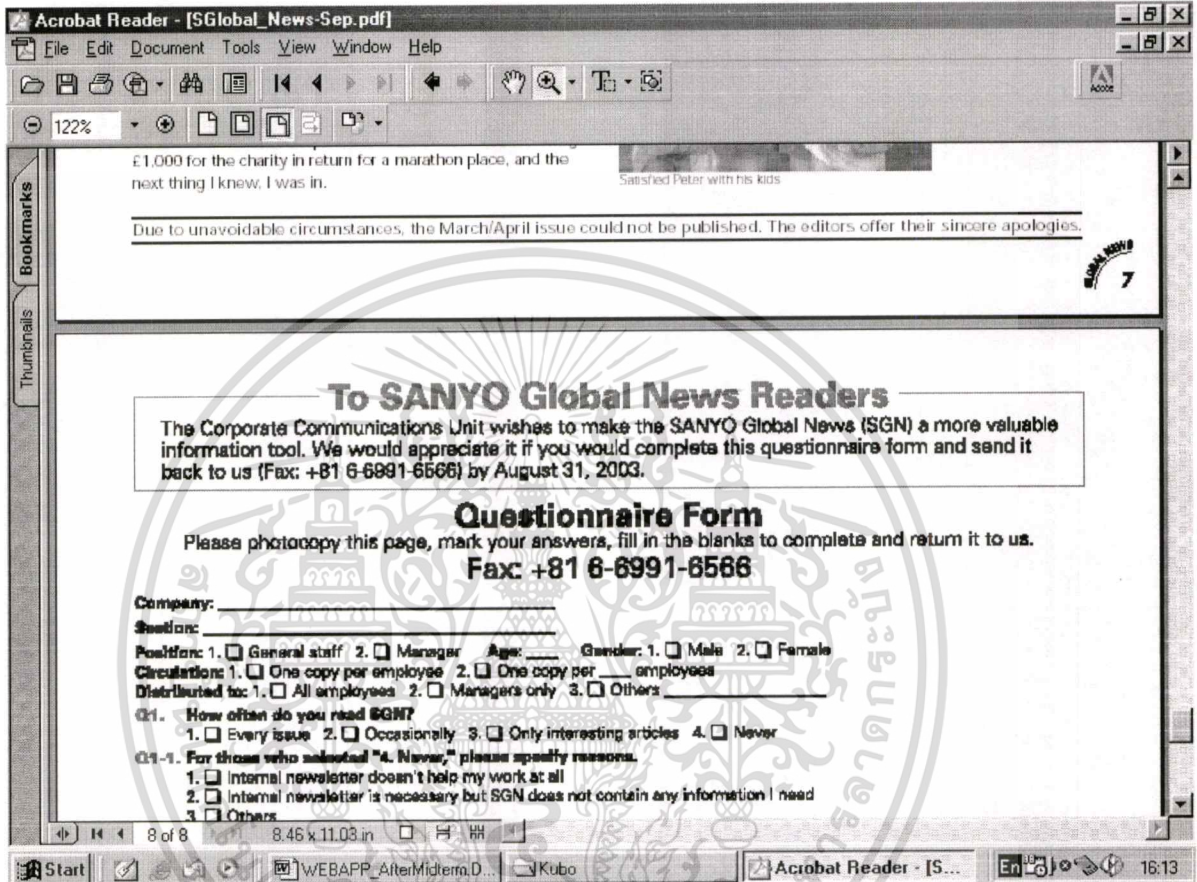
สำหรับขั้นตอนการพิมพ์รายงานด้วยไฟล์ PDF ผ่านเว็บแอปพลิเคชันมีดังต่อไปนี้

- นักพัฒนาโปรแกรมเว็บแอปพลิเคชัน ใช้ความสามารถของเว็บแอปพลิเคชัน ทำการสร้างไฟล์ PDF ขึ้นมา จากข้อมูลที่ต้องการให้ออกเป็นผลลัพธ์ เพื่อเป็นข้อมูลสำหรับพิมพ์ ซึ่งจำเป็นต้องเขียนโปรแกรมให้ทำการสร้างไฟล์ PDF เพื่อให้ผู้ที่ต้องการพิมพ์ดาวน์โหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เครื่องคอมพิวเตอร์ที่ต้องการพิมพ์รายงาน ทำการดาวน์โหลดไฟล์ PDF ที่เป็นรายงานที่ต้องการ แล้วก็สั่งพิมพ์ด้วยคำสั่งพิมพ์ ออกจากโปรแกรม Acrobat Reader ซึ่งเป็นฟรีแวร์ ดังรูปที่ 2.4



รูปที่ 2.4 รายงานที่สามารถพิมพ์ออกจาก Acrobat Reader

ซึ่งมีข้อดีของการพิมพ์จากไฟล์ PDF ที่ประมวลผลจาก Web Application คือ ได้มาซึ่งความแน่นอนของการตั้งค่าน้ำกระดาษ สามารถสั่งพิมพ์ได้จากทุกเว็บเบราว์เซอร์ สามารถใช้ร่วมกันได้กับทุกเว็บแอปพลิเคชัน และสามารถใช้ร่วมกันได้กับทุกเว็บเซิร์ฟเวอร์ แต่ก็มีข้อเสียคือ ไม่สะดวกในการพัฒนาเว็บแอปพลิเคชัน และ จำเป็นต้องลงโปรแกรมพิเศษเพิ่มเติม (Acrobat Reader ที่เครื่องที่ต้องการสั่งพิมพ์)

ซึ่งจากการวิเคราะห์เปรียบเทียบกับเทคโนโลยีอื่นๆ ในปัจจุบันแล้ว วิธีการพิมพ์รายงานด้วยไฟล์ PDF เป็นวิธีที่ดีที่สุด เพราะฉะนั้น ในการพัฒนาโปรแกรมนี้จึงเลือกใช้วิธีนี้ในการพัฒนาโปรแกรม

2.3 พิมพ์จากโปรแกรม Crystal Report

เป็นหนึ่งในโปรแกรมของค่ายไมโครซอฟท์ในปัจจุบัน ซึ่งสามารถออกรายงานผ่านเว็บ ได้อย่างง่ายดาย และสะดวกในการพัฒนาโปรแกรม จากรูปที่ 2.5 เป็นตัวอย่างของเว็บแอปพลิเคชัน ที่พัฒนาจาก ASP ทำหน้าที่ประมวลผลข้อมูลเพื่อส่งไปให้ Crystal Report เพื่อทำหน้าที่ออกรายงานผ่านเว็บ ทำให้เครื่องผู้ใช้งานสามารถสั่งพิมพ์รายงานได้ตามต้องการ

รูปที่ 2.5 หน้าจอที่ทำการรอร์รับค่าก่อนประมวลผล

และรูปที่ 2.6 จะแสดงรายงานที่เป็นผลลัพธ์จากโปรแกรม Crystal Report โดยแสดงผล ลัพท์ทางกายภาพที่ชัดเจน เพื่อให้ผู้ที่ต้องการพิมพ์รายงานได้รับงานพิมพ์ที่ต้องการได้เห็นก่อน ทำการพิมพ์

มีข้อดีของการพิมพ์รายงานผ่าน Crystal Report คือ ทำงานได้หลากหลายภาษา ได้มาซึ่ง ความแน่นอนของการตั้งค่าหน้ากระดาษ สามารถสั่งพิมพ์ได้จากทุกโปรแกรมเว็บเบราว์เซอร์ และ สะดวกในการพัฒนาเว็บแอปพลิเคชัน (ชนพล ฉันทวีชัย, 2546)

Address http://s-hq-sqj01/kaizen/KZDept.asp

Preview
SSC

บริษัท สยามเซมเพอร์เมต จำกัด

KAIZEN แยกตามแผนก ตั้งแต่วันที่ 1/3/2546 ถึง 10/3/2546

หมายเลข Kaizen	Subject	SuggestNo	SuggestName
บริษัท สยามเซมเพอร์เมต จำกัด			
แผนก ผลิตภัณฑ์			
K1SSC00182	SLOPE เรือโบ tank 5 แบบ tank dip	033136004	เพ็ญศรี ลิขิต
K1SSC00035	*มันไต้ขึ้นค้อน steam	006058003	ฉวรงค์ รณิง
K1SSC00040	*ฉิลโร้ตริงเดี่ยว	006145123	จิรพันธ์ ไชยะ
K1SSC00019	*รถเข็นลิกตะกรีน	014128001	สุพิศ สอนแสง
K1SSC00017	*ลดปัญหาการมีดศคพื้น	001202060	ระนอง พลรัตน
K1SSC00181	มีมแทรกประตังขึ้นป้องกัน ได้ฉายาไร	001202101	ภิญโญ สามารถ
K1SSC00020	*ใช้เฟืองขับประตังลอกลิ้งลิ้ม	001202133	เดชา ทังสา
K1SSC00180	พลังงานอาคาร	001202175	ฉฉฉฉ ฉฉฉฉ
K1SSC00008	*ทำ former มีตรนเครื่องรีด	301340011	วิไล ทุมจำ
K1SSC00039	*รับเฟอร์ tank dip	302345050	ศวิศล แก้วเชิด
K1SSC00038	*เดินพอนเมนไล Tank mix	302345077	พงษ์วิสิทธิ์ คำทะลา

รูปที่ 2.6 รายงานที่พร้อมสำหรับการพิมพ์ผ่านเว็บ จากโปรแกรม Crystal Report

แต่ก็มีข้อเสียของการพิมพ์รายงานผ่าน Crystal Report คือ ไม่สามารถใช้ร่วมกันได้กับทุกเว็บเซิร์ฟเวอร์ เนื่องจากสามารถใช้ได้กับ Microsoft Internet Information Services เท่านั้น ซึ่งไม่สามารถใช้ร่วมกันได้กับทุกเว็บแอปพลิเคชัน (ในทางปฏิบัติเพื่อการแปลผลที่รวดเร็วจำเป็นต้องใช้ร่วมกับ ASP เท่านั้น) และ จำเป็นต้องลงโปรแกรมพิเศษเพิ่มเติมซึ่งก็คือ Crystal Report ที่เครื่องเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์และออกแบบโปรแกรม

หลังจากที่ได้ทำการศึกษาและเก็บข้อมูลการทำงานของเทคโนโลยีที่มีอยู่แล้วในปัจจุบันอย่างละเอียดแล้ว ทำให้ได้สามารถวิเคราะห์ถึงความต้องการและออกแบบโปรแกรมใหม่ได้ดังนี้

3.1 ความต้องการของโปรแกรม

โปรแกรมการสร้างเอกสารการพิมพ์รายงานที่สวยงามจากเว็บแอปพลิเคชันที่จะทำการพัฒนาขึ้นนี้ ควรตอบสนองความต้องการของโปรแกรม ตามต้องการของผู้ใช้ ซึ่งมีรายละเอียดดังต่อไปนี้

- โปรแกรมต้องสามารถสร้างรายงานแบบ โดยเพื่อความสะดวกในการใช้งาน ควรทำการแปลงผลลัพธ์ของ HTML ไปเป็นรายงานได้ทันที
- ผลลัพธ์จากการประมวลผล ต้องสามารถเรียกใช้ผ่านเว็บเบราว์เซอร์ได้
- โปรแกรมต้องใช้งานง่าย ไม่ซับซ้อนจนเกินไป
- โปรแกรมต้องใช้เวลาประมวลผลรวดเร็ว

Company Logo Here

Company Text Logo
55 EST, Company Address
Montreal, Quebec H1H H2H
Tel 123-1234 Fax 123-1234

Invoice No: INV000101
TPS 123456789
TVQ 1234567890

Client Name: John Little
Address: _____
City: _____ Prov: _____ Zip: _____
Tele: 123-4567

INVOICE
Date: Oct 18, 2002
Staff ID: 2
Staff Name: Michael
Dept: _____

Qty	Description	Unit Price	Total
2	Company Fan	15.00	30.00

Payment: Cash
Sign: _____

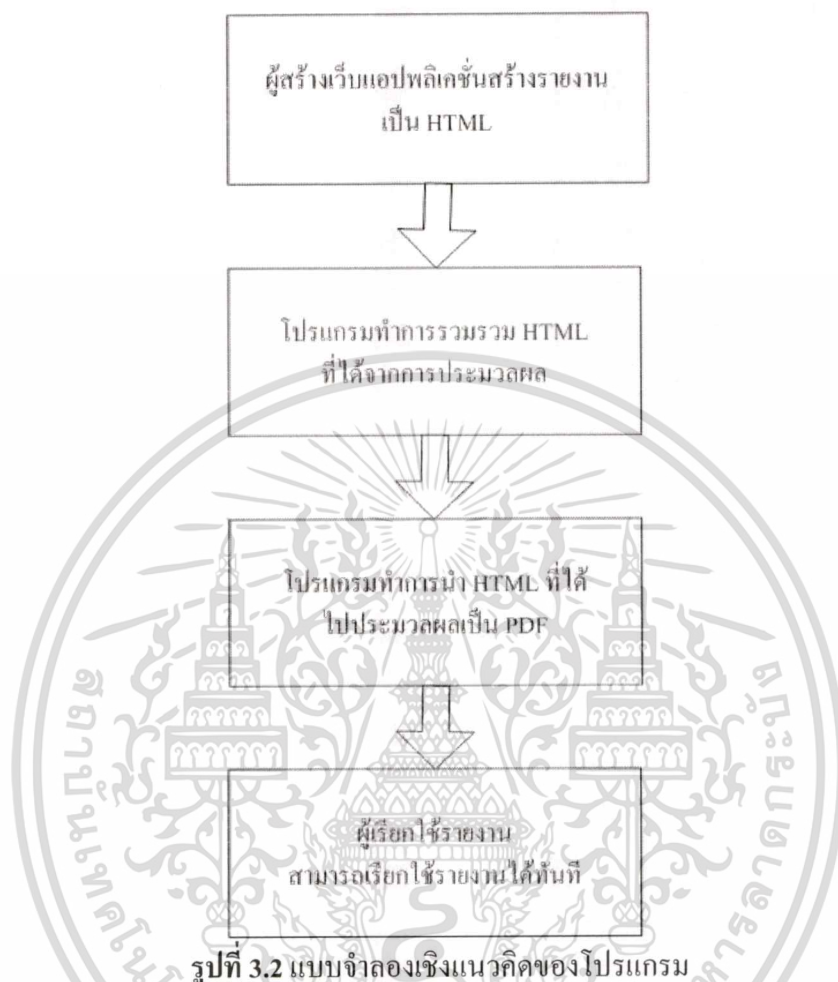
SubTotal \$ 30.00
Shipping \$ 0.00
Taxes GST 7.0% \$ 2.10
QST 7.5% \$ 2.41
Total \$ 34.51
Deposit \$ 0.00
Balance \$ 34.51

- Aucun remboursement en argent. - Des frais de remise en inventaire de 20% seront appliqués sur toute marchandise retournée approuvée. (en dedans de 7 jours suivant la date de réception) - Toutes les marchandises demeurent la propriété de Infotech jusqu'à leur paiement en entier. - Tout retour ou service de garantie doit être accompagné d'une copie de la facture d'achat. - Aucun retour de marchandise est accepté après 15 jours. - Les comptes passés dus seront sujets a des frais d'intérêt de 1.5% par mois. - Un mois de garantie pour CPU. - Une an garantie dans les systems + deux ans de main d'oeuvre pour les systems. - Une an garantie dans les pieces.
- No cash refund. - 20% restocking charge will be applied on all approved returns within 7 days. - All merchandise title remains with Infotech until full payment has been made. - All returns or warranty services must be accompanied by the sales invoices. - No return for goods purchased over 15 days. - Overdue accounts subject to monthly interest of 1.5%. one month guarantee for all CPU. - One year guarantee and 2 years of labor for all system. - One-year guarantee for all part.

เอกสารนี้รูปที่ 3.1 ตัวอย่างแบบฟอร์มที่ต้องพิมพ์ลงตำแหน่งที่ถูกต้อง (Inventory Control Software, 2007) การค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 แบบจำลองเชิงแนวคิดในการใช้งานโปรแกรม



จากรูปที่ 3.2 มีรายละเอียดในแต่ละขั้นตอนดังนี้

- ผู้สร้างเว็บแอปพลิเคชัน ทำการสร้างรายงานเพื่อให้ผลลัพธ์ออกมาเป็น HTML เพื่อใช้ในการแสดงผลเหมือนปรกติ ซึ่งแสดงผลทางหน้าจอคอมพิวเตอร์ด้วยโปรแกรมประเภทเว็บเบราว์เซอร์ทั่วไป
- ผู้สร้างเว็บแอปพลิเคชันทำการติดตั้งโปรแกรมเพื่อให้ผู้ใช้งานสามารถเรียกใช้เว็บแอปพลิเคชันผ่าน โปรแกรม ซึ่งได้ติดตั้งไว้ในพื้นที่เดียวกับเว็บแอปพลิเคชันบนเซิร์ฟเวอร์ โปรแกรมนี้จะทำงานโดยการครอบคลุมการทำงานของเว็บแอปพลิเคชัน ซึ่งจะทำงาน โดยการ ส่งต่อผลลัพธ์ที่ได้จากการประมวลผลของเว็บแอปพลิเคชันส่งต่อไปให้โปรแกรมนำไปประมวลผลต่อไป
- โปรแกรมทำการนำข้อมูลที่ได้รับ มาทำการอ่านและประมวลผลเพื่อแปลงเป็นไฟล์ PDF เพื่อให้ผู้ใช้งานเว็บแอปพลิเคชันสามารถใช้ในการพิมพ์รายงานได้อย่างสวยงาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ผู้ใช้งานเว็บแอปพลิเคชัน เรียกดูรายงานผ่านเว็บเบราว์เซอร์ หรือเรียกรายงานออกมาพิมพ์ได้ทันที

3.3 ขอบเขตของโปรแกรม

จากการศึกษาโปรแกรมที่มีอยู่แล้วในปัจจุบัน และความต้องการของโปรแกรมเพื่อพิมพ์รายงานผ่านเว็บแอปพลิเคชัน สามารถสรุปได้ว่า การพิมพ์รายงานที่สวยงามผ่านเว็บแอปพลิเคชันนั้น ควรจะพิมพ์ผ่านไฟล์ PDF และจากการค้นคว้าหาวิธีการที่จะทำให้สามารถประมวลผลเป็นไฟล์ PDF ได้ในปัจจุบันนั้น จึงได้ทราบว่า มี Library ประเภทนี้อยู่ ซึ่งก็คือ FPDF และ HTML2FPDF ซึ่งเป็น Library ที่อยู่ในกลุ่ม Open Source ที่ถูกพัฒนาขึ้น โดยชาวต่างชาติ แต่ Library เหล่านี้ ไม่สามารถที่จะทำงานร่วมกับภาษาไทยได้ เพราะฉะนั้นขอบเขตของโปรแกรมที่จะทำการพัฒนานี้จึงมีขอบเขตงานทั้งหมดดังนี้

- ดัดแปลง Library ของ FPDF ให้สามารถใช้งานร่วมกับภาษาไทยได้
- ดัดแปลง Library ของ HTML2FPDF ให้สามารถใช้งานร่วมกับภาษาไทยได้
- ทำการสร้าง Font ภาษาไทยเพื่อใช้ใน โปรแกรม
- ทำการสร้างโปรแกรมเพื่อดึงข้อมูล HTML จากเว็บแอปพลิเคชัน โดยอัตโนมัติ เพื่อส่งไปประมวลผลต่อด้วย Library ของ FPDF และ HTML2FPDF เป็นเอกสาร PDF เพื่อใช้ในการพิมพ์รายงานเป็นภาษาไทยได้

3.4 แบบจำลองทางกายภาพของการพัฒนาโปรแกรม

โปรแกรมจะประกอบไปด้วยขั้นตอนหลายขั้นตอน ซึ่งเริ่มจากรวมรวมข้อมูล HTML แล้วส่งต่อไปยัง Library ที่ชื่อว่า HTML2FPDF ซึ่ง Library นี้ จะทำหน้าที่แปลงข้อมูล HTML ไปเป็นฟังก์ชันต่างๆ ซึ่งฟังก์ชันเหล่านี้จะถูกเรียกเพื่อใช้ต่อใน Library ที่ชื่อว่า FPDF และจากนั้น FPDF จะทำการประมวลผลต่อไปเป็นไฟล์ PDF และได้ผลลัพธ์เพื่อให้ผู้ใช้งานโปรแกรมนำไปใช้ในการพิมพ์รายงานออกเครื่องพิมพ์ ซึ่งสามารถแสดงได้ด้วยแบบจำลองทางกายภาพของการพัฒนาโปรแกรมดังรูปที่ 3.3



รูปที่ 3.3 แบบจำลองทางกายภาพของการพัฒนาโปรแกรม

โดยการทำงานทั้งหมดนี้จะทำงานครอบอยู่บนเว็บแอปพลิเคชัน เพื่อที่จะทำให้สามารถที่จะดึงข้อมูล HTML จากเว็บแอปพลิเคชัน ได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การสร้างโปรแกรม

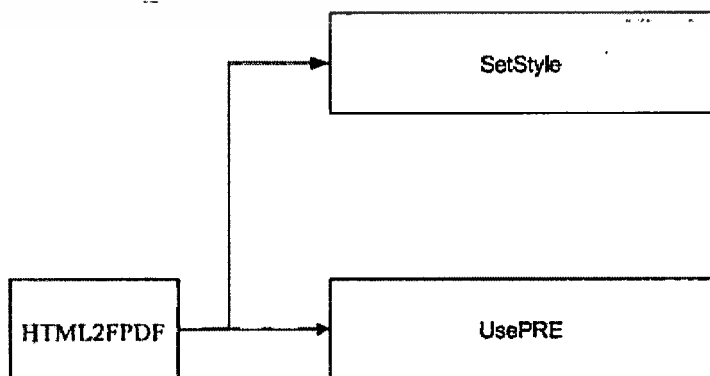
หลังจากที่วิเคราะห์และออกแบบโปรแกรมไว้แล้ว จึงเริ่มทำการสร้างโปรแกรมเพื่อทำการพิมพ์รายงานที่สวยงามผ่านเว็บแอปพลิเคชัน ซึ่งประกอบไปด้วย 5 ส่วน คือ

- ศึกษาการทำงานของ Library HTML2FPDF
- ดัดแปลง Library HTML2FPDF เพื่อให้สามารถทำงานร่วมกับภาษาไทยได้
- ศึกษาการทำงานของ Library FPDF
- ดัดแปลง Library FPDF เพื่อให้สามารถทำงานร่วมกับภาษาไทยได้ และสามารถทำงานกับ Font ภาษาไทยที่เป็นที่นิยมได้
- สร้างโปรแกรมเพื่อเพื่อประมวลผลจาก HTML ไปเป็น PDF โดยอัตโนมัติ

โดยภาพรวมการทำงานของโปรแกรมคือ จะเริ่มจากการทำความเข้าใจ Library HTML2FPDF และ FPDF จากนั้นนำมาดัดแปลงแก้ไขให้สามารถใช้กับภาษาไทยได้ หลังจากนั้นจึงทำการเพิ่ม Font ที่นิยมใช้ในการพิมพ์รายงานภาษาไทย และสร้างโปรแกรมเพื่อดึงข้อมูล HTML จากเว็บแอปพลิเคชันเพื่อนำไปประมวลผลเป็น PDF ซึ่งในแต่ละส่วนมีรายละเอียดดังนี้

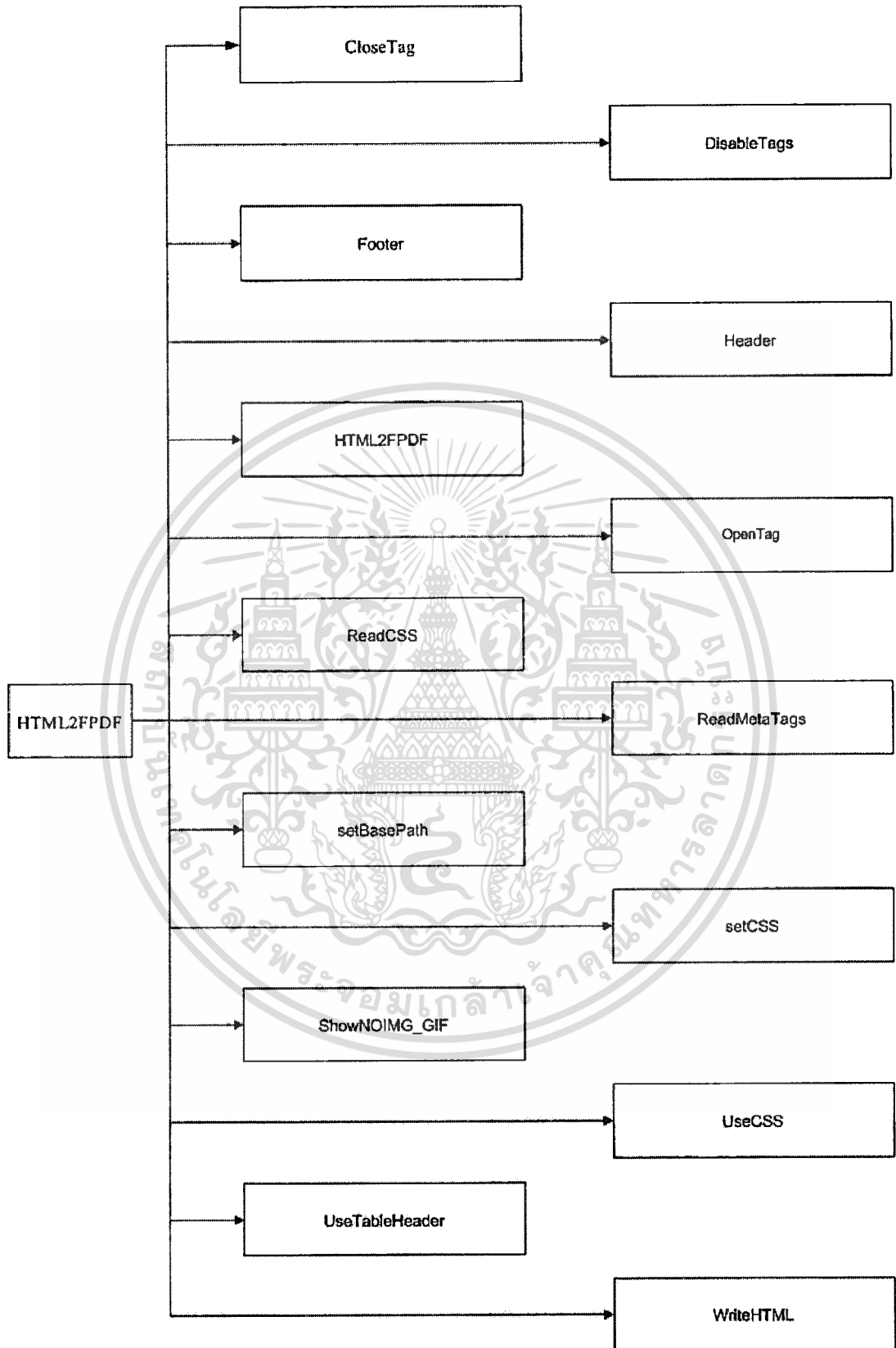
4.1 ศึกษาการทำงานของ Library HTML2FPDF

เริ่มทำการศึกษาทุกฟังก์ชันที่มีเพื่อทำความเข้าใจหลักการทำงาน ขั้นตอนการทำงานในแต่ละฟังก์ชันของ Library HTML2FPDF โดยสามารถแสดงเป็นแผนภาพได้ดังนี้ (HTML2FPDF Library, 2007)



รูปที่ 4.1 แผนภาพฟังก์ชันของ Library HTML2FPDF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสามารถแสดงหน้าที่ของแต่ละฟังก์ชันใน Library HTML2FPDF ได้ดังนี้
(HTML2FPDF Library. 2007)

ตารางที่ 4.1 ฟังก์ชันภายใน Library HTML2FPDF

	ชื่อฟังก์ชัน	ทำหน้าที่
1	CloseTag	เป็นฟังก์ชันทำหน้าที่แยกข้อมูล HTML TAG ต่างๆที่เป็น TAG ปิด เช่น </p> แยกออกมาเพื่อใช้ในการแปลงเป็นฟังก์ชันเพื่อใช้กับ Library FPDF ต่อไป
2	DisableTags	เป็นฟังก์ชันทำหน้าที่กำหนดว่า HTML ใดจะถูกใช้ หรือไม่ถูกใช้ในระหว่างการแปลงเป็นไฟล์ PDF
3	Footer	เป็นฟังก์ชันที่ทำหน้าที่เขียน HEADER ลงบนไฟล์ PDF
4	Header	เป็นฟังก์ชันที่ทำหน้าที่เขียน HEADER ลงบนไฟล์ PDF
5	HTML2FPDF	เป็นฟังก์ชันที่ทำหน้าที่กำหนดค่าตัวแปร เพื่อให้ฟังก์ชันอื่นๆใน Library เรียกใช้ต่อไป เช่น ขนาดกระดาษ ระยะห่าง Margin ซ้าย ขวา บน ล่าง ของหน้ากระดาษ
6	OpenTag	เป็นฟังก์ชันทำหน้าที่แยกข้อมูล HTML TAG ต่างๆที่เป็น TAG เปิด เช่น <p> แยกออกมาเพื่อใช้ในการแปลงเป็นฟังก์ชันเพื่อใช้กับ Library FPDF ต่อไป
7	ReadCSS	เป็นฟังก์ชันทำหน้าที่อ่าน CSS ที่อยู่ใน HTML เพื่อนำมาใช้ร่วมในการแปลงเป็นไฟล์ PDF
8	ReadMetaTags	เป็นฟังก์ชันทำหน้าที่อ่าน HTML TAG ในส่วนของ <Meta> เพื่อนำข้อมูลใน Meta ไปใช้เป็นข้อมูลอ้างอิงในไฟล์ PDF
9	SetBasePath	เป็นฟังก์ชันที่ทำหน้าที่กำหนด Path อ้างอิงของ Library ที่ใช้ในการเรียกซ้ำหลังจากการประมวลผลเสร็จสิ้น
10	SetCSS	เป็นฟังก์ชันทำหน้าที่นำข้อมูล CSS ที่อยู่ใน HTML มาใช้ในการแปลงเป็นไฟล์ PDF
11	SetStyle	เป็นฟังก์ชันทำหน้าที่นำข้อมูล Syle ที่อยู่ใน HTML มาใช้ในการแปลงเป็นไฟล์ PDF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 (ต่อ)

	ชื่อฟังก์ชัน	ทำหน้าที่
12	ShowNOIMG_GIF	เป็นฟังก์ชันที่ทำหน้าที่แสดงรูปภาพทดแทนเมื่อรูปภาพใน HTML ไม่สามารถถูกเรียกขึ้นมาแสดงผลใน PDF
13	UseCSS	เป็นฟังก์ชันที่ทำหน้าที่กำหนดให้ Library เลือกที่จะอ่านข้อมูล CSS ใน HTML หรือไม่
14	UsePRE	เป็นฟังก์ชันที่ทำหน้าที่กำหนดให้ใช้งาน HTML <PRE> หรือไม่
15	UseTableHeader	เป็นฟังก์ชันที่ทำหน้าที่กำหนดให้ Header แสดงในทุกๆหน้าหรือไม่
16	WriteHTML	เป็นฟังก์ชันทำหน้าที่แปลง HTML ซึ่งเป็นมาตรฐานที่มีหลายมาตรฐานให้ไปเป็น XHTML ซึ่งมีมาตรฐานมากกว่าเพื่อใช้ XHTML ในการสร้างฟังก์ชันเป็นไฟล์ PDF ต่อไป

4.2 ดัดแปลง Library HTML2FPDF เพื่อให้สามารถทำงานร่วมกับภาษาไทยได้

จากการศึกษาฟังก์ชันที่มีอยู่ใน Library HTML2FPDF เพื่อทำความเข้าใจหลักการทำงาน และ ขั้นตอนการทำงานของ Library นี้ ทำให้ทราบได้ว่า เหตุที่ทำให้ Library นี้ไม่สามารถทำงานร่วมกับภาษาไทยได้คือ ในฟังก์ชัน WriteHTML ได้มีการกำหนดการ Encoding ของข้อมูลไว้เป็น ISO-8859-1 ซึ่งเป็น Encoding สำหรับภาษาอังกฤษ เพราะฉะนั้นเพื่อให้ Library สามารถทำงานเป็นภาษาไทยได้นั้น จำเป็นต้องใช้ Encoding ที่เป็นภาษาไทย ซึ่งก็คือ ISO-8859-11

ISO-8859-1																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENO	ACK	BEL	BS	TAB	LF	VI	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	ES	GS	RS	US
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8x	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
9x	QCS	PU1	PU2	STS	CGH	MW	SPA	EPA	SOS	SGCI	SCJ	CSJ	ST	QSC	PM	APC
Ax	NBSP	ı	ç	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY	®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

รูปที่ 4.2 แผนภาพรหัส Encoding ISO-8859-1 (Chars and Strs. 2007)

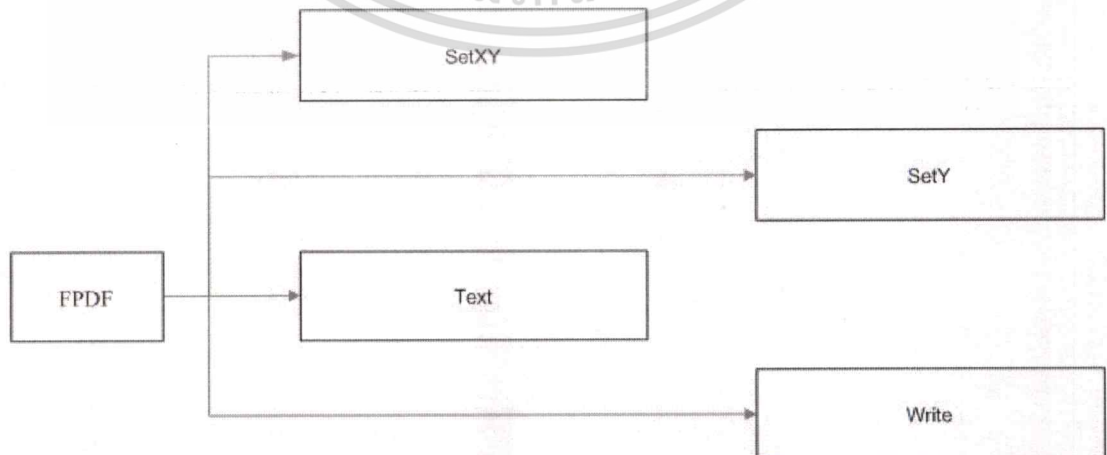
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ISO/IEC 8859-11																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	unused															
1x																
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8x	unused															
9x																
Ax	NBSP	ก	ข	ฃ	ค	ฅ	ง	ง	จ	ฉ	ช	ฌ	ญ	ฎ	ฏ	
Bx	ร	ท	ฒ	ณ	ด	ด	ถ	ท	ธ	น	บ	ป	ฝ	ฝ	ฟ	ฟ
Cx	ภ	ม	ย	ร	ฤ	ล	ภ	ว	ศ	ษ	ห	ฬ	อ	ฮ	ฯ	
Dx	ั	ำ	า	ิ	ี	ึ	ุ	ู	เ	แ	อ	ใ	ไ	ใ	ใ	อ
Ex	เ	แ	อ	ใ	ไ	ใ	ใ	ใ	ใ	ใ	ใ	ใ	ใ	ใ	ใ	อ
Fx	๐	๑	๒	๓	๔	๕	๖	๗	๘	๙	๐	๑	๒	๓	๔	๕

รูปที่ 4.3 แผนภาพรหัส Encoding ISO-8859-11 (Non-ASCII Characters. 2007)

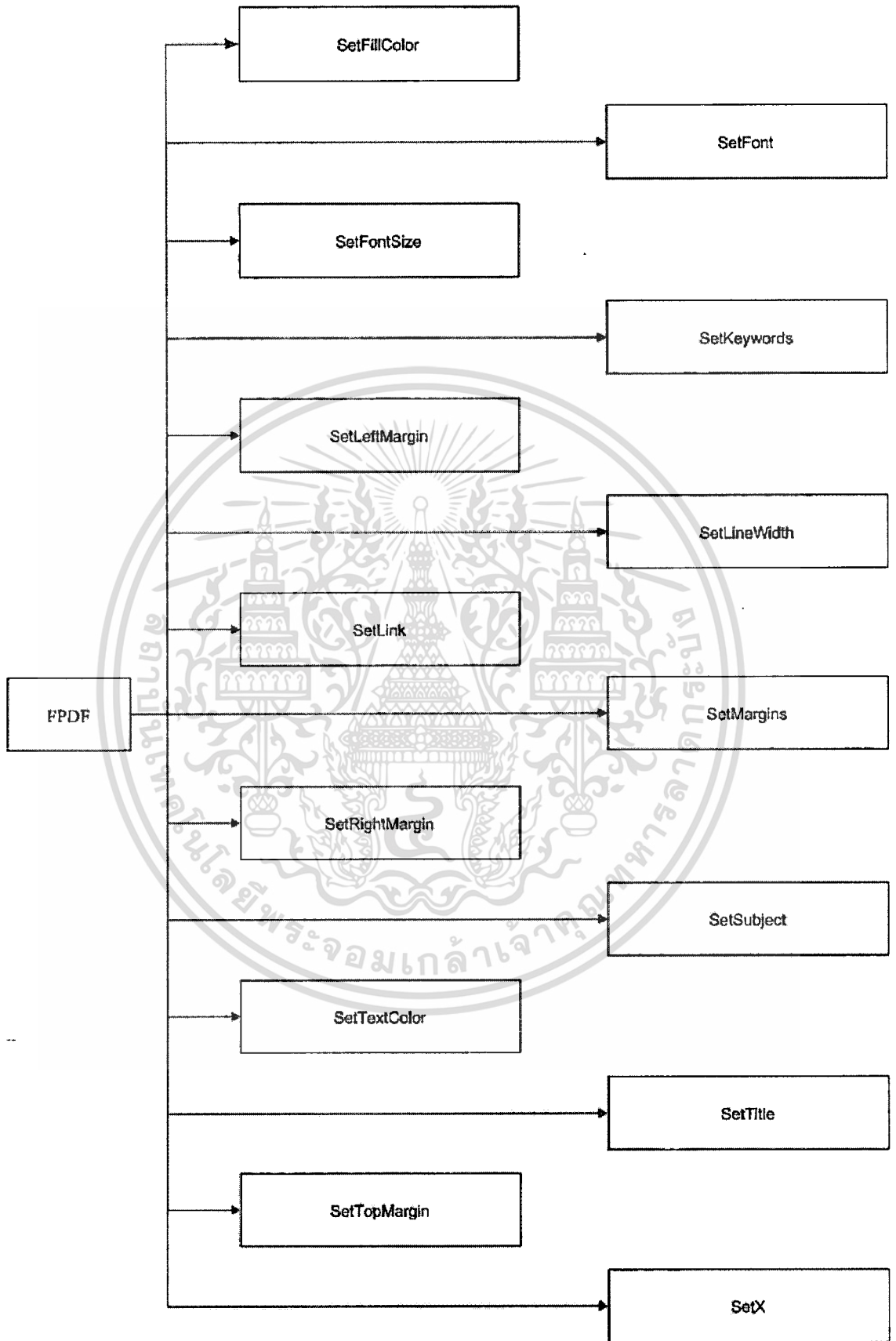
4.3 ศึกษาการทำงานของ Library FPDF

เริ่มทำการศึกษาดูฟังก์ชันที่มีเพื่อทำความเข้าใจหลักการการทำงาน ขั้นตอนการทำงานในแต่ละฟังก์ชันของ Library FPDF โดยสามารถแสดงเป็นแผนภาพได้ดังนี้ (FPDF Library. 2007)



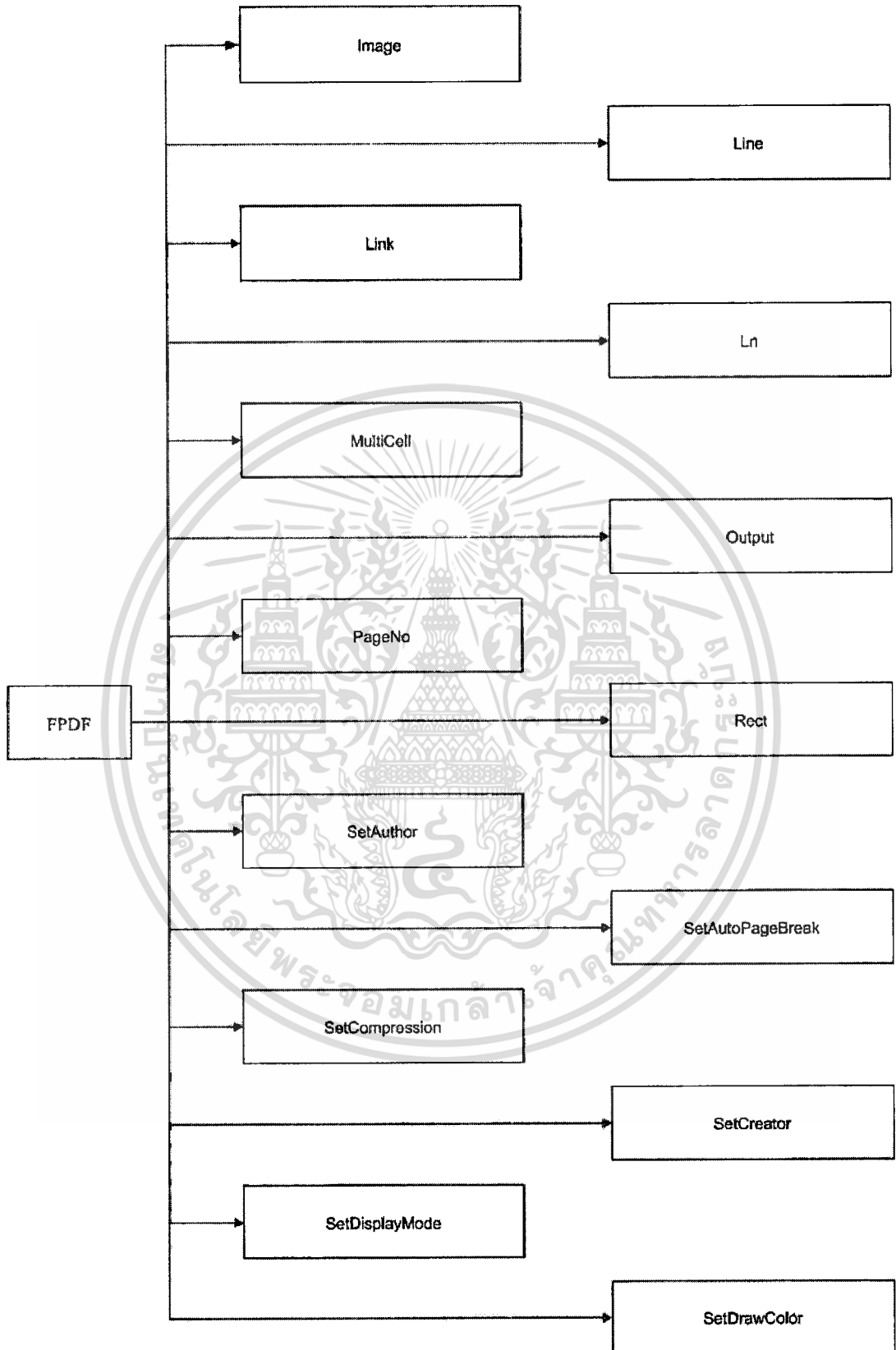
รูปที่ 4.4 แผนภาพฟังก์ชันของ Library FPFD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



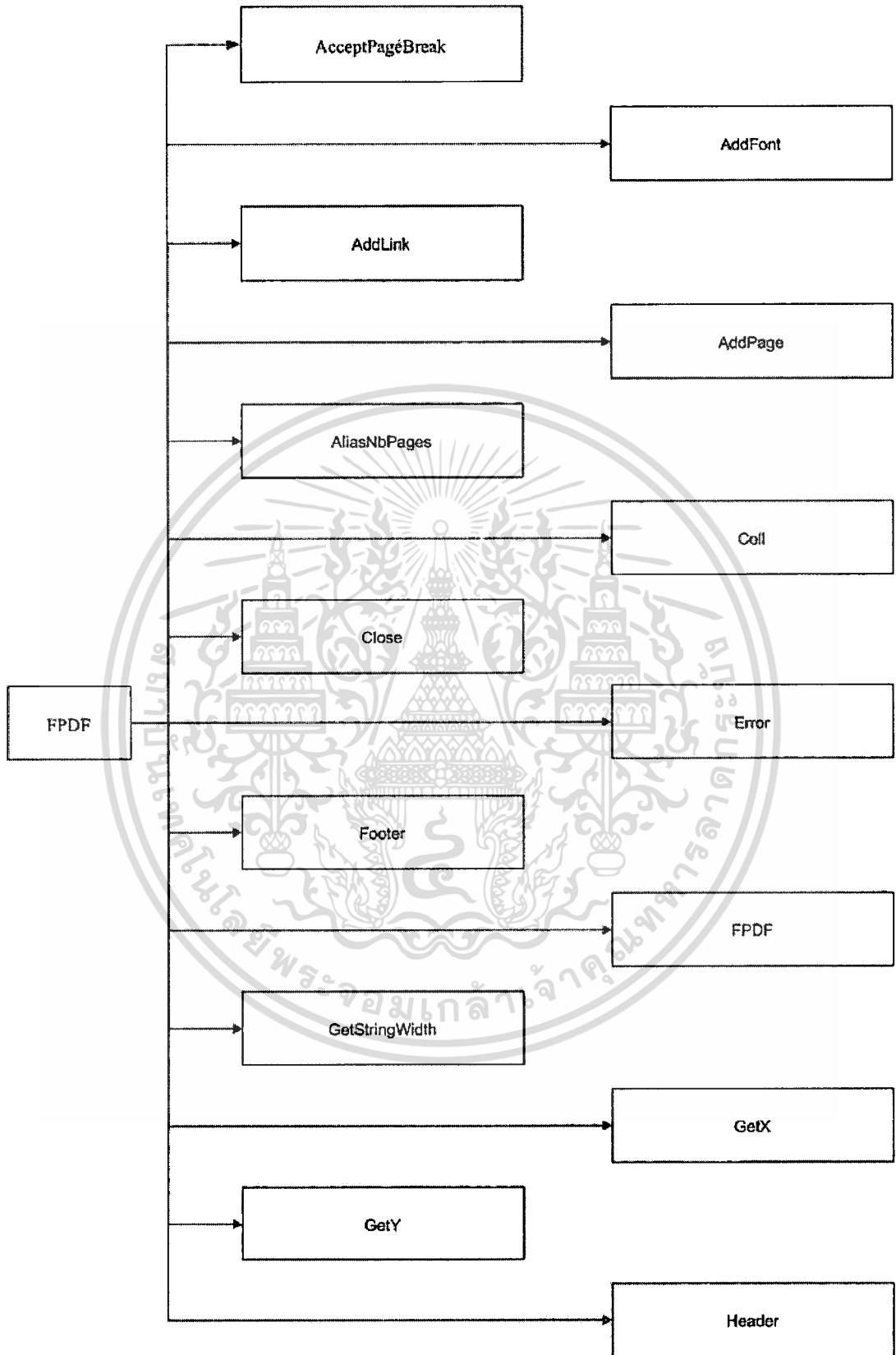
รูปที่ 4.4 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสามารถแสดงหน้าที่ของแต่ละฟังก์ชันใน Library FPDF ได้ดังนี้ (FPDF Library.

2007)

ตารางที่ 4.2 ฟังก์ชันภายใน Library FPDF

	ชื่อฟังก์ชัน	ทำหน้าที่
1	AcceptPageBreak	เป็นฟังก์ชันที่จะทำงานร่วมกับฟังก์ชัน SetAutoPageBreak() เมื่อต้องการให้โปรแกรมแทรกข้อมูลจบบรรทัดทันทีลงไปภายในเอกสารในกรณีที่ต้องการให้โปรแกรมทำการแทรกข้อมูลจบบรรทัดโดยอัตโนมัติ
2	AddFont	เป็นฟังก์ชันเพื่อเพิ่ม Font เข้าไปให้โปรแกรมพร้อมใช้งานก่อนการประมวลผล โดยก่อนที่จะทำการเรียกฟังก์ชันนี้ ต้องมีการสร้าง Font ให้พร้อมใช้งานด้วยโปรแกรม makefont.php ก่อน
3	AddLink	เป็นฟังก์ชันสร้าง Link เพื่อใช้ภายในเอกสาร PDF โดยเมื่อมีการ AddLink เข้าไปแล้ว จุดที่ถูก Addlink จะเป็นข้อความที่อำนวยความสะดวกให้ผู้อ่านเอกสารสามารถที่จะกดแล้ว โปรแกรม Acrobat Reader จะนำไปยังจุดที่ link สั่ง ซึ่งจำเป็นต้องใช้ฟังก์ชัน setlink() ก่อน เพื่อที่จะกำหนดว่าจุดใดจะเป็นจุดที่ต้องการให้ link นั้นข้ามไปถึง
4	AddPage	เป็นฟังก์ชันขึ้นหน้าเอกสารใหม่ โดยเมื่อเรียกใช้ฟังก์ชันนี้แล้ว ฟังก์ชัน footer() และ ฟังก์ชัน header() ก็จะถูกเรียกอัตโนมัติ และรูปแบบ Font ก่อนการเรียกฟังก์ชันนี้ก็จะถูกนำมาใช้ต่อโดยไม่ต้องเรียกใหม่
5	AliasNbPages	เป็นฟังก์ชันเพิ่มข้อมูลชื่อปลอมเข้าไปในเอกสาร เพื่อที่จะใช้อ้างอิงหมายเลขหน้าเอกสารนอกเหนือจากหมายเลขหน้าที่แท้จริง
6	Cell	เป็นฟังก์ชันใส่ข้อมูลตัวอักษรลงในตำแหน่งในหน้าเอกสาร โดยอ้างอิงจากจุดท้ายสุดของ cell ก่อนหน้า หรือถ้าไม่มีสิ้นจากจุดบนซ้ายสุดของหน้าเอกสาร แต่ถ้ามีการใช้งานฟังก์ชัน AutoPageBreak ก่อนหน้านี้ ฟังก์ชันนี้ก็จะทำการทำงานหลังจาก AutoPageBreak ซึ่งเทียบเท่ากับการขึ้นหน้าเอกสารใหม่
7	Close	เป็นฟังก์ชันสั่งให้จบเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 (ต่อ)

	ชื่อฟังก์ชัน	ทำหน้าที่
8	Error	เป็นฟังก์ชันที่จะทำงานก็ต่อเมื่อเกิดข้อผิดพลาดขึ้นในระหว่างการประมวลผล โดยสามารถสร้างข้อความรายงานผลข้อผิดพลาดผ่าน msg ได้เอง
9	Footer	เป็นฟังก์ชันสร้าง footer ให้กับเอกสาร โดยฟังก์ชันนี้จะถูกเรียกอัตโนมัติเมื่อมีการขึ้นหน้าใหม่ด้วยฟังก์ชัน AddPage() หรือ จบเอกสารด้วยฟังก์ชัน Close()
10	FPDF	เป็นฟังก์ชันเพื่อกำหนดค่าต่างๆจะใช้ต่อไปในฟังก์ชันอื่นๆ เช่น ขนาดกระดาษ รูปแบบการพิมพ์
11	GetStringWidth	เป็นฟังก์ชันอำนวยความสะดวกเพื่อนำผลลัพธ์ที่ได้จากฟังก์ชันนี้ไปใช้ต่อ โดยผลลัพธ์คือความกว้างของข้อความ s ที่ต้องการทราบ
12	GetX	เป็นฟังก์ชันอำนวยความสะดวกเพื่อนำผลลัพธ์ที่ได้จากฟังก์ชันนี้ไปใช้ต่อ โดยผลลัพธ์คือความระยะตามแนวราบในหน้าเอกสารของ Current Position
13	float GetY	เป็นฟังก์ชันอำนวยความสะดวกเพื่อนำผลลัพธ์ที่ได้จากฟังก์ชันนี้ไปใช้ต่อ โดยผลลัพธ์คือความระยะตามแนวตั้งในหน้าเอกสารของ Current Position
14	Header	เป็นฟังก์ชันสร้าง header ให้กับเอกสาร โดยฟังก์ชันนี้จะถูกเรียกอัตโนมัติเมื่อมีการขึ้นหน้าใหม่ด้วยฟังก์ชัน AddPage()
15	Image	เป็นฟังก์ชันใส่รูปภาพลงไปในเอกสาร โดยสามารถระบุตำแหน่งหรือ เลือกรูปภาพได้จากไฟล์ JPG หรือ PNG
16	Line	เป็นฟังก์ชันทำหน้าที่วาดเส้นระหว่างตำแหน่งสองตำแหน่ง
17	Link	เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูล link ลงหน้าเอกสาร PDF เพื่อประโยชน์ในการทำให้ข้อความ หรือรูปภาพ เป็นจุดที่สามารถเชื่อมโยงไปยังจุดอื่นจึงสามารถที่จะคลิกแล้วไปยังจุดนั้นๆ ได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 (ต่อ)

	ชื่อฟังก์ชัน	ทำหน้าที่
18	Ln	เป็นฟังก์ชันทำหน้าที่แทรกบรรทัดลงในหน้าเอกสาร สามารถที่จะเลือกได้ว่าบรรทัดที่จะแทรกนั้นมีความสูงเท่าไร
19	MultiCell	เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูลตัวอักษรลงในตำแหน่งหน้าเอกสาร พร้อมด้วยข้อมูลบรรทัดอัตโนมัติทันทีที่หมดบรรทัด โดยอ้างอิงจากจุดท้ายสุดของ cell ก่อนหน้า หรือถ้าไม่มีก็นับจากจุดบนซ้ายสุดของหน้าเอกสาร
20	Output	เป็นฟังก์ชันทำหน้าที่สั่งให้โปรแกรมประมวลผล แล้วส่งออกผลลัพธ์เป็นไฟล์ หรือว่า จะส่งไปยังบราวเซอร์
21	PageNo	เป็นฟังก์ชันทำหน้าที่ให้ผลลัพธ์ค่าเลขที่หน้าปัจจุบันที่ Current Position อยู่
22	Rect	เป็นฟังก์ชันวาดสี่เหลี่ยมในหน้าเอกสาร
23	SetAuthor	เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูลอ้างอิงว่าใครเป็นเจ้าของไฟล์นี้ด้วย author
24	SetAutoPageBreak	เป็นฟังก์ชันทำหน้าที่ให้โปรแกรมทำการแทรกข้อมูลสิ้นสุดหน้ากระดาษของแต่ละแผ่น โดยอัตโนมัติ สามารถเลือกได้ว่าจะเป็น auto หรือ ใช้การคำนวณระยะห่าง margin จากด้านล่างของหน้ากระดาษ
25	SetCompression	เป็นฟังก์ชันที่ทำหน้าที่บีบอัดข้อมูลเอกสาร โดยจะทำงานร่วมกับ Zlib ซึ่งเป็น โปรแกรมที่มีอยู่ใน โปรแกรม PHP
26	SetCreator	เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูลผู้สร้างไฟล์ลงในไฟล์เอกสาร PDF
27	SetDisplayMode	เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูลอ้างอิงให้ โปรแกรม Acrobat Reader เรียกใช้ในการเปิดอ่านไฟล์เอกสาร
28	SetDrawColor	เป็นฟังก์ชันทำหน้าที่ตั้งค่าสีที่ต้องการสำหรับวาดวัตถุต่างๆในเอกสาร
29	SetFillColor	เป็นฟังก์ชันทำหน้าที่ตั้งค่าสีที่ต้องการสำหรับพื้นหลังของวัตถุต่างๆในเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 (ต่อ)

	ชื่อฟังก์ชัน	ทำหน้าที่
30	SetFont	เป็นฟังก์ชันกำหนดชื่อ Font เพื่อให้ใส่ข้อมูลลงเอกสารตามขนาดรูปแบบที่ Font ที่กำหนด
31	FontSize	เป็นฟังก์ชันกำหนดขนาด Font เพื่อให้ใส่ข้อมูลลงเอกสารตามขนาด Font ที่กำหนด
32	SetKeywords	เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูล keyword ลงในไฟล์เอกสาร PDF สำหรับให้โปรแกรมค้นหาต่างๆ ได้นำไปใช้
33	SetLeftMargin	เป็นฟังก์ชันทำหน้าที่กำหนดระยะขอบซ้ายของหน้าเอกสาร
34	SetLineWidth	เป็นฟังก์ชันทำหน้าที่กำหนดความกว้างของเส้นที่จะวาด ถ้าไม่กำหนดโปรแกรมจะตั้งเป็น 0.2 mm
35	SetLink	เป็นฟังก์ชันทำหน้าที่กำหนดจุดที่ต้องการให้ Link ทำการกระโดดไปถึง
36	SetMargins	เป็นฟังก์ชันทำหน้าที่กำหนดระยะของ Margin หน้ากระดาษทุกๆ หน้าของเอกสาร ซึ่งถ้าไม่ตั้งโปรแกรมจะกำหนดเป็น 10mm สำหรับ ขอบซ้าย ขอบบน ขอบขวา
37	SetRightMargin	เป็นฟังก์ชันทำหน้าที่กำหนดระยะของ Margin ขอบขวาของหน้าเอกสารที่ Currency Position อยู่
38	SetSubject	เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูลชื่อเรื่องลงในไฟล์เอกสาร PDF สำหรับให้โปรแกรมค้นหาต่างๆ ได้นำไปใช้
39	SetTextColor	เป็นฟังก์ชันทำหน้าที่ตั้งค่าสีที่ต้องการสำหรับข้อความๆ ในเอกสาร
40	SetTitle	เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูลชื่อของเอกสารลงในไฟล์เอกสาร PDF สำหรับให้โปรแกรมค้นหาต่างๆ ได้นำไปใช้
41	SetTopMargin	เป็นฟังก์ชันทำหน้าที่กำหนดระยะของ Margin ขอบบนของหน้าเอกสารที่ Currency Position อยู่
42	SetX	เป็นฟังก์ชันเลื่อน Current Position ไปยังตำแหน่งในแนวราบ ณ.บรรทัดที่ Current Position อยู่

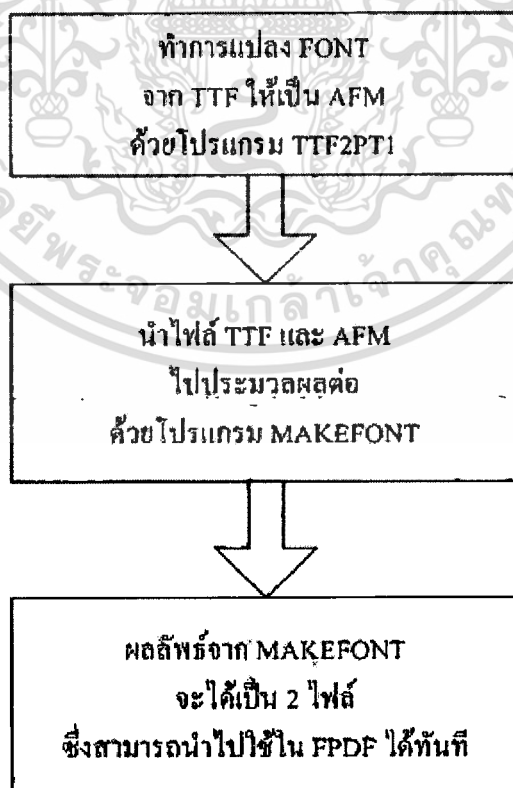
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 (ต่อ)

	ชื่อฟังก์ชัน	ทำหน้าที่
43	SetXY	เป็นฟังก์ชันเลื่อน Current Position ไปยังตำแหน่งในหน้าเอกสาร
44	SetY	เป็นฟังก์ชันเลื่อน Current Position ไปยังตำแหน่งในแนวตั้ง
45	Text	เป็นฟังก์ชันให้ใส่ข้อความลงในเอกสาร โดยสามารถระบุได้ว่าจะใส่ลง ณ.ตำแหน่งใด
46	Write	เป็นฟังก์ชันให้ใส่ข้อความลงในเอกสาร ณ.ตำแหน่งที่ Current Position อยู่

4.4 ดัดแปลง Library FPDF เพื่อให้สามารถทำงานร่วมกับภาษาไทยได้

เนื่องจาก Library FPDF ที่ทำหน้าที่สร้างไฟล์ PDF ยังไม่มีความสามารถด้านภาษาไทย ทำให้ต้องดัดแปลง Library FPDF เพื่อให้สามารถทำงานร่วมกับภาษาไทยได้ และทำการสร้าง Font ภาษาไทย เพื่อให้สามารถใช้กับ Font ภาษาไทยได้ ซึ่งการที่จะสร้าง Font เพื่อที่จะให้ใช้ร่วมกับโปรแกรม FPDF ได้นั้น มีหลายขั้นตอน ซึ่งจะแสดงด้วยรูปที่ 4.5



รูปที่ 4.5 แผนภาพขั้นตอนการสร้าง Font

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าต้องใช้ทั้งหมด 2 โปรแกรม ซึ่งก็คือโปรแกรม TTF2PT1 (True Type Font to Postscript Type 1 Converter. 2007) และ โปรแกรม MAKEFONT ซึ่งเป็นส่วนหนึ่งของ Library FPDF โดยหน้าที่ของแต่ละโปรแกรมคือ

- โปรแกรม TTF2PT1 เป็นโปรแกรมหนึ่งที่ใช้กันแพร่หลายในการแปลง TTF ไปเป็น PT1 สามารถดาวน์โหลดได้ที่ <http://ttf2pt1.sourceforge.net/> โดยโปรแกรมนี้ทำหน้าที่แปลงไฟล์ Font จากรูปแบบ TTF ซึ่งเป็นไฟล์ Font ที่ใช้กันแพร่หลายทั่วไปในโปรแกรมปฏิบัติการของไมโครซอฟท์ (TTF Definition. 2007) มี Font มากมายให้เลือกใช้งานซึ่งรวมถึง Font ภาษาไทยที่มีอยู่มากมายด้วยเช่นกัน โดยโปรแกรมนี้จะแปลงจาก TTF ไปเป็น AFM (Adobe Font Metrics) ซึ่งเป็นไฟล์ Font ร่วมของ Font ในรูปแบบ Postscript ใช้บอกตำแหน่ง Font สำหรับไฟล์ PDF เนื่องจากไฟล์ PDF จะเก็บข้อมูล Font เป็นรูปแบบ Postscript เท่านั้น
- โปรแกรม MAKEFONT เป็นโปรแกรมส่วนหนึ่งของโปรแกรม FPDF ทำหน้าที่แปลง Font ให้เป็น 2 ไฟล์ เพื่อให้โปรแกรม PDF เข้าใจและนำ Font ไปใช้ในโปรแกรม FPDF ได้ทันที ซึ่ง 2 ไฟล์นี้ จะประกอบไปด้วย
 - ชื่อไฟล์. PHP เป็นไฟล์เพื่อบอก Font Definition ให้โปรแกรม FPDF เข้าใจ เช่น Font นี้ชื่ออะไร ทำงานอย่างไร มีตัวอักษรโดยอยู่ที่ตำแหน่งใดบนไฟล์ Font จริง
 - ชื่อไฟล์. Z เป็นไฟล์ Font จริง ซึ่งเป็นไฟล์ในรูปแบบ Postscript และถูกบีบอัดไว้ สามารถพร้อมใช้งานได้จากโปรแกรม FPDF ได้

สำหรับขั้นตอนของการสร้าง Font ทางเทคนิค มีรายละเอียดดังต่อไปนี้ (True Type Font to Postscript Type 1 Converter. 2007)

- เรียกโปรแกรม "ttf2pt1 -a FontName.ttf -NewFont" ผ่าน DOS โดย FontName.ttf คือ Font ที่ต้องการจะสร้าง โดยทำการสร้างจากไฟล์ Font ในรูปแบบ TTF และ NewFont คือ ชื่อ Font ใหม่ ซึ่งจะได้ผลลัพธ์เป็นไฟล์ NewFont.AFM เพื่อนำไปใช้ต่อไป
- เรียกโปรแกรม MakeFont(string fontfile, string afmfile [, string enc]) ผ่านโปรแกรม PHP โดยมี Parameter ให้ใส่ดังต่อไปนี้
 - fontfile : Path ที่อยู่ของไฟล์ TTF และชื่อไฟล์ TTF ในเครื่องคอมพิวเตอร์
 - afmfile : Path ที่อยู่ของไฟล์ AFM และชื่อไฟล์ AFM ในเครื่องคอมพิวเตอร์
 - enc : รหัส Encoding ที่ใช้ โดยสำหรับภาษาไทยจะต้องใช้รหัส Encoding

ISO-8859-11 ดังรูปภาพที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง Font ภาษาไทย ที่นิยมใช้ในปัจจุบัน ได้แก่ AngsanaNew CordiaNew BowaliaNew และ JasmineUPC

- นำไฟล์ที่ได้จากผลลัพธ์ของโปรแกรม MakeFont ซึ่งมี 2 ไฟล์ ไปใช้กับโปรแกรม FPDF โดยผ่านฟังก์ชัน AddFont เพื่อทำการเพิ่ม Font ภาษาไทยเข้าไปในโปรแกรม เพื่อเตรียมสำหรับการประมวลผลเป็นเอกสารภาษาไทย

หลังจากที่ได้ทำการสร้าง Font ภาษาไทยแล้ว ซึ่งจะได้ไฟล์ Font แต่ละ Font เป็น Font ละ 2 ไฟล์ ซึ่งก็คือไฟล์ FontName.PHP และ FontName.Z ซึ่งก่อนที่จะใช้ Font ภาษาไทยเหล่านี้ จำเป็นต้องทำการคัดลอกไฟล์ Font เหล่านี้ไปไว้ที่ Path ที่ Library FPDF อยู่ และ แก้ไข Source Code ของโปรแกรม FPDF โดยเพิ่มฟังก์ชัน AddFont เพื่อทำการเพิ่ม Font เข้าไปในโปรแกรม ซึ่งฟังก์ชัน AddFont สามารถเรียกใช้ได้โดย AddFont(string family [, string style [, string file]]) โดยสามารถใส่ Parameter ดังต่อไปนี้

- family : ชื่อ Font
- style : รูปแบบ Font เช่น r (Regular) b (Bold) i (Italic) และ bi (Bold Italic)
- file : ชื่อ ไฟล์ PHP ซึ่งเป็น Definition ของ Font ที่ใช้สำหรับ FPDF เท่านั้น

และในการดัดแปลง Library FPDF มีรายละเอียดในการเพิ่ม Font ด้วยฟังก์ชัน AddFont ดังต่อไปนี้

```
function SetThaiFont(){
    $this->AddFont('AngsanaNew','','angsa.php');
    $this->AddFont('AngsanaNew','B','angsab.php');
    $this->AddFont('AngsanaNew','I','angsai.php');
    $this->AddFont('AngsanaNew','IB','angfaz.php');
    $this->AddFont('CordiaNew','','cordia.php');
    $this->AddFont('CordiaNew','B','cordiab.php');
    $this->AddFont('CordiaNew','I','cordiai.php');
    $this->AddFont('CordiaNew','IB','cordiaz.php');
    $this->AddFont('BrowalliaNew','','browa.php');
    $this->AddFont('BrowalliaNew','B','browab.php');
    $this->AddFont('BrowalliaNew','I','browai.php');
    $this->AddFont('BrowalliaNew','IB','browaz.php');
    $this->AddFont('JasmineUPC','','jasm.php');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ \$this->AddFont('JasmineUPC','B','jasmb.php'); นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

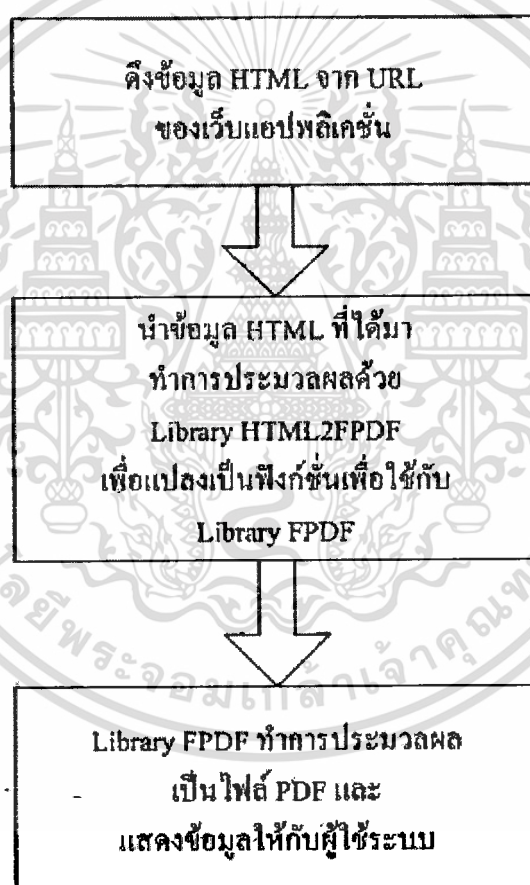
```

$this->AddFont('JasmineUPC','I','jasmi.php');
$this->AddFont('JasmineUPC','IB','jasmz.php');
}

```

4.5 สร้างโปรแกรมเพื่อประมวลผลจาก HTML ไปเป็น PDF โดยอัตโนมัติ

หลังจากที่ได้ดัดแปลง Library HTML2FPDF และ Library FPDF เพื่อให้สามารถใช้งานร่วมกับภาษาไทยได้แล้ว จึงทำการสร้างโปรแกรมเพื่อที่จะรับส่งข้อมูลเข้าไปยัง Library ทั้งสองเพื่อใช้ในการประมวลผลจาก HTML ไปเป็น PDF ได้โดยอัตโนมัติ ซึ่งโปรแกรมจะมีขั้นตอนการทำงานโดยสามารถแสดงเป็นรูปภาพดังนี้

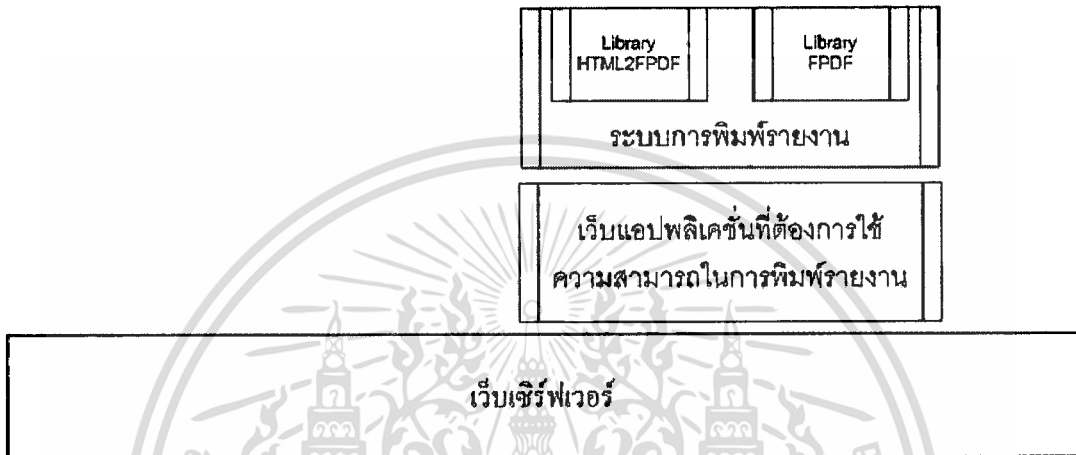


รูปที่ 4.6 แผนภาพขั้นตอนการทำงานของโปรแกรม

จากรูปที่ 4.6 จะเห็นได้ว่าจะต้องทำการดึงข้อมูล HTML ของเว็บแอปพลิเคชันมาทำการประมวลผล ซึ่งการที่จะดึงข้อมูล HTML เหล่านี้ได้ นั้น สามารถดึงได้จาก URL ปัจจุบันที่เว็บแอปพลิเคชันทำงานอยู่ โดยการที่จะอ่านค่าจาก URL ปัจจุบันที่เว็บแอปพลิเคชันอยู่นั้น สามารถทำได้ด้วยการเรียกใช้เว็บแอปพลิเคชันที่ต้องการใช้งานการพิมพ์รายงานผ่านทางโปรแกรมนี้ ซึ่งโปรแกรมนี้จะเหมือนโปรแกรมที่ทำงานครอบเว็บแอปพลิเคชัน ซึ่งจะเป็ผลทำให้โปรแกรมไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถรับทราบได้ว่าเว็บแอปพลิเคชันกำลังใช้งานอยู่นั้น กำลังใช้งานอยู่ที่ URL อะไร และหลังจากนั้น จึงนำค่า URL ที่กำลังทำงานอยู่นั้นไปดึงข้อมูล HTML แล้วส่งต่อไปประมวลผลด้วย Library HTML2FPDF และ PDF ต่อไป

จากการหลักทำงานนี้ สามารถแสดงเป็นรูปภาพเมื่อโปรแกรมกำลังใช้งานอยู่บนเซิร์ฟเวอร์ ร่วมกับเว็บแอปพลิเคชัน ได้ดังนี้



รูปที่ 4.7 แผนภาพเมื่อโปรแกรมกำลังใช้งานบนเซิร์ฟเวอร์

จากรูปที่ 4.7 จะเห็นได้อย่างชัดเจนว่า โปรแกรมการพิมพ์รายงานจะทำงานครอบคลุมอยู่บนเว็บแอปพลิเคชันเพื่อที่จะทำการอ่านค่า URL ปัจจุบันที่เว็บแอปพลิเคชันกำลังทำงานอยู่ โดยเทคนิคที่เหมาะสมในการทำงานนี้คือการใช้ฟังก์ชัน IFRAME ของ HTML เพื่อเรียกโปรแกรมการพิมพ์รายงานได้โดยใช้เว็บเซิร์ฟเวอร์เป็นตัวประมวลผล จากนั้นโปรแกรมการพิมพ์รายงานจะทำการดึงค่า URL ปัจจุบันที่อยู่ใน IFRAME ออกมาแล้วส่งต่อไปประมวลผลต่อ ซึ่งจะได้ผลลัพธ์ที่ได้จะเป็นข้อมูล HTML จากนั้นโปรแกรมจะนำข้อมูล HTML ที่ได้มา ทำการส่งต่อไปประมวลผลด้วย Library HTML2FPDF และ Library FPDF ตามลำดับ จึงได้ผลลัพธ์สุดท้ายออกมาเป็นข้อมูล PDF สำหรับผู้ใช้งานโปรแกรมได้ใช้งานได้ทันที โดยส่วนที่ทำหน้าที่ดึงข้อมูล HTML จากเว็บแอปพลิเคชัน จะทำงานแบบ CGI ซึ่งประกอบไปด้วยไฟล์ 2 ไฟล์ ซึ่งมีรายละเอียดในการทำงานร่วมกันดังนี้

- ไฟล์ INDEX.PHP ทำหน้าที่หลักคือส่ง Parameter ซึ่งเป็นข้อมูล URL ไปยัง GETURL.PHP ซึ่งจะเริ่มทำงานเมื่อเมื่อผู้ใช้งานเว็บแอปพลิเคชันทำการกดปุ่มเพื่อให้โปรแกรมเริ่มทำการประมวลผลหน้ารายงาน HTML ไปเป็น PDF เพื่อส่งพิมพ์
- ไฟล์ GETURL.PHP ทำหน้าที่ดึงข้อมูล HTML จาก URL ที่ได้รับจาก INDEX.PHP ซึ่งเมื่อได้ข้อมูล HTML ออกมาแล้วก็นำไปใส่ในตัวแปร แล้วจากนั้นจึงส่งต่อไปประมวลผลด้วย

Library HTML2PDF และ Library FPDF จนได้ผลลัพธ์ออกมาเป็นไฟล์ PDF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การใช้งานโปรแกรม

ผู้ใช้งาน โปรแกรม สามารถใช้งานได้ 2 ลักษณะ คือ ให้โปรแกรมทำการสร้างข้อมูล PDF โดยการเรียกใช้ Library FPDF โดยตรง ซึ่งจะมีรายละเอียดการใช้งานอยู่ที่ภาคผนวก หรือ ให้โปรแกรมทำงานโดยระบบอัตโนมัติด้วยการแปลงข้อมูล HTML ไปเป็น PDF ซึ่งมีรายละเอียดการใช้งานโปรแกรมดังนี้

5.1 การติดตั้งโปรแกรม

เมื่อผู้พัฒนาเว็บแอปพลิเคชัน ต้องการที่จะใช้โปรแกรมนี้นี้ร่วมกับเว็บแอปพลิเคชันที่ได้สร้างขึ้นมา ก็จำเป็นที่ผู้สร้างเว็บแอปพลิเคชันจะต้องทำหน้าที่ติดตั้งโปรแกรมนี้นี้ เพื่อให้ไปทำงานครอบทั้งเว็บแอปพลิเคชันที่ต้องการจะใช้งาน โดยติดตั้งบนเครื่องเซิร์ฟเวอร์เดียวกับเว็บแอปพลิเคชันนั้นๆ ที่ต้องการที่จะใช้ในการพิมพ์รายงาน ซึ่งวิธีการติดตั้งนั้น สามารถทำได้โดย

- คัดลอกไฟล์ทั้งหมดของระบบนี้ ไปยัง Root Directory ที่เว็บแอปพลิเคชันอยู่
- แก้ไขไฟล์ INDEX.PHP เพื่อให้เว็บแอปพลิเคชันทำการเรียกระบบนี้ไปครอบการทำงานของเว็บแอปพลิเคชัน ซึ่งภายใน INDEX.PHP มี Source Code ดังนี้

```
1 <head>
2 <title>HTML -> PDF</title>
3 <script>
4 function changePage(strPage) {
5     frames['bodyiframe'].location.href = strPage;
6 }
7 function generatePDF(strPage) {
8 window.open('geturl.php?Parameter='+strPage,'pdfWindow','width=800,
height=600,left=0,top=100,screenX=0,screenY=100')
9 }
10 </script>
11 </head>
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ © 2012 โดย บริษัท เทคโนโลยี จำกัด และสงวนลิขสิทธิ์ในชื่อของเอกสารนี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

13
14 <div id="LayerLink" style="visibility:show; position:absolute; width:30px;
height:10px; z-index:2; right: 20px; top: 2px; background-color: #FF3366; layer-
background-color: #FF0000; border: 1px none #000000;"
onMouseOver="document.getElementById('LayerMenu').style.visibility = 'visible'">
15 <b><font color="#FFFFFF">
16 PDF
17 </font></b>
18 </div>
19
21 <div id="LayerMenu" style="visibility:hidden; position:absolute; width:100px;
height:100px; z-index:3; right: 20px; top: 2px; border: 1px none #000000;">
22 <center>
23 <form name="PASSURL">
24 <!-- URL : <input name="INPUTURL" type="text" size="30">
25 <input type="button" value="Go"
onClick="changePage(PASSURL.INPUTURL.value)">
26 </form>
27 -->
28 <a href="">
29 
30 </a>
32 </center>
33 </div>
34 <div id="LayerMenu" style="visibility:show; position:absolute; width:100%;
height:100%; z-index:1; left: 0px; right: 0px; top: 0px; background-color: #FFFFFF;
layer-background-color: #FFFFFF; border: 0px none #000000;">
35 <iframe src="http://localhost/test/home.php"
36 id="bodyiframe"
37 name="bodyiframe"
38 style="width: 101.5%;

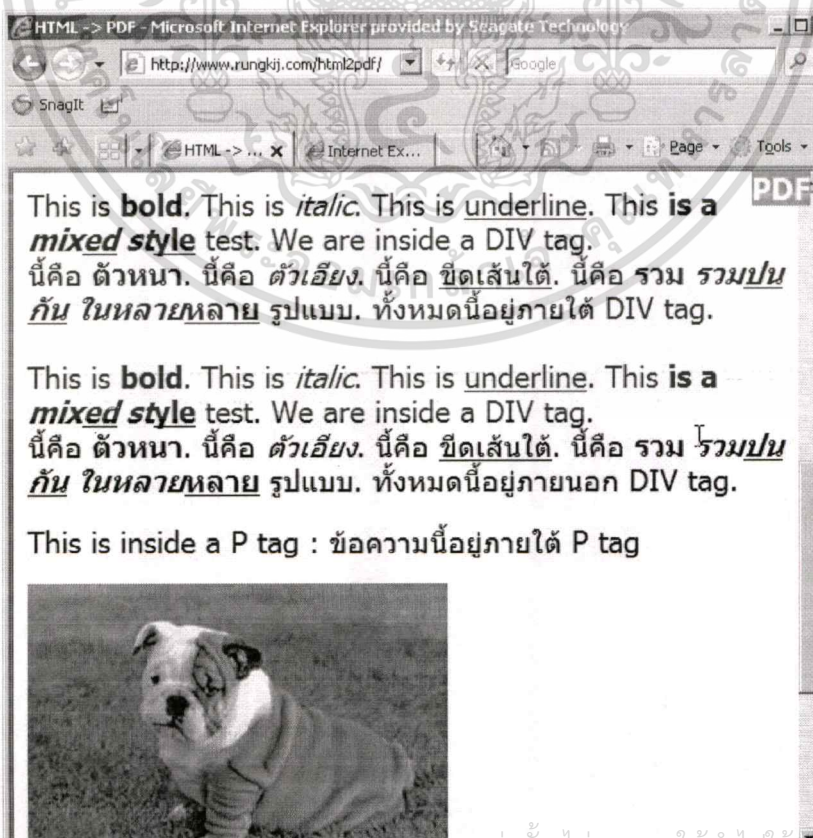
```

```

39         height: 100%;
40         SCROLLING="AUTO"
41         border: 1px solid #000000;"
42     >
43 </iframe>
44 </div>
45 </center>
46 </body>
47 </html>

```

จากนั้นให้ทำการแก้ไขข้อมูล <IFRAME> ในบรรทัดที่ 35 โดยกำหนด SRC เพื่อชี้ไปยังชื่อไฟล์แรกที่เว็บแอปพลิเคชันใช้ในการทำงาน ดังตัวอย่าง ระบบจะเรียกไฟล์ที่ชื่อว่า "http://localhost/test/home.php" หลังจากนั้น ระบบก็จะพร้อมทำงานทันที โดยสามารถตรวจสอบได้ว่าโปรแกรมได้ติดตั้งสมบูรณ์หรือไม่ด้วยการเรียกใช้เว็บแอปพลิเคชัน โดยระหว่างที่เว็บแอปพลิเคชันทำงาน ก็จะมีปุ่มเล็ก ๆ ด้านบนขวามือของเว็บเบราว์เซอร์ตลอดเวลา เพื่อให้ผู้ใช้งานระบบสามารถกดเพื่อสั่งพิมพ์ดังตัวอย่างในรูปที่ 5.1

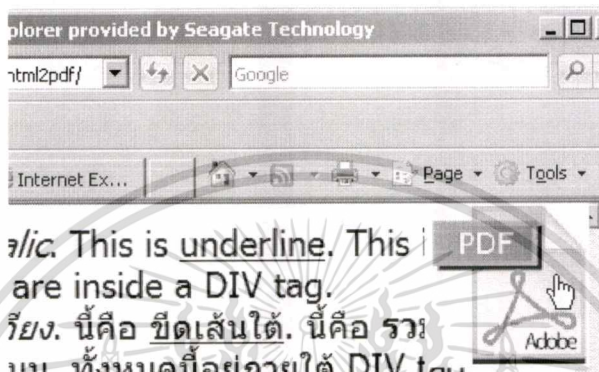


เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และใช้เพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 5.1 หน้าจอการทำงานของโปรแกรม

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

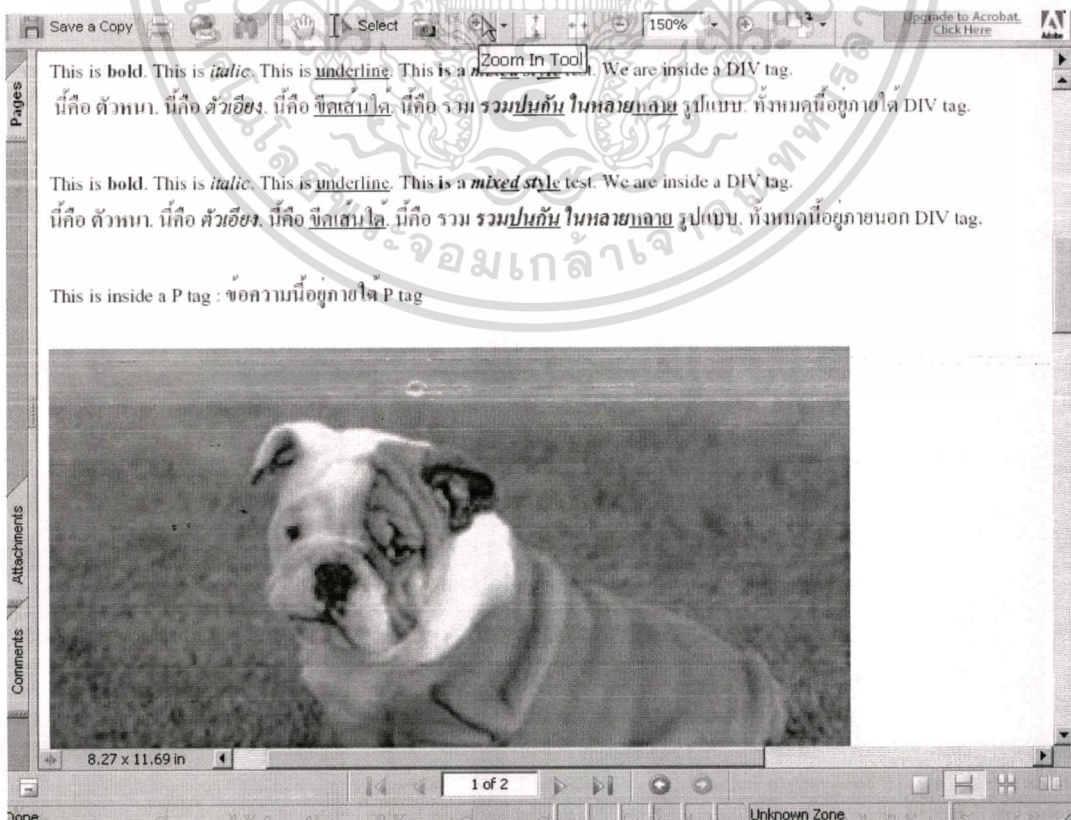
5.2 การใช้งานโปรแกรมสำหรับผู้ใช้งานเว็บแอปพลิเคชัน

ในระหว่างใช้เว็บแอปพลิเคชัน ถ้าผู้ใช้งานต้องการพิมพ์หน้าใดๆในเว็บแอปพลิเคชัน ก็เพียงเลื่อนเมาส์ไปยังปุ่มที่เขียนว่า PDF ด้านบนขวามือของเว็บเบราว์เซอร์ จากนั้นโปรแกรมจะขยายปุ่ม PDF ให้ใหญ่ขึ้นเพื่อให้ผู้ใช้งานระบบสามารถคลิกเมาส์ได้สะดวกยิ่งขึ้นดังรูปที่ 5.2



รูปที่ 5.2 ปุ่มสั่งให้โปรแกรมเริ่มทำงาน

และเมื่อผู้ใช้งานระบบกดปุ่มนี้ ระบบก็จะเริ่มทำการประมวลผลเป็น PDF โดยได้ผลลัพธ์เป็นดังรูปที่ 5.3 เพื่อให้ผู้ใช้งานเว็บแอปพลิเคชันสามารถสั่งพิมพ์จากไฟล์ PDF ได้ทันที



รูปที่ 5.3 ผลลัพธ์หน้ารายงานผ่าน โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การใช้งานโปรแกรมสำหรับผู้สร้างเว็บแอปพลิเคชัน

การทำงานของโปรแกรมนี้คือ จะทำการอ่านข้อมูล HTML ของเว็บแอปพลิเคชันที่ใช้งานเว็บแอปพลิเคชันกำลังใช้งานอยู่ จากนั้นจึงนำข้อมูล HTML มาทำการแปลงเป็นข้อมูล PDF ซึ่งมีข้อดี คือ จะทำให้เครื่องคอมพิวเตอร์สามารถแสดงผลเป็นรายงานในรูปแบบ PDF เพื่อให้สามารถพิมพ์ได้ตรงตามตำแหน่งของหน้ากระดาษ ได้อย่างที่ผู้สร้างเว็บแอปพลิเคชันต้องการ โดยโปรแกรมนี้มีความสามารถที่จะอ่านและเข้าใจ HTML ทั่วไป เพื่อแปลงเป็นข้อมูล PDF ได้ทันที แต่การที่โปรแกรมนี้จะทำงานเพื่อออกผลลัพธ์ข้อมูล PDF ได้ตรงตามตำแหน่งของหน้ากระดาษ อย่างที่ผู้สร้างเว็บแอปพลิเคชันต้องการหรือไม่ นั่น ก็ขึ้นอยู่กับข้อมูล HTML ที่ผู้สร้างเว็บแอปพลิเคชันได้เขียนโปรแกรมขึ้นมาตัวเอง ซึ่งการที่จะทำให้ได้ผลลัพธ์ที่ดีนั้น มีเทคนิค 2 เทคนิค คือ การจำกัดข้อมูลในแนวตั้งของหน้ากระดาษ และ การจำกัดข้อมูลในแนวขวางของหน้ากระดาษ

- การจำกัดข้อมูลในแนวตั้งของหน้ากระดาษ สามารถทำได้โดย โปรแกรมนี้ได้มีคำสั่ง HTML TAG พิเศษ เพิ่มขึ้นมา 1 คำสั่ง ซึ่งก็คือคำสั่งขึ้นหน้าใหม่โดยสามารถเรียกใช้ได้โดยแทรก <NEWPAGE> เข้าไปใน HTML ซึ่งคำสั่งนี้จะไม่มีอยู่ในมาตรฐาน HTML ทั่วไป แต่คำสั่งนี้จะใช้เพื่อแทรกเข้าไปในหน้า HTML ของเว็บแอปพลิเคชัน โดยผู้สร้างเว็บแอปพลิเคชัน สามารถที่จะใช้แทรกเข้าไปในหน้าแสดงผล HTML ของหน้าเว็บแอปพลิเคชันที่ต้องการให้ใช้งานเว็บแอปพลิเคชันสั่งพิมพ์ เพื่อประโยชน์ในการที่จะทำให้โปรแกรมนี้สามารถทำการขึ้นหน้าใหม่ในข้อมูลผลลัพธ์ PDF ตามที่ผู้สร้างเว็บแอปพลิเคชันต้องการ ซึ่งสามารถแสดงตัวอย่างของประโยชน์ของคำสั่ง <NEWPAGE> ดังรูปที่ 5.6
- การจำกัดข้อมูลในแนวขวางของหน้ากระดาษ สามารถทำได้โดยใช้ HTML TAG ที่ชื่อว่า <TABLE> มาครอบการแสดงผลของหน้ารายงาน โดย <TABLE> สามารถที่จะกำหนดความกว้างของตารางให้พอดีกับหน้ากระดาษ ได้ดังรูปที่ 5.10 ซึ่งจะเห็นได้ว่า ผลลัพธ์ก่อนพิมพ์เมื่อกำหนดความกว้าง จะไม่มีปัญหาเกินหน้ากระดาษตามแนวขวาง ซึ่งค่าความกว้างที่แนะนำคือต้องไม่เกิน 700px จึงจะพอดีหน้ากระดาษตามแนวขวางของกระดาษขนาด A4

ซึ่งเทคนิคการจำกัดข้อมูลในแนวตั้งของหน้ากระดาษ สามารถแสดงให้เห็นได้อย่างชัดเจน ด้วยการแสดงโดยรูปด้านล่างนี้

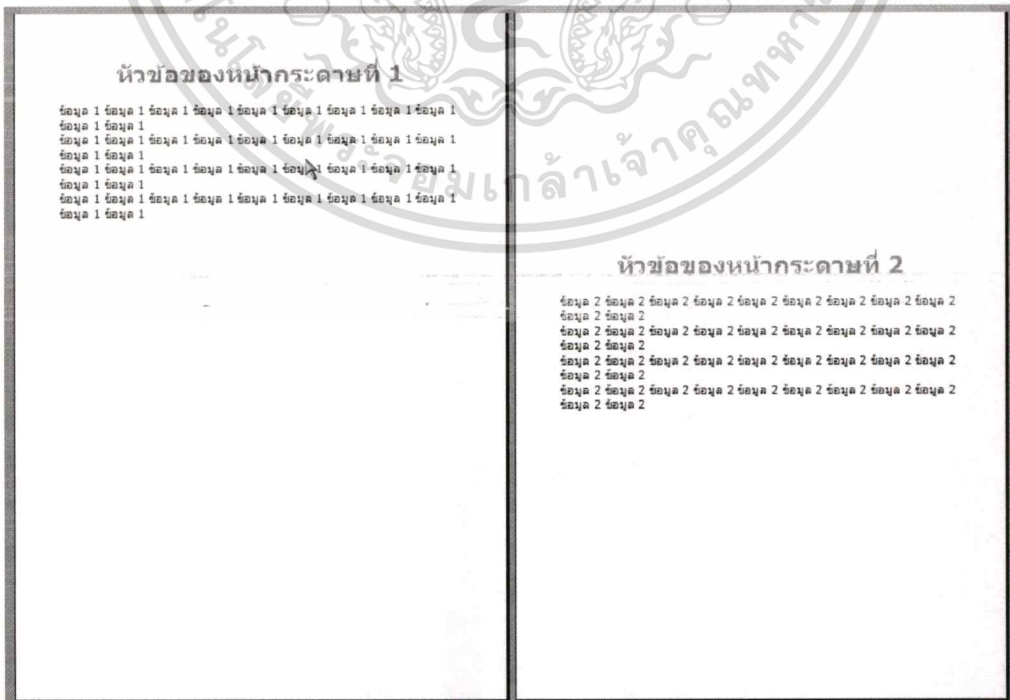
```

1 <html> <head>
2 <title> Title </title>
3 <meta http-equiv="Content-Type" content="text/html; charset=windows-874">
4 </head> <body>
5 <h1><center><font color="red"> หัวข้อของหน้ากระดาษที่ 1</font></center></h1>
6 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1<br />
7 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1<br />
8 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1<br />
9 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1<br />
10 <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br />
11 <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br />
12 <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br />
13 <br /><br /></br />
14 <h1><center><font color="red"> หัวข้อของหน้ากระดาษที่ 2</font></center></h1>
15 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2<br />
16 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2<br />
17 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2<br />
18 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2<br />
19 </body></html>

```

รูปที่ 5.4 เมื่อไม่ใช้คำสั่งขึ้นหน้าใหม่

จาก Source Code ในรูปที่ 5.4 เมื่อผู้ใช้งานระบบสั่งพิมพ์ด้วยเว็บเบราว์เซอร์ หรือ โปรแกรมนี้ ก็จะแสดงผลดังรูปที่ 5.5 ซึ่งจะเห็นได้ว่าแต่ละหน้ากระดาษจะมีการเลื่อมล้ำกันไม่สวยงามถูกต้องตามตำแหน่งที่ควรจะเป็น



รูปที่ 5.5 ผลลัพธ์ก่อนพิมพ์เมื่อไม่ใช้คำสั่งขึ้นหน้าใหม่

แต่เมื่อใช้คำสั่งขึ้นหน้าใหม่ <NEWPAGE> แทรกเข้าไปใน Source Code ของ HTML ดังรูปที่ 5.5 ก็จะทำให้โปรแกรมสามารถที่จะรู้ว่าจะขึ้นหน้าใหม่ได้เมื่อไรนั่นเอง

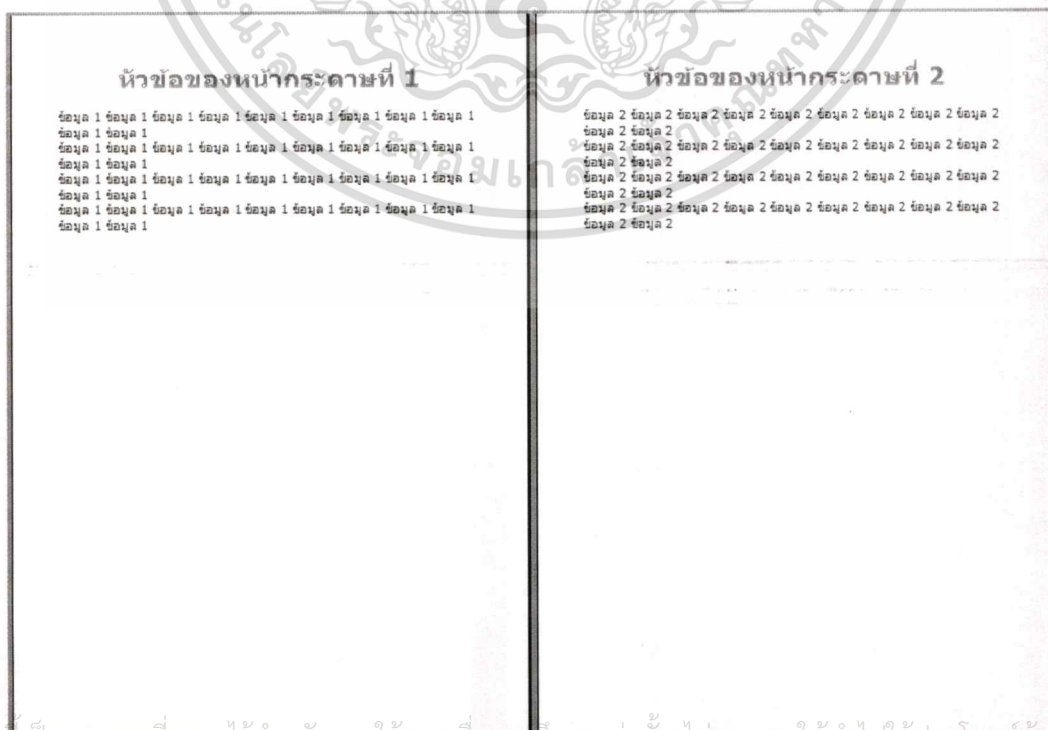
```

1 <html> <head>
2 <title> Title </title>
3 <meta http-equiv="Content-Type" content="text/html; charset=windows-874">
4 </head> <body>
5 <h1><center><font color="red"> หัวข้อของหน้ากระดาษที่ 1</font></center></h1>
6 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1<br />
7 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1<br />
8 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1<br />
9 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1 ข้อมูล 1<br />
10 <NEWPAGE>
11 <h1><center><font color="red"> หัวข้อของหน้ากระดาษที่ 2</font></center></h1>
12 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2<br />
13 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2<br />
14 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2<br />
15 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2 ข้อมูล 2<br />
16 </body></html>

```

รูปที่ 5.6 เมื่อใช้คำสั่งขึ้นหน้าใหม่

ซึ่งหลังจากใช้คำสั่งขึ้นหน้าใหม่ <NEWPAGE> ก็จะทำให้โปรแกรมสามารถขึ้นหน้าใหม่ได้อย่างถูกต้องตามตำแหน่งได้อย่างสวยงามดังรูปที่ 5.6



รูปที่ 5.7 ผลลัพธ์ก่อนพิมพ์เมื่อใช้คำสั่งขึ้นหน้าใหม่

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเทคนิคการจำกัดข้อมูลในแนววางของหน้ากระดาษ สามารถแสดงให้เห็นได้อย่างชัดเจน ด้วยการแสดงโดยรูปด้านล่างนี้

```

1 <html> <head>
2 <title> Title </title>
3 <meta http-equiv="Content-Type" content="text/html; charset=windows-874">
4 </head> <body>
5 <h1><center><font color="red"> หัวข้อของหน้ากระดาษที่ 1</font></center></h1>
6 <table border="1">
7 <tr><td> หัวข้อช่องที่ 1 </td><td> หัวข้อช่องที่ 2 </td>
8 <td> หัวข้อช่องที่ 3 </td><td> หัวข้อช่องที่ 4 </td>
9 </tr>
10 <tr><td> เนื้อหา 1 </td><td> เนื้อหา 2222 </td>
11 <td> เนื้อหา 333333 </td><td> เนื้อหา 44444444 </td>
12 </tr>
13 <tr><td> เนื้อหา 1 </td><td> เนื้อหา 2222 </td>
14 <td> เนื้อหา 333333 </td><td> เนื้อหา 44444444 </td>
15 </tr>
16 </table>
17 <br />
18 </body></html>
19

```

รูปที่ 5.8 เมื่อ ไม่กำหนดความกว้าง

จะเห็นได้ว่าผลลัพธ์ก่อนพิมพ์เมื่อ ไม่กำหนดความกว้างจะมีปัญหาเกินหน้ากระดาษตามแนววางดังรูปที่ 5.8

หัวข้อของหน้ากระดาษที่ 1			
หัวข้อช่องที่ 1	หัวข้อช่องที่ 2	หัวข้อช่องที่ 3	หัวข้อช่องที่ 4
เนื้อหา 1	เนื้อหา 2222	เนื้อหา 333333	เนื้อหา 4444
เนื้อหา 1	เนื้อหา 2222	เนื้อหา 333333	เนื้อหา 4444

รูปที่ 5.9 ผลลัพธ์ก่อนพิมพ์เมื่อ ไม่กำหนดความกว้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่เมื่อใช้คำสั่งกำหนดความกว้างของตาราง `<TABLE WIDTH="700">` แทรกเข้าไปใน Source Code ของ HTML ดังรูปที่ 5.5 ก็จะทำให้โปรแกรมสามารถที่จะรู้ว่าจำกัดความกว้างแนววางที่เท่าไร จึงทำให้ผลลัพธ์ก่อนพิมพ์ไม่เกินหน้ากระดาษตามแนววาง

```

1 <html> <head>
2 <title> Title </title>
3 <meta http-equiv="Content-Type" content="text/html; charset=windows-874">
4 </head> <body>
5 <h1><center><font color="red"> หัวข้อของหน้ากระดาษที่ 1</font></center></h1>
6 <table border="1" width="700">
7 <tr><td> หัวข้อของที่ 1 </td><td> หัวข้อของที่ 2 </td>
8 <td> หัวข้อของที่ 3 </td><td> หัวข้อของที่ 4 </td>
9 </tr>
10 <tr><td> เนื้อหา 1 </td><td> เนื้อหา 2222 </td>
14 <td> เนื้อหา 333333 </td><td> เนื้อหา 44444444 </td>
12 </tr>
13 <tr><td> เนื้อหา 1 </td><td> เนื้อหา 2222 </td>
14 <td> เนื้อหา 333333 </td><td> เนื้อหา 44444444 </td>
15 </tr>
16 </table>
17 <br/>
18 </body></html>

```

รูปที่ 5.10 เมื่อกำหนดความกว้าง

จะเห็นได้ว่าผลลัพธ์ก่อนพิมพ์เมื่อกำหนดความกว้างจะไม่มีปัญหาเกินหน้ากระดาษตามแนววางดังรูปที่ 5.8 ซึ่งค่าความกว้างที่แนะนำคือไม่เกิน 700px จึงจะไม่เกิดหน้ากระดาษตามแนววางของกระดาษขนาดมาตรฐาน A4

หัวข้อของหน้ากระดาษที่ 1

หัวข้อของที่ 1	หัวข้อของที่ 2	หัวข้อของที่ 3	หัวข้อของที่ 4
เนื้อหา 1	เนื้อหา 2222	เนื้อหา 333333	เนื้อหา 44444444
เนื้อหา 1	เนื้อหา 2222	เนื้อหา 333333	เนื้อหา 44444444

รูปที่ 5.11 ผลลัพธ์เมื่อกำหนดความกว้างตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุป

การพัฒนาโปรแกรมการพิมพ์เอกสารการพิมพ์ที่สวยงามจากเว็บแอปพลิเคชัน มีวัตถุประสงค์เพื่อช่วยสนับสนุนการใช้งานเว็บแอปพลิเคชัน เพื่อให้สามารถพิมพ์เอกสารลงตามตำแหน่งต่างๆของกระดาษผ่านเครื่องพิมพ์ด้วยโปรแกรมเว็บเบราว์เซอร์ทั่วไป เพื่ออำนวยความสะดวกในทำงานของผู้ใช้งานเว็บแอปพลิเคชัน และ ผู้สร้างเว็บแอปพลิเคชัน ซึ่งจากเดิม การที่จะพิมพ์เอกสารผ่านเว็บแอปพลิเคชันนั้น ส่วนมากจะใช้คำสั่งพิมพ์ที่มีอยู่ใน โปรแกรมเว็บเบราว์เซอร์ทั่วไป ซึ่งก็สามารถที่จะพิมพ์ได้ในระดับหนึ่ง แต่จะมีปัญหาว่า เอกสารที่ได้จากการพิมพ์ด้วยคำสั่งพิมพ์ที่มีอยู่ในเว็บเบราว์เซอร์นี้ จะไม่สวยงาม โดยเครื่องพิมพ์จะพิมพ์ข้อมูลลงกระดาษซึ่งไม่สามารถที่จะพิมพ์ลงตามตำแหน่งที่ต้องการ ได้ทั้งหมด และเมื่อมีผู้ใช้งานมีหลายคน เปิดจากเครื่องคอมพิวเตอร์หลายๆเครื่อง ก็อาจมีความเป็นไปได้ว่าค่าของการตั้งค่าน้ำกระดาษในโปรแกรมเว็บเบราว์เซอร์ไม่ตรงกัน และ เครื่องพิมพ์ต่างรุ่นต่างยี่ห้อกัน ทั้งหมดนี้มีส่วนทำให้เอกสารที่ได้จากการพิมพ์ไม่มีความสวยงามซึ่งเกิดจากที่ไม่สามารถกำหนดให้ข้อมูลตัวอักษรลงตามตำแหน่งในกระดาษที่ต้องการ ได้นั่นเอง เป็นผลทำให้เมื่อผู้ใช้งานเว็บแอปพลิเคชันต้องการที่จะพิมพ์เอกสารพิเศษ เช่น พิมพ์ใบเสร็จรับเงิน พิมพ์ใบส่งของ พิมพ์ใบแจ้งหนี้ จะไม่สามารถที่จะกำหนดให้เครื่องพิมพ์ทำการพิมพ์ลงในตำแหน่งต่างๆที่วางแผนไว้ในกระดาษได้ทั้งหมด

ทั้งหมดนี้จึงเป็นที่มาของการเริ่มต้นคิดที่ต้องการพัฒนาโปรแกรมการพิมพ์ที่สวยงามจากเว็บแอปพลิเคชันนี้เอง ซึ่งจากการที่วิเคราะห์ที่ผ่านมา มีหลากหลายวิธีให้เลือกใช้ในการพิมพ์เอกสารการพิมพ์ที่สวยงามจากเว็บแอปพลิเคชัน จึงได้เลือกวิธีที่คิดว่าดีที่สุดที่จะทำ ให้ได้พิมพ์เอกสารการพิมพ์ที่สวยงามจากเว็บแอปพลิเคชัน ซึ่งก็คือ การใช้วิธีการส่งออกไฟล์ไปเป็น PDF โดยการที่เครื่องคอมพิวเตอร์เปิดอ่านไฟล์ PDF แล้วสั่งพิมพ์ จึงทำให้เครื่องพิมพ์ทำการพิมพ์ข้อมูลต่างๆลงในหน้ากระดาษ ได้ถูกต้องตามตำแหน่งที่ต้องการ ได้ทั้งหมด

ซึ่งวิธีการส่งออกไฟล์ไปเป็น PDF นี้ ได้มีผู้พัฒนาโปรแกรมในต่างประเทศมากมาย ได้เลือกใช้วิธีนี้เช่นกัน และปัจจุบันมี Library ที่จะสามารถทำงานในลักษณะนี้มากมายใน Open Source จึงทำให้เกิดความคิดว่า เพื่อความรวดเร็วในการพัฒนาโปรแกรม ควรเลือกใช้ Library ที่มีใน Open Source แล้วนำมาพัฒนาต่อให้เป็น โปรแกรมที่ดีขึ้น สุดท้ายจึงได้เลือกใช้ Library ที่ชื่อว่า HTML2FPDF และ Library FPDF มาพัฒนาต่อ ซึ่งจากการศึกษาทั้งสองตัวนี้ จะมีจุดด้อยอยู่ที่ไม่สามารถที่จะทำการประมวลผลเป็นภาษาไทยได้ จึงได้ทำการตัดสินใจ และทำการแก้ไข Library ทั้งสองตัว โดยทำการเพิ่มความสามารถภาษาไทย เพื่อให้สามารถที่จะทำงานร่วมกับภาษาไทยได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการสร้างโปรแกรมเริ่มจากการใช้โปรแกรม MakeFont.php ซึ่งเป็น โปรแกรมส่วนหนึ่งของ Library FPDF และ โปรแกรม TTF2PS1 ที่สามารถเตรียม Font ให้กับ Library FPDF ใช้ งานโดยโปรแกรมนี้สามารถแปลง Font ที่มีอยู่ในปัจจุบัน ซึ่งส่วนมากเป็น Font ในรูปแบบ TTF (True Type Font) ให้เป็น Font ในรูปแบบ *.PHP และ *.Z ที่ Library FPDF เข้าใจ และสามารถ นำไปใช้งานได้ จากนั้นทำการเพิ่ม Font เข้าไปในโค้ดของ FPDF ด้วยฟังก์ชัน AddFont โดยฟังก์ชันนี้ทำหน้าที่บอกให้โปรแกรม FPDF เข้าใจว่าจะมี Font อะไร ชื่ออะไรบ้างที่จะใช้ในการ ประมวลผล

เมื่อ Library FPDF มี Font ภาษาไทยเรียบร้อยแล้ว Library FPDF ก็สามารถที่จะพร้อมใช้ งานเพื่อประมวลผลเป็นภาษาไทยได้ทันที โดยผู้สร้างรายงานสามารถเขียน โค้ดเพื่อเรียกใช้ฟังก์ชันปรกติของ FPDF ได้ และใช้ร่วมกับภาษาไทยได้ด้วย Font ที่มีอยู่ใน โปรแกรม ซึ่งจากขอบเขตของโปรแกรมใหม่ จะทำการสร้าง Font เฉพาะที่เป็นที่นิยมในการพิมพ์รายงานภาษาไทยเท่านั้น ซึ่ง ได้แก่ AngsanaNew CordiaNew BowaliaNew และ JasmineUPC

และเพื่อความสะดวกในการใช้งานจึงได้ใช้ Library อีกหนึ่งตัว ซึ่งก็คือ HTML2FPDF เพื่อทำหน้าที่ดึงข้อมูล HTML แล้วนำมาแปลงเป็นฟังก์ชัน เพื่อส่งต่อไปยัง Library FPDF เพื่อให้มีความสามารถในการประมวลผลจากรายงานบนหน้าเว็บ ซึ่งแสดงผลทาง HTML อยู่แล้ว โดยโปรแกรมทั้งหมดจะทำงานคลุมการทำงานของทั้งเว็บแอปพลิเคชัน และจะเริ่มทำงานเมื่อผู้ใช้งาน กดปุ่มเพื่อสั่งให้โปรแกรมแปลงเป็น PDF เท่านั้น ซึ่งจากการพัฒนาโปรแกรมทั้งหมดนี้ มีจุดที่น่าสนใจซึ่งสามารถสรุปได้ดังนี้

6.1 สรุปผลการพัฒนาโปรแกรม

เทคนิคที่น่าเสนอในโครงการนี้ เป็นเทคนิคหนึ่งในหลายๆเทคนิค ที่ช่วยในการพิมพ์รายงานผ่านเว็บแอปพลิเคชัน ซึ่งจากการทดลองใช้งาน โปรแกรมนี้สามารถทำการพิมพ์รายงานลงในกระดาษได้ตรงตามตำแหน่งที่ต้องการได้อย่างแม่นยำ สามารถพิมพ์ตัวอักษรภาษาไทยได้ และสามารถพิมพ์ตัวอักษรภาษาไทยด้วย Font ไทยที่นิยมซึ่งได้สร้างไว้ในโปรแกรมได้อย่างถูกต้อง ซึ่งได้แก่ AngsanaNew CordiaNew BowaliaNew JasmineUPC และ โปรแกรมยังสามารถประมวลผลได้ด้วยความเร็ว

6.2 ข้อจำกัดของโปรแกรมที่พัฒนาขึ้น

เนื่องจากโปรแกรมนี้จะทำการแปลงจากข้อมูล HTML ซึ่งปัจจุบันมี HTML TAG มากมาย และในอนาคตอาจจะมี TAG ใหม่ๆ เกิดเพิ่มขึ้นอีก ซึ่งมีความเป็นไปได้ว่า HTML TAG ที่จะเกิดขึ้นในอนาคตจะไม่สามารถทำงานร่วมกับโปรแกรมนี้ได้ เพราะฉะนั้นผู้พัฒนาเว็บแอปพลิเคชัน ที่ต้องการนำโปรแกรมนี้ไปใช้ร่วมกับเว็บแอปพลิเคชันที่ตนเองเขียนขึ้น จึงควรทดลองออกรายงานไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขโค้ดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จริงก่อน เพื่อตรวจสอบว่าโปรแกรมทำงานตรงตามตำแหน่งหน้าที่กระดาษได้อย่างถูกต้องอย่างแท้จริงหรือไม่

6.3 ข้อเสนอแนะในการพัฒนาต่อ

เนื่องจากผู้พัฒนาเว็บแอปพลิเคชัน ที่ต้องการนำโปรแกรมนี้ไปใช้ในการออกเอกสารการพิมพ์รายงาน ต้องติดตั้งโปรแกรมนี้เข้าไปที่เว็บเซิร์ฟเวอร์ เพื่อให้ทำงานอยู่บนเว็บแอปพลิเคชันที่ต้องการใช้โปรแกรมนี้ จึงมีจุดอ่อนอยู่ที่ จะทำให้ไม่สามารถทำงานข้ามเครื่องเซิร์ฟเวอร์กันได้ เพราะฉะนั้นถ้ามีเว็บแอปพลิเคชันเดียวกันแต่ทำงานร่วมกันหลายเซิร์ฟเวอร์ ทำให้จำเป็นต้องติดตั้งโปรแกรมนี้เข้าไปที่ทุกๆเซิร์ฟเวอร์ ซึ่งอาจจะไม่สะดวกถ้ามีเซิร์ฟเวอร์หลายตัว จึงควรหาวิธีพัฒนาให้สามารถใช้งานได้ทุกๆเซิร์ฟเวอร์โดยการติดตั้งที่เซิร์ฟเวอร์เครื่องเดียวเป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ทวีชัย สุริยะทองธรรม. 2545. **FLASH Web Programming**. กรุงเทพฯ:

ซัคเซส มีเดีย.

ชนพล ฉันทวีชัย. 2546. **Crystal Report 8.5 สร้างรายงานอย่างมืออาชีพ**. กรุงเทพฯ:

ซีอีคยูเคชั่น.

Chars and Strs. 2007. [Online], Available:

<http://www.prowiki.org/wiki4d/wiki.cgi?CharsAndStrs>.

FPDF Library. 2007. [Online], Available:

<http://sourceforge.net/projects/fpdfgenerator/>.

HTML2FPDF Library. 2007. [Online], Available:

<http://html2fpdf.sourceforge.net/>.

Inventory Control Software. 2007. [Online], Available:

<http://www.inventory-control-software.net/screenshots.htm>.

Non-ASCII Characters. 2007. [Online], Available:

<http://homepage3.nifty.com/tsato/xfig/latin/latin-e.html>.

Portable Document Format. 2007. [Online], Available:

http://th.wikipedia.org/wiki/Portable_Document_Format/.

True Type Font to Postscript Type 1 Converter. 2007. [Online], Available:

<http://ttf2pt1.sourceforge.net/>.

TTF Definition. 2007. [Online], Available: <http://en.wikipedia.org/wiki/TTF>.



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานฟังก์ชันของ Library FPDF เพื่อประมวลผลเป็น PDF โดยตรง

นอกเหนือจากการใช้งานโปรแกรมด้วยการแปลงข้อมูล HTML เป็น PDF ในระบบอัตโนมัติดังรายละเอียดการใช้งานโปรแกรมในบทที่ 6 แต่ผู้สร้างเว็บแอปพลิเคชันยังสามารถใช้วิธีอื่นอีกวิธีหนึ่งในการออกเอกสารการพิมพ์ ซึ่งก็คือ การเรียกใช้งานฟังก์ชันของ Library FPDF เพื่อประมวลผลเป็น PDF ได้โดยตรง โดยใช้โปรแกรมประเภท Text Editor เช่น Notepad หรือ Editplus เป็นต้น เพื่อทำการสร้าง Source File เพื่อทำการเรียกใช้งานฟังก์ชันต่างๆ เพื่อประมวลผลเป็น PDF ซึ่ง Library FPDF มีฟังก์ชันพร้อมให้เรียกใช้งานดังต่อไปนี้

1.1 boolean AcceptPageBreak()

เป็นฟังก์ชันที่จะทำงานร่วมกับฟังก์ชัน SetAutoPageBreak() เมื่อต้องการให้ระบบแทรกข้อมูลจบบรรทัดทันทีลงไปภายในเอกสารในกรณีที่ต้องการให้ระบบทำการแทรกข้อมูลจบบรรทัดโดยอัตโนมัติ

1.2 AddFont(string family [, string style [, string file]])

เป็นฟังก์ชันเพื่อเพิ่ม Font เข้าไปให้ระบบพร้อมใช้งานก่อนการประมวลผล โดยก่อนที่จะทำการเรียกฟังก์ชันนี้ ต้องมีการสร้าง Font ให้พร้อมใช้งานด้วยโปรแกรม makefont.php ก่อน และเมื่อได้ไฟล์ Font ที่พร้อมใช้งานแล้วให้ใส่ Parameter ดังต่อไปนี้

- family : ชื่อ Font
- style : รูปแบบ Font เช่น r (Regular) b (Bold) i (Italic) และ bi (Bold Italic)
- file : ชื่อไฟล์ของ Font

1.3 int AddLink()

เป็นฟังก์ชันสร้าง Link เพื่อใช้ภายในเอกสาร PDF โดยเมื่อมีการ Addlink เข้าไปแล้ว จุดที่ถูก Addlink จะเป็นข้อความที่อำนวยความสะดวกให้ผู้อ่านเอกสารสามารถที่จะกดแล้วโปรแกรม Acrobat Reader จะนำไปยังจุดที่ link ตั้ง ซึ่งจำเป็นต้องใช้ฟังก์ชัน setlink() ก่อน เพื่อที่จะกำหนดว่าจุดใดจะเป็นจุดที่ต้องการให้ link นั้นข้ามไปถึง

1.4 AddPage([string orientation])

เป็นฟังก์ชันขึ้นหน้าเอกสารใหม่ โดยเมื่อเรียกใช้ฟังก์ชันนี้แล้ว ฟังก์ชัน footer() และ ฟังก์ชัน header() ก็จะถูกเรียกอัตโนมัติ และรูปแบบ Font ก่อนการเรียกฟังก์ชันนี้ก็จะถูกนำมา

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นโดยอัตโนมัติ โดยไม่ต้องเรียกใหม่ โดยมี Parameter ให้ใช้กับฟังก์ชันนี้คือ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- orientation : เป็นการเลือกว่าต้องการหน้าใหม่แบบแนวตั้ง p (Portrait) หรือแนวนอน l (Landscape)

1.5 AliasNbPages([string alias])

เป็นฟังก์ชันเพิ่มข้อมูลชื่อปลอมเข้าไปในเอกสาร เพื่อที่จะใช้อ้างอิงหมายเลขหน้าเอกสาร นอกเหนือจากหมายเลขหน้าที่แท้จริง โดยมี Parameter ให้เรียกใช้งานคือ

- alias : เป็นชื่อปลอมของหน้าเอกสารที่ต้องการใช้

1.6 Cell(float w [, float h [, string txt [, mixed border [, int ln [, string align [, int fill [, mixed link]]]]]]])

เป็นฟังก์ชันใส่ข้อมูลตัวอักษรลงในตำแหน่งในหน้าเอกสาร โดยอ้างอิงจากจุดท้ายสุดของ cell ก่อนหน้า หรือถ้าไม่มีก็นับจากจุดบนซ้ายสุดของหน้าเอกสาร แต่ถ้ามีการใช้งานฟังก์ชัน AutoPageBreak ก่อนหน้านี้ ฟังก์ชันนี้จะทำการทำงานหลังจาก AutoPageBreak ซึ่งเทียบเท่ากับการขึ้นหน้าเอกสารใหม่ โดยมี Parameters ให้เรียกใช้ดังต่อไปนี้

- w : ความกว้างของ cell ซึ่งถ้ากำหนดเป็น 0 จะหมายถึงกว้างจนสุดขอบขวาของเอกสาร
- h : ความสูงของ cell ซึ่งถ้ากำหนดเป็น 0 จะหมายถึงสูงเท่าที่ข้อความต้องใช้พื้นที่
- txt : ข้อความที่ต้องการใส่ลงในเอกสาร
- border : ต้องการให้มีกรอบรอบ cell หรือไม่ ถ้าต้องการให้ใส่ค่าเป็น 1 ถ้าไม่ต้องการให้ใส่ค่าเป็น 0 และถ้าต้องการเฉพาะขอบด้านใดด้านหนึ่ง ให้เติม l (Left) หรือ t (Top) หรือ r (Right) หรือ b (Bottom) ได้ทันที
- ln : หลังจากฟังก์ชันนี้ถูกเรียกแล้ว จะให้ current position ไปอยู่ที่จุดใด โดยสามารถเลือกได้คือ 0 (ไปทางขวาของ cell) หรือ 1 (ขึ้นบรรทัดใหม่ซ้ายสุด) หรือ 2 (ขึ้นบรรทัดใหม่ให้อยู่ใต้ cell พอดี)
- align : จัดตำแหน่งข้อความภายใน cell โดยมีค่าให้เลือกใช้คือ l (Left) หรือ c (Center) หรือ r (Right)
- fill : เลือกว่าต้องการให้ cell นี้มีพื้นหลัง หรือว่าจะให้พื้นหลังเป็นแบบใส โดยมีค่าให้เลือกใช้คือ 1 ให้มีพื้นหลัง หรือ 2 ให้พื้นหลังเป็นแบบใส
- link : เป็นข้อมูล URL หรือ ข้อมูล Addlink เพื่อเรียกใช้ภายในเอกสาร

1.7 Close()

เป็นฟังก์ชันสั่งให้จบเอกสาร

1.8 Error(string msg)

เป็นฟังก์ชันที่จะทำงานก็ต่อเมื่อเกิดข้อผิดพลาดขึ้นในระหว่างการประมวลผล โดยสามารถสร้างข้อความรายงานผลข้อผิดพลาดผ่าน msg ได้เอง

1.9 Footer()

เป็นฟังก์ชันสร้าง footer ให้กับเอกสาร โดยฟังก์ชันนี้จะถูกเรียกอัตโนมัติเมื่อมีการขึ้นหน้าใหม่ด้วยฟังก์ชัน AddPage() หรือ จบเอกสารด้วยฟังก์ชัน Close()

1.10 ThaiPDF([string orientation [, string unit [, mixed format]])

เป็นฟังก์ชันเพื่อกำหนดค่าต่างๆจะใช้ต่อไปในฟังก์ชันอื่นๆ โดยมี Parameter ให้ตั้งค่าดังต่อไปนี้

- orientation : รูปแบบหน้ากระดาษเอกสาร ซึ่งสามารถเลือกได้ว่าการแนวตั้ง p (Portrait) หรือแนวนอน l (Landscape)
- unit : หน่วยวัดระยะความกว้าง ยาว สูง สามารถกำหนดค่าได้คือ pt (Point) หรือ mm (millimeter) หรือ cm (centimeter) หรือ in (inch)
- format : ขนาดหน้ากระดาษ สามารถเลือกได้ว่าจะเป็น A3 หรือ A4 หรือ A5 หรือ Letter หรือ Legal

1.11 float GetStringWidth(string s)

เป็นฟังก์ชันคำนวณความสะดวกเพื่อนำผลลัพธ์ที่ได้จากฟังก์ชันนี้ไปใช้ต่อ โดยผลลัพธ์คือความกว้างของข้อความ s ที่ต้องการทราบ

1.12 float GetX()

เป็นฟังก์ชันคำนวณความสะดวกเพื่อนำผลลัพธ์ที่ได้จากฟังก์ชันนี้ไปใช้ต่อ โดยผลลัพธ์คือความระยะตามแนวราบในหน้าเอกสารของ Current Position

1.13 float GetY()

เป็นฟังก์ชันคำนวณความสะดวกเพื่อนำผลลัพธ์ที่ได้จากฟังก์ชันนี้ไปใช้ต่อ โดยผลลัพธ์คือความระยะตามแนวตั้งในหน้าเอกสารของ Current Position

1.14 Header()

เป็นฟังก์ชันสร้าง header ให้กับเอกสาร โดยฟังก์ชันนี้จะถูกเรียกอัตโนมัติเมื่อมีการขึ้นหน้าใหม่ด้วยฟังก์ชัน AddPage()

1.15 Image(string file, float x, float y [, float w [, float h [, string type [, mixed link]]]])

เป็นฟังก์ชันใส่รูปภาพลงไปภายในเอกสาร โดยสามารถระบุตำแหน่ง หรือ เลือกรูปภาพได้จากไฟล์ JPG หรือ PNG มี parameter ให้ใช้ได้ดังต่อไปนี้

- file : ชื่อไฟล์รูปภาพ
- x : ตำแหน่งแนวนอนที่ต้องการให้ภาพไปวางในเอกสาร โดยตำแหน่งที่ระบุนี้จะเป็นจุดบนซ้ายสุดของรูปภาพ
- y : ตำแหน่งแนวตั้งที่ต้องการให้ภาพไปวางในเอกสาร โดยตำแหน่งที่ระบุนี้จะเป็นจุดบนซ้ายสุดของรูปภาพ
- w : ความกว้างของรูปภาพ ถ้าใส่ค่าเป็น 0 จะหมายถึงนำความกว้างของรูปมาใช้โดยอัตโนมัติ
- h : ความสูงของรูปภาพ ถ้าใส่ค่าเป็น 0 จะหมายถึงนำความกว้างของรูปมาใช้โดยอัตโนมัติ
- type : รูปแบบไฟล์รูปภาพ สามารถเลือกเป็น JPG หรือ JPEG หรือ PNG. ถ้าไม่กำหนด ระบบจะนำค่านามสกุลของไฟล์มาใช้
- link : URL หรือ จุดอ้างอิงที่เรียกใช้ภายในเอกสาร PDF ซึ่งกำหนดค่าจาก ฟังก์ชัน AddLink()

1.16 Line(float x1, float y1, float x2, float y2)

เป็นฟังก์ชันทำหน้าที่วาดเส้นระหว่างตำแหน่งสองตำแหน่ง มี parameter ให้ใส่ค่าดังต่อไปนี้

- x1 : ตำแหน่งแนวนอนของจุดแรก
- y1 : ตำแหน่งแนวตั้งของจุดแรก
- x2 : ตำแหน่งแนวนอนของจุดที่สอง
- y2 : ตำแหน่งแนวตั้งของจุดที่สอง

1.17 Link(float x, float y, float w, float h, mixed link)

เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูล link ลงหน้าเอกสาร PDF เพื่อประโยชน์ในการทำให้ข้อความ หรือรูปภาพ เป็นจุดที่สามารถเชื่อมโยงไปยังจุดอื่น จึงสามารถที่จะคลิกแล้วไปยังจุดนั้นๆ ได้ทันที มี parameter ให้ใส่ค่าดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- x : ขอบเขตของตำแหน่งแนวนอนโดยอ้าง ณ.จุดบนซ้ายสุด
- y : ขอบเขตของตำแหน่งแนวตั้งโดยอ้าง ณ.จุดบนซ้ายสุด
- w : ความกว้างของขอบเขต
- h : ความสูงของขอบเขต
- link : URL หรือ จุดอ้างที่ที่เรียกใช้ภายในเอกสาร PDF ซึ่งกำหนดค่าจาก ฟังก์ชัน AddLink()

1.18 Ln([float h])

เป็นฟังก์ชันทำหน้าที่แทรกบรรทัดลงในหน้าเอกสาร สามารถที่จะเลือกได้ว่าบรรทัดที่จะแทรกนั้นมีความสูงเท่าไรได้โดยตั้งค่าผ่าน h

1.19 MultiCell(float w, float h, string txt [, mixed border [, string align [, int fill]])

เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูลตัวอักษรลงในตำแหน่งหน้าเอกสาร พร้อมด้วยข้อมูลจบบรรทัดอัตโนมัติที่ที่หมดบรรทัด โดยอ้างอิงจากจุดท้ายสุดของ cell ก่อนหน้า หรือถ้าไม่มีก็นับจากจุดบนซ้ายสุดของหน้าเอกสาร มี Parameters ให้เรียกใช้ดังต่อไปนี้

- w : ความกว้างของ cell ซึ่งถ้ากำหนดเป็น 0 จะหมายถึงกว้างจนสุดขอบขวาของเอกสาร
- h : ความสูงของ cell ซึ่งถ้ากำหนดเป็น 0 จะหมายถึงสูงเท่าที่ข้อความต้องใช้พื้นที่
- txt : ข้อความที่ต้องการใส่ลงในเอกสาร
- border : ต้องการให้มีกรอบครอบ cell หรือไม่ ถ้าต้องการให้ใส่ค่าเป็น 1 ถ้าไม่ต้องการให้ใส่ค่าเป็น 0 และถ้าต้องการเฉพาะขอบด้านใดด้านหนึ่ง ให้เติม 1 (Left) หรือ t (Top) หรือ r (Right) หรือ b (Bottom) ได้ทันที
- ln : หลังจกฟังก์ชันนี้ถูกเรียกแล้ว จะให้ current position ไปอยู่ที่จุดใด โดยสามารถเลือกได้คือ 0 (ไปทางขวาของ cell) หรือ 1 (ขึ้นบรรทัดใหม่ซ้ายสุด) หรือ 2 (ขึ้นบรรทัดใหม่ให้อยู่ได้ cell พอดี)
- align : จัดตำแหน่งข้อความภายใน cell โดยมีค่าให้เลือกใช้คือ l (Left) หรือ c (Center) หรือ r (Right)
- fill : เลือกว่าต้องการให้ cell นี้มีพื้นหลัง หรือว่าจะให้พื้นหลังเป็นแบบใส โดยมีค่าให้เลือกใช้คือ 1 ให้มีพื้นหลัง หรือ 2 ให้พื้นหลังเป็นแบบใส
- link : เป็นข้อมูล URL หรือ ข้อมูล Addlink เพื่อเรียกใช้ภายในเอกสาร

1.20 string Output([string name [, string dest]])

เป็นฟังก์ชันทำหน้าที่สั่งให้ระบบประมวลผล แล้วส่งออกผลลัพธ์เป็นไฟล์ หรือว่า จะส่งไปยังบราวเซอร์ มี Parameters ให้เรียกใช้ดังต่อไปนี้

- name : ชื่อไฟล์ที่ต้องการให้ระบบส่งออก ถ้าไม่ใส่ระบบจะตั้งชื่อไฟล์เป็น doc.pdf
- dest : รูปแบบการส่งออกผลลัพธ์ สามารถเลือกตั้งได้ว่า

I : ส่งไฟล์ผ่านเว็บเบราว์เซอร์จากนั้นให้เว็บเบราว์เซอร์ใช้โปรแกรม Acrobat Reader เปิดทันที

D : ส่งไฟล์ผ่านเว็บเบราว์เซอร์จากนั้นให้เว็บเบราว์เซอร์ทำหน้าที่ Download ไฟล์ผลลัพธ์

F : บันทึกไฟล์ลงใน path ของเครื่องเซิร์ฟเวอร์

S : ส่งข้อมูลผลลัพธ์หลังจากการประมวลผลไปยังเว็บเบราว์เซอร์โดยตรง

1.21 int pageNo()

เป็นฟังก์ชันทำหน้าที่ให้ผลลัพธ์ค่าเลขที่หน้าปัจจุบันที่ Current Position อยู่

1.22 Rect(float x, float y, float w, float h [, string style])

เป็นฟังก์ชันวาดสี่เหลี่ยมในหน้าเอกสาร โดยมี Parameters ให้เรียกใช้ดังต่อไปนี้

- x : ตำแหน่งแนวนอนของจุดมุมบนซ้ายของสี่เหลี่ยม
- y : ตำแหน่งแนวตั้งของจุดมุมบนซ้ายของสี่เหลี่ยม
- w : ความกว้างของสี่เหลี่ยม
- h : ความสูงของสี่เหลี่ยม
- style : ลักษณะของสี่เหลี่ยม โดยสามารถเลือกได้ว่าเป็น Draw (ให้วาดขอบของสี่เหลี่ยม) หรือ Fill (ให้ใส่พื้นหลัง)

1.23 SetAuthor(string author)

เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูลอ้างอิงว่าใครเป็นเจ้าของไฟล์นี้ด้วย author

1.24 SetAutoPageBreak(boolean auto [, float margin])

เป็นฟังก์ชันทำหน้าที่ให้ระบบทำการแทรกข้อมูลสิ้นสุดหน้ากระดาษของแต่ละแผ่นโดยอัตโนมัติ สามารถเลือกได้ว่าจะเป็น auto หรือ ใช้การคำนวณระยะห่าง margin จากด้านล่างของหน้ากระดาษ

1.25 SetCompression(boolean compress)

เป็นฟังก์ชันที่ทำหน้าที่บีบอัดข้อมูลเอกสาร โดยจะทำงานร่วมกับ Zlib ซึ่งเป็นโปรแกรมที่มีอยู่ในระบบ PHP

1.26 SetCreator(string creator)

เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูลผู้สร้างไฟล์ลงในไฟล์เอกสาร PDF

1.27 SetDisplayMode(mixed zoom [, string layout])

เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูลอ้างอิงให้โปรแกรม Acrobat Reader เรียกใช้ในการเปิดอ่านไฟล์เอกสาร สามารถตั้งค่า Parameters ได้ดังต่อไปนี้

- zoom : ระยะใกล้หรือไกลในการแสดงผลด้วยโปรแกรม Acrobat Reader
 - fullpage: แสดงหน้าจอต่อนหน้าเอกสาร
 - fullwidth: แสดงหน้าจอตอทั้งความกว้างหน้าเอกสาร
 - real: แสดงตามขนาดจริงของเอกสาร เทียบเท่ากับซูมด้วย 100%
 - default: ใช้ค่าที่ Acrobat Reader ตั้งไว้แล้ว
- layout : รูปแบบการแสดงผลเอกสารต่อเนื่อง
 - single: แสดงทีละหน้าไม่มีการต่อเนื่อง
 - continuous: แสดงต่อเนื่องทีละหน้า
 - two: แสดงทีละสองหน้า
 - default: ใช้ค่าที่ Acrobat Reader ตั้งไว้แล้ว

1.28 SetDrawColor(int r [, int g, int b])

เป็นฟังก์ชันทำหน้าที่ตั้งค่าสีที่ต้องการสำหรับวัตถุต่างๆในเอกสาร สามารถส่งค่า Parameter ได้ดังต่อไปนี้

- r : ผสมสีแดงมีค่าตั้งแต่ 0 - 255
- g : ผสมสีเขียวมีค่าตั้งแต่ 0 - 255
- b : ผสมสีน้ำเงินมีค่าตั้งแต่ 0 - 255

1.29 SetFillColor(int r [, int g, int b])

เป็นฟังก์ชันทำหน้าที่ตั้งค่าสีที่ต้องการสำหรับพื้นหลังของวัตถุต่างๆในเอกสาร สามารถรับค่า Parameter ได้ดังต่อไปนี้

- r : ผสมสีแดงมีค่าตั้งแต่ 0 - 255

- g : ผสมสีเขียวมีค่าตั้งแต่ 0 - 255
- b : ผสมสีน้ำเงินมีค่าตั้งแต่ 0 - 255

1.30 SetFont(string family [, string style [, float size]])

เป็นฟังก์ชันกำหนดชื่อ Font เพื่อให้ใส่ข้อมูลลงเอกสารตามขนาดรูปแบบที่ Font ที่กำหนด สามารถรับค่า Parameter ได้ดังต่อไปนี้

- family : ชื่อ Font
- style : รูปแบบ Font เช่น r (Regular) b (Bold) i (Italic) และ bi (Bold Italic)
- size : ขนาดของ Font

1.31 SetFontSize(float size)

เป็นฟังก์ชันกำหนดขนาด Font เพื่อให้ใส่ข้อมูลลงเอกสารตามขนาด Font ที่กำหนด

1.32 SetKeywords(string keywords)

เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูล keyword ลงในไฟล์เอกสาร PDF สำหรับให้โปรแกรมค้นหาต่างๆ ได้นำไปใช้

1.33 SetLeftMargin(float margin)

เป็นฟังก์ชันทำหน้าที่กำหนดระยะขอบซ้ายของหน้าเอกสาร

1.34 SetLineWidth(float width)

เป็นฟังก์ชันทำหน้าที่กำหนดความกว้างของเส้นที่จะวาด ถ้าไม่กำหนด ระบบจะตั้งเป็น 0.2 mm

1.35 SetLink(int link [, float y [, int page]])

เป็นฟังก์ชันทำหน้าที่กำหนดจุดที่ต้องการให้ Link ทำการกระโดดไปถึง สามารถรับค่า Parameter ได้ดังต่อไปนี้

- link : ข้อความที่ต้องการให้แสดง
- y : ระยะแนวตั้งของหน้าที่ต้องการให้ Link กระโดดไปถึง
- page : หมายเลขหน้าที่ต้องการให้ Link กระโดดไปถึง

1.36 SetMargins(float left, float top [, float right])

เป็นฟังก์ชันทำหน้าที่กำหนดระยะของ Margin หน้ากระดาษทุกๆหน้าของเอกสาร ซึ่งถ้าไม่ตั้งระบบจะกำหนดเป็น 10mm สำหรับ ขอบซ้าย ขอบบน ขอบขวา ซึ่งสามารถรับค่า Parameter ได้ดังต่อไปนี้

- left : ระยะขอบซ้าย
- top : ระยะขอบบน
- right : ระยะขอบขวา

1.37 SetRightMargin(float margin)

เป็นฟังก์ชันทำหน้าที่กำหนดระยะของ Margin ขอบขวาของหน้าเอกสารที่ Currency Position อยู่

1.38 SetSubject(string subject)

เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูลชื่อเรื่องลงในไฟล์เอกสาร PDF สำหรับให้โปรแกรมค้นหาต่างๆได้นำไปใช้

1.39 SetTextColor(int r [, int g, int b])

เป็นฟังก์ชันทำหน้าที่ตั้งค่าสีที่ต้องการสำหรับข้อความในเอกสาร โดยสามารถรับค่า Parameter ได้ดังต่อไปนี้

- r : พสมสีแดงมีค่าตั้งแต่ 0 - 255
- g : พสมสีเขียวมีค่าตั้งแต่ 0 - 255
- b : พสมสีน้ำเงินมีค่าตั้งแต่ 0 - 255

1.40 SetTitle(string.title)

เป็นฟังก์ชันทำหน้าที่ใส่ข้อมูลชื่อของเอกสารลงในไฟล์เอกสาร PDF สำหรับให้โปรแกรมค้นหาต่างๆได้นำไปใช้

1.41 SetTopMargin(float margin)

เป็นฟังก์ชันทำหน้าที่กำหนดระยะของ Margin ขอบบนของหน้าเอกสารที่ Currency Position อยู่

1.42 SetX(float x)

เป็นฟังก์ชันเลื่อน Current Position ไปยังตำแหน่งในแนวราบ ณ.บรรทัดที่ Current Position อยู่

1.43 SetXY(float x, float y)

เป็นฟังก์ชันเลื่อน Current Position ไปยังตำแหน่งในหน้าเอกสาร โดยสามารถรับค่า Parameter ได้ดังต่อไปนี้

- x : ตำแหน่งแนวนอนที่ต้องการให้ Current Position ไป
- y : ตำแหน่งแนวตั้งที่ต้องการให้ Current Position ไป

1.44 SetY(float y)

เป็นฟังก์ชันเลื่อน Current Position ไปยังตำแหน่งในแนวตั้ง

1.45 Text(float x, float y, string txt)

เป็นฟังก์ชันให้ใส่ข้อความลงในเอกสาร โดยสามารถระบุได้ว่าจะใส่ลง ณ.ตำแหน่งใด ซึ่งสามารถรับค่า Parameter ได้ดังต่อไปนี้

- x : ตำแหน่งแนวนอนที่ต้องการ
- y : ตำแหน่งแนวตั้งที่ต้องการ
- txt : ข้อความที่ต้องการใส่ในเอกสาร

1.46 Write(float h, string txt [, mixed link])

เป็นฟังก์ชันให้ใส่ข้อความลงในเอกสาร ณ.ตำแหน่งที่ Current Position อยู่ โดยสามารถรับค่า Parameter ได้ดังต่อไปนี้

- h : ความสูงของบรรทัดที่ข้อความอยู่
- txt : ข้อความที่ต้องการใส่ในเอกสาร
- link : URL หรือ จุดอ้างอิงที่เรียกใช้ภายในเอกสาร PDF ซึ่งกำหนดค่าจาก ฟังก์ชัน AddLink()

ประวัติผู้เขียน

ชื่อผู้เขียน	นายสุทธิศักดิ์ เหมาะะประไพพันธุ์
วันเกิด	29 มีนาคม พ.ศ. 2521
สถานที่เกิด	จังหวัดกรุงเทพมหานคร
ประวัติการศึกษา	
ระดับประถมศึกษา	โรงเรียนอัสสัมชัญ สำโรง
ระดับมัธยมศึกษาตอนต้น	โรงเรียนอัสสัมชัญ สำโรง
ระดับปริญญาตรี	คณะวิศวกรรมศาสตร์ ภาควิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยอัสสัมชัญ
ประสบการณ์การทำงาน	บริษัท ซีเกต เทคโนโลยี จำกัด บริษัท เบนคิว (ไทยแลนด์) จำกัด บริษัท ชิสเต็มดีช่ายเจแปน จำกัด บริษัท ชัน โย (ไทยแลนด์) จำกัด บริษัท เคเอสซี อินเทอร์เน็ต จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้