

การพัฒนาแอปพลิเคชันสำหรับธุรกิจ
THE DEVELOPED APPLICATION FOR BUSINESS

โดย

นางสาวชิตชนก แพทย์กิจ
นางสาวชุตานา สถาพรเพ็ญชาติ
นางสาวศุภสุดา อยู่แสนสุข

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

การพัฒนาแอปพลิเคชันสำหรับธุรกิจ
THE DEVELOPED APPLICATION FOR BUSINESS

โดย

นางสาวชิตชนก	แพทย์กิจ	56010308
นางสาวชุตานา	สถาพรเพ็ญชาติ	56010319
นางสาวศุภสุดา	อยู่แสนสุข	56011256

อาจารย์ที่ปรึกษา

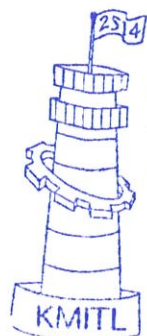
รศ.ดร.สุวิพล สิริชีวะภาค

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



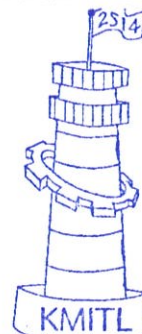
ผ่านการตรวจรูปเล่มแล้ว

(.....)
อาจารย์ที่ปรึกษา

19/พ.ย. 60

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ปีการศึกษา 2559



ผ่านการตรวจชิ้นงานแล้ว

(.....)
กรรมการผู้ตรวจชิ้นงาน

19/พ.ย. 60

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ปริญญาานิพนธ์ปีการศึกษา 2559

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาแอปพลิเคชันสำหรับธุรกิจ

THE DEVELOPED APPLICATION FOR BUSINESS

ผู้จัดทำ

- | | | | |
|----|---------------|---------------|----------|
| 1. | นางสาวชิตชนก | แพทย์กิจ | 56010308 |
| 2. | นางสาวชุตานา | สถาพรเพ็ญชาติ | 56010319 |
| 3. | นางสาวศุภสุดา | อยู่แสนสุข | 56011256 |

.....

(รศ.ดร.สุวิพล สิริธิชีวะภาค)

อาจารย์ที่ปรึกษา

กิตติกรรมประกาศ

ปริญญาานิพนธ์ครั้งนี้สำเร็จลุล่วงได้ด้วยดี ด้วยคำแนะนำ และคำปรึกษาจากอาจารย์ที่ปรึกษา รศ.ดร.สุวิพล สิริชีวะภาค และขอบคุณ คุณกมล ธีระกาญจน์ ที่ให้คำปรึกษา คณะผู้จัดทำขอขอบพระคุณเป็นสูง และขอบคุณเพื่อนๆที่คอยให้กำลังใจ และคำปรึกษาเกี่ยวกับโครงการจนสำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ผู้จัดทำขอขอบพระคุณ บิดา มารดา และครอบครัวของคณะผู้จัดทำที่ให้การสนับสนุนช่วยเหลือผู้จัดทำ และให้กำลังใจตลอดมา คณะผู้จัดทำหวังว่าปริญญาานิพนธ์เรื่องนี้จะมีส่วนประโยชน์และสามารถนำไปประยุกต์ใช้สืบต่อไป

นางสาวชิตชนก	แพทย์กิจ
นางสาวชุตานา	สถาพรเพ็ญชาติ
นางสาวศุภสุดา	อยู่แสนสุข

การพัฒนาแอปพลิเคชันสำหรับธุรกิจ

THE DEVELOPED APPLICATION FOR BUSINESS

โดย	นางสาวชิตชนก	แพทย์กิจ	56010308
	นางสาวชุตานา	สถาพรเพ็ญชาติ	56010319
	นางสาวศุภสุดา	อยู่แสนสุข	56011256

อาจารย์ที่ปรึกษา รศ.ดร.สุวิพล ลิทธิชีวะภาค

บทคัดย่อ

ปฏิญานิพนธ์การพัฒนาแอปพลิเคชันสำหรับการประกอบธุรกิจนี้ เป็นการศึกษาวิจัยและพัฒนาเพื่อให้แอปพลิเคชันสามารถใช้งานได้จริง โดยจะเห็นว่าในปัจจุบันแอปพลิเคชันที่ใช้ในธุรกิจอีคอมเมิร์ซนั้นมีอยู่อย่างแพร่หลาย ซึ่งในปฏิญานิพนธ์นี้จะเน้นทางด้านการใช้ข้อมูลสินค้าที่ละเอียด เพื่อให้ผู้บริโภคนั้นมั่นใจและยอมรับในตัวผลิตภัณฑ์ที่นำมาจำหน่ายซึ่งจุดเด่นของแอปพลิเคชันนี้จะอยู่ที่การให้ข้อมูลที่น่าเชื่อถือ มีขั้นตอนการสั่งซื้อ การจ่ายเงินที่ปลอดภัย และอำนวยความสะดวกแก่ผู้ใช้งานในแง่ของการประหยัดเวลารวมถึงค่าใช้จ่ายในการเดินทางได้อีกด้วย และระบบหลังร้านที่ช่วยให้การบริการจัดการสินค้าง่ายต่อผู้ประกอบการซึ่งผู้ประกอบการสามารถตรวจสอบคลังสินค้าได้ผ่านทางแอปพลิเคชัน ทั้งยังแสดงสถิติของสินค้ายอดนิยมได้อัตโนมัติอีกด้วย

ABSTRACT

This thesis The Developed Application for Business. This are education research and development so that the application can actually work. You will see that Applications that use the e-commerce business are widely available. The project will focus on using information to give consumers the confidence and acceptance of the products, which were sold. Highlights of the app, this option will be available to provide the information credible, the process of ordering and paying

is safe and convenient for customers, to save time and expense of travel. And Back-N that help easy product management services to entrepreneurs. The operator can check the stock via the application. It also shows the statistics of popular products automatically.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VIII
สารบัญตาราง	XI
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 ภาษาจาวา	2
2.1.1 ข้อดีของภาษาจาวา	2
2.1.2 ข้อเสียของ ภาษาจาวา	3
2.2 ประวัติความเป็นมาของภาษาพีเอชพี	3
2.2.1 ภาษาพีเอชพี	4
2.2.2 คุณสมบัติของภาษาพีเอชพี	4
2.2.3 การรองรับพีเอชพี	5
2.2.4 คำสั่งเบื้องต้น	6
2.3 ภาษาเอชทีเอ็มแอล (HTML)	8
2.3.1 การใช้งานเบื้องต้นภาษาเอชทีเอ็มแอล	9
2.3.2 โครงสร้างของหลักของเอชทีเอ็มแอล	9
2.3.3 การแสดงผลที่เว็บเบราว์เซอร์	10

สารบัญ (ต่อ)

	หน้า
2.4 ภาษาเอสคิวแอล (SQL)	10
2.5 อาปาเช่ (APACHE)	11
2.6 ระบบฐานข้อมูล (Database System)	12
2.7 โมเดล วิว คอนโทรลเลอร์ (MODEL VIEW CONTROLLER)	13
2.7.1 ขั้นตอนการทำงานของ MVC	13
2.8 ระบบปฏิบัติการแอนดรอยด์	13
2.8.1 ประวัติความเป็นมา	14
2.8.2 โครงสร้างของแอนดรอยด์	14
2.8.3 โปรแกรมและเครื่องมือในการพัฒนา	17
2.8.4 การทดลองรันแอปพลิเคชัน	18
2.8.5 การทำงานของแอกติวิตี้	18
2.8.6 การจัดเก็บข้อมูลในแอกติวิตี้	19
2.8.7 การใช้งานผู้ดำเนินการ (Handler)	19
2.9 เว็บเซิร์ฟเวอร์	19
บทที่ 3	
การออกแบบและการจัดทำปฏิญญานิพนธ์	20
3.1 การออกแบบ	20
3.1.1. ภาพรวมของระบบ	20
3.1.2. บล็อกไดอะแกรมรวมการทำงานของเว็บไซต์	21
3.1.3 การเก็บฐานข้อมูลเบื้องต้น	22
3.1.4 ออกแบบฐานข้อมูลสินค้า	23
3.1.4.1 ทำการออกแบบฐานข้อมูลสินค้า	23
3.1.4.2 ทำการสร้างฐานข้อมูลสินค้าสร้างฐานข้อมูล	25
3.1.5 ลงระบบปฏิบัติการให้เซิร์ฟเวอร์	25
3.1.5.1 ทำการลงระบบปฏิบัติการให้แก่เซิร์ฟเวอร์	25

สารบัญ (ต่อ)

	หน้า
3.1.6 เว็บไซต์ (WEBSITE)	26
3.1.6.1 ทดลองเขียนหน้าเว็บโดยใช้ภาษา HTML	26
3.1.6.2 ปรับแต่ง PHP ให้รันพร้อม FRAMEWORK	27
3.1.6.3 สร้างหน้าควบคุมเพื่อเรียกใช้งานหน้าเว็บเพจ	28
3.1.6.4 หน้าแรกโดยใช้ MVC	28
3.1.6.5 หน้าลงชื่อเข้าใช้	29
3.1.6.6 หน้ารายละเอียดสินค้า	30
3.1.6.7 หน้าตะกร้าสินค้า	31
3.1.6.8 หน้าเช็คเอาท์	32
3.1.6.9 หน้า BACK-END	33
3.1.6.10 หน้า BACK-END	34
3.1.6.11 หน้า BACK-END เพิ่มสินค้า	35
3.1.7 การจัดทำแอปพลิเคชัน	35
3.1.7.1 บล็อกไดอะแกรมการทำงานแอปพลิเคชัน	35
3.1.7.2 หน้า ACTIVITY SPLASH SCREEN	38
3.1.7.3 หน้า ACTIVITY_MAIN.	39
3.1.7.4 หน้าสมัครสมาชิก	40
3.1.7.5 หน้าสินค้าของฉัน	41
3.1.7.6 หน้าแจ้งการโอนเงิน	43
3.1.7.7 หน้าวิธีการโอนเงิน	44
3.1.7.8 หน้าวิธีการสั่งซื้อ	45
3.2 เครื่องมือที่ใช้ในการทดลอง	46
3.3 การจัดเก็บผลการทดลอง	47
3.3.3 ทดสอบการนำข้อมูลมาแสดง	47
3.3.3.1 ทดสอบบนเว็บไซต์	47
3.3.3.2 ทดสอบบนแอปพลิเคชัน	47

สารบัญ (ต่อ)

		หน้า
บทที่ 4	ผลการทดลอง	48
	4.1 ผลการดำเนินงานของเว็บไซต์	48
	4.1.1 การเก็บฐานข้อมูลเบื้องต้น	48
	4.1.2 ผลในการสร้างตารางฐานข้อมูลในเซิร์ฟเวอร์	49
	4.2 ผลการทดสอบ	51
	4.2.1 ผลการทดลองออกแบบหน้าเว็บโดยใช้ภาษาHTML	51
	4.2.2 ผลการทดลองจากการสร้างหน้าแรก	51
	4.2.3 ผลการทดลองสร้างหน้าล็อกอินสำหรับหน้าแรก	52
	4.2.4 ผลการทดลองสร้างหน้ารายละเอียดสินค้า	53
	4.2.5 ข้อมูลจากฐานข้อมูลเมื่อมีการกดสินค้าเข้าตะกร้า	53
	4.2.6 การแสดงผลหน้าเช็คเอาท์	55
	4.2.7 ผลการทดสอบหน้าล็อกอินเข้าสู่หน้าแอดมิน	57
	4.3 ผลการดำเนินงานแอปพลิเคชัน	58
	4.3.1 หน้า ACTIVITY SPLASH SCREEN	58
	4.3.2 หน้า ACTIVITY_MAIN.	59
	4.3.3 หน้าสมัครสมาชิกและข้อมูลสมาชิก	59
	4.2.4 หน้าสินค้าของฉัน	60
	4.2.5 หน้าแจ้งการโอนเงิน	60
	4.2.6 หน้าวิธีการโอนเงิน	61
	4.2.7 หน้าวิธีการสั่งซื้อ	61
บทที่ 5	สรุปผลและข้อเสนอแนะ	62
	5.1 สรุปผล	62
	5.2 ข้อเสนอแนะ	62
บรรณานุกรม		63
ภาคผนวก		64

สารบัญรูป

รูปที่		หน้า
2.1	โครงสร้างของระบบปฏิบัติการแอนดรอยด์	14
2.2	การทำงานของแคตติวิตี	17
3.1	ผังการทำงานของระบบ	20
3.2	บล็อกไดอะแกรมรวมการทำงานของเว็บไซต์	21
3.3	XAMPP CONTROL PANEL	22
3.4	SERVER จำลองในเครื่องคอมพิวเตอร์	25
3.5	ลงระบบปฏิบัติการให้กับเซิร์ฟเวอร์	25
3.6	ภาษา HTML ในการสร้างรูปแบบของเว็บไซต์	26
3.7	ภาษา HTML ในการสร้างรูปแบบของเว็บไซต์	26
3.8	MODEL VIEW CONTROLLER (MVC) (ขยายรูปที่)	27
3.9	สร้างตารางฐานข้อมูลสินค้า	27
3.10	สร้างหน้าควบคุมเพื่อเรียกใช้งานหน้าเว็บเพจ	28
3.11	ผังงานการทำงานของกรเรียกหน้าเว็บเพจมาแสดง	28
3.12	ผังงานการทำงานของหน้าลือกอินหน้าเว็บเพจ	29
3.13	ผังงานแสดงการทำงานเมื่อกดเข้ามาที่สินค้า	30
3.14	ผังงานแสดงการทำงานเมื่อมีการกดสั่งซื้อเพิ่มคำสั่งซื้อลงในดาต้าเบส	31
3.15	ผังงานแสดงการทำงานเมื่อต้องการชำระเงิน	33
3.16	ผังงานการทำงานของหน้าลือกอินหน้าBACK-END	34
3.17	ผังงานแสดงการทำงานตรวจสอบการโอนเงิน	39
3.18	ผังงานแสดงการทำงานการเพิ่มสินค้าของผู้ประกอบการ	35

3.19	ผังการทำงานของแอปพลิเคชัน (ก)	36
3.20	ผังการทำงานของแอปพลิเคชัน (ข)	37
3.21	ผังการทำงานของแอปพลิเคชันหน้า SPLASH SCREEN	38
3.22	การออกแบบแอปพลิเคชันหน้า SPLASH SCREEN	38
3.23	ผังการทำงานของแอปพลิเคชันหน้า ACTIVITY_MAIN.	39
3.24	การออกแบบแอปพลิเคชันหน้า ACTIVITY_MAIN.	39
3.25	ผังการทำงานของแอปพลิเคชันหน้าสมัครสมาชิก	40
3.26	การออกแบบแอปพลิเคชันหน้าสมัครสมาชิกและข้อมูลสมาชิก	41
3.27	ผังการทำงานของแอปพลิเคชันหน้าสินค้าของฉัน	42
3.28	การออกแบบแอปพลิเคชันหน้าสินค้าของฉัน	42
3.28	ผังการทำงานของแอปพลิเคชันหน้าแจ้งการโอนเงิน	43
3.29	การออกแบบแอปพลิเคชันหน้าแจ้งการโอนเงิน	43
3.30	ผังการทำงานของแอปพลิเคชันหน้าวิธีการโอนเงิน	44
3.31	การออกแบบแอปพลิเคชันหน้าวิธีการโอนเงิน	44
3.32	ผังการทำงานของแอปพลิเคชันหน้าวิธีการสั่งซื้อ	45
3.33	การออกแบบแอปพลิเคชันหน้าวิธีการสั่งซื้อ	45
4.1	ฐานข้อมูลในเซิร์ฟเวอร์จำลอง	48
4.2	จากตารางที่ 3.1 การออกแบบฐานข้อมูล PRODUCT_DATA	49
4.3	จากตารางที่ 3.2 การออกแบบตารางฐานข้อมูล PRODUCT_ORDER	49
4.4	จากตารางที่ 3.3 การออกแบบตารางฐานข้อมูล CART	50
4.5	จากตารางที่ 3.4 การออกแบบตารางฐานข้อมูล USER_ID	50
4.6	หน้าเว็บไซต์ที่ได้ทำการออกแบบขึ้นมาเบื้องต้น	51
4.7	ผลการทดลองแสดงผลหน้าแรกของเว็บไซต์	51
4.8	ผลการทดลองจากฐานข้อมูล	52
4.9	หน้าล็อกอินเข้าสู่ระบบ	52

4.10	แสดงผลหน้ารายละเอียดสินค้าเมื่อกดเข้ามาที่สินค้า	53
4.11	แสดงการทำงานเมื่อมีการกดสั่งซื้อเพิ่มคำสั่งซื้อลงในตาต้าเบส	53
4.12	ฐานข้อมูลแสดงเมื่อมีการกดสั่งซื้อมาที่หน้าตะกร้าสินค้า	54
4.13	แสดงเมื่อมีการกดสั่งซื้อมาที่หน้าตะกร้าสินค้า	54
4.14	หน้าแสดงการกรอกข้อมูลที่อยู่การออกใบเสร็จ	55
4.15	หน้าแสดงการกรอกข้อมูลที่อยู่จัดส่ง	55
4.16	แสดงหน้าวิธีการชำระเงิน	56
4.17	แสดงหน้าตรวจสอบรายละเอียดต่างๆ	56
4.18	หน้าล็อกอินเข้าสู่หน้าแอดมิน	57
4.19	แสดงผลเมื่อล็อกอินผิดพลาด	57
4.20	แสดงผลเมื่อล็อกอินเข้าสู่ระบบสำเร็จ	58
4.21	แอปพลิเคชันหน้า SPLASH SCREEN	58
4.22	แอปพลิเคชันหน้า ACTIVITY_MAIN.	59
4.23	แอปพลิเคชันหน้าสมัครสมาชิกและข้อมูลสมาชิก	59
4.24	แอปพลิเคชันหน้าสินค้าของฉัน	60
4.25	แอปพลิเคชันหน้าแจ้งการโอนเงิน	60
4.26	แอปพลิเคชันหน้าวิธีการโอนเงิน	61
4.27	แอปพลิเคชันหน้าวิธีการสั่งซื้อ	61

สารบัญตาราง

ตารางที่		หน้า
1	การออกแบบฐานข้อมูล PRODUCT_DATA	23
2	การออกแบบตารางฐานข้อมูล PRODUCT_ORDERการออกแบบตาราง	23
3	การออกแบบฐานข้อมูล CART	24
4	การออกแบบตารางฐานข้อมูล USER_ID	24

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันนี้ ผู้คนมากมายต่างกำลังให้ความสนใจและใช้งานสิ่งที่เรียกว่า “แอปพลิเคชัน” กันอย่างแพร่หลาย เราอาจจะไม่เข้าใจความหมายของมันว่ามันคืออะไรกันแน่ “แอปพลิเคชัน” คือซอฟต์แวร์ประเภทหนึ่งที่ทำให้เราสามารถกระทำการบางอย่างได้ตามความต้องการของเรา แอปพลิเคชันสำหรับใช้งานบนเครื่องคอมพิวเตอร์ตั้งโต๊ะและโน้ตบุ๊กนั้น เรียกว่า “เดสก์ท็อป แอปพลิเคชัน (Desktop Applications)” ส่วนแอปพลิเคชันที่ทำงานบนเครื่องอุปกรณ์พกพาทั้งหลาย เรียกว่า “โมบายล์ แอปพลิเคชัน (Mobile Applications)” เมื่อเรารันแอปพลิเคชัน มันจะทำงานอยู่ภายใต้ระบบปฏิบัติการตลอดเวลาจนกว่าเราจะทำการปิดมันไป ภายในเวลาเดียวกัน อาจมีหลาย แอปพลิเคชันที่กำลังทำงานพร้อมกันในระบบปฏิบัติการ เราเรียกกระบวนการนี้ว่า “มัลติแทสกกิง (Multitasking)”

ในปัจจุบันแอปพลิเคชันมีบทบาทในการดำรงชีวิตของมนุษย์มากขึ้นจะเห็นได้จากการนำแอปพลิเคชัน มาใช้งานในหลายด้านด้วยกัน ในที่นี้รวมถึงธุรกิจอีคอมเมิร์ซ หรือการดำเนินธุรกิจโดยใช้สื่ออิเล็กทรอนิกส์ เพื่อให้บรรลุเป้าหมายทางธุรกิจที่องค์กรได้วางไว้ โดยธุรกิจอีคอมเมิร์ซสามารถเปิดดำเนินการค้า 24 ชั่วโมง ดำเนินการค้าอย่างไร้พรมแดนทั่วโลก ใช้งบประมาณลงทุนน้อย ตัดปัญหาด้านการเดินทาง และง่ายต่อการประชาสัมพันธ์โดย สามารถประชาสัมพันธ์ได้ทั่วโลก

เมื่อพิจารณาแล้วเห็นว่า “แอปพลิเคชันสำหรับการกระจายสินค้าทั่วไป” นั้นจะทำให้เราก้าวทันธุรกิจอีคอมเมิร์ซในตลาดโลกมากยิ่งขึ้น ดังนั้นวิศวกรรมศาสตรบัณฑิตสาขาวิศวกรรมโทรคมนาคม ฉบับนี้ จึงมุ่งศึกษาการจัดทำเว็บไซต์ และการทำแอปพลิเคชัน เพื่อสามารถนำไปศึกษาและพัฒนาแอปพลิเคชันอีคอมเมิร์ซที่ล้ำสมัยต่อไป

1.2 วัตถุประสงค์

1. เพื่อให้เว็บไซต์และแอปพลิเคชันที่สร้างขึ้นสามารถใช้งานได้จริง
2. เพื่อจัดทำหน้าเว็บไซต์โดยใช้ภาษาพีเอชพี
3. เพื่อจัดทำแอปพลิเคชันสำหรับมือถือโดยใช้โปรแกรมแอนดรอยด์สตูดิโอ

1.3 ขอบเขตของปริญญาบัตร

1. ศึกษาเกี่ยวกับการจัดทำแอปพลิเคชันโดยโปรแกรมแอนดรอยด์สตูดิโอ
2. ศึกษาการดูแลฐานข้อมูล
3. ทำแอปพลิเคชันเพื่อให้ใช้งานได้จริง

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ภาษาจาวา

จาวา(Java) ภาษาโปรแกรมเชิงวัตถุ พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่บริษัท ซัน ไมโครซิสเต็มส์ ภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส(C++) โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทน จุดเด่นของภาษา Java อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) มาพัฒนาโปรแกรมของตนด้วยจาวาได้

ภาษาจาวาเป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้นคลาสคือที่เก็บเมทอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม (Behavior)

2.1.1 ข้อดีของภาษาจาวา

- ภาษาจาวา เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบสมบูรณ์ ซึ่งเหมาะสำหรับพัฒนาระบบที่มีความซับซ้อน การพัฒนาโปรแกรมแบบวัตถุจะช่วยให้เราสามารถใช้อำนาจหรือชื่อ ต่าง ๆ ที่มีอยู่ในระบบงานนั้นมาใช้ในการออกแบบโปรแกรมได้ ทำให้เข้าใจได้ง่ายขึ้น
- โปรแกรมที่เขียนขึ้นโดยใช้ภาษาจาวา จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน ไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่องรันโปรแกรมนั้นก็สามารถถูกประมวลผลและ รันบนเครื่องพีซีธรรมดาได้
- ภาษาจาวามีการตรวจสอบข้อผิดพลาดทั้งตอนเวลาประมวลผลและ เวลาทำงานทำให้ลดข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรม และช่วยให้ดีบัคโปรแกรมได้ง่าย
- ภาษาจาวามีความซับซ้อนน้อยกว่าภาษาซีพลัสพลัส เมื่อเปรียบเทียบโค้ด ของโปรแกรมที่เขียนขึ้นโดยภาษาจาวา กับภาษาซีพลัสพลัส พบว่าโปรแกรมที่เขียนโดยภาษาจาวา จะมีจำนวนโค้ด น้อยกว่าโปรแกรมที่เขียนโดยภาษาซีพลัสพลัส ทำให้ใช้งานได้ง่ายกว่าและลดความผิดพลาดได้มากขึ้น

- ภาษาจาวาถูกออกแบบมาให้มีความปลอดภัยสูงตั้งแต่แรก ทำให้โปรแกรมที่เขียนขึ้นด้วยจาวามีความปลอดภัยมากกว่าโปรแกรมที่เขียนขึ้น ด้วยภาษาอื่น เพราะจาวามี ความปลอดภัย ทั้งระดับต่ำ และ ระดับสูง

2.1.2 ข้อเสียของ ภาษาจาวา

- ทำงานได้ช้ากว่าเนทีฟโค้ด (native code) (โปรแกรมที่ compile ให้อยู่ในรูปของ ภาษาเครื่อง) หรือโปรแกรมที่เขียนขึ้นด้วยภาษาอื่น อย่างเช่น ซี หรือ ภาษาซีพลัสพลัส ทั้งนี้ก็ เพราะว่าโปรแกรมที่เขียนขึ้นด้วยภาษาจาวาจะถูกแปลงเป็นภาษากลาง ก่อน แล้วเมื่อโปรแกรม ทำงานคำสั่งของภาษากลางนี้จะถูกเปลี่ยนเป็นภาษาเครื่องอีก ทีหนึ่ง ทีละคำสั่ง (หรือกลุ่มของ คำสั่ง) ณ เวลาทำงาน ทำให้ทำงานช้ากว่าเนทีฟโค้ด ซึ่งอยู่ในรูปของภาษาเครื่องแล้วตั้งแต่ compile โปรแกรมที่ต้องการความเร็วในการทำงานจึงไม่นิยมเขียนด้วยจาวา

- เครื่องมือ ที่มีในการใช้พัฒนาโปรแกรมจาวามักไม่ค่อยเก่ง ทำให้หลายอย่าง โปรแกรมเมอร์จะต้องเป็นคนทำเอง ทำให้ต้องเสียเวลาทำงานในส่วนที่ เครื่องมือ ทำไม่ได้ ถ้าเราดู เครื่องมือของ เอ็มเอส (MS) จะใช้งานได้ง่ายกว่า และพัฒนาได้เร็วกว่า

2.2 ประวัติความเป็นมาของภาษาพีเอชพี

พีเอชพี เป็นภาษาจำพวก ภาษาสคริปต์ คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่าสคริปต์ (script) และเวลาใช้งานต้องอาศัยตัวแปลชุดคำสั่ง ตัวอย่างของภาษาสคริปก็เช่น จาวาสคริปต์ (JavaScript), ภาษาเพิร์ล (Perl) เป็นต้น ลักษณะของพีเอชพี ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ พีเอชพีได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบเอชทีเอ็มแอล โดย สามารถ สอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวว่าพีเอชพี เป็นภาษาที่เรียกว่า เซิร์ฟเวอร์ไซด์ (server-side) เป็นเครื่องมือที่สำคัญชนิดหนึ่ง ที่ช่วยให้เราสามารถสร้างเอกสารแบบ ไดนามิก เอชทีเอ็มแอล (Dynamic HTML) ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น ถ้าใครรู้จัก เซิร์ฟเวอร์ไซด์ อินคูด (Server Side Include - SSI) ก็จะสามารถเข้าใจการทำงานของพีเอชพี ได้ไม่ ยาก สมมุติว่า เราต้องการจะแสดงวันเวลาปัจจุบันที่ผู้เข้ามาเยี่ยมชมเว็บไซต์ในขณะนั้น ในตำแหน่ง ใดตำแหน่งหนึ่งภายในเอกสารเอชทีเอ็มแอล ที่เราต้องการ อาจจะใช้คำสั่งในรูปแบบนี้ เช่น อนุมัติ ก่อนที่จะส่งไปยังผู้อ่านอีกทีหนึ่งอาจจะกล่าวได้ว่าพีเอชพี ได้รับการพัฒนาขึ้นมา เพื่อแทนที่

เซิร์ฟเวอร์ไซด์อินคูช รูปแบบใหม่ๆ โดยให้มีความสามารถ และมีส่วนเชื่อมต่อกับเครื่องมือชนิดอื่นมากขึ้น เช่น ติดต่อกับคลังข้อมูลหรือ database เป็นต้น

พีเอชพี เป็นผลงานที่เติบโตมาจากกลุ่มของนักพัฒนาในเชิงเปิดเผยรหัสต้นฉบับ หรือ โอเพ่นซอร์ส(Open Source) ดังนั้น พีเอชพี จึงมีการพัฒนาไปอย่างรวดเร็ว และแพร่หลาย โดยเฉพาะอย่างยิ่งเมื่อใช้ร่วมกับอาปาเซเว็บเซิร์ฟเวอร์ ระบบปฏิบัติการอย่างเช่น ลินุกซ์ (Linux) หรือ ฟรีบีเอสดี(FreeBSD) เป็นต้น ในปัจจุบันพีเอชพี สามารถใช้ร่วมกับ เว็บเซิร์ฟเวอร์ หลากๆตัวบนระบบปฏิบัติการอย่างเช่น Windows 95/98/NT เป็นต้น เนื่องจากว่าพีเอชพี ไม่ได้เป็นส่วนหนึ่งของตัว เว็บเซิร์ฟเวอร์ ดังนั้นถ้าจะใช้พีเอชพี ก็จะต้องดูก่อนว่า เว็บเซิร์ฟเวอร์ นั้นสามารถใช้สคริปต์พีเอชพี ได้หรือไม่ ยกตัวอย่างเช่น พีเอชพีสามารถใช้ได้กับอาปาเซเว็บเซิร์ฟเวอร์ และ เว็บเซิร์ฟเวอร์ส่วนตัว(PWP) สำหรับระบบปฏิบัติการ Windows 95/98/NT

ในกรณีของอาปาเซเราสามารถใช่ พีเอชพี ได้สองรูปแบบคือ ในลักษณะของซีจีไอ และ โมดูลอาปาเซ ความแตกต่างอยู่ตรงที่ว่า ถ้าใช้พีเอชพี เป็นแบบโมดูลพีเอชพี จะเป็นส่วนหนึ่งของ อาปาเซ หรือเป็นส่วนขยายในการทำงานนั่นเอง ซึ่งจะทำงานได้เร็วกว่าแบบที่เป็น ซีจีไอ เพราะว่า ถ้าเป็น ซีจีไอแล้ว ตัวแปลชุดคำสั่งของ พีเอชพี ถือว่าเป็นแคโปรแกรมภายนอกซึ่งอาปาเซจะต้องเรียกขึ้นมาทำงานทุกครั้ง ที่ต้องการใช้ พีเอชพี ดังนั้น ถ้ามองในเรื่องของประสิทธิภาพในการทำงาน การใช้ พีเอชพี แบบที่เป็นโมดูลหนึ่งของ อาปาเซ จะทำงานได้มีประสิทธิภาพมากกว่า

2.2.1 ภาษาพีเอชพี

พีเอชพี (PHP) คือ ภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ โดยลิขสิทธิ์อยู่ในลักษณะโอเพ่นซอร์ส ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบเอชทีเอ็มแอล โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และ ภาษาเพิร์ล ซึ่งภาษาพีเอชพี นั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจ ที่มีความตอบโต้ได้อย่างรวดเร็ว

2.2.2 คุณสมบัติของภาษาพีเอชพี

การแสดงผลของพีเอชพี จะปรากฏในลักษณะเอชทีเอ็มแอล ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่พีเอชพีแตกต่างจากภาษาในลักษณะไคลเอนต์-ไซด์ สคริปต์ เช่น ภาษาจาวาสคริปต์ ที่ผู้ชมเว็บไซต์สามารถอ่าน ดูและคัดลอกคำสั่งไปใช้เองได้ นอกจากนี้พีเอชพียังเป็น

ภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยาก โดยมีเครื่องมือช่วยเหลือและคู่มือที่สามารถหาอ่านได้ฟรีบนอินเทอร์เน็ต ความสามารถการประมวลผลหลักของพีเอชพี ได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผล การอ่านข้อมูลจากดาต้าเบส ความสามารถจัดการกับคุกกี้ ซึ่งทำงานเช่นเดียวกับโปรแกรมในลักษณะซีจีไอ คุณสมบัติอื่นเช่น การประมวลผลตามบรรทัดคำสั่ง (command line scripting) ทำให้ผู้เขียนโปรแกรมสร้างสคริปต์พีเอชพี ทำงานผ่านพีเอชพีพาร์เซอร์ (PHP parser) โดยไม่ต้องผ่านเซิร์ฟเวอร์หรือเบราว์เซอร์ ซึ่งมีลักษณะเหมือนกับครอน (ใน ยูนิกซ์หรือลินุกซ์) สคริปต์เหล่านี้สามารถนำไปใช้ในแบบซิมเปิลเท็กซ์ โปรเซสซิง แทส (Simple text processing tasks) ได้

การแสดงผลของพีเอชพี ถึงแม้ว่าจุดประสงค์หลักใช้ในการแสดงผลเอชทีเอ็มแอล แต่ยังสามารถสร้างเอ็กซ์เอชทีเอ็มแอล (XHTML) หรือ เอ็กซ์เอ็มแอล (XML) ได้ นอกจากนี้สามารถทำงานร่วมกับคำสั่งเสริมต่างๆ ซึ่งสามารถแสดงผลข้อมูลหลักพีดีเอฟ แฟลช พีเอชพีมีความสามารถอย่างมากในการทำงานเป็นประมวลผลข้อความ จาก โฟชัน เอ็กเทน หรือ รูปแบบเพิร์ล ทั่วไป เพื่อแปลงเป็นเอกสารเอ็กซ์เอ็มแอล ในการแปลงและเข้าสู่เอกสารเอ็กซ์เอ็มแอล เรารองรับมาตรฐานเอสเอเอ็กซ์ (SAX) และดีไอเอ็ม (DOM) สามารถใช้รูปแบบเอ็กซ์เอสแอลที (XSLT) ของเราเพื่อแปลงเอกสาร เอ็กซ์เอ็มแอล

2.2.3 การรองรับพีเอชพี

คำสั่งของพีเอชพี สามารถสร้างผ่านทางโปรแกรมแก้ไขข้อความทั่วไป เช่น โน้ตแพด ซึ่งทำให้การทำงานพีเอชพี สามารถทำงานได้ในระบบปฏิบัติการหลักเกือบทั้งหมด โดยเมื่อเขียนคำสั่งแล้วนำมาประมวลผล อาปาเซ, ไมโครซอฟท์ อินเทอร์เน็ต อินฟอร์มเมชัน เซอร์วิส (Microsoft Internet Information Services (IIS)) , เพอร์ซันนอล เว็บ เซิร์ฟเวอร์ (Personal Web Server) และอื่นๆ อีกมากมาย. สำหรับส่วนหลักของพีเอชพี ยังมีโมดูล ในการรองรับซีจีไอ มาตรฐาน ซึ่งพีเอชพี สามารถทำงานเป็นตัวประมวลผลซีจีไอได้ด้วย และด้วยพีเอชพี, คุณมีอิสรภาพในการเลือกระบบปฏิบัติการ และ เว็บเซิร์ฟเวอร์ นอกจากนี้คุณยังสามารถใช้สร้างโปรแกรมโครงสร้าง สร้างโปรแกรมเชิงวัตถุ (OOP) หรือสร้างโปรแกรมที่รวมทั้งสองอย่างเข้าด้วยกัน แม้ว่าความสามารถของคำสั่ง สร้างโปรแกรมเชิงวัตถุ มาตรฐานในเวอร์ชันนี้ยังไม่สมบูรณ์ แต่ตัวไลบรารีทั้งหลายของโปรแกรม และตัวโปรแกรมประยุกต์ (รวมถึงเพิร์ล ไลบรารี) ได้ถูกเขียนขึ้นโดยใช้รูปแบบการเขียนแบบ สร้างโปรแกรมเชิงวัตถุ เท่านั้น

พีเอชพีสามารถทำงานร่วมกับฐานข้อมูลได้หลายชนิด ซึ่งฐานข้อมูลส่วนหนึ่งที่รองรับ ได้แก่ ออราเคิล ดีเบส โพสต์เกรสคิวแอล ไอบีเอ็ม ดีบี2 มายเอสคิวแอล อินฟอร์มิกซ์ โอดีบีซี โครงสร้างของฐานข้อมูลแบบ ดีบีเอ็ช ซึ่งทำให้พีเอชพีใช้กับฐานข้อมูลอะไรก็ได้ที่รองรับรูปแบบนี้ และพีเอชพี ยังรองรับ โอดีบีซี (Open Database Connection) ซึ่งเป็นมาตรฐานการเชื่อมต่อ ฐานข้อมูลที่ใช้กันแพร่หลายอีกด้วย คุณสามารถเชื่อมต่อกับฐานข้อมูลต่างๆ ที่รองรับมาตรฐานโลก นี้ได้

พีเอชพียังสามารถรองรับการสื่อสารกับการบริการในโพรโทคอลต่างๆ เช่น แอลแต่ป ไอแมพ เอสเอ็นเอ็มพี เอ็นเอ็นพีที พีอ็อป3 เอชทีทีพี คอม และอื่นๆ อีกมากมาย คุณสามารถเปิดซอคเก็ต บนเครือข่ายโดยตรง และ ตอบโต้โดยใช้ โพรโทคอลใดๆ ก็ได้พีเอชพี มีการรองรับสำหรับการ แลกเปลี่ยนข้อมูลแบบ วีดีอีเอ็ช คอมเพล็กซ์ กับเว็บ โปรแกรม อื่นๆ ทั่วไปได้ พุดถึงในส่วน อินเทอร์เน็ตคอนเน็คชัน , พีเอชพีมีการรองรับสำหรับจาวาสคริป ให้เปลี่ยนเป็นพีเอชพี แล้วใช้งาน

2.2.4 คำสั่งเบื้องต้น

2.2.4.1 คำสั่งแสดงผล

echo เป็นคำสั่งแสดงผล ที่ไม่มีการ return ค่ากลับ ดังนั้นทำให้ระบบมีการทำงานในการแสดงผลเร็วกว่า print และแสดงผลในรูปแบบคำสั่งประโยคที่ไม่ซับซ้อน

Code ตัวอย่าง

```
<?php
echo' <input type="hidden" name="productID" value="'. $product_detail[0]->
Product_ID.'"/>'; ?>
```

2.2.4.2 การเชื่อมต่อคำ

ในภาษา PHP จะใช้ "." (dot) ในการเชื่อมต่อคำให้เป็นประโยค ก่อนที่จะทำการแสดงผลออกมา เพื่อสื่อสารกับผู้ใช้งานโปรแกรม

Code ตัวอย่าง

```
<?php
echo' <div class="title-detail">'. $product_detail[0]->product_name.'</div>';
echo' <table class="table table-detail">';
echo' <tbody>';
```

```
echo'      <form action ="".base_url().'index.php/Detail/addtocart" method="post"
accept-charset="utf-8">;
?>
```

2.2.4.3 กำหนดตัวแปรPHPไว้เก็บค่า

ตัวแปรเป็นสัญลักษณ์ที่ใช้ในการแทนค่าเพื่อนำไปประมวลผล ตามที่ต้องการที่จะให้โปรแกรมทำงานให้ตามที่ต้องการ เช่น \$value=1; โดยเราสามารถกำหนดตัวแปรไว้เรียกใช้งานได้ตลอดหน้า PHP นั้น

Code ตัวอย่าง

```
<?php
$total = 0.0;
$total += ($row->price - $row->price_sale)*$row->Quantity;
echo '      <td colspan="2"><b>'.$total.'</b></td>;
?>
```

2.2.4.4 คำสั่งเงื่อนไข

คำสั่ง if else นั้นเป็นการกำหนดทางเลือกอื่นๆ หากกรณีโปรแกรมนั้นไม่ตรงกับเงื่อนไขที่อยู่ใน if โปรแกรมก็จะทำงานคำสั่งต่างๆที่อยู่ในวงเล็บปีกกาของ else แทน

Code ตัวอย่าง

```
<?php
if($product_detail[0]->stock > 0)
{
echo'<td><span class="label label-success arrowed">Ready Stock</span></td>;
}
else
{
echo'<td><span class="label label-danger arrowed">Out of Stock</span></td>;
}
?>
```

2.2.4.5 คำสั่ง foreach

คำสั่ง Foreach loop เป็นคำสั่งที่ยอมให้เราเข้าไปถึงข้อมูลที่อยู่ภายในอะเรย์ได้โดยตรง

Code ตัวอย่าง

```
<?php
foreach($feature_product as $row)
{
echo ' <a href="http://128.199.209.131 /ajsom/index.php/ Detail/productview/
'.$row->Product_ID.">';
echo '';
echo ' </a>';
}
?>
```

2.3 ภาษาเอชทีเอ็มแอล (HTML)

เอชทีเอ็มแอล (Hypertext Markup Language) เป็นภาษาหลักที่ใช้ในการสร้างไฟล์เว็บเพจ โดยมีแนวคิดจากการสร้างเอกสารไฮเปอร์เท็กซ์ (Hypertext Document) ซึ่งพัฒนาขึ้นมาจากภาษา เอสจีเอ็มแอล (Standard Generalized Markup Language) โดย ทิม เบอร์เนอร์ ลี (Tim Berners-Lee) เป็นภาษามาตรฐานที่ใช้พัฒนาเอกสารในรูปแบบของเว็บเพจเผยแพร่บนระบบเครือข่ายอินเทอร์เน็ต มีโครงสร้างการเขียนที่อาศัยตัวกำกับ เรียกว่า แท็ก (Tag) ควบคุมการแสดงผลของข้อความ, รูปภาพ หรือวัตถุอื่นๆ เรียกใช้เอกสารเหล่านี้โดยการใช้โปรแกรมเว็บเบราว์เซอร์ (Web Browser) เช่น มอซิลลา ไฟร์ฟอกซ์ (Mozilla Firefox), โอเปราOpera , เน็ตสเคปเนวิกเกตอร์(Netscape navigator), อินเทอร์เน็ตเอ็กซ์พลอเรอร์ (Internet Explorer) เป็นต้น

2.3.1 การใช้งานเบื้องต้นภาษาเอชทีเอ็มแอล

ภาษาเอชทีเอ็มแอล เป็นภาษาคอมพิวเตอร์รูปแบบหนึ่ง ที่มีโครงสร้างการเขียนโดยอาศัยตัวกำกับ (Tag) ควบคุมการแสดงผลข้อความ รูปภาพ หรือวัตถุ อื่น ๆ ผ่าน บรรทัดแต่ละ แท็ก อาจจะมีส่วนขยาย เรียกว่า แอตทริบิวต์ (Attribute) สำหรับระบุ หรือควบคุมการแสดงผล ของเว็บได้ด้วยเอชทีเอ็มแอล เป็นภาษาที่ถูกพัฒนาโดยเวิร์ลด์ ไรด์ เว็บ คอนซอร์เทียม (World Wide Web Consortium : W3C) จากแม่แบบของ ภาษาเอสจีเอ็มแอล (Standard Generalized Markup Language) โดยตัดความสามารถบางส่วนออกไป เพื่อให้สามารถทำความเข้าใจและเรียนรู้ได้ง่าย และด้วยประเด็นดังกล่าว ทำให้บริการ เวิร์ลด์ ไรด์ เว็บ (www) เติบโตขยายตัวอย่างกว้างขวาง

2.3.1.1 คำสั่ง หรือแท็กเป็นลักษณะเฉพาะของภาษาเอชทีเอ็มแอล ใช้ในการระบุรูปแบบคำสั่ง หรือการลงรหัสคำสั่งเอชทีเอ็มแอล ภายในเครื่องหมาย less-than bracket (<) และ greater-than bracket (>) โดยที่แท็กเอชทีเอ็มแอล แบ่งได้ 2 ลักษณะ คือ

- 1) แท็กเดี่ยว เป็นแท็กที่ไม่ต้องมีการปิดรหัส เช่น <HR>,
 เป็นต้น
- 2) แท็กเปิด/ปิด รูปแบบของแท็กนี้จะเป็นแบบ <tag>.... </tag> โดยที่ <tag> เราเรียกว่าแท็กเปิด ส่วน </tag> เราเรียกว่าแท็กปิด

2.3.1.2 แอตทริบิวต์ เป็นตัวบอกรายละเอียดของแท็กนั้นเช่น ... เป็นการบอกว่าให้อักขรที่อยู่ในแท็กนี้ชิดซ้าย

2.3.1.3 not case sensitive หมายถึง คุณจะพิมพ์
 หรือ
 ก็ได้ ผลลัพธ์ออกมาไม่ต่างกัน

2.3.2 โครงสร้างของหลักของเอชทีเอ็มแอล

```
<html>
<head>
<title>หัวข้อเว็บเพจ</title>
</head >
<body>
...ส่วนของเนื้อหา หรือข้อความที่จะใช้แสดงเนื้อหาในเว็บเพจ ...
</body>
</html>
```

- 1) `<html>...</html>` เป็นแท็กแรกที่ต้องมีในภาษาเอชทีเอ็มแอล ซึ่งบอกให้ทราบว่านี่คือ ภาษาเอชทีเอ็มแอล ใช้ในการกำหนดจุดเริ่มและจุดสิ้นสุดของเอกสารเอชทีเอ็มแอล
- 2) `<head>...</head>` กำหนดรายละเอียดหัวเอกสารเอชทีเอ็มแอล
- 3) `<title>...</title>` เป็นแท็กที่ใช้กำหนดชื่อเว็บเพจ ซึ่งภายแท็กจะเป็นชื่อเรื่องเว็บที่ต้องการ ซึ่งความยาวไม่เกิน 64 ตัวอักษร ข้อความนี้จะปรากฏที่ title bar ของเบราว์เซอร์ที่เราใช้งานอยู่
- 4) `<body>...</body>` ส่วนที่เป็นเนื้อหาเว็บเพจของเรา ไม่ว่าจะเป็นข้อความ รูปภาพ ตาราง และการเชื่อมโยงต่อไปยังเอกสารอื่น ๆ

2.3.3 การแสดงผลที่เว็บเบราว์เซอร์

หลังจากมีการพิมพ์โปรแกรมนี้เสร็จเรียบร้อยแล้ว ให้บันทึกเป็น ไฟล์ที่มีนามสกุล .htm หรือ .html จากนั้นให้เรียกโปรแกรมเว็บเบราว์เซอร์ขึ้นมาทำการทดสอบ ข้อมูลที่เราสร้างจะถูก นำมาที่ออกมาแสดงที่จอภาพ ถ้าไม่เขียนอะไรผิด บนจอภาพก็จะแสดงผลตามนั้น

ถ้าเรามีการปรับปรุงแก้ไขข้อมูลในโปรแกรมเดิม ให้อยู่ในรูปของ โปรแกรมใหม่ ก็จำเป็นต้องโหลดโปรแกรมขึ้นมาใหม่ เพียงแต่เลื่อนเมาส์ไปคลิกที่ปุ่ม รีเฟรช (Refresh) โปรแกรมก็จะทำการ ประมวลผลและแสดงผลออกมาใหม่ ในคำสั่งเอชทีเอ็มแอล ส่วนใหญ่ใช้ตัวเปิด เป็นเครื่องหมายน้อยกว่า (<) ตามด้วยคำสั่ง และปิดท้ายด้วยเครื่องหมายมากกว่า (>) และมีตัวปิดที่มีรูปแบบเหมือนตัวเปิดเสมอ เพียงแต่จะมีเครื่องหมาย / อยู่หน้าคำสั่งนั้นๆ เช่น คำสั่ง <BODY> จะมี </BODY> เป็นคำสั่งปิด เมื่อใดที่ผู้เขียนลืมหรือพิมพ์คำสั่งผิด จะส่งผลให้การทำงานของโปรแกรมผิดพลาดทันที

2.4 ภาษาเอสควิแอล (SQL)

SQL ย่อมาจาก structured query language คือภาษาที่ใช้ในการเขียนโปรแกรมเพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์และเป็นระบบเปิด (open system) หมายถึงเราสามารถใส่คำสั่ง sql กับฐานข้อมูลชนิดใดก็ได้ และ คำสั่งงานเดียวกันเมื่อสั่งงานผ่าน ระบบฐานข้อมูลที่แตกต่างกันจะได้ ผลลัพธ์เหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูล ชนิดใดก็ได้โดยไม่ติดขัดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง นอกจากนี้แล้ว SQL ยังเป็นชื่อโปรแกรมฐานข้อมูล ซึ่งโปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของ

ภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่งซึ่งแบ่งการทำงานได้เป็น 4 ประเภท ดังนี้

1. Select query ใช้สำหรับดึงข้อมูลที่ต้องการ จะมีการค้นหารายการจากตารางในฐานข้อมูล ตั้งแต่หนึ่งตารางขึ้นไป ตามเงื่อนไขที่สั่ง ผลลัพธ์ที่ได้จะเป็นเซตของข้อมูลที่สามารถสร้างเป็นตารางใหม่ หรือใช้แสดงออกมาทางจอภาพเท่านั้น โดยมีรูปแบบดังนี้ Select รายละเอียดที่เลือก From ตารางแหล่งที่มา Where กำหนดเงื่อนไขฐานข้อมูลที่เลือก Group by ชื่อคอลัมน์
2. Update query ใช้สำหรับการแก้ไขข้อมูลในตาราง โดยแก้ไขในคอลัมน์ที่มีค่าตรงตามเงื่อนไข
3. Insert query ใช้สำหรับการเพิ่มข้อมูลใหม่ ๆ เข้าไปในฐานข้อมูล มีรูปแบบดังนี้ Insert Into ชื่อตาราง [= ชื่อ คอลัมน์ 1,2,..] Values [ค่าที่จะใส่ลงใน คอลัมน์ 1,2,..]
4. Delete query ใช้ลบข้อมูลออกจากตาราง มีรูปแบบดังนี้ Delete From ชื่อตาราง Where เงื่อนไข

2.5 อาปาเช่ (Apache)

อาปาเช่ คือ เว็บเซิร์ฟเวอร์ พัฒนามาจาก เอชทีทีพีดี เว็บเซิร์ฟเวอร์ (HTTPD Web Server) โดยเจ้า อาปาเช่ นี้จะทำหน้าที่ในการจัดเก็บ โฮมเพจ และส่ง โฮมเพจไปยัง บราวเซอร์ (Browser) ที่มีการเรียกเข้ายังเว็บเซิร์ฟเวอร์ที่เก็บ โฮมเพจ นั้นอยู่ ซึ่งปัจจุบันจัดได้ว่าเป็น เว็บเซิร์ฟเวอร์ ที่มีความน่าเชื่อถือมาก เนื่องจากเป็นที่นิยมใช้กันทั่วโลก อีกทั้งอาปาเช่ยังเป็นซอฟต์แวร์แบบ โอเพ่นซอร์ส ที่เปิดให้บุคคลทั่วไปสามารถเข้ามาร่วมพัฒนาส่วนต่างๆ ของอาปาเช่ได้ ซึ่งทำให้เกิดเป็น โมดูล ที่เกิดประโยชน์มากมาย เช่น mod_perl, mod_python หรือ mod_php และทำงานร่วมกับภาษาอื่นได้ แทนที่จะเป็นเพียงเซิร์ฟเวอร์ที่ให้บริการเพียงแค่ HTML อย่างเดียว นอกจากนี้อาปาเช่เองยังมีความสามารถอื่นๆ ด้วย เช่น การยืนยันตัวบุคคล (mod_auth, mod_access, mod_digest) หรือเพิ่มความปลอดภัยในการสื่อสารผ่าน โปรโตคอล เอชทีทีพีเอส (https - mod_ssl) และยังมีโมดูลอื่นๆ ที่ได้รับความนิยมใช้ เช่น mod_vhost ทำให้สามารถสร้างโฮสต์เสมือน ภายในเครื่องเดียวกันได้ หรือ mod_rewrite ซึ่งเป็นเครื่องมือที่จะช่วยให้ยูเอแอล (url) ของเว็บนั้นอ่านง่ายขึ้น

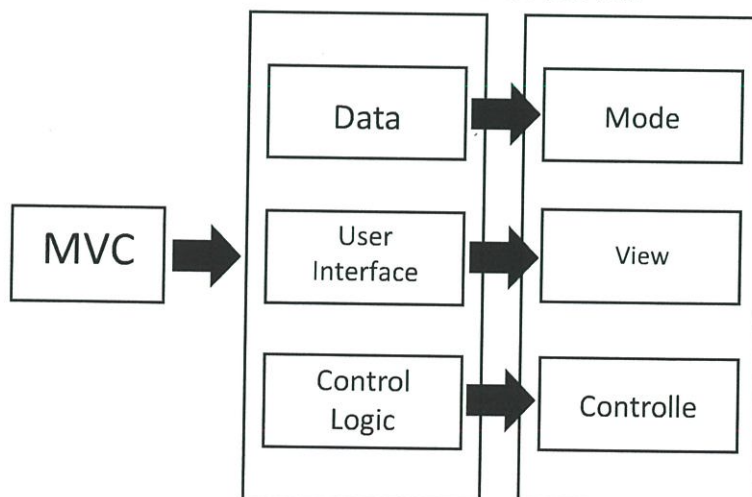
2.6 ระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูล หมายถึง โครงสร้างสารสนเทศที่ประกอบด้วยรายละเอียดของข้อมูลที่เกี่ยวข้องกันที่จะนำมาใช้ในระบบต่าง ๆ ร่วมกัน ระบบฐานข้อมูล จึงนับว่าเป็นการจัดเก็บข้อมูลอย่างเป็นระบบ ซึ่งผู้ใช้สามารถจัดการกับข้อมูลได้ในลักษณะต่าง ๆ ทั้งการเพิ่ม การแก้ไข การลบ ตลอดจนการเรียกดูข้อมูล ซึ่งส่วนใหญ่จะเป็นการประยุกต์นำเอาระบบคอมพิวเตอร์เข้ามาช่วยในการจัดการฐานข้อมูล

2.7 โมเดล วิว คอนโทรลเลอร์ (Model View Controller)

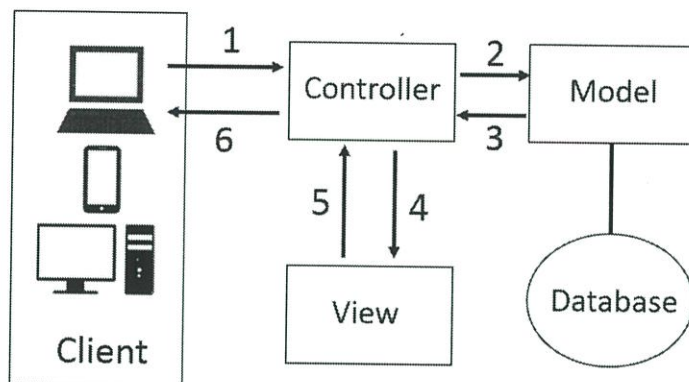
โมเดล วิว คอนโทรลเลอร์ คือ สถาปัตยกรรมซอฟต์แวร์ชนิดหนึ่ง ซึ่งปัจจุบันมี Frameworks สำหรับสร้าง Web Apps จำนวนมาก แต่เกือบทั้งหมดมีโครงสร้างแบบ MVC ซึ่งแต่ละ Web Apps จะถูกแบ่งออกเป็นสามส่วน คือ

1. Model คือ ส่วนที่ใช้ในการติดต่อกับฐานข้อมูล ซึ่งทำหน้าที่ในการดึงข้อมูลขึ้นมาจากฐานข้อมูลขึ้นมา และทำการจัดการข้อมูลต่างๆไว้ในรูปแบบที่เหมาะสม
2. View คือส่วนที่จะนำข้อมูลจาก Model ไปใช้แสดงผลให้ผู้ใช้เห็นผลลัพธ์ออกมาใน User Interface
3. Controller คือส่วนที่จะคอยรับ Input จาก Client เข้ามาแล้วนำคำสั่งไปประมวลผลเพื่อสั่งงาน View และ Model ให้ประมวลผลออกมาเป็นอย่างไร



รูปที่ 2.1 การทำงานของเอ็มวีซี

2.7.1 ขั้นตอนการทำงานของ MVC



รูปที่ 2.2 ขั้นตอนการทำงานของเอ็มวีซี

1. เริ่มจาก Client ส่ง Request ไปที่ Web Application ซึ่งจะถูกส่งต่อให้ Controller ทำการตรวจสอบข้อมูลที่มาให้ (Request Method, Request Parameters)
2. แล้ว Controller จะเรียก Method ให้ทำงานเพื่อจัดการ Request นั้น
3. Model จะทำการคำนวณและอาจติดต่อกับ Database เพื่อจัดการกับ Request นั้น แล้วส่งผลลัพธ์กลับไป Controller
4. เมื่อ Controller ได้ผลลัพธ์จาก Model แล้วก็ใช้ผลลัพธ์นั้นส่งต่อให้ View ทำงาน
5. View จะสร้าง Page สำหรับแสดงผลลัพธ์นั้น แล้วส่ง page กลับไปที่ Controller
6. Controller ส่ง Page นั้น (เป็น Response) กลับไปยัง Client

2.8 ระบบปฏิบัติการแอนดรอยด์

ระบบปฏิบัติการแอนดรอยด์ (Android) คือระบบปฏิบัติการแบบเปิดเผยซอร์ฟแวร์ต้นฉบับ (Open Source) โดยบริษัท กูเกิ้ล (Google Inc.) ที่ได้รับความนิยมเป็นอย่างสูง เนื่องจากอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์ มีจำนวนมาก อุปกรณ์มีหลากหลายระดับ หลากราคา รวมทั้งสามารถทำงานบนอุปกรณ์ที่มีขนาดหน้าจอ และความละเอียดแตกต่างกันได้ ทำให้ผู้บริโภคสามารถเลือกได้ตามต้องการ และหากมองในทิศทางสำหรับนักพัฒนาโปรแกรม (Programmer) แล้วนั้น การพัฒนาโปรแกรมเพื่อใช้งานบนระบบปฏิบัติการแอนดรอยด์ ไม่ใช่เรื่องที่ยาก เพราะมีข้อมูลในการพัฒนารวมทั้งแอนดรอยด์ เอสดีเค (Android SDK) เตรียมไว้ให้กับนักพัฒนาได้เรียนรู้

และเมื่อนักพัฒนาต้องการจะเผยแพร่หรือจำหน่ายโปรแกรมที่พัฒนาแล้วเสร็จ แอนดรอยด์ก็ยังมีตลาดในการเผยแพร่โปรแกรม ผ่านแอนดรอยด์มาร์เก็ต แต่หากจะกล่าวถึงโครงสร้างภาษาที่ใช้ในการพัฒนานั้น สำหรับ แอนดรอยด์ เอสดีเค จะยึดโครงสร้างของภาษาจาวา (Java language) ในการเขียนโปรแกรม เพราะโปรแกรมที่พัฒนามาได้จะต้องทำงานอยู่ภายใต้ เดลวิก วิชวล แมชชีน (Dalvik Virtual Machine) เช่นเดียวกับโปรแกรมจาวา ที่ต้องทำงานอยู่ภายใต้ จาวา วิชวล แมชชีน (Java Virtual Machine)

นอกจากนั้นแล้ว แอนดรอยด์ ยังมีโปรแกรมแอมที่เปิดเผยซอร์ฟแวร์ต้นฉบับ (Open Source) เป็นจำนวนมาก ทำให้นักพัฒนาที่สนใจ สามารถนำซอร์ฟแวร์ต้นฉบับ มาศึกษาได้อย่างไม่ยาก ประกอบกับความนิยมของแอนดรอยด์ได้เพิ่มขึ้นอย่างมาก

2.8.1 ประวัติความเป็นมา

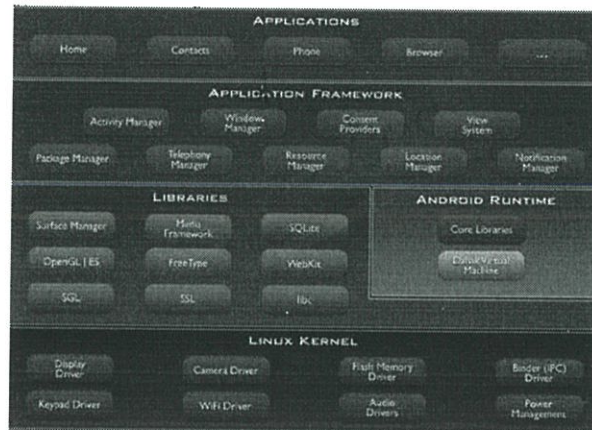
เริ่มต้นระบบปฏิบัติการแอนดรอยด์ ถูกพัฒนามาจากบริษัท แอนดรอยด์ เมื่อปี พ.ศ 2546 โดยมีนาย แอนดี้ รูบิน (Andy Rubin) ผู้ให้กำเนิดระบบปฏิบัติการนี้ และถูกบริษัท กูเกิล ซื้อกิจการเมื่อ เดือนสิงหาคม ปี พ.ศ 2548 โดยบริษัทแอนดรอยด์ ได้กลายเป็นมาบริษัทลูก ของบริษัท กูเกิล และยังมีนาย แอนดี้ รูบิน ดำเนินงานอยู่ในทีมพัฒนาระบบปฏิบัติการต่อไป

ระบบปฏิบัติการแอนดรอยด์ เป็นระบบปฏิบัติการที่พัฒนามาจากการนำเอา แกนกลางของระบบปฏิบัติการลินุกซ์ (Linux Kernel) ซึ่งเป็นระบบปฏิบัติการที่ออกแบบมาเพื่อทำงานเป็นเครื่องให้บริการ (Server) มาพัฒนาต่อ เพื่อให้กลายเป็นระบบปฏิบัติการบนอุปกรณ์พกพา (Mobile Operating System)

ต่อมาเมื่อเดือน พฤศจิกายนปีพ.ศ 2550 บริษัทกูเกิล ได้ทำการก่อตั้งสมาคมโอเอสเอ (Open Handset Alliance) เพื่อเป็นหน่วยงานกลางในการกำหนดมาตรฐานกลาง ของอุปกรณ์พกพาและระบบปฏิบัติการแอนดรอยด์ โดยมีสมาชิกในช่วงก่อตั้งจำนวน 34 รายเข้าร่วม ซึ่งประกอบไปด้วยบริษัทชั้นนำที่ดำเนินธุรกิจด้านการสื่อสาร เช่น โรงงานผลิตอุปกรณ์พกพา, บริษัทพัฒนาโปรแกรม, ผู้ให้บริการสื่อสาร และผู้ผลิตอะไหล่อุปกรณ์ด้านสื่อสาร

2.8.2 โครงสร้างของแอนดรอยด์

การทำความเข้าใจโครงสร้างของระบบปฏิบัติการแอนดรอยด์ ถือว่าเป็นสิ่งสำคัญเพราะถ้านักพัฒนาโปรแกรม สามารถมองภาพโดยรวมของระบบได้ทั้งหมด จะให้สามารถเข้าใจถึงกระบวนการทำงาน ได้ดียิ่งขึ้น และสามารถนำไปช่วยในการออกแบบโปรแกรมที่ต้องการพัฒนา เพื่อให้เกิดประสิทธิภาพในการทำงานทั้งนี้แสดงโครงสร้างของแอนดรอยด์ดังรูปที่ 2.1



รูปที่ 2.3 โครงสร้างของระบบปฏิบัติการแอนดรอยด์

จากรูปที่ 2.1 โครงสร้างของระบบปฏิบัติการแอนดรอยด์ จะสังเกตเห็นได้ว่าการแบ่งออกมาเป็นส่วนๆ ที่มีความเกี่ยวเนื่องกัน โดยส่วนบนสุดจะเป็นส่วนที่ผู้ใช้งานทำการติดต่อโดยตรง ซึ่งก็คือส่วนของ (Applications) จากนั้นก็จะลำดับลงมาเป็นองค์ประกอบอื่นๆ ตามลำดับ และสุดท้ายจะเป็นส่วนที่ติดต่อกับอุปกรณ์โดยผ่านทางลินุกซ์ เคอร์เนล (Linux Kernel) โครงสร้างของแอนดรอยด์ พอที่จะอธิบายเป็นส่วนๆ ได้ดังนี้

2.8.2.1 แอปพลิเคชัน ส่วน แอปพลิเคชัน หรือส่วนของโปรแกรมที่มีมาจกระบบปฏิบัติการ หรือเป็นกลุ่มของโปรแกรมที่ผู้ใช้งานได้ทำการติดตั้งไว้ โดยผู้ใช้งานสามารถเรียกใช้โปรแกรมต่างๆ ได้โดยตรง ซึ่งการทำงานของแต่ละโปรแกรมจะเป็นไปตามที่ผู้พัฒนาโปรแกรมได้ออกแบบและเขียนโค้ดโปรแกรมเอาไว้

2.8.2.2 แอปพลิเคชันเฟรมเวิร์ค เป็นส่วนที่มีการพัฒนาขึ้นเพื่อให้นักพัฒนาสามารถพัฒนาโปรแกรมได้สะดวก และมีประสิทธิภาพมากยิ่งขึ้น โดยนักพัฒนาไม่จำเป็นต้องพัฒนาในส่วนที่มีความยุ่งยากมากๆ เพียงแค่ทำการศึกษาถึงวิธีการเรียกใช้งาน แอปพลิเคชันเฟรมเวิร์ค ในส่วนที่ต้องการใช้งาน แล้วนำมาใช้งาน ซึ่งมีหลายกลุ่มด้วยกัน

1) แอคติวิตี้ เมเนเจอร์ (Activities Manager) เป็นกลุ่มของชุดคำสั่งที่จัดการเกี่ยวกับวงจรการทำงานของหน้าต่างโปรแกรม(Activity)

2) คอนเทนท์ โพรไวเดอร์ (Content Providers) เป็นกลุ่มของชุดคำสั่งที่ใช้ในการเข้าถึงข้อมูลของโปรแกรมอื่น และสามารถแบ่งปันข้อมูลให้โปรแกรมอื่นเข้าถึงได้

3) วิว ซิสเต็ม (View System) เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับการจัดการโครงสร้างของหน้าจอที่แสดงผลในส่วนที่ติดต่อกับผู้ใช้งาน (User Interface)

4) เทเลโฟนี เมเนเจอร์ (Telephony Manager) เป็นกลุ่มของชุดคำสั่งที่ใช้ในการเข้าถึงข้อมูลด้านโทรศัพท์ เช่น หมายเลขโทรศัพท์ เป็นต้น

5) รีซอร์ส เมเนเจอร์ (Resource Manager) เป็นกลุ่มของชุดคำสั่งในการเข้าถึงข้อมูลที่เป็น ข้อความ, รูปภาพ

6) โลเคชัน เมเนเจอร์ (Location Manager) เป็นกลุ่มของชุดคำสั่งที่เกี่ยวข้องกับตำแหน่งทางภูมิศาสตร์ ที่ระบบปฏิบัติการได้รับค่าจากอุปกรณ์

7) นอติฟิเคชัน เมเนเจอร์ (Notification Manager) เป็นกลุ่มของชุดคำสั่งที่จะถูกเรียกใช้เมื่อโปรแกรม ต้องการแสดงผลให้กับผู้ใช้งาน ผ่านทางแถบสถานะ (Status Bar) ของหน้าจอ

2.8.2.3 ไลบรารี เป็นส่วนของชุดคำสั่งที่พัฒนาด้วย ภาษาซี/ภาษาซีพลัสพลัส โดยแบ่งชุดคำสั่งออกเป็นกลุ่มตามวัตถุประสงค์ของการใช้งาน เช่น เซอร์เฟส เมเนเจอร์ (Surface Manage) จัดการเกี่ยวกับการแสดงผล, มีเดีย เฟรมเวิร์ค (Media Framework) จัดการเกี่ยวกับการการแสดงผลภาพและเสียง, โอเพ่น จีแอล | อีเอส(ES) และ เอสจีแอล (SGL) จัดการเกี่ยวกับภาพ 3 มิติ และ 2 มิติ, เอสคิวแอล ไลต์ (SQLite) จัดการเกี่ยวกับระบบฐานข้อมูล เป็นต้น

2.8.2.4 แอนดรอยด์ รันไทม์ (Android Runtime) จะมีเดลวิก วิชวล แมชชีน ที่ถูกออกแบบมา เพื่อให้ทำงานบนอุปกรณ์ที่มี หน่วยความจำ(Memory), หน่วยประมวลผลกลาง(CPU) และพลังงาน(Battery)ที่จำกัด ซึ่งการทำงานของเดลวิก วิชวล แมชชีน จะทำการแปลงไฟล์ที่ต้องการทำงาน ไปเป็นไฟล์ .DEX ก่อนการทำงาน เหตุผลก็เพื่อให้มีประสิทธิภาพเพิ่มขึ้นเมื่อใช้งานกับ หน่วยประมวลผลกลางที่มีความเร็วไม่มาก ส่วนต่อมาก็คือคอร์ไลบรารี (Core Libraries) ที่เป็นส่วนรวบรวมคำสั่งและชุดคำสั่งสำคัญ โดยถูกเขียนด้วยภาษาจาวา (Java Language)

2.8.2.5 ลินุกซ์ เคอเนล (Linux Kernel) เป็นส่วนที่ทำหน้าที่หัวใจสำคัญ ในจัดการกับบริการหลักของระบบปฏิบัติการ เช่น เรื่องหน่วยความจำ พลังงาน ติดต่อกับอุปกรณ์ต่างๆ ความปลอดภัย เครือข่าย โดยแอนดรอยด์ได้นำเอาส่วนนี้มาจากระบบปฏิบัติการลินุกซ์ รุ่น 2.6 (Linux 2.6 Kernel) ซึ่งได้มีการออกแบบมาเป็นอย่างดี

2.8.3 โปรแกรมและเครื่องมือในการพัฒนา

โปรแกรมหรือเครื่องมือที่ใช้พัฒนาแอปพลิเคชัน แอนดรอยด์ นั้นมีให้เลือกใช้ หลายค่ายหลายทางเลือกด้วยกัน แต่สำหรับโครงการชิ้นนี้จะใช้โปรแกรมที่ถูกเลือกแนะนำ ซึ่งฟรีและเป็นที่ยอมรับกัน อย่างแพร่หลายสำหรับนักพัฒนาทั่วไป ซึ่งโปรแกรมหรือเครื่องมือที่ต้องติดตั้งมีดังนี้

2.8.3.1 ชุดของเครื่องมือที่ใช้ในการพัฒนาโปรแกรมจาวา ใช้ในการพัฒนาโปรแกรมจาวาของบริษัทซันไมโครซิสเต็มส์ ซึ่งใครก็ตามที่ต้องการจะพัฒนาโปรแกรมโดยใช้ภาษาจาวา

2.8.3.2 อีคิปส์ (Eclipse) คือโปรแกรมที่ใช้สำหรับพัฒนาภาษาจาวา ซึ่งโปรแกรม อีคิปส์เป็นโปรแกรมหนึ่งที่ใช้ในการพัฒนาแอปพลิเคชัน เซิร์ฟเวอร์ ได้อย่างมีประสิทธิภาพ และเนื่องจากอีคิปส์เป็นซอฟต์แวร์ โอเพ่นซอร์ส (Open Source) ที่พัฒนาขึ้นเพื่อใช้โดยนักพัฒนาเอง ทำให้ความก้าวหน้าในการพัฒนาของ อีคิปส์ เป็นไปอย่างต่อเนื่องและรวดเร็ว

2.8.3.3 แอนดรอยด์ เอสดีเค ซึ่งเป็นชุดโปรแกรมที่ทางกูเกิลพัฒนาออกมาเพื่อแจกจ่ายให้นักพัฒนาแอปพลิเคชัน หรือผู้สนใจทั่วไปดาวน์โหลดไปใช้งานกันได้โดยไม่มีค่าใช้จ่ายใดๆ (ฟรีสำหรับการใช้งานโปรแกรม) ซึ่งนี่ก็เป็นหนึ่งในปัจจัยที่ทำให้แอปพลิเคชันบนแอนดรอยด์ นั้นเพิ่มขึ้น อย่างรวดเร็ว ในชุดเอสดีเค นั้นจะมีโปรแกรมและไลบรารีต่างๆ ที่จำเป็นต่อการพัฒนาแอปพลิเคชันบนแอนดรอยด์ อย่างเช่นอีมูเลเตอร์(Emulator) ซึ่งทำให้ผู้พัฒนาโปรแกรมสามารถสร้างแอปพลิเคชัน และนำมาทดลองใช้งานบนอีมูเลเตอร์ ก่อนโดยมีสภาวะแวดล้อมเหมือนกับการนำไปใช้งานโทรศัพท์มือถือที่ใช้ระบบ ปฏิบัติการแอนดรอยด์จริงๆ

2.8.3.4 การพัฒนาอุปกรณ์แอนดรอยด์ (Android Development Tools) คือเครื่องมือที่ใช้พัฒนาแอนดรอยด์ ในการพัฒนาแอปพลิเคชัน บนระบบ แอนดรอยด์ ไอเอส จะใช้ภาษาจาวา โดยต้องติดตั้งส่วนเสริม ซึ่งก็คือเจ้า เอดีที หรือ เครื่องมือสำหรับการพัฒนาแอนดรอย (Android Development Tools) ซึ่งเป็นส่วนเสริมของไอดีอี หรือที่หลายคนมักเรียกมันว่าเป็น ปลั๊กอินของโปรแกรม อีคิปส์ นั่นเอง ซึ่งใช้ในการเขียนโปรแกรม และ เอดีที นี้ก็รวมอยู่เป็นส่วนหนึ่งของ แอนดรอยด์เอสดีเค

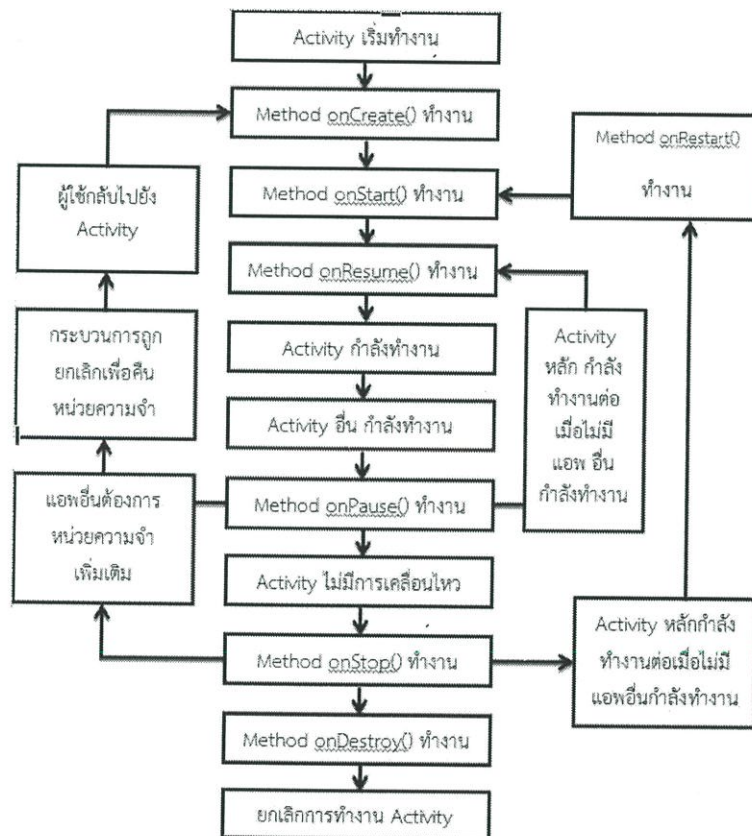
2.8.4 การทดลองรันแอปพลิเคชัน

สามารถทำได้ 2 วิธีคือ

1. รันบน อีมูเลเตอร์ คือการจำลองเครื่องคอมพิวเตอร์ให้เป็นสมาร์ทโฟน หรือ แท็บเล็ต(Tablet)ซึ่งตัวจำลองนี้เรียกว่า แอนดรอยด์ วิชวล ดีไวซ์ (Android Virtual Device)
2. รันบนอุปกรณ์จริง โดยส่งไฟล์แอปพลิเคชัน ไปรันบนสมาร์ทโฟน แท็บเล็ต โดยการติดตั้งไดรเวอร์(Driver) สำหรับอุปกรณ์นั้นๆ

2.8.5 การทำงานของแอกติวิตี้

การทำงานของแอกติวิตี้ มีลักษณะดังรูปที่ 2.3



รูปที่ 2.4 การทำงานของแอกติวิตี้

ซึ่งความหมายของแต่ละเมธอด มีดังนี้

1. onCreate() คือ การเริ่มต้นทำงานของโปรแกรม
2. onStart() คือ ทำงานเมื่อแสดงหน้าจอหลักของโปรแกรม
3. onResume() คือ ทำงานเมื่อมีการใช้งานกับผู้ใช้

4. onPause() คือ ทำงานเมื่อต้องการหยุดโปรแกรม
5. onStop() คือ ทำงานเมื่อโปรแกรมไม่ได้แสดงผลหน้าจอ
6. onDestroy() คือ ทำงานก่อนที่โปรแกรมจะถูกปิด
7. onRestart() คือ ทำงานใหม่อีกครั้ง หลังจากถูก onStop()

2.8.6 การจัดเก็บข้อมูลในแอสทิติวตี้

แอปพลิเคชันของแอนดรอยด์ จะมีการทำงานร่วมกับข้อมูลต่างๆ อยู่เสมอ ในระบบแอนดรอยด์มีไลบรารี ที่เกี่ยวข้องอยู่หลายตัว เช่น ข้อมูลประเภท แชร์เพอร์เฟอร์เรนซ์ (Share Preferences) เหมาะสำหรับข้อมูลที่มีขนาดไม่ใหญ่ เช่น ค่าเซ็ทอัพต่างๆ ฐานข้อมูลเอสคิวไลต์ SQLite เหมาะกับข้อมูลประเภททะเบียนที่รายการจำนวนมากๆ ไฟล์ข้อมูลประเภทสตรีมโดยใช้คำสั่ง InputStream และ OutputStream ในการอ่านเขียนข้อมูล

2.8.7 การใช้งานผู้ดำเนินการ (Handler)

โดยปกติแล้ว เมื่อต้องการจะเปลี่ยนแปลงค่าของวัตถุต่างๆ เช่นการเปลี่ยนข้อความในเท็กวิว (TextView) เปลี่ยนรูปภาพใน อิมเมจวิว (ImageView) เรามักจะกระทำโดยผ่านการโต้ตอบกับวัตถุนั้นๆ หรือวัตถุข้างเคียง เช่น กดปุ่มแล้วทำให้ข้อความเปลี่ยนแปลงแต่หากมีความต้องการที่จะเปลี่ยนแปลงค่าของวัตถุต่างๆ โดยที่เราไม่ได้ทำการโต้ตอบกับวัตถุใดๆ เลย เช่นการเปลี่ยนแปลงค่าโดยผ่านเธรด(Thread) หรือเปลี่ยนแปลงค่าเพราะทำการโหลดข้อมูลมาจากอินเทอร์เน็ต เป็นต้น การกระทำเหล่านี้จะต้องใช้ผู้ดำเนินการเข้ามาช่วย

2.9 เว็ปเซอร์ฟเวอร์

เว็บเซอร์เวอร์ที่ผู้จัดทำใช้บริการคือ ไมโครซอฟท์ แอสเซอร์ (Microsoft Azure) ซึ่งมีรายละเอียดดังนี้

ไมโครซอฟท์ แอสเซอร์เป็นแพลตฟอร์มคลาวด์คอมพิวเตอรส์และโครงสร้างพื้นฐานที่สร้างขึ้นโดยไมโครซอฟท์สำหรับการปรับใช้และการจัดการการใช้งานและบริการผ่านเครือข่ายทั่วโลกของศูนย์ข้อมูลไมโครซอฟท์ที่มีการจัดการมีทั้ง เพลทฟอรม์ที่เป็นการบริการ (Platform as a Service) และบริการ โครงสร้างพื้นฐาน (Infrastructure as a Service) และสนับสนุนหลายภาษาที่แตกต่างกันในการเขียนโปรแกรมเครื่องมือและกรอบรวมทั้งไมโครซอฟท์ที่เฉพาะเจาะจงและบุคคลที่สาม

บทที่ 3

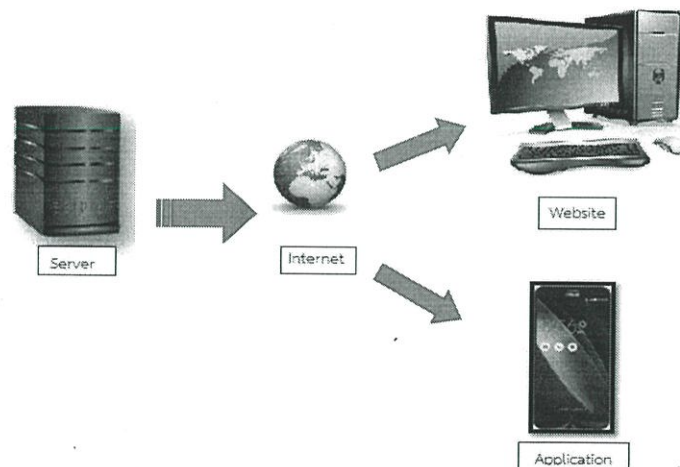
การออกแบบและการจัดทำปฏิญญาพันธ

ในโครงการส่วนนี้จะเป็นส่วนที่แสดงวิธีการออกแบบเว็บไซต์ และแอปพลิเคชัน โดยกล่าวถึง การทำงานภาพรวมของโครงการนี้

3.1 การออกแบบ

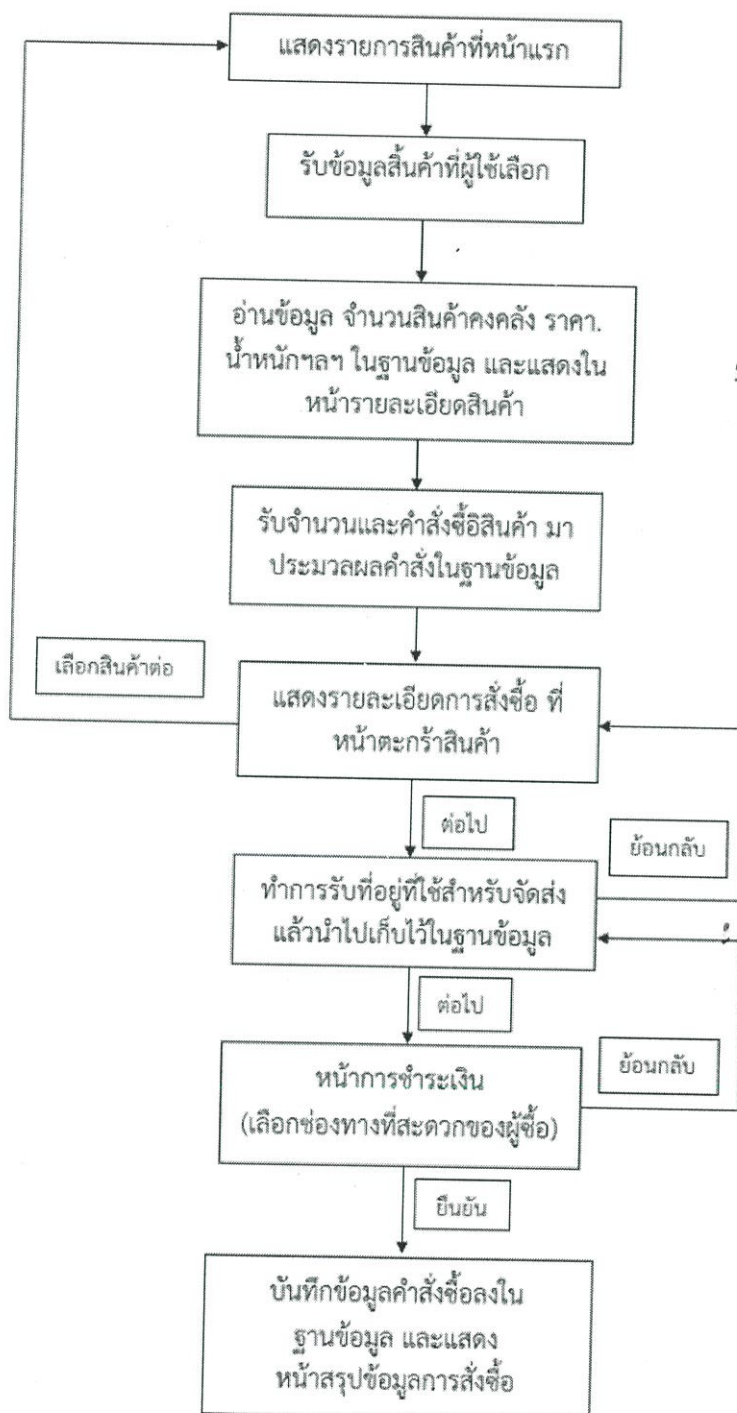
3.1.1. ภาพรวมของระบบ

การทำงานจะแบ่งออกเป็น 3 ส่วนซึ่งประกอบไปด้วยระบบหลังร้าน, ระบบหน้าเว็บ สำหรับผู้ใช้งานและแอปพลิเคชันบนสมาร์ตโฟนในระบบปฏิบัติการแอนดรอยด์ในส่วนการทำงานของระบบจัดการข้อมูลจะใช้ มายเอสคิวแอลในการจัดการฐานข้อมูล และจัดการในส่วนของการจัดการระบบหลังร้านไม่ว่าจะเป็นการแอดข้อมูลเข้าไปในระบบหลังบ้านทั้งหมด เช่น ราคาสินค้า ประเภทสินค้า การคำนวณยอดรวม เป็นต้น ซึ่งจะทำให้การเก็บข้อมูลไว้ใน มายเอสคิวแอล ภายในเซิร์ฟเวอร์ โดยที่ระบบจะทำการเชื่อมโยงไปยังระบบหน้าร้าน ซึ่งจะทำการแสดงข้อมูลโดยใช้ภาษาพีเอชพี ร่วมกับภาษา เอสคิวแอลในการเรียกใช้งานฐานข้อมูล ในส่วนของแอปพลิเคชันนั้น ทางเราใช้โปรแกรมแอนดรอยด์สตูดิโอในการสร้าง แล้วจะเรียกข้อมูลในส่วนของฐานข้อมูลทั้งหมดผ่านเซิร์ฟเวอร์ มาประมวลผล โดยที่เราจะไม่เก็บข้อมูลไว้ที่เครื่องเลยเพื่อป้องกันการขโมยข้อมูล ในการอาศัยช่องว่างที่ไม่ใช้อินเทอร์เน็ตในการจัดการดังรูปที่ 3.1



รูปที่ 3.1 ผังการทำงานของการทำงานของระบบ

3.1.2. บล็อกไดอะแกรมรวมการทำงานของเว็บไซต์

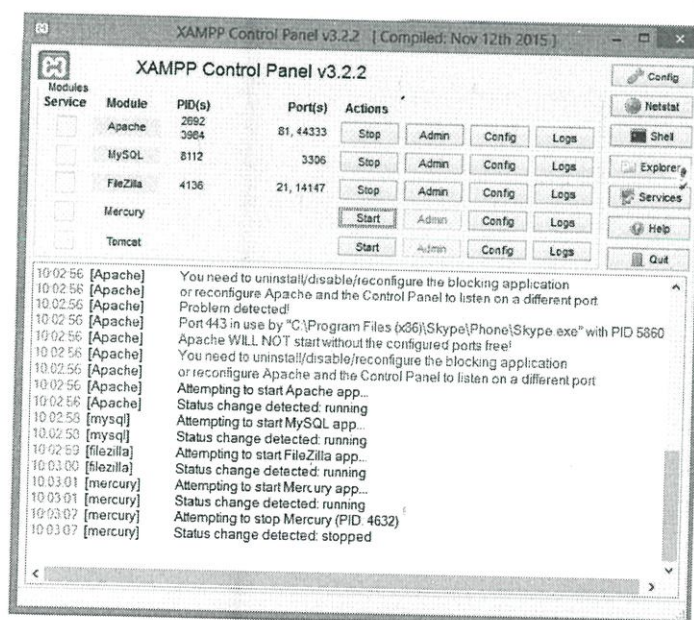


รูปที่ 3.2 บล็อกไดอะแกรมรวมการทำงานของเว็บไซต์

ในการทำงานโดยภาพรวมนั้น เมื่อผู้ที่สนใจได้ทำการเข้ามาในเว็บไซต์ จะเห็นสินค้าที่แสดงอยู่บนเว็บไซต์ ถ้ามีความสนใจในตัวสินค้า และทำการคลิกเลือกสินค้ามา 1 รายการ ในส่วนของเซิร์ฟเวอร์จะทำการรับข้อมูลเข้ามา และทำการอ่านข้อมูล จำนวนสินค้าคงคลัง, ราคา, น้ำหนัก ในฐานข้อมูล แล้วแสดงผลในหน้ารายละเอียดสินค้าตัวนั้น และถ้าทำการกดสั่งซื้อระบบจะรับจำนวนของสินค้าและคำสั่งซื้อมาประมวลผลคำสั่งในฐานข้อมูล แล้วลิงค์ไปยังหน้าตะกร้าสินค้าเพื่อเลือกที่จะซื้อต่อ ถ้าต้องการในส่วนนี้จะกลับไปหน้าแรกของเว็บไซต์ หรือ ดำเนินการต่อไปเพื่อจ่ายเงิน ถ้าต้องการในส่วนนี้จะลิงค์ไปยังหน้าต่อไปเพื่อรับที่อยู่ในการจัดส่งแล้วนำไปเก็บไว้ที่ฐานข้อมูลจากนั้นจะไปต่อยังหน้าชำระเงินเมื่อเลือกช่องทางในการชำระเรียบร้อยแล้วเมื่อกดยืนยันจะไปยังหน้าสรุปข้อมูลการสั่งซื้อและบันทึกคำสั่งซื้อนั้นลงในฐานข้อมูล

3.1.3 การเก็บฐานข้อมูลเบื้องต้น

เป็นขั้นตอนที่ใช้โปรแกรม XAMPP Control Panel ในการจำลองฐานข้อมูลเมื่อทำการลงโปรแกรม XAMPP Control Panel แล้ว ทำการเปิด Apache และ MySQL และทำการเปิด <http://localhost/phpmyadmin/> เพื่อทำการสร้างฐานข้อมูลจำลองภายในเครื่อง และเมื่อทำการอัปเดตข้อมูลเรียบร้อยแล้วก็จะนำข้อมูลไปเก็บไว้ในเซิร์ฟเวอร์ในขั้นตอนต่อไปดังรูปที่ 3.3



รูปที่ 3.3 XAMPP Control Panel

3.1.4 ออกแบบฐานข้อมูลสินค้า

3.1.4.1 ทำการออกแบบฐานข้อมูลสินค้า สำหรับผู้ประกอบการเพื่อรองรับข้อมูล ออเดอร์ที่มีการสั่งซื้อเข้ามาซึ่งประกอบไปด้วย หมายเลขออเดอร์ หมายเลขแสดงตัวตนของผู้ซื้อ ชื่อสินค้า ราคา จำนวนสินค้าที่สั่งซื้อ และที่อยู่สำหรับจัดส่ง ดังตารางด้านล่างนี้

ตารางที่ 1 การออกแบบฐานข้อมูล product_data

Name	Datatype	Comment
Product_ID	INT	หมายเลขสินค้าภายในฐานข้อมูล
product_name	TINYTEXT	ชื่อสินค้าในฐานข้อมูล
price	INT	ราคาสินค้าในฐานข้อมูล
price_sale	INT	ส่วนลดในฐานข้อมูล(บาท)
stock	INT	สินค้าคงคลังในฐานข้อมูล
weight	INT	น้ำหนักสินค้าในฐานข้อมูล
Description	TEXT	รายละเอียดสินค้า

ตารางที่ 2 การออกแบบตารางฐานข้อมูล product_order

Name	Datatype	Comment
order_ID	INT	บอกหมายเลขการสั่งซื้อ
user_id	INT	บอกว่าใครสั่งซื้อ
bill_address	TEXT	ที่อยู่ออกไปเสร็จ
payment	TEXT	สถานะการจ่ายเงิน ให้กำหนดเป็นตัวอักษร จ่ายแล้ว =paid ยังไม่จ่าย=not_pay ปิดตะกร้าแต่ยังไม่จ่าย=close
datetime	DATETIME	เวลาที่เรากดสั่งซื้อ
ship_address	TEXT	ที่อยู่จัดส่ง

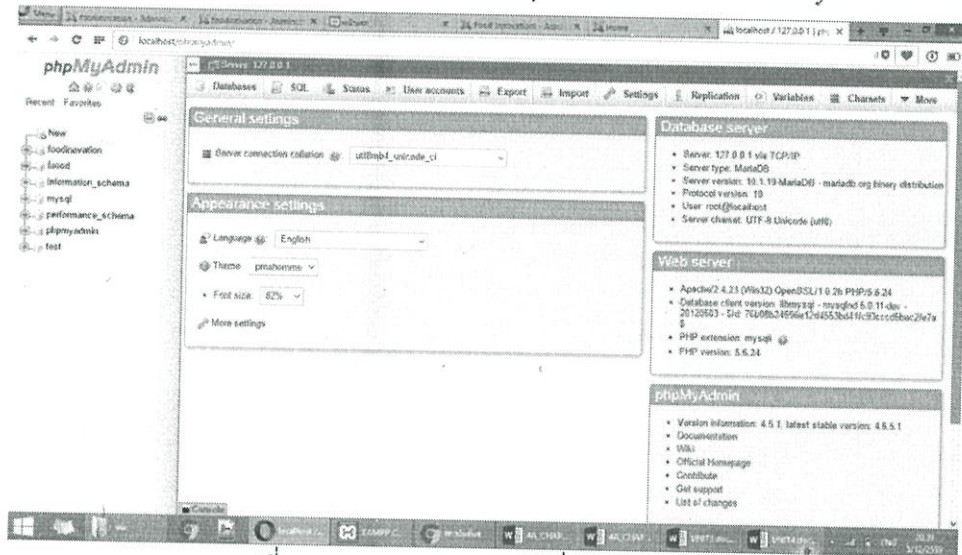
ตารางที่ 3 การออกแบบตารางฐานข้อมูล user_ID

Name	Datatype	Comment
user_id	INT	หมายเลขคนที่ซื้อ
user_name	INT	ชื่อคนที่สั่งซื้อ
password	TEXT	รหัสสำหรับการเข้าระบบสมาชิก
e_mail	TINYTEXT	อีเมลของผู้สมัคร
name_lastname	TEXT	ชื่อ นามสกุล
postcode	INT	รหัสไปรษณีย์
district	TEXT	เขต อำเภอ
province	TEXT	จังหวัด
mobile_number	INT	หมายเลขโทรศัพท์

ตารางที่ 4 การออกแบบตารางฐานข้อมูล cart

Name	Datatype	Comment
product_order_ID	INT	จะเก็บข้อมูลของ product id ว่ามีสินค้าอะไรบ้าง
order_ID	INT	ลำดับการสั่งซื้อ
Product_ID	INT	บอกว่าเป็นสินค้าตัวไหนบ้าง
Quantity	INT	จำนวนสินค้า
payment	TEXT	สถานะการจ่ายเงิน ให้กำหนดเป็นตัวอักษร จ่ายแล้ว =paid ยังไม่จ่าย=not_pay ปิดตะกร้าแต่ยังไม่จ่าย=close
date_order	DATETIME	วันที่สั่งซื้อ
total_price	INT	ราคารวมที่จะต้องจ่าย
total_weight	INT	น้ำหนักทั้งหมดของสินค้า
shipby	VARCHAR	รูปแบบการจัดส่ง ems หรือ ลงทะเบียน

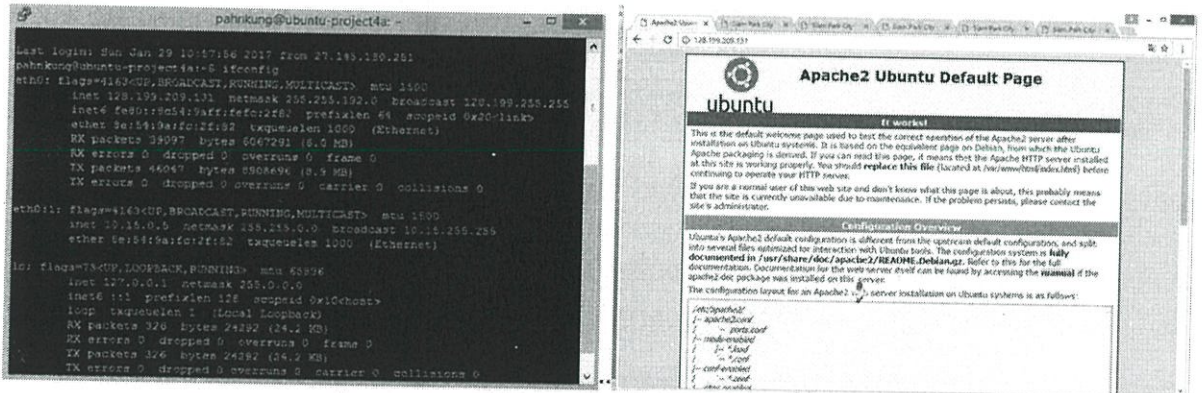
3.1.4.2 ทำการสร้างฐานข้อมูลสินค้าสร้างฐานข้อมูลจำลองภายในเครื่อง ซึ่งทำการทดลองสร้างเว็บไซต์บนเครื่องคอมพิวเตอร์เพื่อศึกษาการออกแบบเว็บไซต์ก่อนที่จะนำไปอัปโหลดบนเซิร์ฟเวอร์ในภายหลังดังรูปที่ 3.4



รูปที่ 3.4 Server จำลองในเครื่องคอมพิวเตอร์

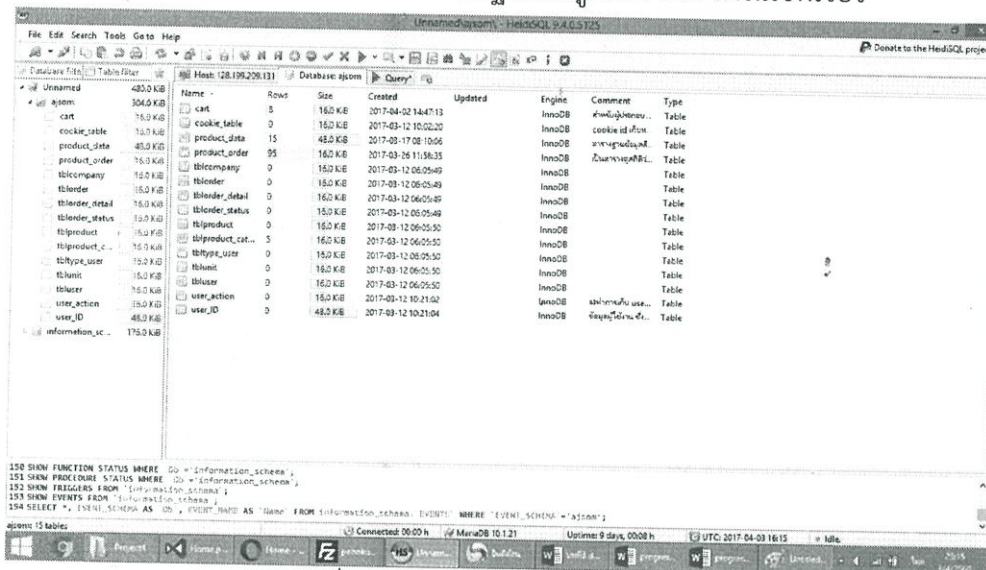
3.1.5 ระบบปฏิบัติการให้เซิร์ฟเวอร์

3.1.5.1 ทำการลงระบบปฏิบัติการให้แก่เซิร์ฟเวอร์ โดยทำการรีโมทลือคอนด้วยโปรแกรมพุตตีในการสั่งการ จากนั้นทำการอัปโหลดข้อมูลต่างๆ ลงในเซิร์ฟเวอร์ดังรูปที่ 3.5



รูปที่ 3.5 ระบบปฏิบัติการให้กับเซิร์ฟเวอร์

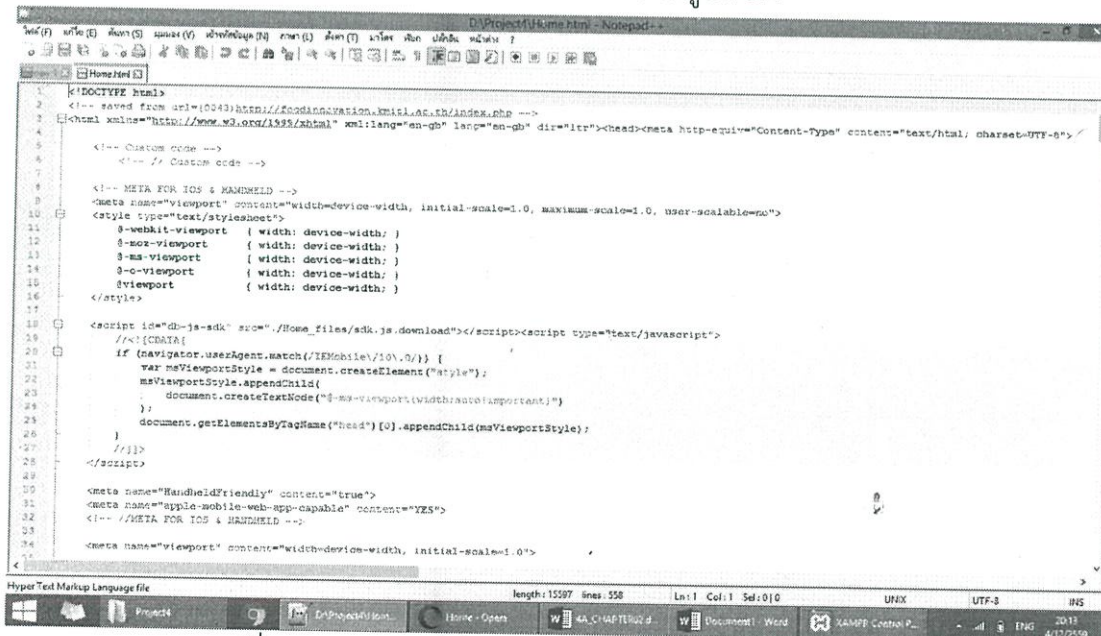
3.1.5.2 ทำการอัปเดตฐานข้อมูลที่สร้างไว้ ลงในเซิร์ฟเวอร์



รูปที่ 3.6 สร้างตารางฐานข้อมูลสินค้า

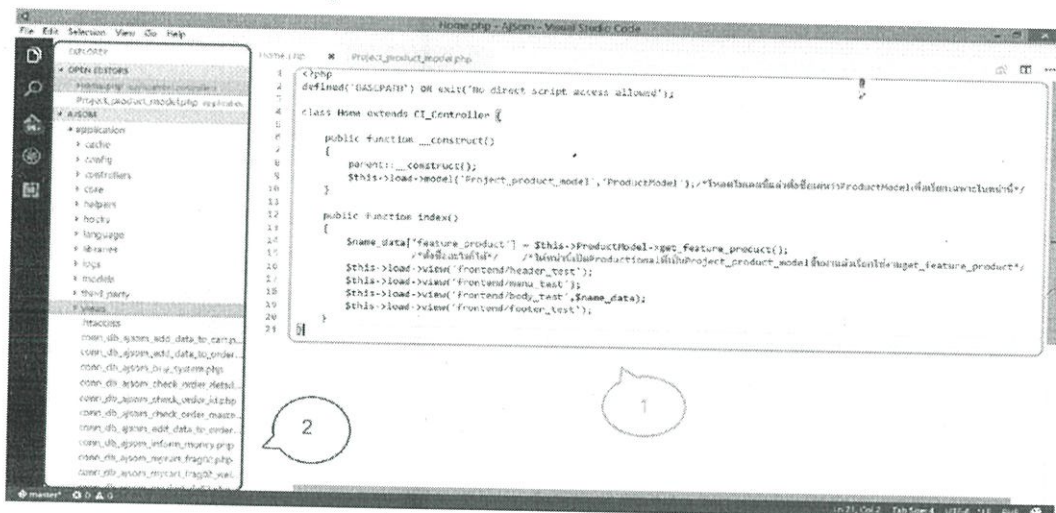
3.1.6 เว็บไซต์ (Website)

3.1.6.1 ทดลองเขียนหน้าเว็บโดยใช้ภาษาเอชทีเอ็มแอล ในการสร้างรูปแบบและตกแต่งหน้าเว็บไซต์ และอัปโหลดบนเซิร์ฟเวอร์ ดังรูปที่ 3.7

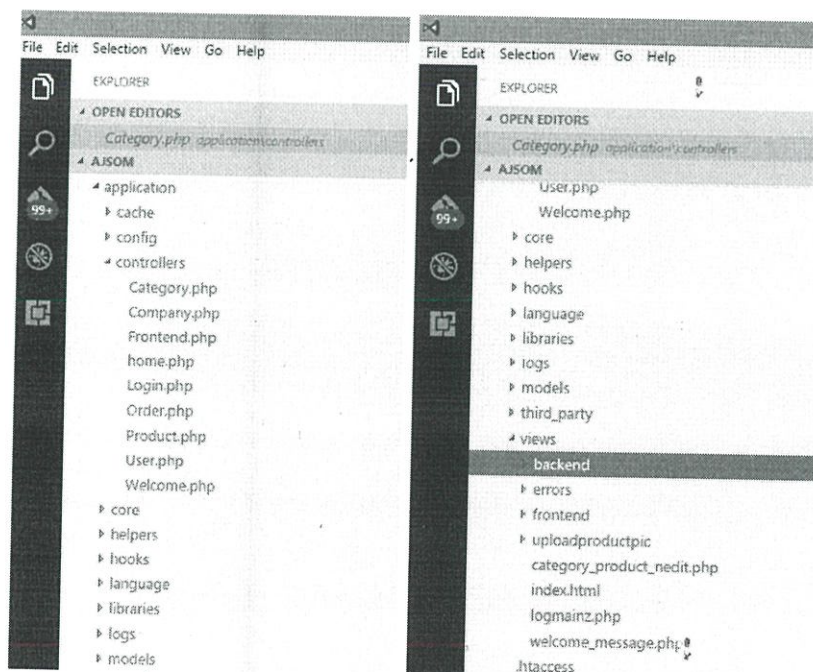


รูปที่ 3.7 ภาษาเอชทีเอ็มแอล ในการสร้างรูปแบบของเว็บไซต์

3.1.6.2 ปรับแต่ง PHP ให้รันพร้อม framework ได้ และทำการออกแบบ Model view controller (MVC) จัดทำหน้าเว็บไซต์โดยมีการปรับแต่ง PHP ให้รันพร้อม framework ได้ (หมายเลขที่1) และออกแบบ Model view controller (MVC) (หมายเลขที่2) เพื่อแยกส่วนซอฟต์แวร์ในส่วน ตรรกะเนื้อหา (domain logic) ได้แก่ความเข้าใจในระบบของผู้ใช้ และ ส่วนการป้อนข้อมูลและแสดงผล (GUI) ซึ่งช่วยให้การพัฒนา การทดสอบ และการดูแลรักษา ซอฟต์แวร์แยกออกจากกัน ดังรูปที่ 3.8



รูปที่ 3.8 Model view controller (MVC)



รูปที่ 3.9 Model view controller (MVC) (ขยายรูปที่ 1)

3.1.6.3 สร้างหน้าควบคุมเพื่อเรียกใช้งานหน้าเว็บเพจโดยการควบคุมหน้าเว็บเพจนั้น เราได้ทำการออกแบบให้มีการเรียกใช้งาน4ส่วนได้แก่ ส่วนหัว ส่วนเมนู ส่วนเนื้อหา และส่วนท้าย ซึ่งในการเรียกใช้งานนั้นจะเรียกโดยใช้ชื่อแทน Class ในหน้านั้นๆ ดังรูปที่ 3.10

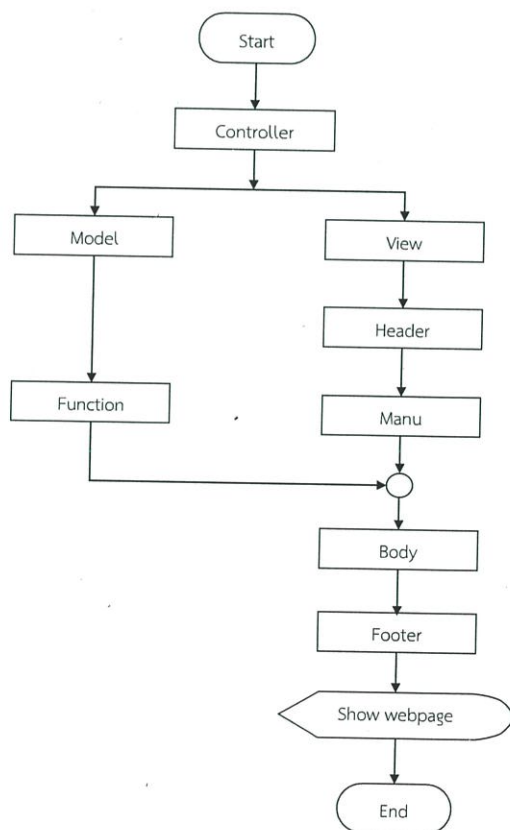
```

Home.php x Project_product_model.php
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Home extends CI_Controller {
5
6     public function __construct()
7     {
8         parent::__construct();
9         $this->load->model('Project_product_model','ProductModel'); /*โหลดโมเดลแล้วดึงข้อมูลจากProductModelเพื่อเรียกหาในหน้า*/
10    }
11
12    public function index()
13    {
14        $name_data['feature_product'] = $this->ProductModel->get_feature_product();
15        /*ดึงชื่ออะไรก็ได้*/ /*ให้หน้านี้เป็นProductionalที่เป็นProject_product_modelเข้ามาแล้วเรียกมีมาget_feature_product*/
16        $this->load->view('frontend/header_test');
17        $this->load->view('frontend/manu_test');
18        $this->load->view('frontend/body_test',$name_data);
19        $this->load->view('frontend/footer_test');
20    }
21

```

รูปที่ 3.10 สร้างหน้าควบคุมเพื่อเรียกใช้งานหน้าเว็บเพจ

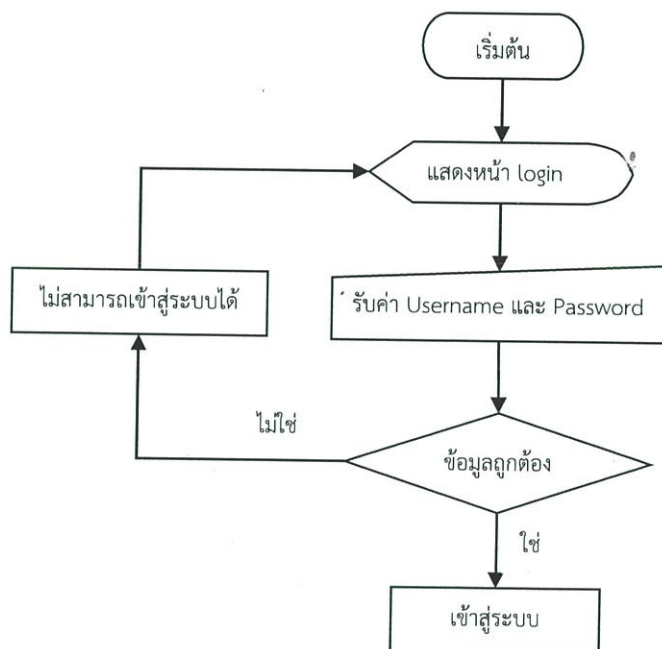
3.1.6.4 หน้าแรกโดยใช้ MVC



รูปที่ 3.11 ผังงาน (Flow chart) การทำงานของการเรียกหน้าเว็บเพจมาแสดง

การทำงานนั้น ส่วนควบคุม (Controller) จะทำการโหลดส่วนโมเดล (Model) สร้างฟังก์ชันมาเพื่อเปิดใช้งานฐานข้อมูล เพื่อเรียกใช้งานฟังก์ชัน `get_feature_product` ที่ได้มีการควิรฐานข้อมูลไว้มาแสดงบนหน้าเว็บเพจในส่วนของเนื้อหา และส่งโหลดส่วนวิว (View) เพื่อเรียกใช้งานหน้าเว็บเพจซึ่งประกอบไปด้วย ส่วนหัว ส่วนเมนู ส่วนเนื้อหา และส่วนท้าย ดังรูปที่ 3.11

3.1.6.5 หน้าลงชื่อเข้าใช้

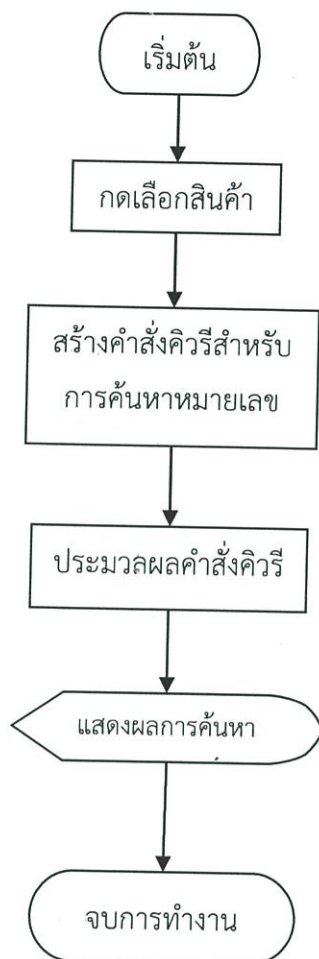


รูปที่ 3.12 ผังงานการทำงานของหน้าล็อกอินหน้าเว็บเพจ

การทำงานของหน้าล็อกอิน แสดงดังรูปที่ เริ่มต้นจะแสดงช่องกรอก Username และ Password ถ้าผู้ใช้ใส่ข้อมูล Username และ Password ถูกต้อง จะทำการเข้าสู่ระบบ แต่ถ้าผู้ใช้ใส่ข้อมูล Username และ Password ไม่ถูกต้อง จะไม่สามารถเข้าสู่ระบบได้และแสดงเป็นหน้าล็อกอินเหมือนเดิม ดังรูปที่ 3.12

หมายเหตุ: ผู้ที่มีสิทธิ์เข้าถึงหน้าแสดงข้อมูล ต้องทำการลงทะเบียนกับทางผู้ดูแลระบบก่อน

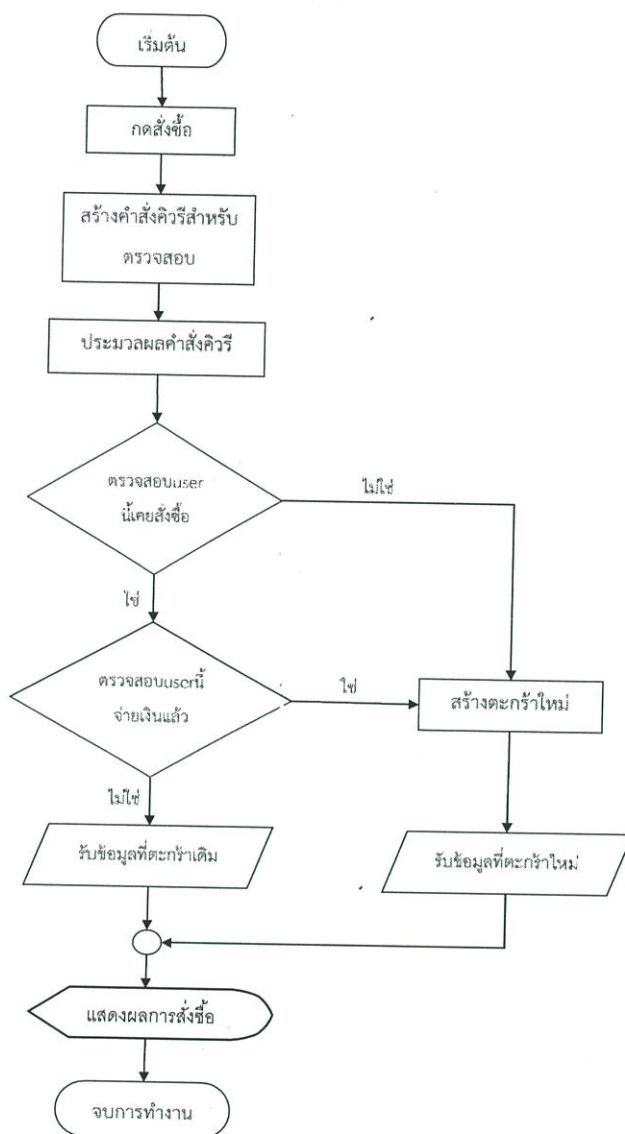
3.1.6.6 หน้ารายละเอียดสินค้า



รูปที่ 3.13 ผังงานแสดงการทำงานเมื่อกดเข้ามาที่สินค้า

การทำงานของหน้ารายละเอียดสินค้า แสดงดังรูปที่ เมื่อมีการกดเลือกสินค้าจากหน้าแรกเข้ามา จากคำสั่งควรีที่ได้ทำการสร้างไว้จะทำการประมวลผล เพื่อเรียกใช้งานฐานข้อมูลมาแสดงยังหน้าเว็บ โดยหลักในการเรียกหน้าเว็บนั้นจะใช้ MVC เป็นหลัก ดังรูปที่ 3.13

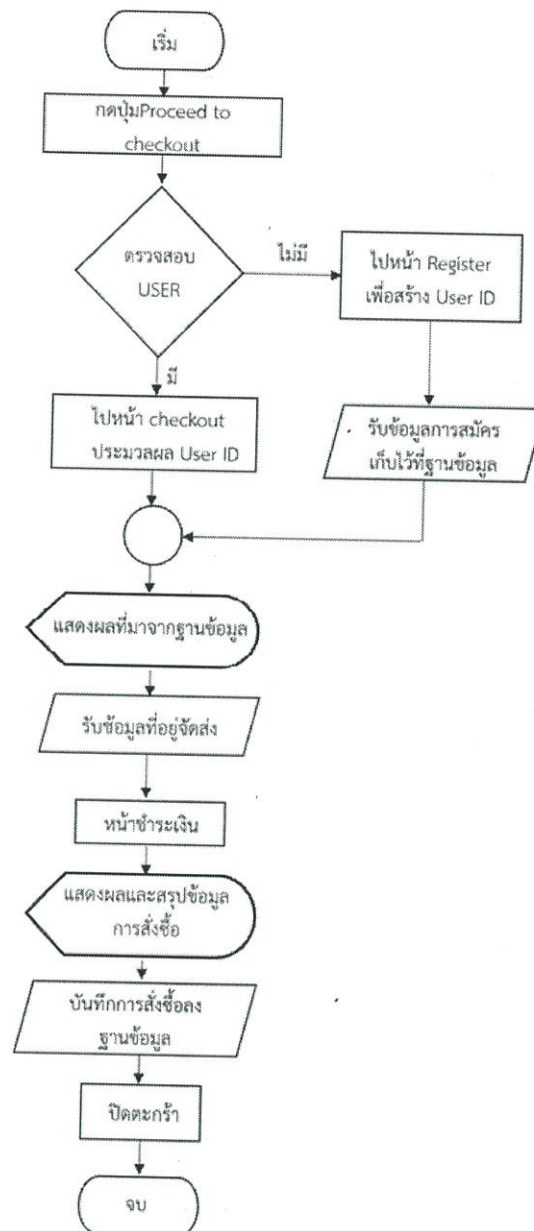
3.1.6.7 หน้าตะกร้าสินค้า



รูปที่ 3.14 ผังงานแสดงการทำงานเมื่อมีการกดสั่งซื้อเพิ่มคำสั่งซื้อลงในดาต้าเบส

เมื่อมีการกดปุ่ม Add to cart จะมีการประมวลผลจากคำสั่งซื้อที่ได้ทำการสร้างไว้ โดยทำการตรวจสอบว่าผู้ใช้หมายเลขไอดีนี้ มีประวัติการสั่งซื้อเข้ามาหรือไม่ ถ้าไม่ใช่ ให้ทำการสร้างตะกร้าสินค้าขึ้นมาใหม่ แล้วรับข้อมูลเข้าที่ตะกร้าใหม่ แต่ถ้าเคยสั่งซื้อแล้ว จะทำการตรวจสอบอีกรอบว่ามีการชำระรายการสินค้านั้นๆไปหรือยัง ถ้าใช่สร้างตะกร้าใหม่ แต่ถ้ายังไม่ชำระ ให้รับข้อมูลเข้าที่ตะกร้าเดิม แล้วแสดงผลการสั่งซื้อที่หน้าตะกร้าสินค้า ดังรูปที่ 3.14

3.1.6.8 หน้าเช็คเอาท์

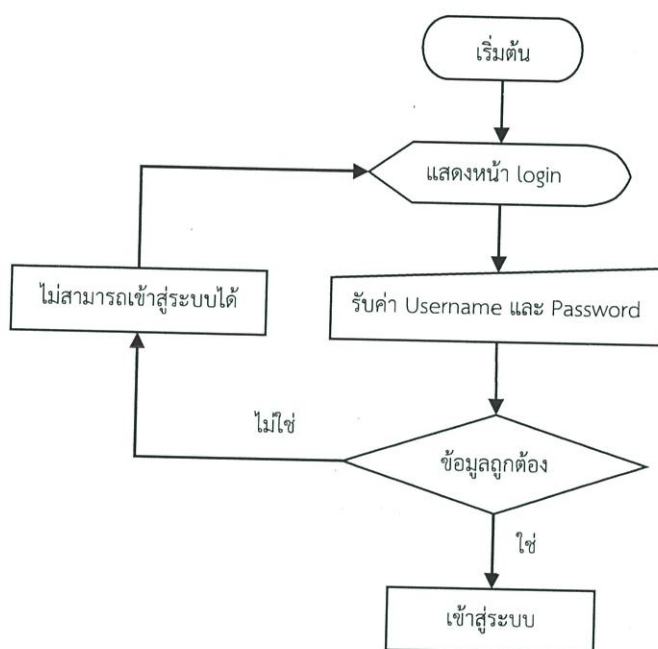


รูปที่ 3.15 ผังงานแสดงการทำงานเมื่อต้องการชำระเงิน

เมื่อกดปุ่ม Proceed to checkout คำสั่งคิวรีจะประมวลผลทำการตรวจสอบ USER ถ้าไม่มี จะไปที่หน้าลงทะเบียนสมาชิก แล้วรับเข้ามาเก็บในฐานข้อมูล แต่ถ้ามีการล็อกอินแล้ว จะไปที่หน้าเช็คเอาท์ จะมีให้กรอกรายละเอียด ที่อยู่จัดส่ง เมื่อกดต่อไปจะแสดงรายละเอียดข้อมูล

จากฐานข้อมูลเพื่อตรวจสอบความถูกต้องเมื่อผู้ใช้กดยืนยันจะทำการบันทึกข้อมูลเก็บไว้ในฐานข้อมูล และทำการปิดตะกร้าโดยแสดงสถานะเป็นรอโอน ดังรูปที่ 3.15

3.1.6.9 หน้า back-end

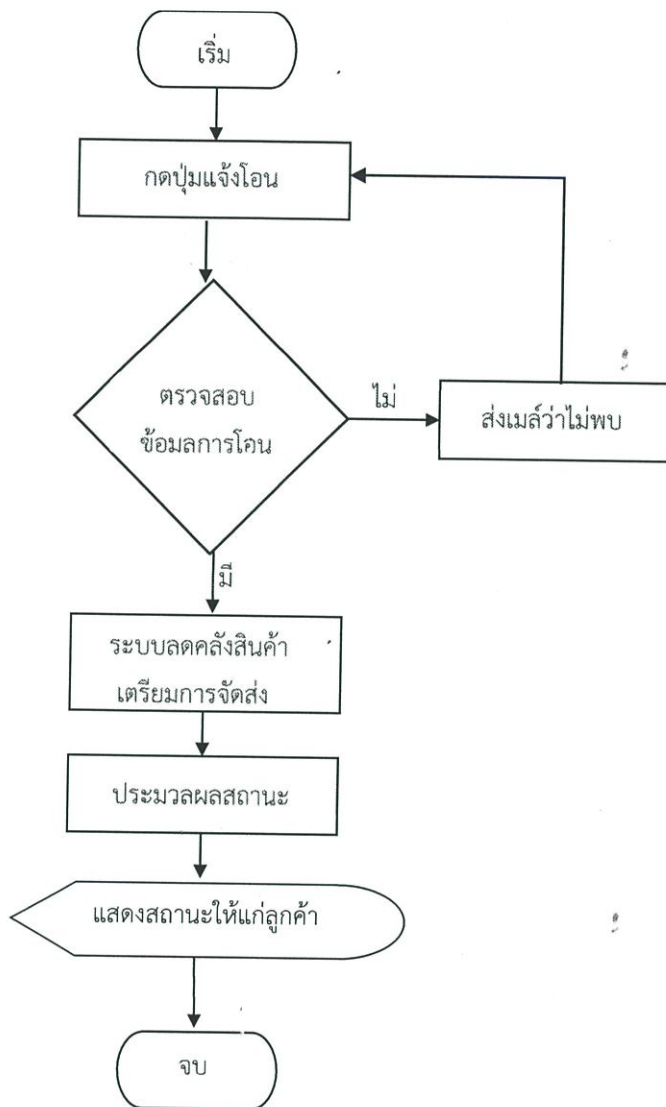


รูปที่ 3.16 ผังงานการทำงานของหน้าล็อกอินหน้าback-end

การทำงานของหน้าล็อกอิน แสดงดังรูปที่ เริ่มต้นจะแสดงช่องกรอก Username และ Password ถ้าผู้ใช้ใส่ข้อมูล Username และ Password ถูกต้อง จะทำการเข้าสู่ระบบ แต่ถ้าผู้ใช้ใส่ข้อมูล Username และ Password ไม่ถูกต้อง จะไม่สามารถเข้าสู่ระบบได้และแสดงเป็นหน้าล็อกอินเหมือนเดิม ดังรูปที่ 3.16

หมายเหตุ: ผู้ที่มีสิทธิ์เข้าถึงหน้าแสดงข้อมูล ต้องทำการลงทะเบียนกับทางผู้ดูแลระบบก่อน

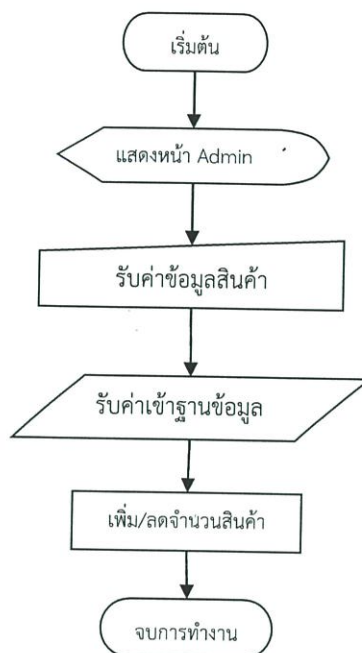
3.1.6.11 หน้า back-end



รูปที่ 3.17 ผังงานแสดงการทำงานตรวจสอบการโอนเงิน

เมื่อกดปุ่มแจ้งโอน ทางเราจะทำการตรวจสอบความถูกต้อง ถ้าไม่พบข้อมูลการโอนเงิน จะทำการส่งอีเมลแจ้งเตือนเนื่องจากอาจใส่ข้อมูลผิดพลาด ถ้าตรวจสอบแล้วตรงกับข้อมูล ระบบจะทำการลดสต็อก และทางเราจะเตรียมการจัดส่งโดยการพิมพ์ที่อยู่จัดส่งให้พร้อม แล้วทำการประมวลผลเพื่ออัปเดตสถานะเป็นเตรียมการจัดส่งแล้วแสดงขึ้นที่หน้าข้อมูลการสั่งซื้อ ดังรูปที่ 3.17

3.1.6.12 หน้า back-end เพิ่มสินค้า



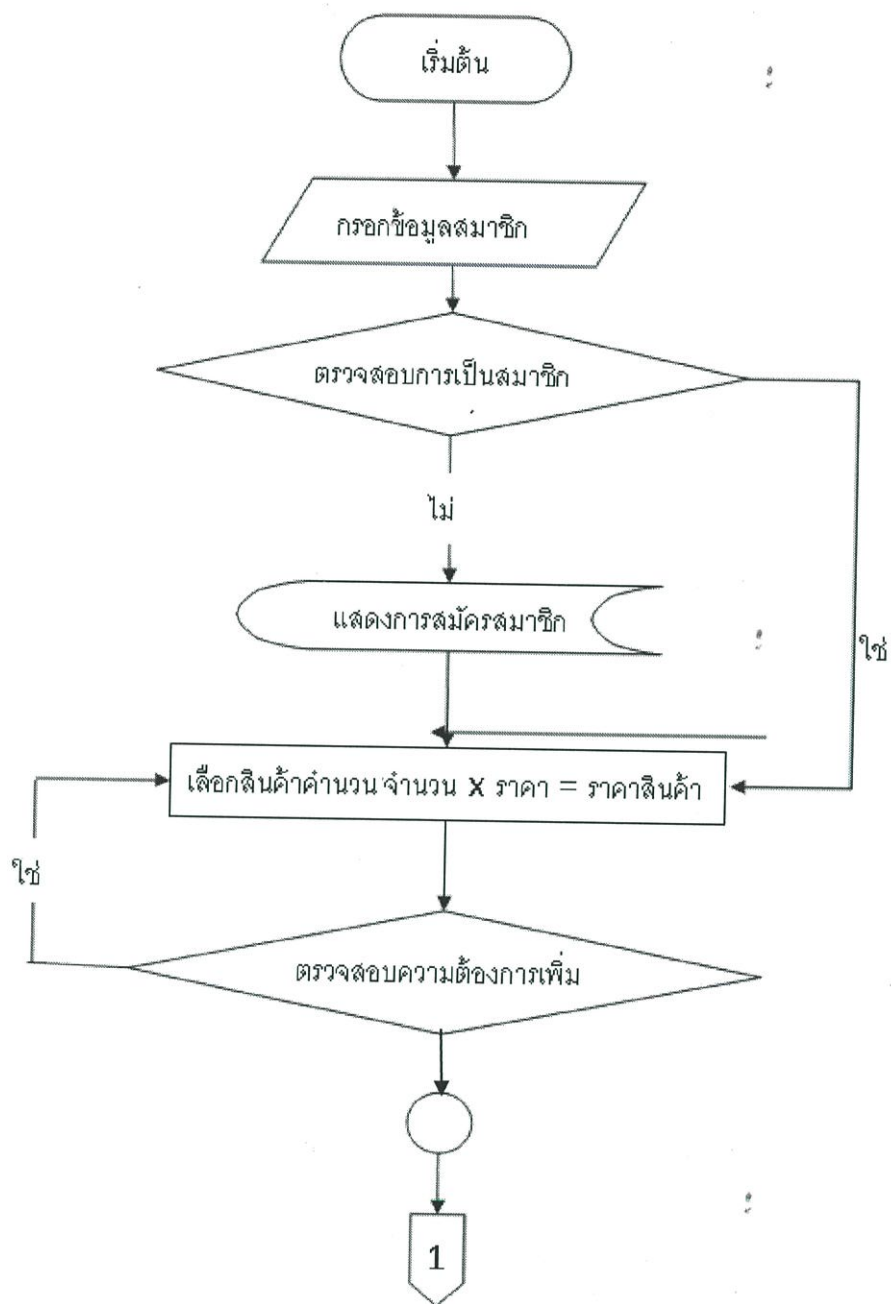
รูปที่ 3.18 ผังงานแสดงการทำงานการเพิ่มสินค้าของผู้ประกอบการ

เมื่อล็อกอินเรียบร้อยแล้ว จะสามารถเข้าสู่หน้าแอดมินจัดการสินค้าได้ และทำการรับค่าจากคีย์บอร์ด เก็บพื้นฐานข้อมูล จากนั้นจำนวนหรือรายละเอียดสินค้า จะอัปเดตทันที

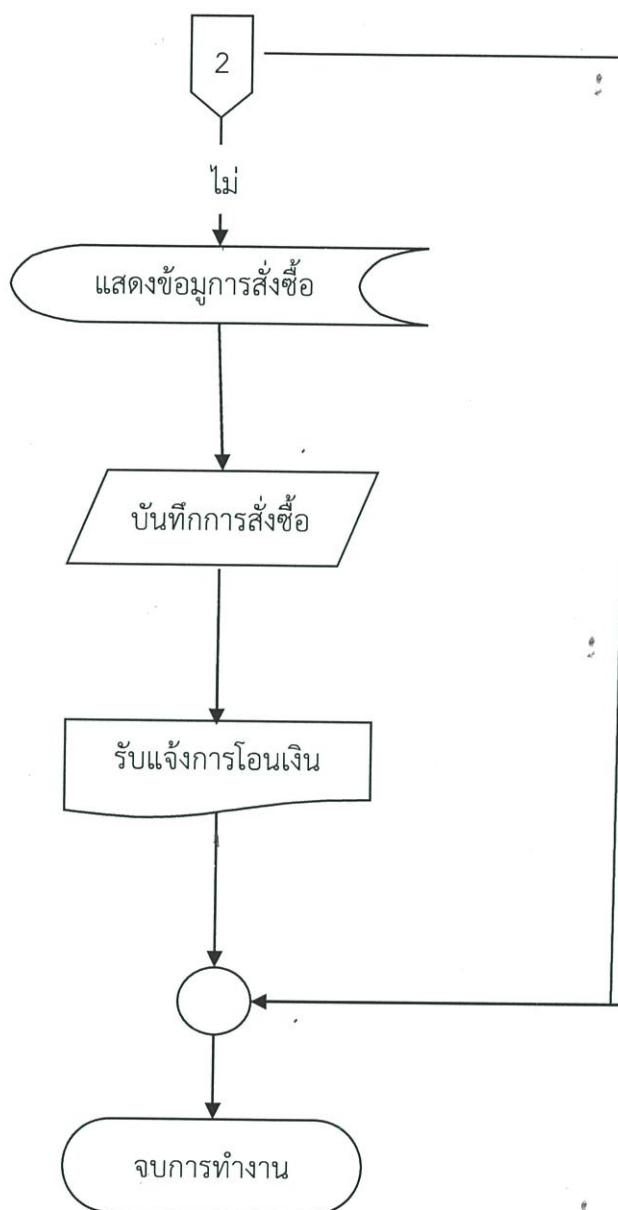
3.1.7 การจัดทำแอปพลิเคชัน

3.1.7.1 บล็อกไดอะแกรมการทำงานของแอปพลิเคชัน

จากบล็อกไดอะแกรมการทำงานของแอปพลิเคชัน เมื่อเข้ามาใช้งานแอปพลิเคชัน ลูกค้าจะต้องเป็นสมาชิก โดยถ้าคนเป็นสมาชิกแล้วสามารถเลือกซื้อสินค้า และทางระบบจะทำการการคำนวณราคาสินค้า ส่วนคนที่ไม่เป็นสมาชิกต้องสมัครสมาชิกก่อน โดยการกรอกข้อมูล และเราจะจัดเก็บข้อมูลลงใน ฐานข้อมูล และจึงเลือกซื้อสินค้าได้ตามปกติ จากนั้นจะต้องมีการตรวจสอบความต้องการการสั่งซื้อเพิ่มถ้า ไม่ก็สรุปข้อมูลการสั่งซื้อและบันทึกและรอรับแจ้งการโอนเงิน เพื่อจัดส่งสินค้าต่อไป แสดงดังรูปที่ 3.19 และ 3.20



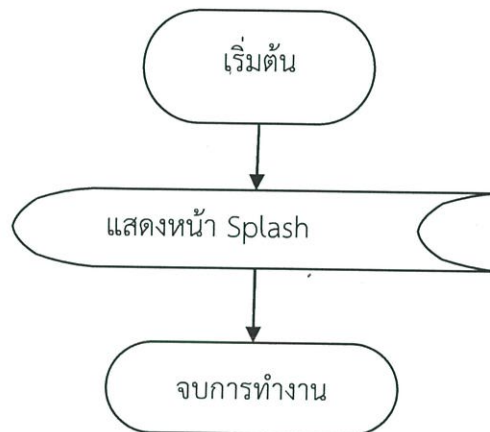
รูปที่ 3.19 ผังการทำงานของแอปพลิเคชัน (ก)



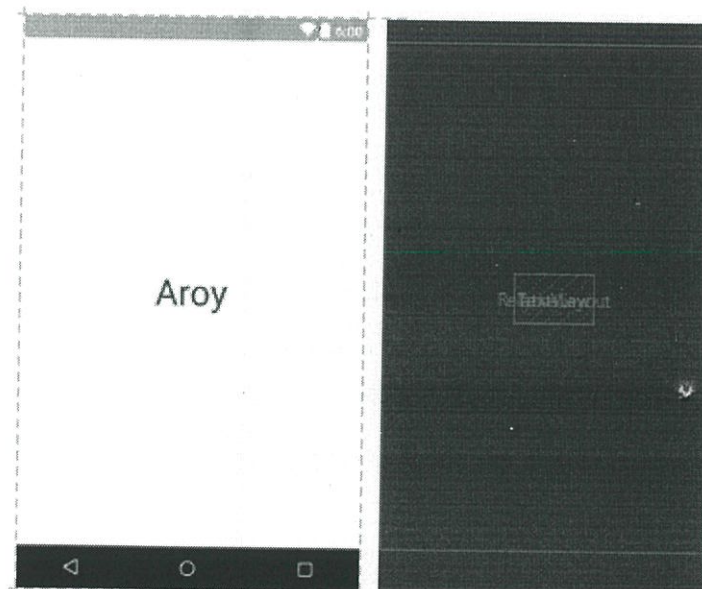
รูปที่ 3.20 ผังการทำงานของแอปพลิเคชัน (ข)

3.1.7.2 หน้า activity Splash Screen

เมื่อทำการเปิดแอปพลิเคชัน จะแสดงหน้า Activity Splash Screen เป็นหน้าแรก ซึ่งเป็นหน้าที่ต้องการให้แสดงก่อนเข้าสู่หน้าลงชื่อใช้งาน โดยหน้านี้จะแสดงชื่อของแอปพลิเคชัน ผังการทำงานของแอปพลิเคชันหน้า Activity Splash Screen แสดงดังรูปที่ 3.1๕ การออกแบบแอปพลิเคชันหน้า Splash Screen แสดงดังรูปที่ 3.21



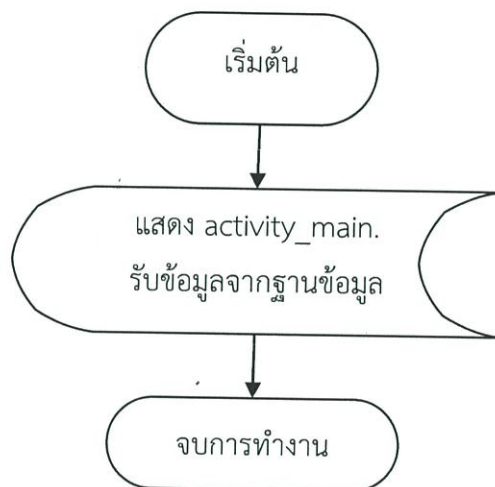
รูปที่ 3.21 ผังการทำงานของแอปพลิเคชันหน้า Splash Screen



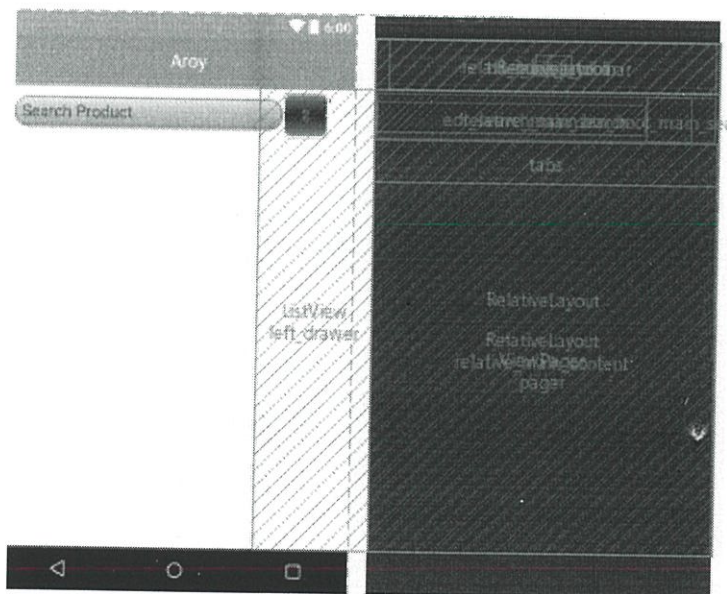
รูปที่ 3.22 การออกแบบแอปพลิเคชันหน้า Splash Screen

3.1.7.3 หน้า activity_main.

แสดงข้อมูลของ activity_main. ผังการทำงานของแอปพลิเคชันหน้าเกี่ยวกับ แสดงดังรูปที่ 3. และการออกแบบแอปพลิเคชัน แสดงดังรูปที่ 3.21 การออกแบบแอปพลิเคชันหน้า activity_main.แสดงดังรูปที่ 3.23



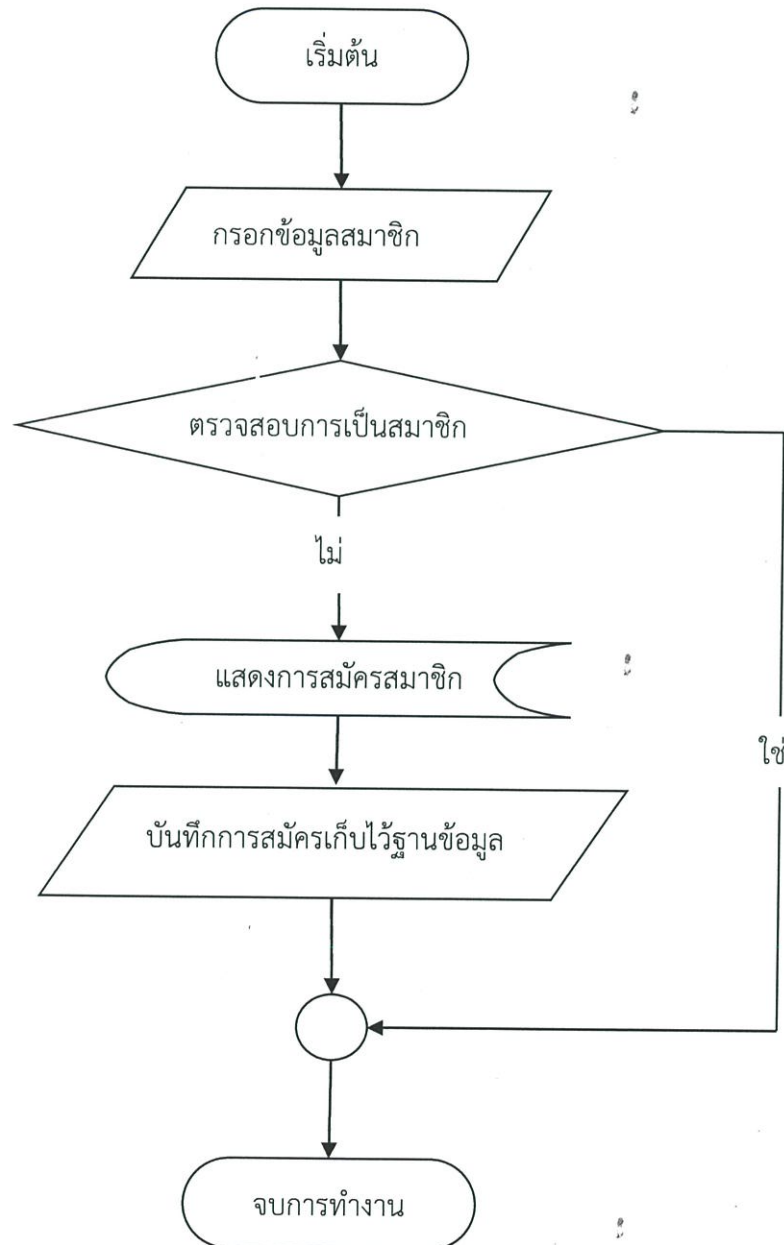
รูปที่ 3.23 ผังการทำงานของแอปพลิเคชันหน้า activity_main.



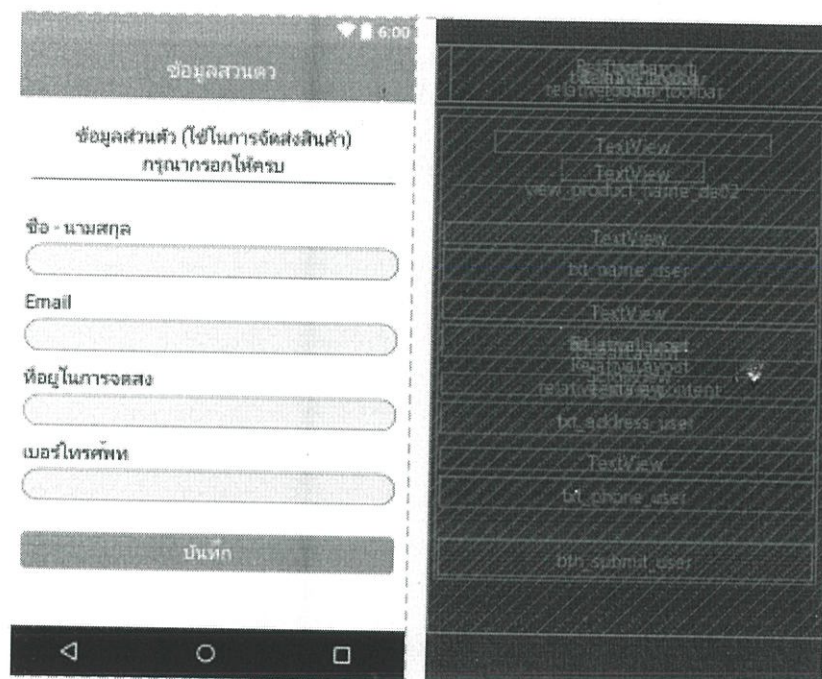
รูปที่ 3.24 การออกแบบแอปพลิเคชันหน้า activity_main.

3.1.7.4 การทำงานของแอปพลิเคชันหน้าสมัครสมาชิก

บล็อกไดอะแกรมการทำงานของหน้าสมาชิกของแอปพลิเคชัน เมื่อเข้ามาใช้งาน จะต้องสมัครสมาชิกก่อน โดยการกรอกข้อมูล และเราจะจัดเก็บข้อมูลลงใน ฐานข้อมูล เพื่อเรียกใช้งานในครั้งถัดไป แสดงดังรูปที่ 3.23 และการออกแบบแอปพลิเคชัน แสดงดังรูปที่ 3.25



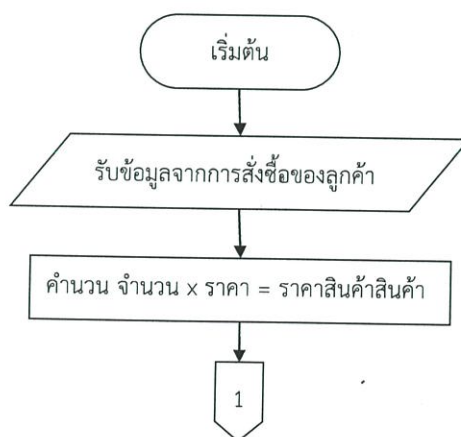
รูปที่ 3.25 ผังการทำงานของแอปพลิเคชันหน้าสมัครสมาชิก

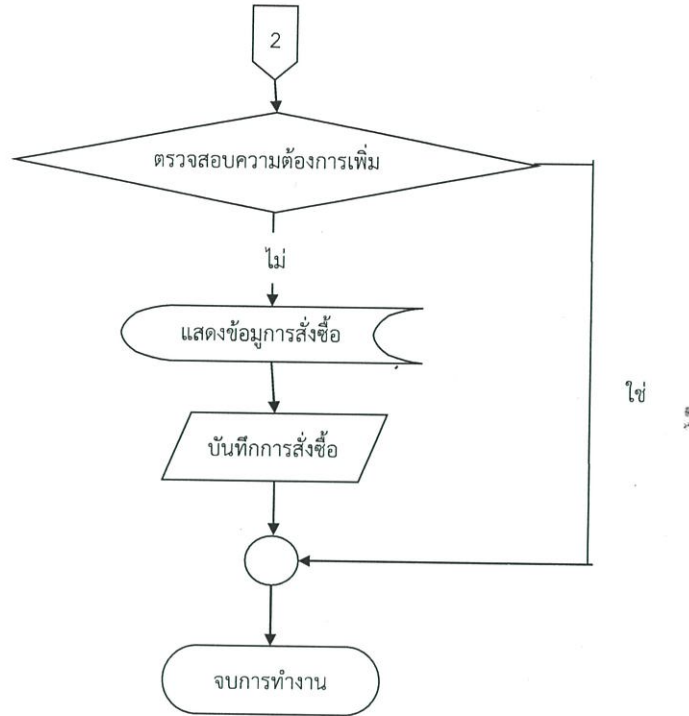


รูปที่ 3.26 การออกแบบแอปพลิเคชันหน้าสมัครสมาชิกและข้อมูลสมาชิก

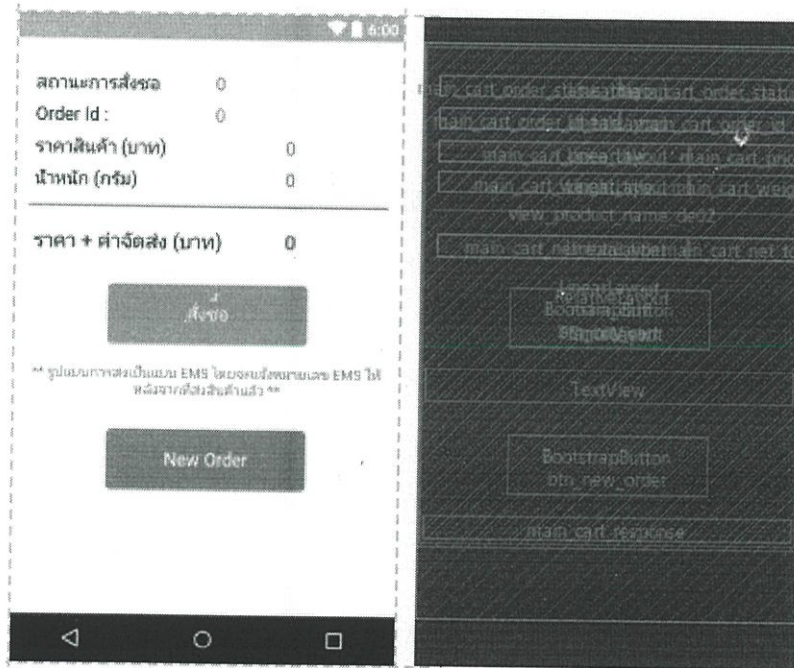
3.1.7.5 หน้าสินค้าของฉัน

ทางระบบจะทำการการคำนวณราคาสินค้าจากนั้นจะต้องมีการตรวจสอบความต้องการการสั่งซื้อเพิ่มถ้าไม่ก็สรุปข้อมูลการสั่งซื้อแสดงดังรูปที่ 3.25 และการออกแบบแอปพลิเคชันแสดงดังรูปที่ 3.27





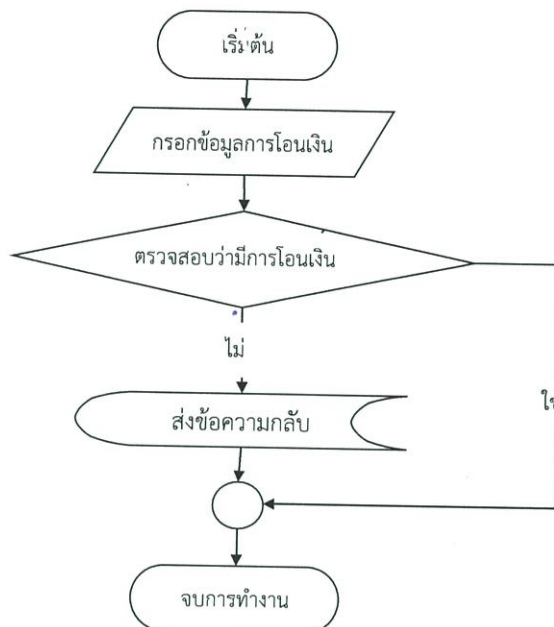
รูปที่ 3.27 ผังการทำงานของแอปพลิเคชันหน้าสินค้าของฉัน



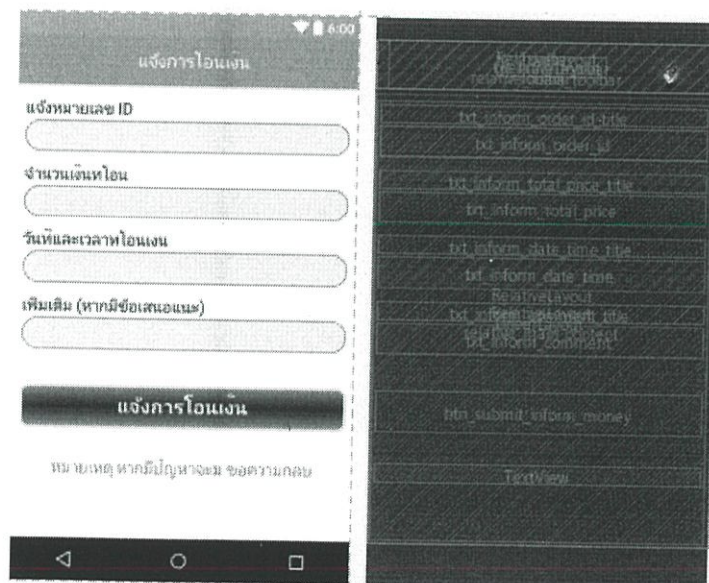
รูปที่ 3.28 การออกแบบแอปพลิเคชันหน้าสินค้าของฉัน

3.1.7.6 หน้าแจ้งการโอนเงิน

เมื่อผู้ซื้อแจ้งการโอนเงิน และเราจะตรวจสอบว่าได้รับจริงและถ้ายังไม่ได้ให้แจ้งกลับไปทางลูกค้าและถ้าได้รับ จึงส่งสินค้าแสดงดังรูปที่ 3.27 และการออกแบบแอปพลิเคชัน แสดงดังรูปที่ 3.29



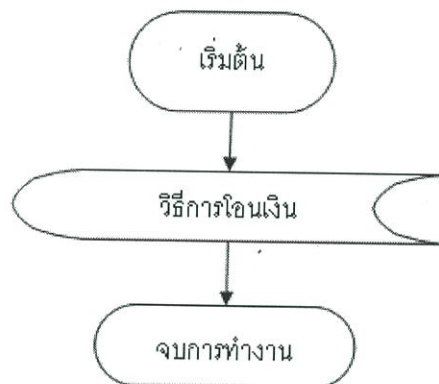
รูปที่ 3.29 ผังการทำงานของแอปพลิเคชันหน้าแจ้งการโอนเงิน



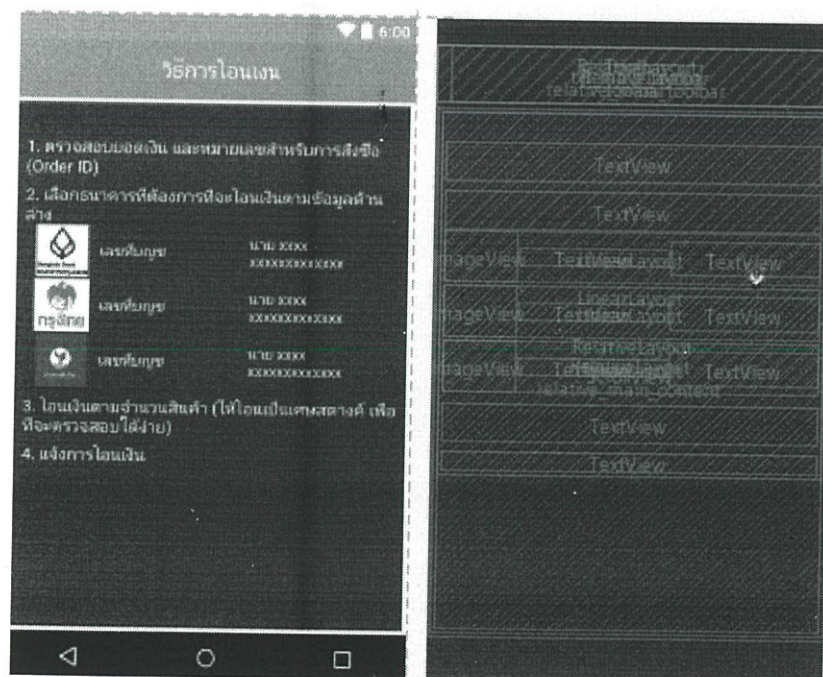
รูปที่ 3.30 การออกแบบแอปพลิเคชันหน้าแจ้งการโอนเงิน

3.1.7.7 หน้าวิธีการโอนเงิน

แสดงวิธีการโอนเงินของแอปพลิเคชัน โดยผู้ใช้งานจะต้องเลื่อนหรือปิดหน้าจอไปทางซ้ายหรือทางขวา เพื่อดูหน้าแสดงวิธีการใช้งาน แสดงดังรูปที่ 3.29 และการออกแบบแอปพลิเคชัน แสดงดังรูปที่ 3.31



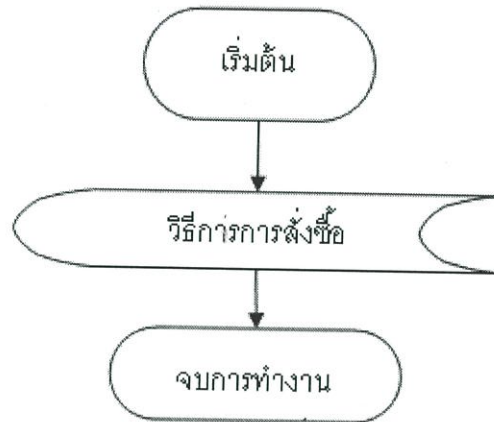
รูปที่ 3.31 ผังการทำงานของแอปพลิเคชันหน้าวิธีการโอนเงิน



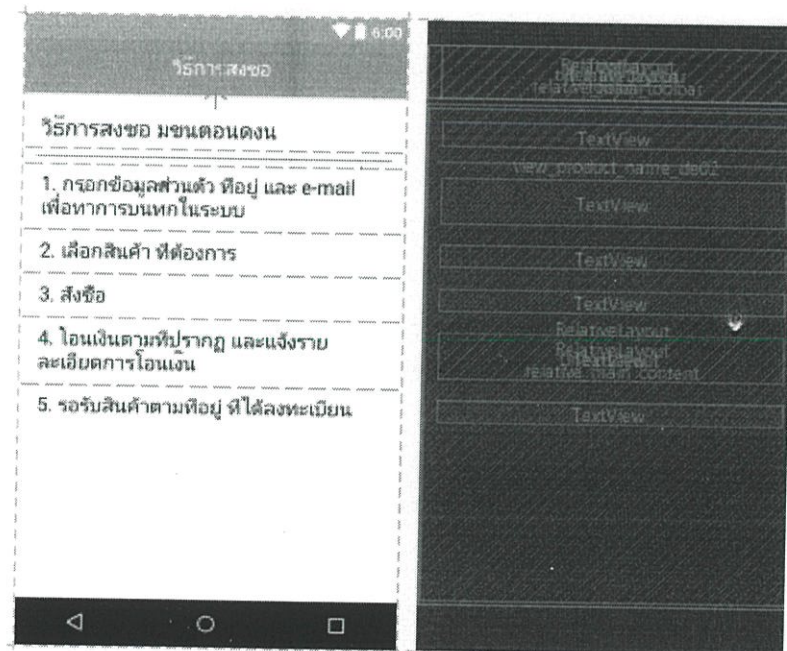
รูปที่ 3.32 การออกแบบแอปพลิเคชันหน้าวิธีการโอนเงิน

3.1.7.8 หน้าวิธีการสั่งซื้อ

แสดงวิธีการการวิธีการสั่งซื้อของแอปพลิเคชัน โดยผู้ใช้งานจะต้องเลื่อนหรือปิดหน้าจอไปทางซ้ายหรือทางขวา เพื่อดูหน้าแสดงวิธีการใช้งาน แสดงดังรูปที่ 3.31 และการออกแบบแอปพลิเคชันแสดงดังรูปที่ 3.33



รูปที่ 3.33 ผังการทำงานของแอปพลิเคชันหน้าวิธีการสั่งซื้อ



รูปที่ 3.34 การออกแบบแอปพลิเคชันหน้าวิธีการสั่งซื้อ

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1	โทรศัพท์มือถือระบบปฏิบัติการแอนดรอย	1	ชุด
3.2.2	คอมพิวเตอร์แบบพกพา	1	ชุด
3.2.3	โปรแกรมที่ใช้ในการทดลอง		
3.2.3.1	เอ็กซ์แอม (xampp)		
3.2.3.2	วิซวลสตูดิโอ (visual Studio Express Products)		
3.2.3.3	โค้ดอีนเตอร์ (codeigniter)		
3.2.3.4	ไฮดี เอสคิวแอล (heidi sql)		
3.2.3.5	เซิร์ฟเวอร์ (server)		
3.2.3.6	จาวาดีเวลลอปเม้นคิต (java development kit)		
3.2.3.7	ไฟล์ซิลล่า (filezilla)		
3.2.3.8	แอนดรอยสตูดิโอ (android studio)		

3.3 การจัดเก็บผลการทดลอง

ได้มีการทดสอบเพื่อความถูกต้องของข้อมูลที่จะถูกนำไปใช้งาน โดยมีการทดสอบฐานข้อมูล, การค้นหาข้อมูลจากฐานข้อมูล และการนำข้อมูลจากฐานข้อมูลมาแสดงผล และได้มีการทดสอบเพื่อความถูกต้องในการแสดงผลข้อมูล

3.3.1 ทดสอบฐานข้อมูล

เมื่อผู้ใช้งานเว็บไซต์ และได้ทำการสมัครสมาชิก หรือทำการสั่งซื้อ ข้อมูลที่รับเข้ามานี้จะถูกจัดเก็บในฐานข้อมูล ซึ่งประกอบไปด้วยชื่อ นามสกุล ที่อยู่ เบอร์โทร อีเมล โดยจะถูกจัดเก็บในตารางที่ชื่อว่า user_ID ทำการบันทึกผลและตรวจสอบ

3.3.2 ทดสอบการค้นหาข้อมูลจากฐานข้อมูล

ทำการทดสอบการค้นหาตารางและคอลัมน์ที่ต้องการจากฐานข้อมูลนั้น โดยโปรแกรม HeidiSQL โดยใช้ภาษา SQL ในการเปิดฐานข้อมูล บันทึกผลและตรวจสอบ

3.3.3 ทดสอบการนำข้อมูลมาแสดง

3.3.3.1 ทดสอบบนเว็บไซต์

ทำการทดลองนำข้อมูลมาแสดงโดยใช้ภาษา PHP ร่วมกับ SQL ในการเปิดฐานข้อมูล โดยให้มีการแสดงสินค้าที่หน้าแรกของเว็บไซต์ รายละเอียดสินค้า ตะกร้าสินค้า และหน้าสรุปการสั่งซื้อ ทดสอบโดยการสั่ง echo ออกมาที่หน้าเว็บไซต์ ทำการบันทึกผลและตรวจสอบ

3.3.3.2 ทดสอบบนแอปพลิเคชัน

ทำการทดลองนำข้อมูลมาแสดงโดยใช้ภาษา PHP ร่วมกับ SQL ในการเปิดฐานข้อมูล โดยให้มีการแสดงสินค้าที่หน้าแรกแอปพลิเคชัน รายละเอียดสินค้า ตะกร้าสินค้า และหน้าสรุปการสั่งซื้อ ทดสอบโดยการกดสั่งซื้อสินค้าในแอปพลิเคชันจริงๆว่าเป็นไปตามที่เขียนฟังก์ชันไว้หรือไม่ ทำการบันทึกผลและตรวจสอบ

บทที่ 4

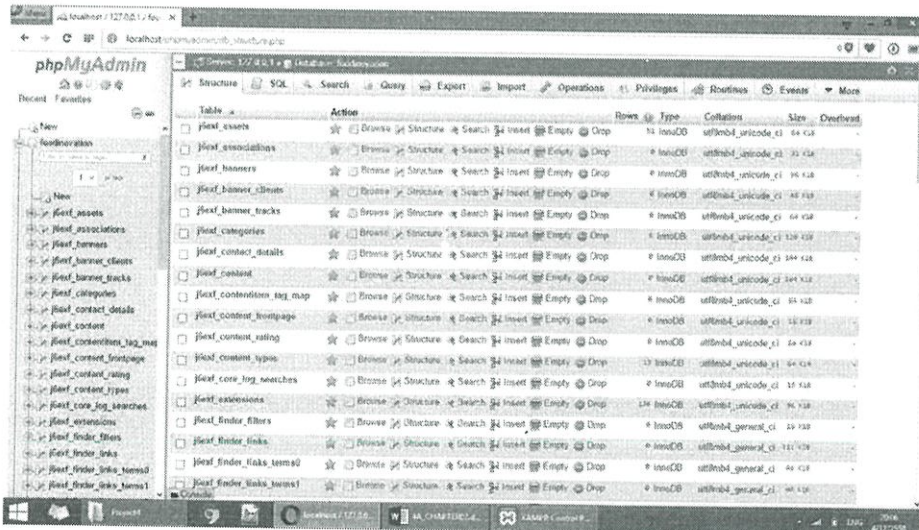
ผลการทดลอง

ในส่วนของการทดลองเขียนเว็บและแอปพลิเคชัน โดยจะแบ่งการทดลองและจัดเก็บผลการทดลองออกเป็นส่วนต่างๆ ได้ดังต่อไปนี้

4.1 ผลการดำเนินงานของเว็บไซต์

4.1.1 การเก็บฐานข้อมูลเบื้องต้น

จากรูปที่ 4.1 เมื่อทำการลงโปรแกรม XAMPP เพื่อทำการสร้างฐานข้อมูลจำลองในเครื่องตนเอง และทำการลง Joomla ใน localhost เพื่อจัดทำเว็บไซต์ เมื่อต้องการแก้ไขข้อมูล จะทำการเข้าไปที่ <http://localhost/food/administrator/index.php> แล้วใส่รหัสผ่านที่ได้ทำการตั้งไว้ จะได้ดังรูปที่ 4.2



รูปที่ 4.1 ฐานข้อมูลในเซิร์ฟเวอร์จำลอง

4.1.2 ผลในการสร้างตารางฐานข้อมูลในเซิร์ฟเวอร์

โดยหลักการทำงานได้กล่าวไปแล้วในบทที่ 3 ศึกษาได้จาก ตารางที่ 1 การออกแบบฐานข้อมูล product_data

#	Name	Datatype	Length/Set	Unsigned	Allow N.	Zerofill	Default	Comment	Collation	Expression
1	Product_ID	INT	11				AUTO_INCREMENT	หมายเลขสินค้าในฐานข้อมูล		
2	product_name	TEXT						ชื่อสินค้าในฐานข้อมูล		utf8_bin_252_v2
3	price	FLOAT					0	ราคาสินค้าในฐานข้อมูล		
4	price_sale	FLOAT					0	ราคาสินค้าในฐานข้อมูล		
5	review_star	FLOAT					0	จำนวนรีวิวสินค้าในฐานข้อมูล		
6	product_type	INT	11				0	ประเภทของสินค้าในฐานข้อมูล		
7	rating_volume	INT	11				0	จำนวนการรีวิวสินค้าในฐานข้อมูล		
8	stock	INT	11				0	จำนวนสินค้าในฐานข้อมูล		
9	cost	FLOAT					0	ต้นทุนสินค้าในฐานข้อมูล		
10	weight	FLOAT					0	น้ำหนักสินค้าในฐานข้อมูล		
11	return	INT	11				0	จำนวนการคืนสินค้าในฐานข้อมูล		
12	Relate_product1	INT	11				0	สินค้าที่เกี่ยวข้องที่ 1		
13	Relate_product2	INT	11				0	สินค้าที่เกี่ยวข้องที่ 2		
14	Relate_product3	INT	11				0	สินค้าที่เกี่ยวข้องที่ 3		
15	Relate_product4	INT	11				0	สินค้าที่เกี่ยวข้องที่ 4		
16	feature_product	INT	11				0	สินค้าที่เกี่ยวข้องที่ 5		
17	Description	TEXT					No default	รายละเอียดสินค้า (ไม่เกิน 16k)		utf8_bin_252_v2

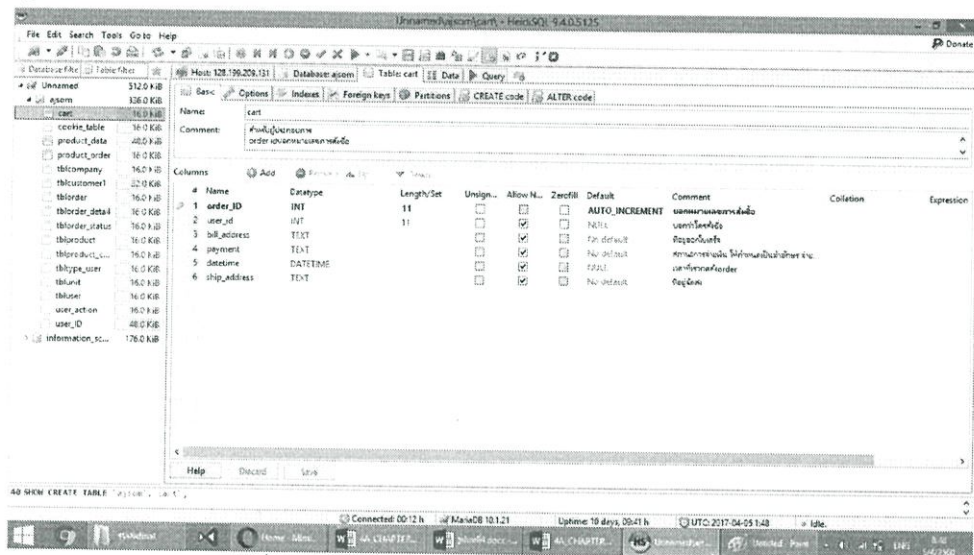
รูปที่ 4.2 การออกแบบฐานข้อมูล product_data

โดยหลักการทำงานได้กล่าวไปแล้วในบทที่ 3 ศึกษาได้จาก ตารางที่ 2 การออกแบบฐานข้อมูล product_order

#	Name	Datatype	Length/Set	Unsigned	Allow N.	Zerofill	Default	Comment	Collation	Expression
1	product_order_ID	INT	11				AUTO_INCREMENT	หมายเลขของ product id ที่สัมพันธ์มา...		
2	order_ID	INT	11				NULL	หมายเลขคำสั่งซื้อที่สัมพันธ์มา...		
3	Product_ID	INT	11				NULL	หมายเลขสินค้าที่สัมพันธ์มา...		
4	Currency	INT	11				No default	สกุลเงินที่ขายสินค้า		
5	payment	TEXT					No default	ข้อมูลการชำระเงิน		
6	date_order	DATETIME					SPRUL	วันที่สั่งซื้อ		
7	total_price	INT	11				NULL	รวมมูลค่ารวม		
8	total_weight	INT	11				NULL	รวมน้ำหนัก		
9	shipby	VARCHAR	50				NULL	ข้อมูลการขนส่ง		
10	comment	VARCHAR	50				NULL	ข้อมูลหมายเหตุ		

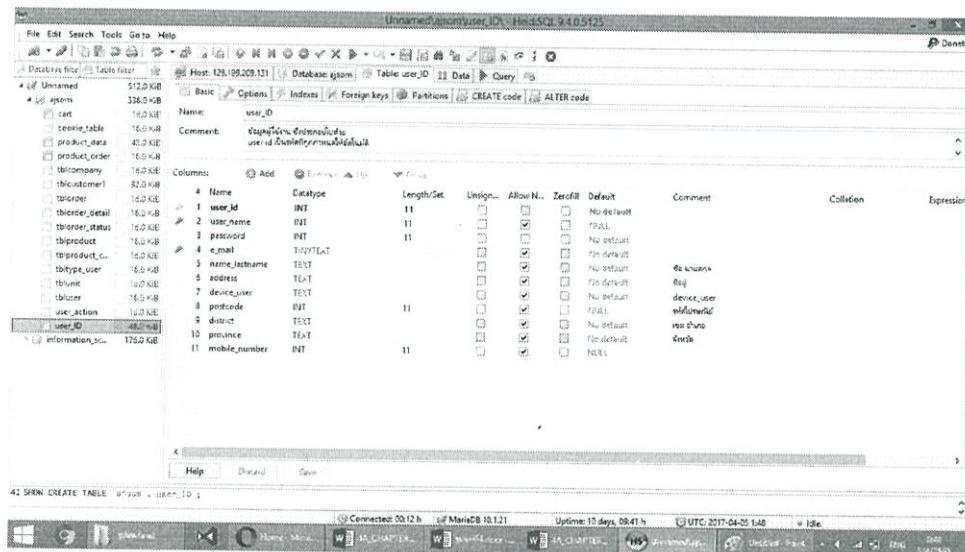
รูปที่ 4.3 การออกแบบตารางฐานข้อมูล product_order

โดยหลักการทำงานได้กล่าวไปแล้วในบทที่ 3 ศึกษาได้จาก ตารางที่ 3 การออกแบบฐานข้อมูล cart



รูปที่ 4.4 การออกแบบตารางฐานข้อมูล cart

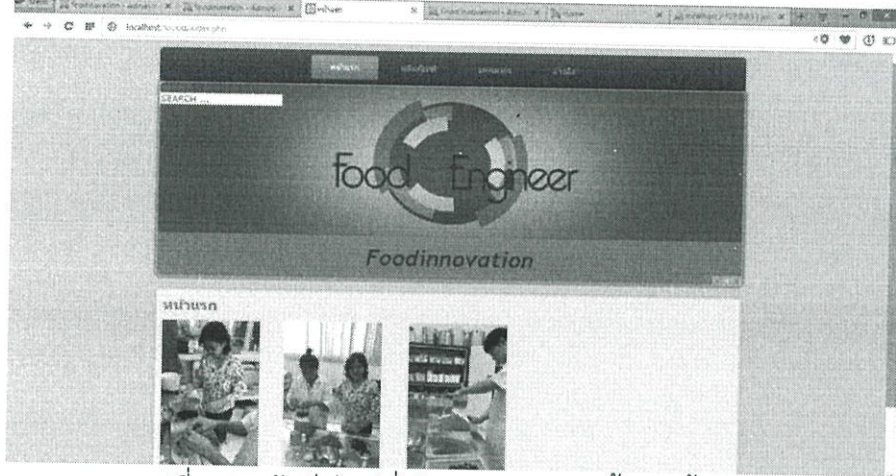
โดยหลักการทำงานได้กล่าวไปแล้วในบทที่ 3 ศึกษาได้จาก ตารางที่ 4 การออกแบบฐานข้อมูล user_ID



รูปที่ 4.5 การออกแบบตารางฐานข้อมูล user_ID

4.2 ผลการทดสอบ

4.2.1 ผลการทดลองออกแบบหน้าเว็บโดยใช้ภาษาเอชทีเอ็มแอล



รูปที่ 4.6 หน้าเว็บไซต์ที่ได้ทำการออกแบบขึ้นมาเบื้องต้น

4.2.2 ผลการทดลองจากการสร้างหน้าแรก

ผลจากการทดลองสร้างหน้าแรก ซึ่งผ่านกระบวนการดึงข้อมูลจากฐานข้อมูลมาใช้งานเรียบร้อยแล้ว ดังรูปที่ 4.7



รูปที่ 4.7 ผลการทดลองแสดงผลหน้าแรกของเว็บไซต์

4.2.2.1 ผลการทดลองจากฐานข้อมูล

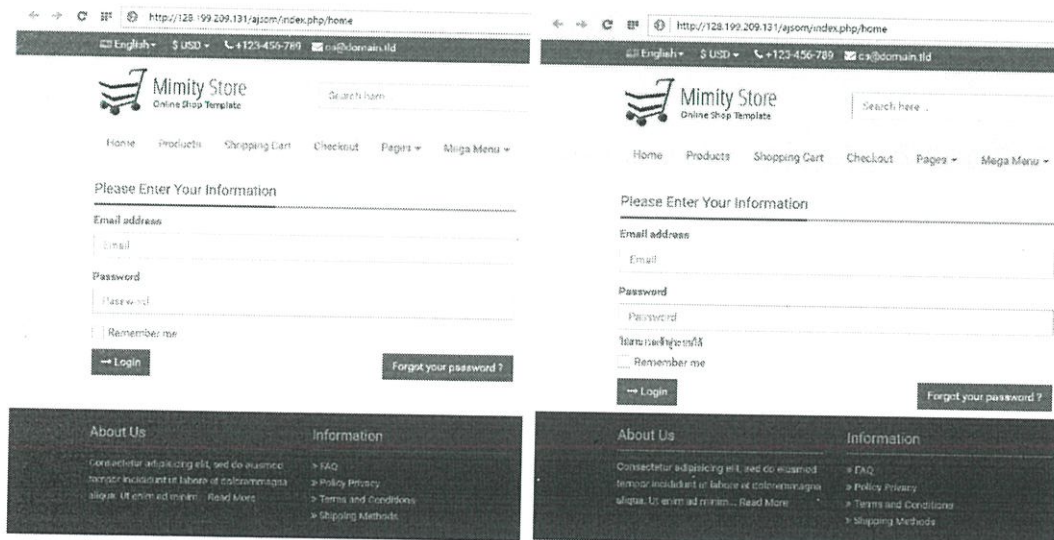
ในการสร้างเว็บไซต์นั้นเราจะทำการเขียน โดยใช้โอเดีย MVC ซึ่งจะใช้ภาษา HTML ในการออกแบบ ภาษาพีเอชพีในการควบคุม และภาษาเอสคิวแอล ในการเรียกใช้งานฐานข้อมูลดังแสดงตามรูปที่ 4.8

Product_ID	product_name	price	price_sale	review_star	product_type	sale_volume	stock	cost	weight	reserve	relate_product
1	โคมไฟโคมะรุมา	100	10	0	0	0	41	0	0	0	0
2	ตุ๊กตาใจเย็น	50	5	0	0	0	100	0	0	0	0
3	นมถั่วเหลือง	55	5.5	0	0	0	34	0	0	0	0
4	สลัดแอปเปิ้ล	60	6	0	0	0	100	0	0	0	0
5	เครื่องดื่มสมุนไพร	40	5	0	0	0	100	0	0	0	0
6	น้ำผลไม้สด	50	20	0	0	0	100	0	0	0	0
7	ช็อคโกแลต	30	3	0	0	0	100	0	0	0	0
8	เครื่องสำอางค์	20	2	0	0	0	100	0	0	0	0
9	ผลไม้ในบ่อผสมผสาน: กล้วยหอมสุก กล้วยน้ำว้าสุก	29	2.9	0	0	0	100	0	0	0	0
10	ชาสมุนไพรเข้มข้น	90	9	0	0	0	100	0	0	0	0
11	ชาจีนพรีเมียม	79	10	0	0	0	100	0	0	0	0
12	ช็อคโกแลตรสผลไม้	129	19	0	0	0	100	0	0	0	0
13	กาแฟพรีเมียม	93	10	0	0	0	100	0	0	0	0
15	น้ำดื่ม	35	5	0	0	0	100	0	0	0	0
16	อาหารสัตว์	59	19	0	0	0	100	0	0	0	0

รูปที่ 4.8 ผลการทดลองจากฐานข้อมูล

4.2.3 ผลการทดลองสร้างหน้าล็อกอินสำหรับหน้าแรก

การทำงานของหน้าล็อกอิน แสดงดังรูปที่ เริ่มต้นจะแสดงช่องกรอก Username และ Password ถ้าผู้ใช้ใส่ข้อมูล Username และ Password ถูกต้อง จะทำการเข้าสู่ระบบ แต่ถ้าผู้ใช้ใส่ข้อมูล Username และ Password ไม่ถูกต้อง จะไม่สามารถเข้าสู่ระบบได้และแสดงเป็นหน้าล็อกอินเหมือนเดิม



รูปที่ 4.9 หน้าล็อกอินเข้าสู่ระบบ

4.2.4 ผลการทดลองสร้างหน้ารายละเอียดสินค้า

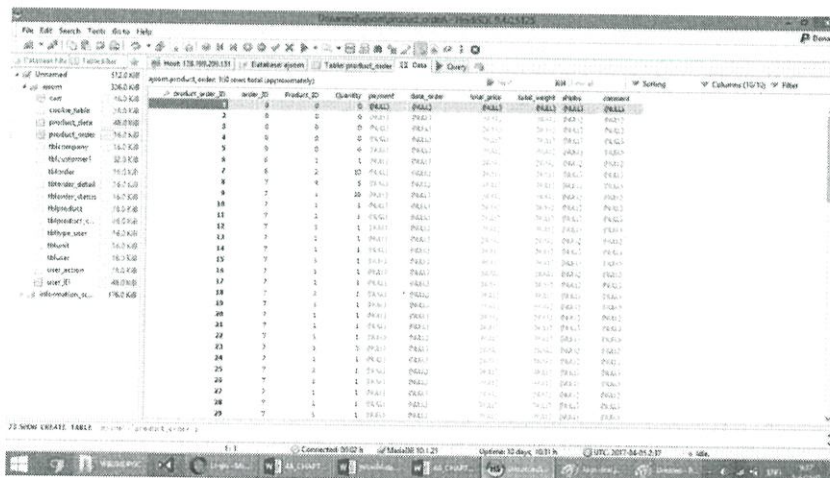
จากที่ได้ทำการทดลองนำข้อมูลมาแสดงโดยใช้ภาษา PHP ร่วมกับ SQL ในการเปิดฐานข้อมูล โดยให้มีการแสดงสินค้าที่หน้ารายละเอียดสินค้า หลักการทำงานได้กล่าวไปแล้วในบทที่ 3 ศึกษาได้จาก รูปที่ 3.13 ผังงานแสดงการทำงานเมื่อกดเข้ามาที่สินค้า และได้จัดเก็บผลการทดสอบตามรูปที่ 4.10



รูปที่ 4.10 แสดงผลหน้ารายละเอียดสินค้าเมื่อกดเข้ามาที่สินค้า

4.2.5 ข้อมูลจากฐานข้อมูลเมื่อมีการกดสินค้าเข้าตะกร้า

จากการทดสอบการจัดเก็บข้อมูลลงในฐานข้อมูล และสามารถนำข้อมูลมาแสดงผลโดยใช้ภาษา PHP ร่วมกับ SQL ในการเปิดฐานข้อมูล โดยให้มีการแสดงสินค้าที่หน้าตะกร้าสินค้าตามรูปที่ 4.11



รูปที่ 4.11 แสดงการทำงานเมื่อมีการกดสั่งซื้อเพิ่มคำสั่งซื้อลงในดาต้าเบส

4.2.5.1 ข้อมูลจากฐานข้อมูล

เมื่อมีการสั่งซื้อระบบจะตรวจสอบ ประวัติการสั่งซื้อเข้ามาหรือไม่ ถ้าไม่ใช่ ให้ทำการสร้างตะกร้าสินค้าขึ้นมาใหม่ แล้วรับข้อมูลเข้าที่ตะกร้าใหม่ แต่ถ้าเคยสั่งซื้อแล้ว จะทำการตรวจสอบอีกรอบว่ามีการชำระเงินรายการสินค้านั้นๆไปหรือยัง ถ้าใช่สร้างตะกร้าใหม่ แต่ถ้ายังไม่ชำระ ให้รับข้อมูลเข้าที่ตะกร้าเดิมตามรูปที่ 4.12

order_ID	user_id	bill_address	payment	datetime	ship_address
1	(NULL)	(NULL)	paid	2017-03-26 08:35:56	(NULL)
2	1	(NULL)	paid	2017-03-26 08:41:27	(NULL)
3	1	(NULL)	paid	2017-03-26 08:41:31	(NULL)
4	1	(NULL)	paid	2017-03-26 11:04:51	(NULL)
5	1	(NULL)	paid	2017-03-26 11:05:21	(NULL)
6	1	(NULL)	paid	2017-03-26 11:45:33	(NULL)
7	1	(NULL)	paid	2017-03-26 12:35:27	(NULL)
8	1	(NULL)	not_pay	2017-04-02 06:27:23	(NULL)

รูปที่ 4.12 ฐานข้อมูลแสดงเมื่อมีการกดสั่งซื้อมาที่หน้าตะกร้าสินค้า

4.2.5.2 ผลการทดลองสร้างหน้าตะกร้าสินค้าโดย หลักการทำงานได้

กล่าวไปแล้วในบทที่ 3 ศึกษาได้จาก รูปที่ 3.14 ผังงานแสดงการทำงานเมื่อมีการกดสั่งซื้อเพิ่มคำสั่งซื้อลงในดาต้าเบส

Product	Description	Quantity	Unit price	SubTotal	Action
	ข้าวเหนียวสุกแล้ว	5	90	450	
	นมสดรสจืด	1	49.5	49.5	
	ขนมปังกรอบเนยถั่ว	2	81	162	
				Total	661.5

รูปที่ 4.13 แสดงเมื่อมีการกดสั่งซื้อมาที่หน้าตะกร้าสินค้า

4.2.6 การแสดงผลหน้าเช็คเอาท์

The screenshot shows a web browser window with the URL `file:///C:/Users/User/Desktop/WB094DPGC/checkout2.html`. The page title is "Checkout" and it features a progress bar with five steps: 1. Billing, 2. Shipping, 3. Payment, 4. Review, and 5. Finish. The form contains the following fields:

- First Name (*)
- Last Name
- Email Address (*)
- Phone Number (*)
- Address (*)
- Country (*) (United Kingdom)
- Region / State (*) (Please Select)
- City (*)
- Post Code (*)

A "Next" button is located at the bottom left of the form.

รูปที่ 4.14 หน้าแสดงการกรอกข้อมูลที่อยู่การออกใบเสร็จ

The screenshot shows a web browser window with the URL `http://128.199.209.131/checkout2.html`. The page title is "Checkout" and it features a progress bar with five steps: 1. Billing, 2. Shipping, 3. Payment, 4. Review, and 5. Finish. The form contains the following fields:

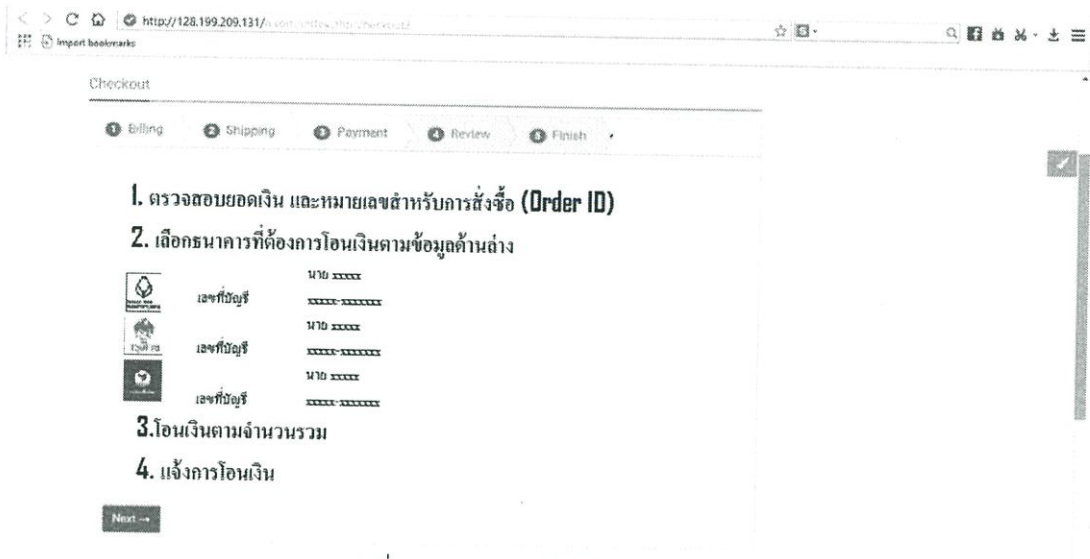
- Address (*)
- Country (*) (United Kingdom)
- Region / State (*) (Please Select)
- City (*)
- Post Code (*)

Below the address fields, there is a "Delivery Option" section with three radio button options:

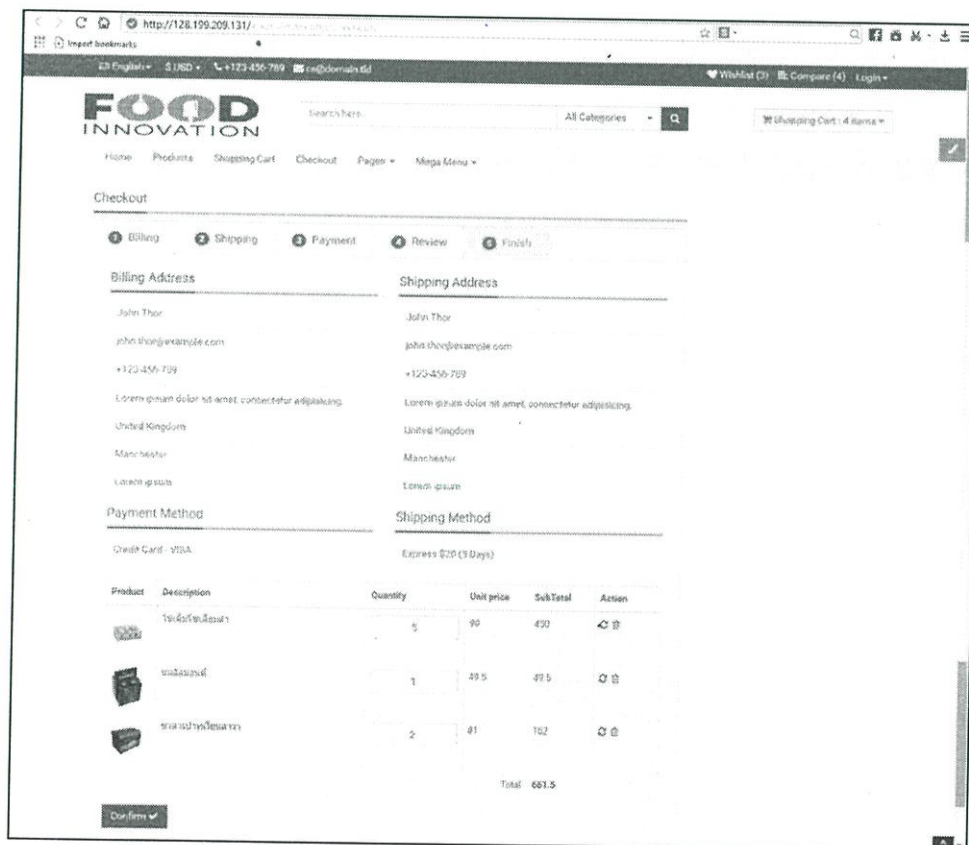
- Regular \$10 (14 Days)
- Priority \$15 (7 Days)
- Express \$20 (3 Days)

A "Next" button is located at the bottom left of the form.

รูปที่ 4.15 หน้าแสดงการกรอกข้อมูลที่อยู่จัดส่ง



รูปที่ 4.16 แสดงหน้าวิธีการชำระเงิน

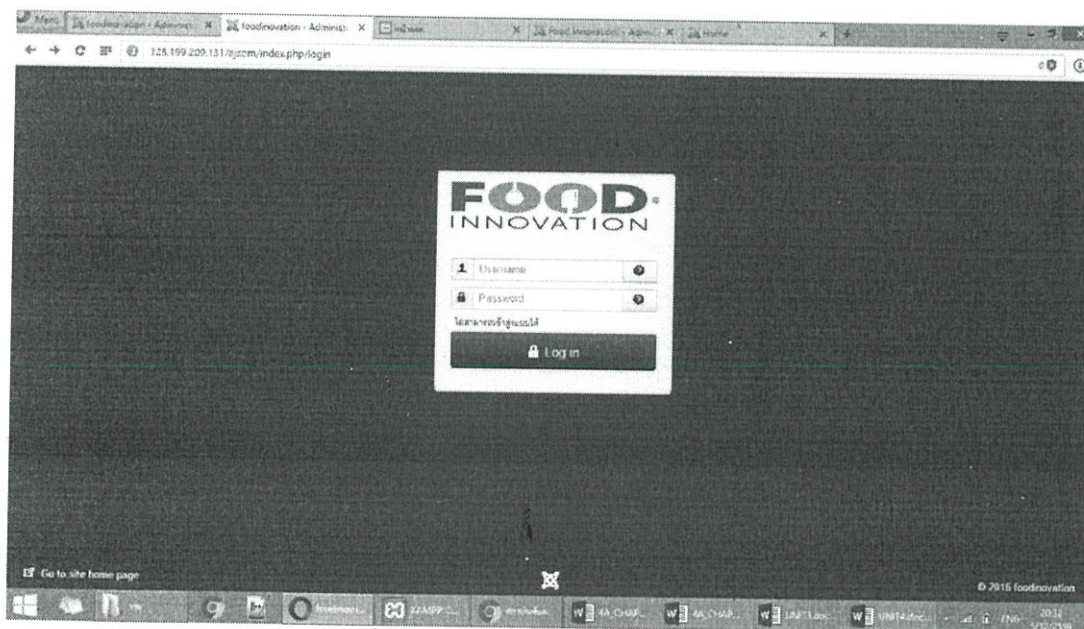


รูปที่ 4.17 แสดงหน้าตรวจสอบรายละเอียดต่างๆ

4.2.7 ผลการทดสอบหน้าล็อกอินเข้าสู่หน้าแอดมิน



รูปที่ 4.18 หน้าล็อกอินเข้าสู่หน้าแอดมิน



รูปที่ 4.19 แสดงผลเมื่อล็อกอินผิดพลาด



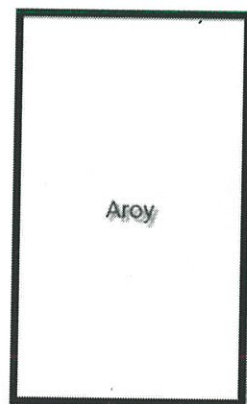
รูปที่ 4.20 แสดงผลเมื่อล็อกอินเข้าสู่ระบบสำเร็จ

4.3 ผลการดำเนินงานของแอปพลิเคชัน

จากที่ได้ศึกษาการทำแอปพลิเคชันเราจึงได้ทำเขียน และทำการออกแบบหน้าตาของแอปพลิเคชันเพื่อให้มาใช้งานจริงในการทำธุรกิจ ได้ดังนี้

4.3.1 หน้า activity Splash Screen

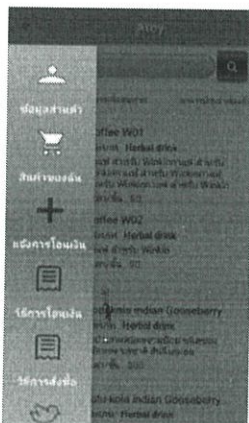
หน้า activity Splash Screen เป็นหน้าเริ่มแรก ก่อนเข้าแอปพลิเคชัน โดยหลักการทำงานได้กล่าวไปแล้วในบทที่ 3 ศึกษาได้จากผังการทำงานของแอปพลิเคชันหน้า Activity Splash Screen แสดงดังรูปที่ 3.19 และหน้าแอปพลิเคชันหน้า แสดงดังรูปที่ 4.21



รูปที่ 4.21 แอปพลิเคชันหน้า Splash Screen

4.3.2 หน้า activity_main.

แสดงข้อมูลของ activity_main. โดยจะแสดงถึงสินค้าในธุรกิจ และไอคอนของระบบของการจัดการของธุรกิจที่ได้สร้างขึ้น โดยหลักการทำงานได้กล่าวไปแล้วในบทที่ 3 ศึกษาได้จากผังการทำงานของแอปพลิเคชันหน้า activity_main แสดงดังรูปที่ 3.22 และหน้าแอปพลิเคชัน แสดงดังรูปที่ 4.22



รูปที่ 4.22 แอปพลิเคชันหน้า activity_main.

4.3.3 การทำงานของแอปพลิเคชันหน้าสมัครสมาชิกและข้อมูลสมาชิก

หน้าสมาชิกของแอปพลิเคชัน จะมีการให้ผู้รับบริการกรอกข้อมูลเพื่อจัดเก็บข้อมูลเพื่อเวลาผู้รับบริการนั้นใช้บริการของแอปพลิเคชัน เราจะได้บริการผู้รับบริการได้อย่างเต็มที่ โดยหลักการทำงานได้กล่าวไปแล้วในบทที่ 3 ศึกษาได้จากผังการทำงานของแอปพลิเคชันหน้า สมัครสมาชิกและข้อมูลสมาชิกแสดงดังรูปที่ 3.23 และหน้าแอปพลิเคชัน แสดงดังรูปที่ 4.23

 A screenshot of a registration form titled 'ข้อมูลส่วนตัว' (Personal Information). The form includes a sub-header 'ข้อมูลส่วนตัว (ใช้ในการจัดส่งสินค้า) กรุณากรอกให้ครบ' (Personal information (used for product delivery) please fill in completely). Below this are several input fields: 'ชื่อ - นามสกุล' (Name - Surname) with 'p', 'Email' with 'p', 'ที่อยู่ในการจัดส่ง' (Delivery address) with 'No Address00', and 'เบอร์โทรศัพท์' (Phone number) with 'No Phone'. At the bottom is a 'บันทึก' (Save) button.

รูปที่ 4.23 แอปพลิเคชันหน้าสมัครสมาชิกและข้อมูลสมาชิก

4.2.4 หน้าสินค้าของฉัน

เป็นหน้าคำนวณราคาสินค้า สรุปการสั่งซื้อสินค้า และสามารถยืนยันการสั่งซื้อได้ในหน้านี้เต็มที่ โดยหลักการทำงานได้กล่าวไปแล้วในบทที่ 3 ศึกษาได้จากผังการทำงานของแอปพลิเคชันหน้าสินค้าของฉัน แสดงดังรูปที่ 3.25 และหน้าแอปพลิเคชัน แสดงดังรูปที่ 4.24

สถานะการสั่งซื้อ	ยืนยันการสั่งซื้อ
Order Id :	2
ราคาสินค้า (บาท)	420
น้ำหนัก (กรัม)	1200
ราคา + ค่าจัดส่ง (บาท)	502.0

สั่งซื้อ

** กรุณาทำการส่งเป็นแบบ EMS โดยจะจัดส่งแบบ EMS ให้หลังจากจัดส่งสินค้าแล้ว **

Add data to detail Complete

รูปที่ 4.24 แอปพลิเคชันหน้าสินค้าของฉัน

4.2.5 หน้าแจ้งการโอนเงิน

หน้าแจ้งการโอน ผู้รับบริการจะต้องโอนเงินเข้ามาเมื่อเลือกซื้อสินค้าเรียบร้อยแล้ว เพื่อสตวกต่อผู้ประกอบการในการตรวจสอบ โดยหลักการทำงานได้กล่าวไปแล้วในบทที่ 3 ศึกษาได้จากผังการทำงานของแอปพลิเคชันหน้าแจ้งการโอนเงินแสดงดังรูปที่ 3.27 และหน้าแอปพลิเคชัน แสดงดังรูปที่ 4.25

แจ้งหมายเลข ID

จำนวนเงินที่โอน

วันที่และเวลาที่โอนเงิน

เพิ่มเติม (หากมีข้อเสนอนะ)

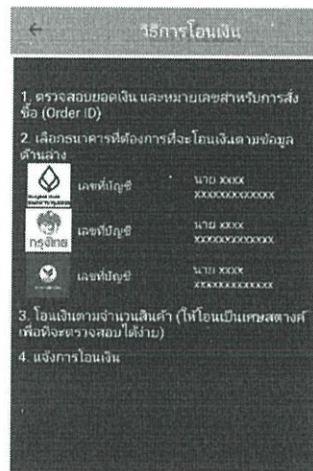
แจ้งการโอนเงิน

หมายเหตุ หากมีปัญหาก็มี ข้อความกลับ

รูปที่ 4.25 แอปพลิเคชันหน้าแจ้งการโอนเงิน

4.2.6 หน้าวิธีการโอนเงิน

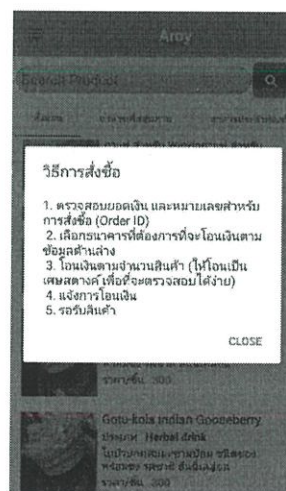
แสดงวิธีการโอนเงินของแอปพลิเคชัน โดยหลักการทำงานได้กล่าวไปแล้วใน บทที่ 3 ศึกษาได้จากผังการทำงานของแอปพลิเคชันหน้าวิธีการโอนเงิน แสดงดังรูปที่ 3.29 และ หน้าแอปพลิเคชัน แสดงดังรูปที่ 4.26



รูปที่ 4.26 แอปพลิเคชันหน้าวิธีการโอนเงิน

4.2.7 หน้าวิธีการสั่งซื้อ

แสดงวิธีการสั่งซื้อของแอปพลิเคชัน ชั้น โดยหลักการทำงานได้กล่าวไปแล้วในบทที่ 3 ศึกษาได้จากผังการทำงานของแอปพลิเคชันหน้าวิธีการสั่งซื้อ แสดงดังรูปที่ 3.31 และหน้าแอปพลิเคชัน แสดงดังรูปที่ 4.27



รูปที่ 4.27 แอปพลิเคชันหน้าวิธีการสั่งซื้อ

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

โครงการชิ้นนี้ได้ประสบความสำเร็จในการจัดทำเว็บไซต์และจัดทำแอปพลิเคชัน โดยในส่วนการเขียนเว็บ เราได้อัพโหลดเว็บไซต์ไว้ที่เซิร์ฟเวอร์ และจัดเก็บข้อมูลสินค้าพร้อมรายละเอียดต่างๆลงในฐานข้อมูล ผู้ใช้งานจึงสามารถเยี่ยมชมหรือเข้าไปสั่งซื้อสินค้าจากทางเว็บไซต์เราได้ ส่วนในด้านแอปพลิเคชัน เมื่อผู้ใช้งานเข้าถึงแอปพลิเคชัน ผู้ใช้งานสามารถดูรายละเอียดสินค้าและสั่งซื้อสินค้าผ่านทางแอปพลิเคชันนี้ได้เช่นกัน ซึ่งในส่วนแอปพลิเคชันนี้มีความเหมาะสมแก่ผู้ใช้งานที่พกโทรศัพท์มือถือทำให้สะดวกในการสั่งซื้อสินค้า

5.2 ข้อเสนอแนะ

ทั้งเว็บไซต์และแอปพลิเคชันที่ทางเราได้จัดขึ้นนั้นเพื่อเป็นการศึกษาและทดสอบให้ใช้งานได้จริง แต่ยังมีส่วนที่ยังไม่สมบูรณ์ครบถ้วน ดังนั้นทั้งเว็บไซต์และแอปพลิเคชันที่ได้จัดทำขึ้นมานั้นยังคงต้องมีการพัฒนาต่อไปเพื่อให้เหมาะสมแก่การใช้งานอย่างเต็มรูปแบบมากยิ่งขึ้น และในส่วนการติดตั้งแอปพลิเคชัน ยังไม่ได้ดำเนินการให้ดาวน์โหลดได้อย่างทั่วถึงซึ่งยังต้องมีการดำเนินการต่อไป

บรรณานุกรม

- [1] Akexorcist. “[Android Basic] รู้จักการใช้งาน Android Studio แบบพื้นฐาน”
<http://www.akexorcist.com/2015/01/how-to-use-android-studio-part-2.html>
- [2] agri.cmu. “การใช้งาน Joomla เบื้องต้น”
<http://www.agri.cmu.ac.th/upload/download/49080109.pdf>
- [3] irobust. “สอนเขียน PHP เบื้องต้น”
<http://www.irobust.co.th/PHP/Tutorial/tutorial/>

ภาคผนวก

1. CodeIgniter – Overview

CodeIgniter is an application development framework, which can be used to develop websites, using PHP. It is an Open Source framework. It has a very rich set of functionality, which will increase the speed of website development work.

If you know PHP well, then CodeIgniter will make your task easier. It has a very rich set of libraries and helpers. By using CodeIgniter, you will save a lot of time, if you are developing a website from scratch. Not only that, a website built in CodeIgniter is secure too, as it has the ability to prevent various attacks that take place through websites.

CodeIgniter Features

Some of the important features of CodeIgniter are listed below:

- Model-View-Controller Based System
- Extremely Light Weight
- Full Featured database classes with support for several platforms.
- Query Builder Database Support
- Form and Data Validation
- Security and XSS Filtering
- Session Management
- Email Sending Class. Supports Attachments, HTML/Text email, multiple protocols (sendmail, SMTP, and Mail) and more.
- Image Manipulation Library (cropping, resizing, rotating, etc.). Supports GD, ImageMagick, and NetPBM
- File Uploading Class
- FTP Class
- Localization
- Pagination
- Data Encryption
- Benchmarking
- Full Page Caching
- Error Logging
- Application Profiling
- Calendaring Class
- User Agent Class
- Zip Encoding Class

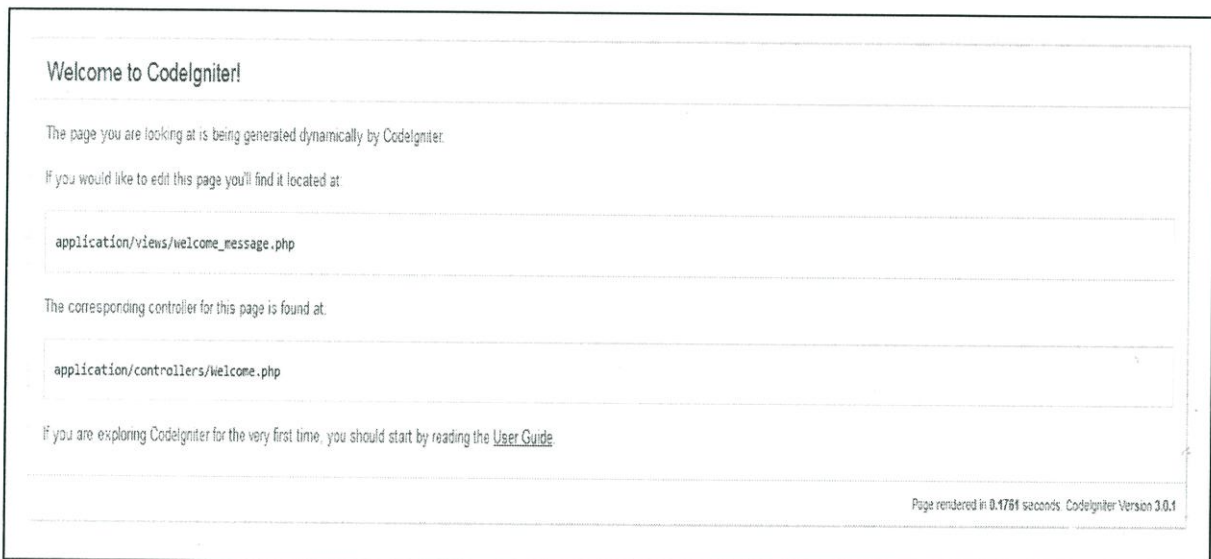
- Template Engine Class
- Trackback Class
- XML-RPC Library
- Unit Testing Class
- Search-engine Friendly URLs
- Flexible URI Routing
- Support for Hooks and Class Extensions
- Large library of "helper" functions

2. Installing CodeIgniter

It is very easy to install CodeIgniter. Just follow the steps given below:

- **Step-1:** Download the CodeIgniter from the link <http://www.codeigniter.com/download>
- **Step-2:** Unzip the folder.
- **Step-3:** Upload all files and folders to your server.
- **Step-4:** After uploading all the files to your server, visit the URL of your server, e.g., www.domain-name.com.

On visiting the URL, you will see the following screen:



3. Application Architecture

The architecture of CodeIgniter application is shown below.

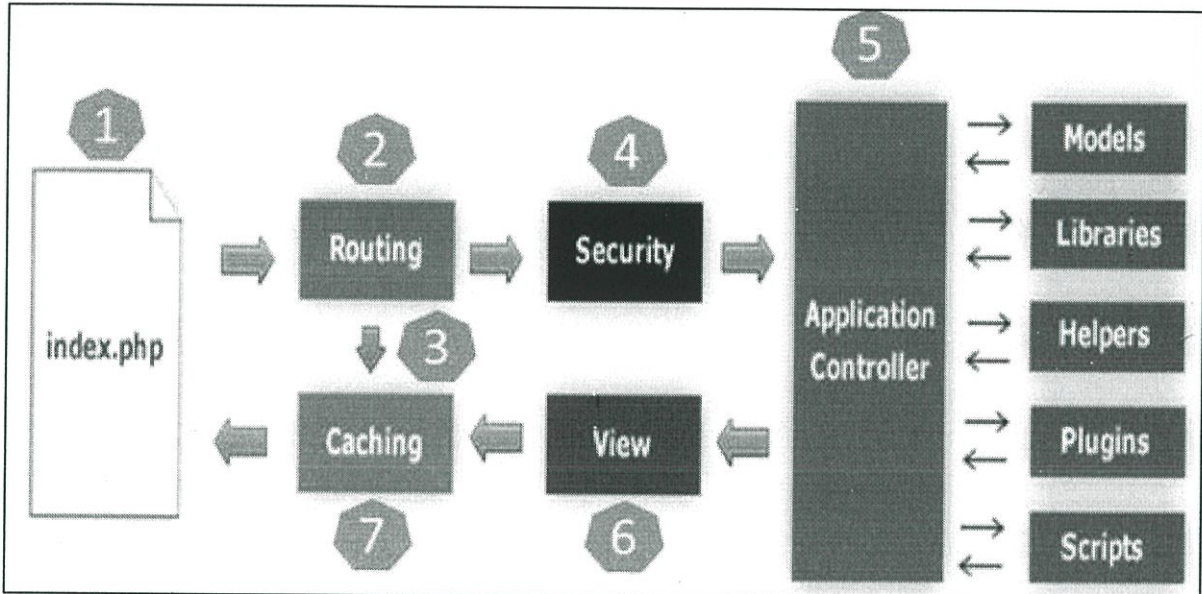


Figure: CodeIgniter Application Flowchart

- As shown in the figure, whenever a request comes to CodeIgniter, it will first go to **index.php** page.
- In the second step, **Routing** will decide whether to pass this request to step-3 for caching or to pass this request to step-4 for security check.
- If the requested page is already in **Caching**, then **Routing** will pass the request to step-3 and the response will go back to the user.
- If the requested page does not exist in **Caching**, then **Routing** will pass the requested page to step-4 for **Security** checks.
- Before passing the request to **Application Controller**, the **Security** of the submitted data is checked. After the **Security** check, the **Application Controller** loads necessary **Models, Libraries, Helpers, Plugins** and **Scripts** and pass it on to **View**.
- The **View** will render the page with available data and pass it on for **Caching**. As the requested page was not cached before so this time it will be cached in **Caching**, to process this page quickly for future requests.

Directory Structure

The image given below shows the directory structure of the CodeIgniter.

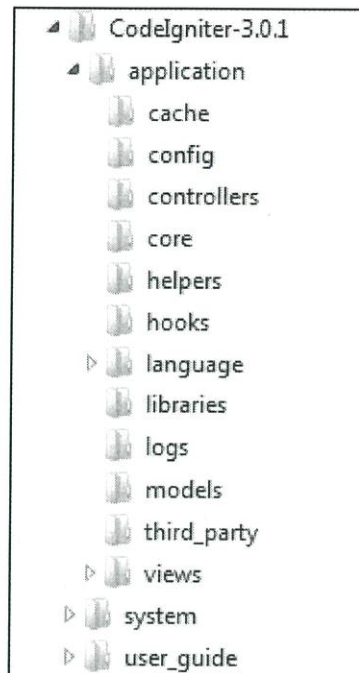


Figure: Directory Structure

CodeIgniter directory structure is divided into 3 folders:

- Application
- System
- User_guide

Application

As the name indicates the Application folder contains all the code of your application that you are building. This is the folder where you will develop your project. The Application folder contains several other folders, which are explained below:

- **Cache:** This folder contains all the cached pages of your application. These cached pages will increase the overall speed of accessing the pages.
- **Config:** This folder contains various files to configure the application. With the help of **config.php** file, user can configure the application. Using **database.php** file, user can configure the database of the application.
- **Controllers:** This folder holds the controllers of your application. It is the basic part of your application.
- **Core:** This folder will contain base class of your application.
- **Helpers:** In this folder, you can put helper class of your application.

- **Hooks:** The files in this folder provide a means to tap into and modify the inner workings of the framework without hacking the core files.
- **Language:** This folder contains language related files.
- **Libraries:** This folder contains files of the libraries developed for your application.
- **Logs:** This folder contains files related to the log of the system.
- **Models:** The database login will be placed in this folder.
- **Third_party:** In this folder, you can place any plugins, which will be used for your application.
- **Views:** Application's HTML files will be placed in this folder.

System

This folder contains CodeIgniter core codes, libraries, helpers and other files, which help make the coding easy. These libraries and helpers are loaded and used in web app development.

This folder contains all the CodeIgniter code of consequence, organized into various folders:

- **Core:** This folder contains CodeIgniter's core class. Do not modify anything here. All of your work will take place in the application folder. Even if your intent is to extend the CodeIgniter core, you have to do it with hooks, and hooks live in the application folder.
- **Database:** The database folder contains core database drivers and other database utilities.
- **Fonts:** The fonts folder contains font related information and utilities.
- **Helpers:** The helpers folder contains standard CodeIgniter helpers (such as date, cookie, and URL helpers).
- **Language:** The language folder contains language files. You can ignore it for now.
- **Libraries:** The libraries folder contains standard CodeIgniter libraries (to help you with e-mail, calendars, file uploads, and more). You can create your own libraries or extend (and even replace) standard ones, but those will be saved in the **application/libraries** directory to keep them separate from the standard CodeIgniter libraries saved in this particular folder.

User_guide

This is your user guide to CodeIgniter. It is basically, the offline version of user guide on CodeIgniter website. Using this, one can learn the functions of various libraries, helpers

and classes. It is recommended to go through this user guide before building your first web app in CodeIgniter.

Beside these three folders, there is one more important file named "**index.php**". In this file, we can set the application environment and error level and we can define system and application folder name. It is recommended, not to edit these settings if you do not have enough knowledge about what you are going to do.

4. CodeIgniter – MVC Framework

CodeIgniter is based on the **Model-View-Controller (MVC) development pattern**. MVC is a software approach that separates application logic from presentation. In practice, it permits your web pages to contain minimal scripting since the presentation is separate from the PHP scripting.

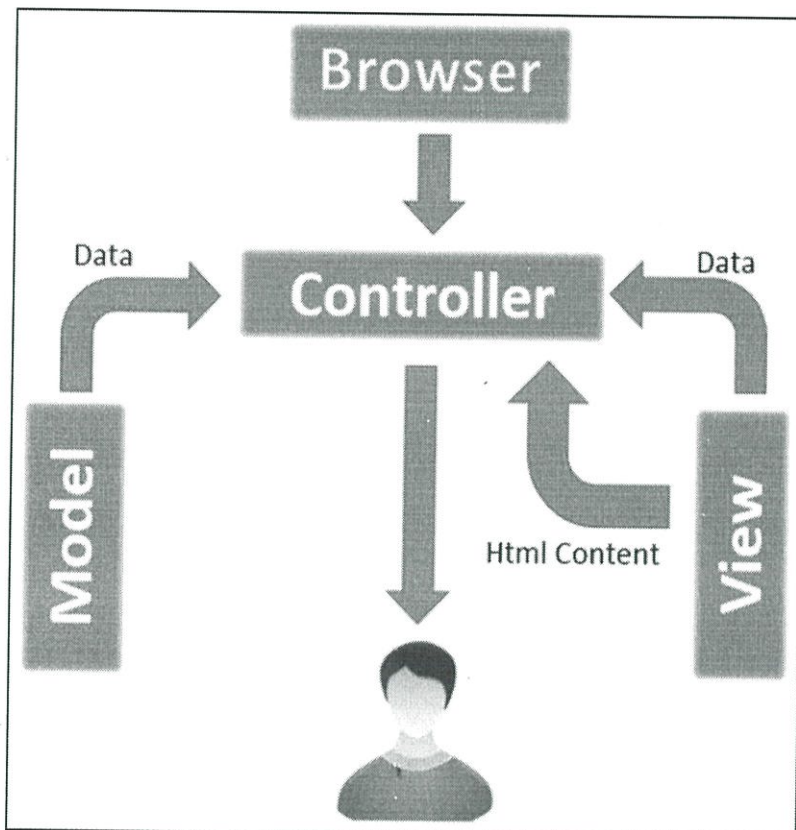


Figure: CodeIgniter – MVC Framework

- The **Model** represents your data structures. Typically, your model classes will contain functions that help you retrieve, insert and update information in your database.
- The **View** is information that is being presented to a user. A View will normally be a web page, but in CodeIgniter, a view can also be a page fragment like a header or footer. It can also be an RSS page, or any other type of "page".
- The **Controller** serves as an intermediary between the Model, the View, and any other resources needed to process the HTTP request and generate a web page.

5. CodeIgniter – Basic Concepts

Controllers

A controller is a simple class file. As the name suggests, it controls the whole application by URI.

Creating a Controller

First, go to **application/controllers** folder. You will find two files there, **index.html** and **Welcome.php**. These files come with the CodeIgniter.

Keep these files as they are. Create a new file under the same path named "**Test.php**". Write the following code in that file:

```
<?php
class Test extends CI_Controller {

    public function index()
    {
        echo "Hello World!";
    }
}
?>
```

The **Test** class extends an in-built class called **CI_Controller**. This class must be extended whenever you want to make your own Controller class.

Calling a Controller

The above controller can be called by URI as follows:

```
http://www.your-domain.com/index.php/test
```

Notice the word "**test**" in the above URI after index.php. This indicates the class name of controller. As we have given the name of the controller "**Test**", we are writing "**test**" after the index.php. The class name must start with **uppercase letter** but we need to write **lowercase letter** when we call that controller by URI. The general syntax for calling the controller is as follows:

```
http://www.your-domain.com/index.php/controller/method-name
```

Creating & Calling Constructor Method

Let us modify the above class and create another method named "hello".

```
<?php
class Test extends CI_Controller {

    public function index()
    {
        echo "This is default function.";
    }

    public function hello()
    {
        echo "This is hello function.";
    }
}
?>
```

We can execute the above controller in the following three ways:

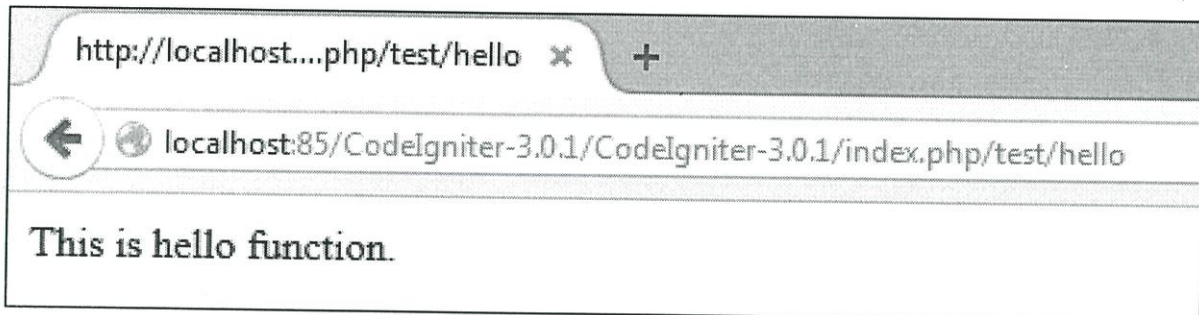
1. <http://www.your-domain.com/index.php/test>
2. <http://www.your-domain.com/index.php/test/index>
3. <http://www.your-domain.com/index.php/test/hello>

After visiting the first URI in the browser, we get the output as shown in the picture given below. As you can see, we got the output of the method "**index**", even though we did not pass the name of the method the URI. We have used only controller name in the URI. In such situations, the CodeIgniter calls the default method "**index**".



Visiting the second URI in the browser, we get the same output as shown in the above picture. Here, we have passed method's name after controller's name in the URI. As the name of the method is "**index**", we are getting the same output.

Visiting the third URI in the browser, we get the output as shown in picture given below. As you can see, we are getting the output of the method "hello" because we have passed "hello" as the method name, after the name of the controller "test" in the URI.



Points to Remember:

- The name of the controller class must start with an uppercase letter.
- The controller must be called with lowercase letter.
- Do not use the same name of the method as your parent class, as it will override parent class's functionality.

Views

This can be a simple or complex webpage, which can be called by the controller. The webpage may contain header, footer, sidebar etc. View cannot be called directly. Let us create a simple view. Create a new file under **application/views** with name "test.php" and copy the below given code in that file.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>CodeIgniter View Example</title>
</head>
<body>
  CodeIgniter View Example
</body>
</html>
```

Change the code of **application/controllers/test.php** file as shown in the below.

Loading the View

The view can be loaded by the following syntax:

```
$this->load->view('name');
```

Where name is the view file, which is being rendered. If you have planned to store the view file in some directory then you can use the following syntax:

```
$this->load->view('directory-name/name');
```

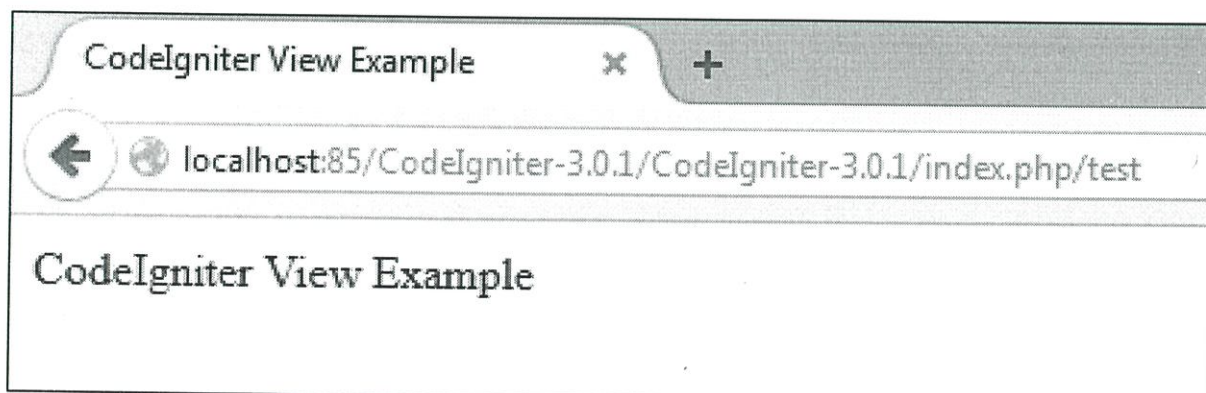
It is not necessary to specify the extension as php, unless something other than .php is used.

The index() method is calling the view method and passing the "test" as argument to view() method because we have stored the html coding in "**test.php**" file under **application/views/test.php**.

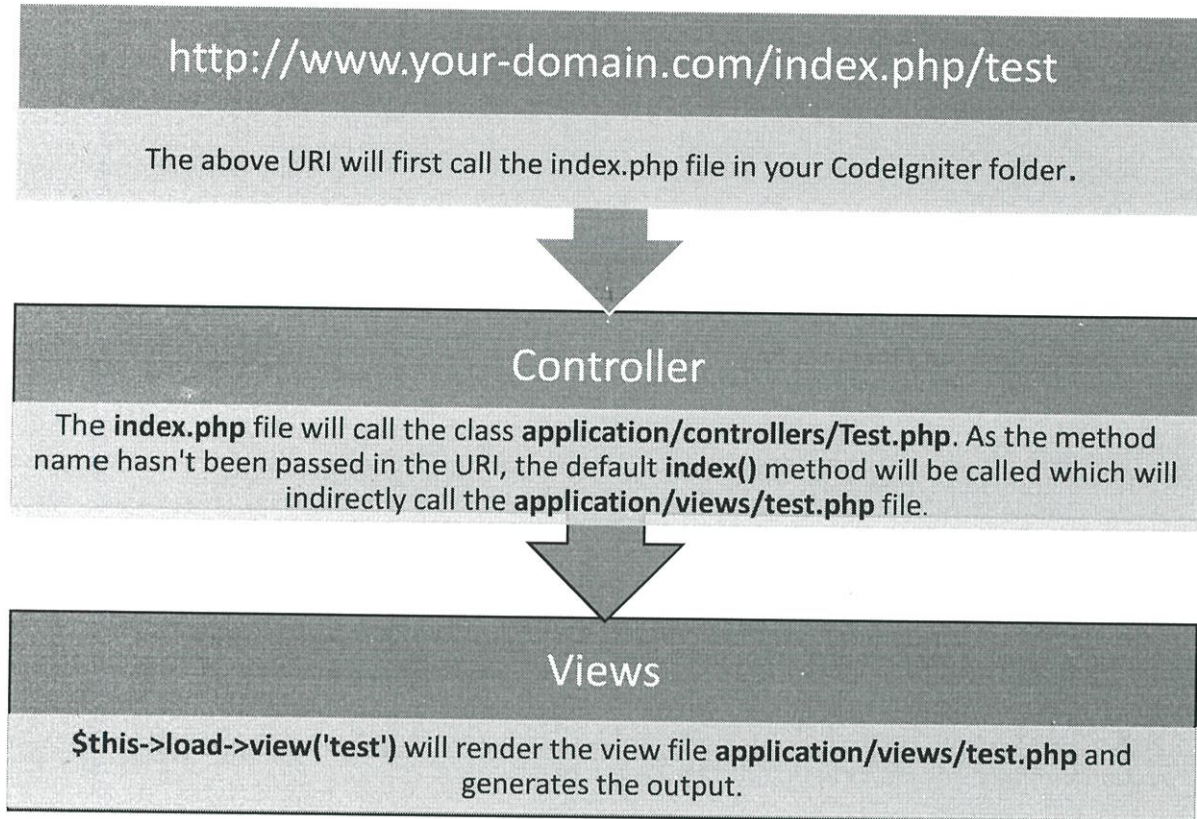
```
<?php
class Test extends CI_Controller {

    public function index()
    {
        $this->load->view('test');
    }
}
```

Here is the output of the above code:



The following flowchart illustrates of how everything works:



Models

Models classes are designed to work with information in the database. As an example, if you are using CodeIgniter to manage users in your application then you must have a model class, which contains functions to insert, delete, update and retrieve your users' data.

Creating Model Class

Model classes are stored in `application/models` directory. Following code shows how to create a model class in CodeIgniter.

```

<?php
Class Model_name extends CI_Model{

    Public function __construct()
    {
        parent::__construct();
    }
}
?>
  
```

Where Model_name is the name of the model class that you want to give. Each model class must inherit the CodeIgniter's CI_Model class. The first letter of the model class must be in capital letter. Following is the code for users' model class.

```
<?php
Class zzzextends CI_Model{

Public function __construct()
{
    parent::__construct();
}

}

?>
```

The above model class must be saved as User_model.php. The class name and file name must be same.

Loading Model

Model can be called in controller. Following code can be used to load any model.

```
$this->load->model('model_name');
```

Where model_name is the name of the model to be loaded. After loading the model you can simply call its method as shown below.

```
$this->model_name->method();
```

Auto-loading Models

There may be situations where you want some model class throughout your application. In such situations, it is better if we autoload it.

```
/*
|-----
| Auto-load Models
|-----
| Prototype:
|
| $autoload['model'] = array('first_model', 'second_model');
|
| You can also supply an alternative model name to be assigned
| in the controller:
|
| $autoload['model'] = array('first_model' => 'first');
*/
$autoload['model'] = array();
```

As shown in the above figure, pass the name of the model in the array that you want to autoload and it will be autoloaded, while system is in initialization state and is accessible throughout the application.

Helpers

As the name suggests, it will help you build your system. It is divided into small functions to serve different functionality. A number of helpers are available in CodeIgniter, which are listed in the table below. We can build our own helpers too.

Helpers are typically stored in your **system/helpers**, or **application/helpers directory**. Custom helpers are stored in **application/helpers** directory and systems' helpers are stored in **system/helpers** directory. CodeIgniter will look first in your **application/helpers directory**. If the directory does not exist or the specified helper is not located, CodeIgniter will instead, look in your global **system/helpers/ directory**. Each helper, whether it is custom or system helper, must be loaded before using it.

Helper Name	Description
Array Helper	The Array Helper file contains functions that assist in working with arrays.
CAPTCHA Helper	The CAPTCHA Helper file contains functions that assist in creating CAPTCHA images.
Cookie Helper	The Cookie Helper file contains functions that assist in working with cookies.
Date Helper	The Date Helper file contains functions that help you work with dates.
Directory Helper	The Directory Helper file contains functions that assist in working with directories.
Download Helper	The Download Helper lets you download data to your desktop.
Email Helper	The Email Helper provides some assistive functions for working with Email. For a more robust email solution, see CodeIgniter's Email Class.
File Helper	The File Helper file contains functions that assist in working with files.
Form Helper	The Form Helper file contains functions that assist in working with forms.
HTML Helper	The HTML Helper file contains functions that assist in working with HTML.
Inflector Helper	The Inflector Helper file contains functions that permits you to change words to plural, singular, camel case, etc.
Language Helper	The Language Helper file contains functions that assist in working with language files.
Number Helper	The Number Helper file contains functions that help you work with numeric data.
Path Helper	The Path Helper file contains functions that permits you to work with file paths on the server.
Security Helper	The Security Helper file contains security related functions.

Smiley Helper	The Smiley Helper file contains functions that let you manage smileys (emoticons).
String Helper	The String Helper file contains functions that assist in working with strings.
Text Helper	The Text Helper file contains functions that assist in working with text.
Typography Helper	The Typography Helper file contains functions that help you format text in semantically relevant ways.
URL Helper	The URL Helper file contains functions that assist in working with URLs.
XML Helper	The XML Helper file contains functions that assist in working with XML data.

Loading a Helper

A helper can be loaded as shown below:

```
$this->load->helper('name');
```

Where name is the name of the helper. For example, if you want to load the URL Helper, then it can be loaded as:

```
$this->load->helper('url');
```

Routing

CodeIgniter has user-friendly URI routing system, so that you can easily re-route URL. Typically, there is a one-to-one relationship between a URL string and its corresponding controller class/method. The segments in a URI normally follow this pattern:

```
your-domain.com/class/method/id/
```

- The **first segment** represents the controller class that should be invoked.
- The **second segment** represents the class function, or method, that should be called.
- The **third**, and any additional segments, represent the ID and any variables that will be passed to the controller.

In some situations, you may want to change this default routing mechanism. CodeIgniter provides facility through which you can set your own routing rules.

Customize Routing Rules

There is a particular file where you can handle all these. The file is located at application/config/routes.php. You will find an array called \$route in which you can customize your routing rules. The key in the \$route array will decide what to route and the value will decide where to route. There are three reserved routes in CodeIgniter.

Reserved Routes	
<code>\$route['default_controller']</code>	This route indicates which controller class should be loaded, if the URI contains no data, which will be the case when people load your root URL. You are encouraged to have a default route otherwise a 404 page will appear, by default. We can set home page of website here so it will be loaded by default.
<code>\$route['404_override']</code>	This route indicates which controller class should be loaded if the requested controller is not found. It will override the default 404 error page. It won't affect to the show_404() function, which will continue loading the default error_404.php file in application/views/errors/error_404.php .
<code>\$route['translate_uri_dashes']</code>	As evident by the Boolean value, this is not exactly a route. This option enables you to automatically replace dashes ('-') with underscores in the controller and method URI segments, thus saving you additional route entries if you need to do that. This is required because the dash is not a valid class or method-name character and will cause a fatal error, if you try to use it.

Routes can be customized by **wildcards** or by using **regular expressions** but keep in mind that these customized rules for routing must come after the reserved rules.

Wildcards

We can use two wildcard characters as explained below:

- **(:num)** – It will match a segment containing only numbers.
- **(:any)** – It will match a segment containing any character.

Example

```
$route['product/:num']='catalog/product_lookup';
```

In the above example, if the literal word "product" is found in the first segment of the URL, and a number is found in the second segment, the "catalog" class and the "product_lookup" method are used instead.

Regular Expressions

Like wildcards, we can also use regular expressions in **\$route array key** part. If any URI matches with regular expression, then it will be routed to the value part set into \$route array.

Example

```
$route['products/([a-z]+)/(\d+)']='$1/id_$2';
```

In the above example, a URI similar to products/shoes/123 would instead call the **"shoes"** controller class and the **"id_123"** method.