



โครงการทางการจัดการเทคโนโลยีสารสนเทศ

CuisineVision (Coffee Shop Management Software)

นายสาโรจน์	แซ่ลี้	5410513069
นายสุวนนท์	ชูจันทร์	5410513077

อาจารย์ที่ปรึกษา

ดร.อารีย์ ลีมวุฒิไกรจิรัฐ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา โครงการทางการจัดการเทคโนโลยีสารสนเทศ 1

(PROJECT IN INFORMATION TECHNOLOGY MANAGEMENT I)

มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

ภาคเรียนที่ 2 ปีการศึกษา 2557

กิตติกรรมประกาศ

โครงการฉบับนี้สำเร็จลุล่วงลงได้ เนื่องด้วยความกรุณาจาก ดร.อารีย์ ถีมวุฒิไกรจิรัฐ อาจารย์ที่ปรึกษางานสัมมนา ที่กรุณาสละเวลาในการช่วยเหลือและให้คำแนะนำข้อคิดเห็นต่างๆ รวมถึงช่วยตรวจสอบข้อบกพร่องในการทำงานโดยตลอดจนสำเร็จไปได้ด้วยดี ขอขอบพระคุณอาจารย์เป็นอย่างสูงไว้ ณ ที่นี้ นอกจากนี้ขอขอบคุณคณะผู้จัดทำที่ให้การสนับสนุน และความร่วมมือตลอดการทำงาน ทำให้งานมีความสำเร็จลุล่วง และท้ายนี้ต้องขอบคุณเจ้าของตำราทุกท่านที่ผู้ทำรายงานเล่มนี้ได้ใช้อ้างอิงผู้จัดทำรายงานจะนำความรู้ที่ได้รับจากการทำรายงานเล่มนี้ไปประยุกต์ใช้ในการทำรายงานอย่างถูกต้องในรายวิชาอื่นต่อไป

คณะผู้จัดทำ

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชาโครงการงานทางการจัดการเทคโนโลยีสารสนเทศ 1 (PROJECT IN INFORMATION TECHNOLOGY MANAGEMENT I) จัดทำขึ้นเพื่อเป็นเอกสารประกอบการนำเสนอโครงการ CuisineVision (Coffee Shop Management Software) โดยเนื้อหาประกอบด้วยขั้นตอนการทำงาน การออกแบบหน้าจอและฐานข้อมูลและรายงานความถี่หน้ารวมถึงปัญหาและอุปสรรค

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่ารายงานเล่มนี้จะเป็นประโยชน์ต่อผู้ที่สนใจ ได้ทราบถึงวัตถุประสงค์และความสำคัญของโครงการนี้ คณะผู้จัดทำขอขอบคุณอาจารย์อารีย์ ลีมวุฒิไกรจิรัฐ ผู้ให้ความรู้ และแนวทางการศึกษา

คณะผู้จัดทำ

เรื่อง	สารบัญ	หน้า
กิตติกรรมประกาศ		๗
คำนำ		๘
สารบัญ		๑๑
สารบัญรูป		๑๒
สารบัญตาราง		๑๓
บทที่ 1 บทนำ		1
หลักการและเหตุผล		1
ขอบเขตของโครงการ		2
กลุ่มผู้ใช้		2
วัตถุประสงค์ของโครงการ		2
ความสามารถ		3
ขั้นตอนการดำเนินงาน		4
แผนการดำเนินงาน		5
เครื่องมือสำหรับดำเนินการและค่าใช้จ่าย		6
ประโยชน์ที่คาดว่าจะได้รับ		7
บทที่ 2 ทบทวนวรรณกรรม		8
Microsoft Visual Studio		8
Microsoft Expression Blend		9
.NET Technology และ .NET Framework		10
WPF (Windows Presentation Foundation)		13
ภาษา C#		14
Microsoft SQL Server		15
ลำดับขั้นตอนการสร้างโปรแกรมบน .NET FRAMEWORK		20

เรื่อง	สารบัญ	หน้า
บทที่ 3 วิเคราะห์และออกแบบระบบ		22
Entity–Relationship Model		22
Data Dictionary		23
Unified Modeling Language(UML)		27
Use Case Scenario		28
Activity Diagram for Use Case		38
Interface Design		48
อ้างอิง		49

สารบัญรูป

เรื่อง	หน้า
รูปที่ 1 : แสดงโลโก้ Microsoft Visual Studio	8
รูปที่ 2 : แสดงโลโก้ Microsoft Expression Blend	9
รูปที่ 3 : แสดงโลโก้ .NET Framework	10
รูปที่ 4 : แสดง .NET Framework Stack	10
รูปที่ 5 : แสดงโลโก้ C#.net	14
รูปที่ 6 : แสดงภาพผู้ได้ชื่อว่าเป็นบิดาของภาษา C# ชื่อว่า Anders Hejlsberg	16
รูปที่ 7 : แสดงโลโก้ Microsoft SQL Server	18
รูปที่ 8 : แสดงขั้นตอนการสร้างโปรแกรมบน .NET Framework	21
รูปที่ 9 : แสดง Entity-Relationship Model ของระบบจัดการร้านกาแฟ	22
รูปที่ 10 : แสดงความสัมพันธ์ Use Case Diagram	27
รูปที่ 11 : แสดง Activity Diagram สำหรับ Use Case เข้าสู่ระบบ	38
รูปที่ 12 : แสดง Activity Diagram สำหรับ Use Case จัดการข้อมูลรายการอาหาร	39
รูปที่ 13 : แสดง Activity Diagram สำหรับ Use Case จัดการข้อมูลพนักงาน	40
รูปที่ 14 : แสดง Activity Diagram สำหรับ Use Case จัดการบัญชีผู้ใช้	41
รูปที่ 15 : แสดง Activity Diagram สำหรับ Use Case ใช้เครื่องคิดเลข	42
รูปที่ 16 : แสดง Activity Diagram สำหรับ Use Case สั่งอาหาร	43
รูปที่ 17 : แสดง Activity Diagram สำหรับ Use Case เรียกชำระค่าบริการ	44
รูปที่ 18 : แสดง Activity Diagram สำหรับ Use Case จัดการรายการอาหารที่ถูกคำสั่ง	45
รูปที่ 19 : แสดง Activity Diagram สำหรับ Use Case รับชำระค่าบริการ	46
รูปที่ 20 : แสดง Activity Diagram สำหรับ Use Case ออกจากระบบ	47
รูปที่ 21 : แสดงหน้า Login Form	48
รูปที่ 22 : แสดงหน้า Customer Form	48
รูปที่ 23 : แสดงหน้า Order Form	49
รูปที่ 24 : แสดงหน้าจอการจัดการสำหรับเจ้าของร้าน	49

สารบัญตาราง

เรื่อง	หน้า
ตารางที่ 1 : แสดงแผนการดำเนินการ	5
ตารางที่ 2 : แสดงเครื่องมือสำหรับดำเนินการและค่าใช้จ่าย	6
ตารางที่ 3 : แสดง Data Dictionary ของตาราง “Product”	23
ตารางที่ 4 : แสดง Data Dictionary ของตาราง “Employee”	23
ตารางที่ 5 : แสดง Data Dictionary ของตาราง “CustomerTable”	24
ตารางที่ 6 : แสดง Data Dictionary ของตาราง “Account”	24
ตารางที่ 7 : แสดง Data Dictionary ของตาราง “Receipt”	25
ตารางที่ 8 : แสดง Data Dictionary ของตาราง “ReceiptInfo”	25
ตารางที่ 9 : แสดง Data Dictionary ของตาราง “Order”	26
ตารางที่ 10 : แสดง Use Case Scenario ของงาน “เข้าสู่ระบบ”	28
ตารางที่ 11 : แสดง Use Case Scenario ของงาน “จัดการข้อมูลอาหาร”	29
ตารางที่ 12 : แสดง Use Case Scenario ของงาน “จัดการข้อมูลพนักงาน”	30
ตารางที่ 13 : แสดง Use Case Scenario ของงาน “จัดการบัญชีผู้ใช้”	31
ตารางที่ 14 : แสดง Use Case Scenario ของงาน “ใช้เครื่องคิดเลข”	32
ตารางที่ 15 : แสดง Use Case Scenario ของงาน “เรียกชำระค่าบริการ”	33
ตารางที่ 16 : แสดง Use Case Scenario ของงาน “สั่งอาหาร”	34
ตารางที่ 17 : แสดง Use Case Scenario ของงาน “จัดการรายการอาหารที่ลูกค้าสั่ง”	35
ตารางที่ 18 : แสดง Use Case Scenario ของงาน “รับชำระค่าบริการ”	36
ตารางที่ 19 : แสดง Use Case Scenario ของงาน “ออกจากระบบ”	37

บทที่ 1

บทนำ

ชื่อโครงการ	CuisineVision (Coffee Shop Management Software)
อาจารย์ที่ปรึกษาโครงการ	ดร.อารีย์ ลิ้มวุฒิไกรจิรัฐ
ผู้รับผิดชอบโครงการ	นายสาโรจน์ แซ่ลี 5410513069 นายสุวนนท์ ชูจันทร์ 5410513077
ระดับการศึกษา	ปริญญาตรีบริหารธุรกิจบัณฑิต
ภาควิชา	บริหารธุรกิจ สาขาวิชา การจัดการเทคโนโลยีสารสนเทศ คณะวิทยาการจัดการ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่
ภาคการศึกษาที่	1 และ 2
ปีการศึกษา	2557

หลักการและเหตุผล

ปัจจุบันร้านกาแฟเป็นร้านอาหารและเครื่องดื่มยอดนิยมในกลุ่มของนักศึกษา และวัยทำงาน ซึ่งมักจะเป็นสถานที่ที่มีการนัดพบ หรือจัดประชุมเล็กๆ กันอยู่เสมอ กลุ่มลูกค้าจึงต้องการบรรยากาศความเป็นส่วนตัวในการทำกิจกรรมต่างๆ

หากมองในมุมมองของผู้ประกอบการ เป้าหมายหลักที่จะทำให้ลูกค้าพึงพอใจของธุรกิจประเภทนี้คือการสร้างบรรยากาศให้เป็นที่พอใจแก่ลูกค้า ส่วนเครื่องดื่มเป็นเพียงส่วนหนึ่งที่จะช่วยในการเสริมสร้างบรรยากาศเท่านั้น คำถามต่อมาคือ ทำอย่างไรให้ลูกค้าได้รับความสะดวกสบายในการใช้บริการมากที่สุด? การมีพนักงานในร้านมากเกินไปอาจเป็นตัวเลือกที่ไม่ดีนักในการเสริมสร้างบรรยากาศความเป็นส่วนตัวของลูกค้า รวมถึงในด้านการลงทุนของผู้ประกอบการด้วย แต่จะทำอย่างไรให้ลูกค้าทุกคนที่มาใช้บริการได้รับการอย่างเท่าเทียมกันทุกคน?

จากปัญหาดังกล่าว ผู้รับผิดชอบโครงการจึงเกิดแนวความคิดในการทำโปรแกรมประยุกต์ชื่อว่า CuisineVision เพื่อรองรับปัญหาและช่วยให้การบริการมีประสิทธิภาพมากขึ้น โดยที่จะใช้โปรแกรมประยุกต์ดังกล่าวจะอำนวยความสะดวกแก่ลูกค้าและผู้ประกอบการในด้านต่างๆ แทนการใช้พนักงาน เช่น การแสดงเมนูสินค้า แสดงลำดับการสั่งสินค้า เรียกชำระค่าบริการ แสดงรายรับ-รายจ่ายของทางร้าน เป็นต้น

ขอบเขตของโครงการ

วิชาเรียนของนักศึกษาชั้นปีที่ 4 คณะวิทยาการจัดการ ภาควิชาบริหารธุรกิจ สาขาการจัดการเทคโนโลยีสารสนเทศ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

กลุ่มผู้ใช้

- ผู้ประกอบการร้านกาแฟ
- พนักงานร้านกาแฟ
- ลูกค้าร้านกาแฟ

วัตถุประสงค์ของโครงการ

- 1) ช่วยสนับสนุนการดำเนินงานของพนักงาน ทำให้สามารถบริการลูกค้าได้รวดเร็ว
- 2) ช่วยประหยัดค่าใช้จ่ายในระยะยาวโดยการใช้คอมพิวเตอร์ทำงานแทนพนักงาน
- 3) ลดความผิดพลาดของการให้บริการ
- 4) ลูกค้าทุกคนมีความเท่าเทียมกันในการได้รับการบริการ (มีระบบแสดงลำดับการให้บริการ)

ความสามารถ

1. สามารถแสดงรายการอาหารและกดสั่งซื้อได้ในหน้าจอสำหรับลูกค้า
2. สามารถแสดงลำดับการให้บริการในหน้าจอสำหรับลูกค้า
3. มีระบบคัดแยกผู้ใช้และจำกัดสิทธิต่างๆ โดยการใช้ชื่อบัญชีและรหัสในการเข้าใช้งาน
4. คำนวณราคาอาหารทั้งหมดขณะสั่งซื้อสินค้าและจะแสดงในหน้าจอสำหรับลูกค้า
5. ลูกค้าสามารถกดปุ่มเพื่อสั่งสินค้า และสามารถเลือกตัวเลือกเสริมเช่น หวานน้อย เป็นต้น
6. ลูกค้าสามารถใช้โปรแกรมเครื่องคิดเลขในหน้าจอสำหรับลูกค้าได้
7. ลูกค้าสามารถเรียกพนักงานเพื่อชำระค่าบริการด้วยการกดปุ่มเพื่อใช้ระบบเรียกชำระค่าบริการในหน้าจอสำหรับลูกค้า
8. ผู้ประกอบการสามารถเพิ่ม ปรับปรุง ลบ รายชื่อพนักงานได้
9. ผู้ประกอบการสามารถเพิ่ม ปรับปรุง ลบ รายการสินค้าได้
10. ผู้ประกอบการสามารถออกเอกสารเกี่ยวกับข้อมูลทางการเงินในรูปแบบไฟล์ Microsoft Excel (.xlsx) ได้
11. ผู้ประกอบการและพนักงานสามารถปรับปรุงประกาศต่างๆ ในหน้าจอสำหรับลูกค้าได้
12. สามารถแสดงรายการอาหารที่ลูกค้าสั่งในหน้าจอพนักงาน
13. สามารถแจ้งเตือนในหน้าจอสำหรับพนักงาน เมื่อมีลูกค้ากดปุ่มเรียกชำระค่าบริการหรือเรียกพนักงาน
14. พนักงานสามารถปรุงปรุงสถานะของสินค้าเช่น มีสินค้า หหมด แนะนำ เป็นต้น
15. มีปุ่มสำหรับกดเมื่อพนักงานทำอาหารในรายการที่ลูกค้าสั่งเสร็จแล้ว เมื่อกดปุ่ม ลำดับการให้บริการที่แสดงในหน้าจอลูกค้าก็จะแสดงเป็นลำดับถัดไป
16. หน้าจอสำหรับลูกค้าจะปรับเป็นค่าเริ่มต้นทั้งหมดหลังจากกดชำระเงิน

ขั้นตอนการดำเนินงาน

- 1) Planning
 - ศึกษาการใช้งานเครื่องมือที่เกี่ยวข้องกับโปรแกรมประยุกต์
- 2) Analysis
 - วิเคราะห์ความต้องการทางด้าน Software และ Hardware ที่ใช้ในการพัฒนาโปรแกรมประยุกต์
 - วิเคราะห์รูปแบบของข้อมูลที่ต้องจัดเก็บ
- 3) Design
 - ออกแบบรูปแบบ interface ของโปรแกรมประยุกต์
 - ออกแบบ interface การแสดงผลการทำงานต่างๆ
- 4) Coding
 - ทำการสร้าง interface ตามรูปแบบที่ทำการออกแบบไว้
 - ทำการสร้างโปรแกรมประยุกต์
- 5) Testing
 - ทดสอบการทำงานของโปรแกรมประยุกต์โดยผ่านผู้ใช้งานจริง
 - ทำการปรับปรุงหากเกิดข้อผิดพลาด
- 6) Implementation
 - นำโปรแกรมประยุกต์ไปใช้งานจริง
- 7) Documentation
 - จัดทำเอกสาร

แผนการดำเนินงาน

การทำงานในแต่ละเดือน	ส.ค.	ก. ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
	2557	2557	2557	2557	2557	2558	2558	2558	2558
Planning ศึกษาและรวบรวมหัวข้อที่เกี่ยวข้องกับแอปพลิเคชัน, อุปกรณ์ที่ใช้ในการออกแบบ									
Analysis วิเคราะห์รูปแบบของแอปพลิเคชันและฐานข้อมูลที่ใช้เก็บข้อมูล									
Design ออกแบบแอปพลิเคชันและกระบวนการทำงานต่าง ๆ									
Coding สร้างแอปพลิเคชัน									
Testing ตรวจสอบความถูกต้อง									
Implementation ทดลองการใช้งานจริง									
Document เอกสาร									

ตารางที่ 1 : แสดงแผนการดำเนินการ

ลำดับ	โปรแกรม/อุปกรณ์ที่ใช้ในการปฏิบัติงาน	ค่าใช้จ่าย(บาท)
1	<p>คอมพิวเตอร์ 2 ชุด</p> <p>ชุดที่ 1 Computer Notebook</p> <p>Intel Core i3,CPU 2.53GHz, RAM 4GB,</p> <p>VGA CARD : ATI Mobility Radeon HD 550v,</p> <p>Hard disk : 320GB 7200RPM</p> <p>OS : Windows 7 64-bit</p> <p>ชุดที่ 2 Computer PC</p> <p>Intel(R) Core(TM)2Duo CPU E7400 2.80GHz,</p> <p>RAM 4 GB, Hard disk : 320GB</p> <p>VGA CARD:NVIDIA GeForce 9500 GT 1 GB</p> <p>OS : Windows 7 64-bit</p>	-
2	Visual Studio 2013 with Expression Blend	-
3	SQL Server 2014 with Management Studio	-
4	Microsoft Excel 2010	-

ตารางที่ 2 : แสดงเครื่องมือสำหรับดำเนินการและค่าใช้จ่าย

ประโยชน์ที่คาดว่าจะได้รับ

1. ทำให้ลูกค้าที่ใช้บริการมีความพึงพอใจ ในด้านการใช้งานโปรแกรมประยุกต์ และการให้บริการ
2. ทำให้พนักงานสามารถให้บริการลูกค้าได้รวดเร็วยิ่งขึ้น
3. ลดความผิดพลาดในการให้บริการ
4. ลดต้นทุนในการจ้างพนักงาน

บทที่ 2

ทบทวนวรรณกรรม

Microsoft Visual Studio



รูปที่ 1 : แสดงโลโก้ Microsoft Visual Studio

Visual Studio คือ Integrated Development Environment พัฒนาขึ้นโดยไมโครซอฟท์ ซึ่งเป็นเครื่องมือที่ช่วยนักพัฒนาซอฟต์แวร์พัฒนาโปรแกรมคอมพิวเตอร์ เว็บไซต์ เว็บแอปพลิเคชัน และ เว็บเซอร์วิส และสามารถเขียนโปรแกรมด้วยความสะดวกสบายขึ้น สามารถแก้ไขข้อผิดพลาดในการเขียนโปรแกรมได้ง่าย รวดเร็ว โดยระบบที่รองรับการทำงานนั้นมีไมโครซอฟท์ วินโดวส์ ฟ็อกเกตพีซี Smartphone และ เว็บเบราว์เซอร์ ในปัจจุบัน Visual Studio นั้นสามารถใช้ภาษาโปรแกรมที่เป็นภาษาคอตเน็ต ในโปรแกรมเดียวกัน เช่น VB.NET C++ C# J# เป็นต้น

Visual Studio รุ่น Express หมายถึง รุ่นที่เปิดให้ดาวน์โหลดไปใช้ได้ฟรี โดยข้อดีของรุ่น express นี้ก็คือ นอกจากจะใช้ในการศึกษาแล้วยังสามารถนำไปพัฒนาโปรแกรมในเชิงพาณิชย์ได้ครับ ประหยัดค่าลิขสิทธิ์ซอฟต์แวร์

ซอฟต์แวร์ของไมโครซอฟท์ที่มีรุ่น express

- Visual Web Developer 2010 express สำหรับผู้สนใจพัฒนาเว็บแอปพลิเคชันด้วยเทคโนโลยี ASP.NET
- Visual C# 2010 express สำหรับผู้สนใจพัฒนาแอปพลิเคชันบนระบบปฏิบัติการวินโดวส์ หรือไลบรารีด้วยภาษา C#
- Visual Basic 2010 express สำหรับผู้สนใจพัฒนาแอปพลิเคชันบนระบบปฏิบัติการวินโดวส์หรือไลบรารีด้วยภาษา VB
- Visual C++ 2010 express สำหรับผู้สนใจพัฒนาแอปพลิเคชันบนระบบปฏิบัติการวินโดวส์ หรือไลบรารีด้วยภาษา C++
- Visual Studio 2010 Express for Windows Phone นื่องใหม่ล่าสุดสำหรับพัฒนาซอฟต์แวร์บน Pocket PC ที่ใช้ระบบปฏิบัติการ Windows Phone 7 ครับ ช่วงที่เขียนบล็อกอยู่นี้กำลังใช้เวอร์ชัน 6.5 อยู่ ส่วนเวอร์ชัน 7 ยังไม่ได้เปิดตัวเป็นทางการครับ

- SQL Server 2008 R2 Express เป็นระบบจัดการฐานข้อมูล ในรุ่น express จะมีเฉพาะ Database engine ที่ใช้ในการจัดการข้อมูลเท่านั้น ส่วนเครื่องมือสร้างรายงาน (Report) และเครื่องมือบริหารจัดการระบบฐานข้อมูล (Management Tool) ต้องติดตั้งเพิ่มต่างหากครับ และจะมีข้อจำกัดในการใช้ทรัพยากรของคอมพิวเตอร์ เช่น หน่วยความจำหลัก และขนาดของฐานข้อมูล สรุปว่าเหมาะกับงานฐานข้อมูลส่วนบุคคล หรือนงานขนาดเล็กครับ
- SQL Server Management Studio Express สำหรับบริหารจัดการระบบฐานข้อมูล SQL Server รุ่น express ครับ ยกตัวอย่างเช่น การสร้างฐานข้อมูล และการสร้าง/แก้ไข/ลบตาราง

Microsoft Expression Blend



รูปที่ 2 : แสดงโลโก้ Microsoft Expression Blend

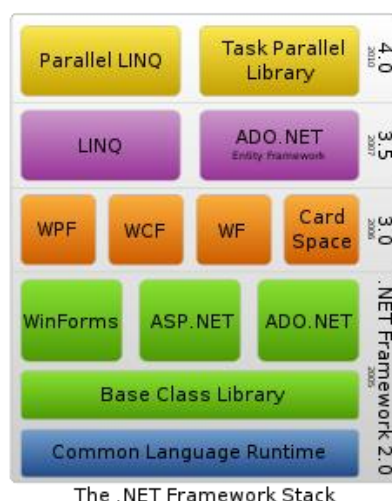
Microsoft Expression Blend เป็นเครื่องมือที่ช่วยในการออกแบบ interface และสร้างกราฟฟิคสำหรับ Web application และ Desktop application จุดเด่นของ Microsoft Expression Blend คือจะช่วยให้การออกแบบ interface บน Windows Phone ของ Silverlight ทำได้ง่ายและสะดวกขึ้น อันเนื่องมาจากมี Tools ที่ช่วยในการออกแบบด้านนี้โดยเฉพาะ เช่นออกแบบภาพเคลื่อนไหว การควบคุมการแสดงผลในรูปแบบต่าง ๆ และสามารถช่วยให้ผู้พัฒนาโปรแกรมเข้าใจรูปแบบการเขียน Silverlight โดยที่ไม่จำเป็นจะต้องมีพื้นฐานการเขียนพวก XML มากก็ตาม

.NET Technology และ .NET Framework



รูปที่ 3 : แสดงโลโก้ .NET Framework

.NET Technology และ .NET Framework คือ รูปแบบการพัฒนาโปรแกรมแบบใหม่ ที่ไมโครซอฟท์ได้พัฒนาออกมาแล้วระยะหนึ่ง โดยมีจุดประสงค์สำคัญคือสามารถใช้งานในสถานะของฮาร์ดแวร์หรือระบบปฏิบัติการ ที่แตกต่างกันได้อย่างไม่มีปัญหา (เช่น เครื่องพีซีกับเครื่องแมค หรือระบบปฏิบัติการวินโดวส์กับลินุกซ์) และสามารถพัฒนาโปรแกรมใหม่ๆ ได้ด้วยภาษาอะไรก็ได้ให้สามารถทำงานร่วมกันได้ (เช่น ภาษา C กับ Java เป็นต้น) รวมถึงเป็นเครื่องมือในการพัฒนาโปรแกรมให้สามารถเชื่อมต่อกับโปรแกรมต่างๆ ของไมโครซอฟท์ได้โดยง่าย ซึ่งก็รวมไปถึงการทำงานภายในของระบบปฏิบัติการวินโดวส์เองด้วย ผู้พัฒนาจึงสามารถพัฒนาโปรแกรมใหม่ๆ ได้โดยง่าย และรวดเร็ว ไม่ติดข้อจำกัดต่างๆ อย่างเช่นการพัฒนาโปรแกรมในสมัยก่อนอีกต่อไป



รูปที่ 4 : แสดง .NET Framework Stack

.NET Framework เป็นแพลตฟอร์มสำหรับพัฒนาซอฟต์แวร์ที่รองรับภาษาคอมพิวเตอร์มากกว่า 40 ภาษา ซึ่งมี Library เป็นจำนวนมากสำหรับการเขียนโปรแกรม รวมถึงบริหารการดำเนินการของโปรแกรมบน .NET Framework โดย Library นั้นได้รวมถึงส่วนต่อประสานกับผู้ใช้ การเชื่อมต่อฐานข้อมูล วิทยาการเข้ารหัสลับ อัลกอริทึม การเชื่อมต่อเครือข่ายคอมพิวเตอร์ และการพัฒนาเว็บแอปพลิเคชัน โดย .NET Framework มีส่วนประกอบ ภายในแบ่งออกเป็น 3 ชั้นใหญ่ๆ คือ

1. **Programming Language** : เป็นรูปแบบของ ภาษา ที่ ออกแบบ มาเพื่อให้สามารถทำงานในสถานะที่เป็น .NET ได้โดยที่ทาง Microsoft ได้เปิดตัว ภาษาหลัก ๆ ที่จะใช้ ในการพัฒนาบน .NET นี้ 3 ภาษา C# เป็น ภาษา ใหม่ที่ Microsoft พัฒนา มาจาก C++ กับ JAVA เป็นหลัก VB.NET เป็น ภาษา ที่ พัฒนา มาจาก Visual Basic ในเวอร์ชัน 6.0 JScript.net เป็น ภาษา ที่ พัฒนา มาจาก JScript ซึ่งเป็น JavaScript ใน เวอร์ชัน ของ Microsoft
2. **Base Classes Library** : Library นั้นเปรียบเสมือน ชุดคำสั่งสำเร็จรูป ย่อยๆที่เพิ่มเข้ามา ซึ่งส่วนใหญ่จะเป็น ชุดคำสั่ง ที่ต้องใช้ใช้งานอยู่เป็นประจำ ดังนั้นจึงมีผู้คิดค้น เครื่องอำนวยความสะดวก ในการเขียนโปรแกรม ซึ่ง Library ใน ภาษา ต่างๆส่วนใหญ่จะอยู่ในรูปแบบไฟล์ include แต่ถ้าเป็น .ASP สิ่งที่เป็น library ก็คือ component ต่างๆนั่นเอง ซึ่งภายในระบบ .NET จะสร้างสิ่งที่เรียกว่าเป็น Library พื้นฐานขึ้น ทำให้ไม่ว่าจะใช้ ภาษา ใด ในการพัฒนา โปรแกรม ก็สามารถที่จะเรียกใช้ Library ที่เป็นตัวเดียวกันได้หมด
3. **Common Language Runtime (CLR)** : นับเป็น สิ่งสำคัญ แทบจะที่สุดของระบบ .NET นี้ก็ว่าได้ เพราะ CLR ที่ว่านี้มีหน้าที่ทำให้ โปรแกรม ที่เขียนขึ้นมาด้วย ภาษา ต่างๆกัน กลายเป็น ภาษา รูปแบบ มาตรฐาน เดียวกัน ทั้งหมด เราเรียก ภาษา ที่ว่านี้ว่า Intermediate language (IL) ซึ่งเมื่อต้องการที่จะรัน โปรแกรม ใด CLR ที่ว่านี้จะ ตรวจสอบ เครื่องที่รันว่ามี สภาพแวดล้อม การทำงาน เช่นใดหลังจากนั้นก็ จะ คอมไพล์ เป็น โปรแกรม ที่เหมาะสมต่อ การทำงาน ของเครื่องนั้น ทำให้เราสามารถใช้งาน โปรแกรม ต่างๆได้อย่างมีประสิทธิภาพสูงสุด ในแต่ละเครื่องผลกระทบเมื่อเลือกใช้งาน .NET Framework Technology

.NET Framework ไม่ใช่เป็น Component ที่ติดตั้งให้ทันทีเมื่อลงระบบปฏิบัติการวินโดวส์ XP หรือวินโดวส์ 2000 แต่เป็น Component หนึ่งที่สามารถติดตั้งเพิ่มจากแผ่นติดตั้งหรือ Download เพื่อติดตั้งเองได้ ทั้งนี้การติดตั้งนี้ไม่มีค่าใช้จ่ายเรื่องลิขสิทธิ์หรือมีผลกระทบต่อโปรแกรมที่ติดตั้งอยู่แต่เดิม

โปรแกรมที่พัฒนา **.NET Application** จะพัฒนาโปรแกรมที่จะได้ภาษากลางที่เรียกว่า Intermediate Language (IL) ที่จะต้องส่งให้ .NET Platform เป็นตัวกลางในการแปลภาษาที่ได้พัฒนาเป็นภาษาเครื่อง (Machine code) อีกทีหนึ่ง ซึ่งแตกต่างจากโปรแกรมที่ไม่ได้พัฒนาด้วย

.NET Technology ที่จะพัฒนาแล้วได้ภาษาเครื่องออกมาทันที ซึ่งมีการประมาณว่า ประสิทธิภาพของโปรแกรมที่ทำงานบน .Net Framework นั้น จะได้ประมาณ 80% ของโปรแกรมที่ไม่ได้พัฒนาด้วย .NET Technology (เช่น delphi หรือ Visual Basic 6.0) ทั้งนี้ความแตกต่างจะเห็นได้ชัดเจนขนาดไหนนั้นจะขึ้นกับทรัพยากรของเครื่องด้วย

โปรแกรมที่พัฒนา .NET Application จะถูกควบคุมให้อยู่ในสภาพแวดล้อมที่ควบคุมไว้บน .NET Framework ซึ่งเป็นข้อดีในแง่ของความน่าเชื่อถือของระบบ และผลกระทบที่อาจเกิดขึ้นว่าจะไม่กระทบต่อการทำงานส่วนอื่นๆ? โปรแกรมที่พัฒนาด้วย .NET Technology นั้น โดยส่วนใหญ่แล้วจะไม่สามารถเชื่อมต่อโดยตรงให้เข้ากับโปรแกรมที่ไม่ได้พัฒนาด้วย .NET Technology ได้ การใช้งานร่วมกันระหว่างโปรแกรมจึงเกิดขึ้นเฉพาะระหว่างโปรแกรมที่พัฒนาด้วย .NET Technology อย่างไรก็ดี ทางไมโครซอฟต์ได้ออกแบบให้มีทางออกในการเชื่อมต่อกับโปรแกรมอื่นๆ ได้โดยง่ายผ่านเทคโนโลยี Web Service ซึ่งทำให้อุปสรรคในการทำงานระหว่างโปรแกรมอยู่ในรูปแบบที่เป็นมาตรฐานและเปิดกว้างมากขึ้นประโยชน์ที่ได้เมื่อเลือกใช้โปรแกรมที่พัฒนาบน .NET Technology

.NET Framework ข้อดีและประโยชน์

1. เป็นระบบที่มี Library ที่เป็น มาตรฐาน เดียวกัน เนื่องจากมี Library ที่เป็น มาตรฐาน เดียวกัน ทั้งหมดทำให้เราไม่ต้องกังวลว่า ภาษา ที่ใช้เขียนนั้นมี Library ตัวนั้นตัวนี้หรือไม่ รวมทั้งไม่ต้องคอยกังวลว่าถ้าใช้ Library ของ ภาษา หนึ่งแล้วอีก ภาษา หนึ่งจะไม่มี Library ตัวนั้น
2. ไม่ขึ้นกับ ระบบปฏิบัติการ (OS) เนื่องจาก ระบบปฏิบัติการ ที่แต่ละ บุคคล หรือ องค์กร ใช้ นั้นย่อมไม่เหมือนกัน แต่ภายใน .NET Framework จะไม่มี ปัญหา นี้ของเพียงแค่มีระบบ .NET Framework ก็จะทำให้สามารถใช้งาน โปรแกรม ต่างๆ ได้ ซึ่งเป็นข้อดีตรงที่เราจะสามารถใช้โปรแกรมต่างๆ ได้ทุก ระบบปฏิบัติการ
3. ใช้ในการพัฒนาได้ทุกภาษา ทำให้เราไม่ต้องคอยมา ศึกษา ภาษา ใหม่ ๆ เมื่อต้อง การสร้าง โปรแกรม ในแต่ละครั้ง
4. มีการควบคุม สิ่งแวดล้อม ในการทำงานเป็นอย่างดี เนื่องจากเป็นระบบที่เป็น มาตรฐาน ทำให้ การควบคุม จัดสรรระบบต่างๆ ทำได้ง่ายขึ้น ไม่ว่าจะเป็นการจัดสรร หน่วยความจำ ด้านการใช้งานเครื่องก็มีความรวดเร็วมากขึ้น ลดโอกาสที่เครื่องจะแฉงได้เป็นอย่างดี
5. ความปลอดภัย ที่มีมากขึ้น .NET Framework สามารถ กำหนดสิทธิ์ การใช้งานหรือ permission ของ ผู้ใช้งาน ได้มากขึ้นทำให้สามารถกำหนดว่า จะให้ โปรแกรม ในส่วนใด ใช้งานได้หรือไม่ได้ แล้วแต่เฉพาะบุคคล

WPF (Windows Presentation Foundation)

WPF ย่อมาจาก Windows Presentation Foundation คือเทคโนโลยีหนึ่งของ Microsoft ในการสร้างซอฟต์แวร์ออกมาได้อย่างสวยงาม และรัน บน Windows เท่านั้น สามารถสร้างสรรค์ Application ในแบบ Rich UI, Supported Render 3D แม้กระทั่งใช้ความสามารถของ DirectX และความสามารถอื่นๆอีกมากมาย มีระบบการทำงานคล้ายๆ Windows Form Application แต่ว่ามีอิสระมากกว่า ทำงานได้หลากหลาย ง่ายต่อการแก้ไข จำกัดการใช้งานที่ .NET Framework 3.0 ขึ้นไป

ข้อดีของ WPF Application

- สามารถสร้างโปรแกรมให้สวยงามได้ด้วย Style (คล้าย CSS ของ Web)
- มีกราฟฟิคที่สวยงาม
- สร้าง Animation ในโปรแกรมได้ง่ายด้วย Storyboard
- สามารถแบ่งหน้าที่การทำงานระหว่าง Designer และ Programmer ได้อย่างชัดเจน
- กำหนด Control(Tool) ของเราเองได้ด้วย User Control
- ไม่ต้องพิมพ์ Code มาก ใช้ interface ตั้งการส่วนใหญ่

XAML เป็นส่วนเพิ่มเติมมาใน Windows Presentation Foundation จะอยู่ใน .NET Framework 3.0 เพื่อใช้สร้าง User Interface โดยหลังจากสร้าง User Interface ตัวโปรแกรมจะ Generate Code ออกมาเป็นไฟล์ Text ซึ่งก็คือ .XAML ที่ง่ายต่อการมองและแก้ไขอย่างยิ่ง โดยไฟล์ XAML นั้นจะใช้อธิบาย User Interface ที่เราสร้างขึ้นได้ทั้งหมดทุกอย่างไม่ว่าจะเป็น สี ตำแหน่ง การวาง ขนาด Animation หรือ Feature ต่างๆ ทั้งหมดจะเก็บอยู่ในนี้ และไฟล์ต่างๆที่ถูกสร้างหรืออยู่บนมาตรฐาน XAML นั้นสามารถใช้ภาษาต่างๆใน .NET สร้างได้เลย ในแง่ของเทคโนโลยี XAML ยินยอมให้ Designer และ Developer สามารถแชร์ข้อมูลกันได้โดยตรงผ่าน XAML ซึ่งทำให้ง่ายในการพัฒนาเป็นอย่างยิ่ง

ภาษา C#



รูปที่ 5 : แสดงโลโก้ C#.net

ภาษา C# จะเขียนว่า C Sharp ก็ได้ (อ่านว่าซีชาร์พ) เครื่องหมาย # ในทางคณิตศาสตร์หมายถึงครึ่งเสียง ส่วนคำว่า Sharp หมายถึงเจียบคม มีไหวพริบ บริษัทไมโครซอฟท์ตั้งชื่อภาษานี้ว่า C# เพื่อจะสื่อให้เห็นว่านี่คือภาษาในตระกูลภาษา C เช่นเดียวกับภาษา C++ คือเป็นภาษาที่มีต้นกำเนิดจากภาษา C หากท่านสังเกตให้ดีจะเห็นว่าสัญลักษณ์ # จะมองให้เป็นเครื่องหมาย + สื่อนำมาซ้อนกันก็ได้ แสดงเป็นนัยว่าภาษานี้ก้าวหน้ากว่าภาษา C++ ไปอีกระดับหนึ่ง (คือเป็นภาษา C+++) สัญลักษณ์ # นี้มีปรากฏในภาษาอื่นๆ ของ .NET ด้วย เช่นภาษา J#(เดิมคือภาษา J++) และภาษา A# (เหมือนภาษา Ada) อาจเป็นเพื่อเหตุผลทางการค้า

ภาษา C# คืออะไร

ภาษา C# เป็นภาษาที่ใหม่มากๆ ปรากฏตัวเป็นครั้งแรกในปี 2000 และถูกอัปเดตเวอร์ชันอยู่เสมอ มันเป็นภาษาที่ได้รับอิทธิพลจากภาษาก่อนหน้าเช่นภาษา Delphi ภาษา C++ ภาษา Java และภาษา Eiffel ในตอนต้นภาษานี้ถูกออกแบบ และกำหนดลักษณะโดยบริษัทไมโครซอฟท์ ต่อมาได้ถูกรับรองจากหน่วยงาน ECMA (หน่วยงานกำหนดมาตรฐานสากลด้านสารสนเทศ) และ ISO แต่ปัจจุบันไมโครซอฟท์ยังพัฒนาภาษานี้อย่างต่อเนื่อง (ปัจจุบันเป็นเวอร์ชัน 3.0)

ภาษา C# ถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework เป็นการนำข้อดีของภาษาต่างๆ (เช่นภาษา Delphi ภาษา C++) มาปรับปรุงเพื่อให้มีความเป็น OOP อย่างถึงที่สุด ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลง (เรียบง่ายกว่าภาษา C++) และมีเครื่องแต่งตัวน้อยลง (เมื่อเทียบกับ Java)

ประวัติความเป็นมาของภาษา C#

ภาษา C# เป็นภาษา โปรแกรมเชิงวัตถุทำงานบน .NET Framework พัฒนาโดยบริษัท ไมโครซอฟท์ และมี Anders Hejlsberg เป็นหัวหน้าโครงการ โดยมีรากฐานมาจากภาษา C++ และ ภาษาอื่นๆ (โดยเฉพาะภาษาเคลไพและจาวา) โดยปัจจุบันภาษา C# เป็นภาษามาตรฐานรองรับโดย ECMA และ ISO

ซึ่งในปัจจุบันได้พัฒนาและปรับปรุงแบบของ ภาษา C# อยู่ตลอดเวลาโดยทาง Microsoft ได้นำภาษา C# ไปอยู่ในชุดพัฒนา software อย่าง visual studio ซึ่งทำให้เป็นที่นิยมเพิ่มมากขึ้น

ไมโครซอฟท์ส่งมาตรฐานภาษา C# ให้กับ ECMA และได้รับการยอมรับเป็นมาตรฐาน ECMA ในเดือนธันวาคม ค.ศ. 2001 ในชื่อว่า ECMA-334 C# Language Specification ใน ค.ศ. 2003 ภาษา C# ได้รับการยอมรับเป็นมาตรฐาน ISO (ISO/IEC 23270) มาตรฐาน ISO/IEC 23270:2003 ระบุรูปแบบ และกำหนดการแปล (ตีความ) โปรแกรมที่เขียนด้วยภาษา C# โดยตัวมาตรฐานได้ระบุ:

- รูปแบบการนำเสนอ
- ไวยากรณ์
- กฎการตีความสำหรับแปลโปรแกรมภาษา C#
- ข้อห้าม และข้อจำกัด ของเครื่องมือที่สร้างตามข้อกำหนดของ C#
- ISO/IEC 23270:2003 ไม่ได้ระบุ
- กลไกในการแปลงโปรแกรมภาษา C# เพื่อใช้ในระบบประมวลผลข้อมูล
- กลไกในการเรียกให้โปรแกรมภาษา C# ทำงาน เพื่อใช้ในระบบประมวลผลข้อมูล
- กลไกในการแปลงข้อมูลเข้า เพื่อใช้กับโปรแกรมภาษา C#
- กลไกในการแปลงข้อมูลออก หลังจากถูกประมวลผลโดยโปรแกรมภาษา C#

นอกจากนี้ตัวมาตรฐานไม่ได้กล่าวถึง โครงสร้างข้อมูล (Data Structure) และตัว Library กลางของ .NET Framework ซึ่งเป็นหัวใจสำคัญของการเขียนโปรแกรมด้วยภาษา C# เลย

ผู้สร้างภาษา C#

ผู้สร้างภาษา C# คือบริษัท ไมโครซอฟท์ แต่บิดาของภาษา C# คือ Anders Hejlsberg (แอนเดรส ฮาเยสเบิร์ก) ไมโครซอฟท์ต้องการให้ภาษา C# เป็น “อะไรที่จะอยู่ไปอีกรุ่น” เหมือนบริษัท รถยนต์โพลล์ที่จ้าง Ferdinand Porsche (เฟอร์ดินันด์ พอร์ช) นักออกแบบรถยนต์มือดีมาออกแบบรถโพลคเต่า (เป็นเหตุการณ์ที่เกิดขึ้นก่อนสงครามโลกครั้งที่สอง) ทำให้มันกลายเป็นรถยนต์คลาสสิกมาจนถึงปัจจุบัน ในทำนองเดียวกันเมื่อต้องการภาษาที่ต้องการให้กลายเป็นภาษา “คลาสสิก” บริษัท ไมโครซอฟท์ตัดสินใจมอบหมายให้ Hejlsberg บรมครูนักออกแบบภาษา ผู้เคยสร้างภาษาที่กลายเป็นตำนานมาแล้วเช่น Turbo Pascal และผู้นำในทีมสร้างภาษา Delphi

ใครๆ ก็สร้างภาษาใหม่ได้ไม่ยาก แต่ภาษาที่จริงจัง นั้นเป็นอีกเรื่องหนึ่ง สำหรับภาษา C# Hejlsberg ไม่เพียงสร้างภาษาที่ใช้งานได้ดีเท่านั้น เขายังเน้น “ความมีรสนิยม” และ “ความสวยงาม” เพราะต้องการให้โปรแกรมเมอร์เขียน โปรแกรมภาษา C# ได้อย่างมีความสุข



รูปที่ 6 : แสดงภาพผู้ได้ชื่อว่าเป็นบิดาของภาษา C# ชื่อว่า Anders Hejlsberg

จุดเด่นของ C#

เป็นรูปแบบของภาษาที่ทำงานเป็นลำดับ(Sequential)และต้องผ่านการแปลงไฟล์ให้อยู่ในรูปแบบพร้อมใช้งานหรือ EXE file (ที่เรียกว่าการ Compile) เช่นเดียวกับภาษา C โดย C# เป็นภาษาที่ถูกผลักดัน

โดยบริษัทไมโครซอฟท์ให้ออกมายิ่งใหญ่โดยมีจุดเด่นดังนี้

ในการเขียน C# มีเครื่องมือในการช่วยเขียนอย่าง Microsoft Visual Studio ที่มีฟีเจอร์ในการไฮไลต์โค้ด และส่วนช่วยในการเขียนโปรแกรมที่สะดวกมาก

- เขียนง่ายจุดเด่นของตัวแปรและอ็อบเจกต์บน C# เน้นชัดในเรื่องของการจัดการคุณสมบัติ (Properties) และการตั้งค่าเริ่มต้นที่ช่วยให้สามารถพัฒนาระบบได้สะดวกรวดเร็วขึ้น
- อ่านง่าย การจัดระเบียบโดยตัว Visual Studio เป็นส่วนที่ผมชอบที่สุดเลยก็ว่าได้คุณเขียนโค้ดไปสักพักหนึ่ง จะเริ่มรู้สึกคล้ายตากับย่อหน้าที่งั้นๆ แต่ Visual Studio โดยปกติเมื่อจะจัดการกับย่อหน้าทั้งหมดเหล่านั้นให้อยู่ในระเบียบสะอาดตาที่สุดเลยก็เดียว
- เป็นอ็อบเจกต์ เนื่องจากภาษา C# นั้นมีแม่แบบมาจากภาษา Java ซึ่งจุดเด่นคือการทำทุกสิ่งให้เป็นวัตถุ (Object) ทำให้สามารถเขียนและพัฒนาได้ง่าย
- ประสิทธิภาพสูง C# เป็นภาษาที่พัฒนาขึ้นมาภายใต้ .NET Framework ซึ่งสามารถดึงเอาความสามารถของเทคโนโลยีบน .NET ออกมาใช้ได้อย่างเต็มประสิทธิภาพ

- สามารถทำงานระดับลึก ภาษา C# สามารถทำงานกับหน่วยความจำรวมถึงระบบคอมพิวเตอร์ในระดับลึก โดยผ่าน พอยเตอร์ (Pointer) หรือทำงานกับโปรโตคอล TCP/IP ที่ต่ำกว่าระดับ4ได้
- เน้นที่ XML C# ออกแบบมาเพื่อใช้งานร่วมกับXMLได้อย่างราบรื่นที่สุดด้วยความช่วยเหลือของ .NET Framework
- คุณสมบัติGeneric type ช่วยยืดหยุ่นในการประกาศตัวแปร
- เทคโนโลยี LINQ ช่วยในการทำงานกับฐานข้อมูลได้อย่างมีประสิทธิภาพ

Microsoft SQL Server



รูปที่ 7 : แสดงโลโก้ Microsoft SQL Server

SQL Server เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (relational database management system หรือ RDBMS) จาก Microsoft ที่ได้รับการออกแบบสำหรับสภาพแวดล้อมวิสาหกิจ SQL Server เรียกใช้บน T-SQL (Transact-SQL) ชุดของส่วนขยายโปรแกรมจาก Sybase และ Microsoft ที่เพิ่มหลายส่วนการทำงานจาก SQL มาตรฐาน รวมถึงการควบคุมทรานแซคชัน, exception และการควบคุมความผิดพลาด, การประมวลผลแถว และการประกาศตัวแปร

Yukon เป็นชื่อรหัสในการพัฒนา SQL Server 2005 ได้รับการเผยแพร่ในเดือนพฤศจิกายน 2548 ผลิตภัณฑ์ 2005 ได้รับการกล่าวหาว่าให้ความยืดหยุ่น ความสามารถเชิงปริมาณ ความน่าเชื่อถือ และความปลอดภัยกับการประยุกต์ฐานข้อมูล และทำให้สร้างและจัดวางง่ายขึ้น ดังนั้น จึงลดความซับซ้อนและความน่าเบื่อเกี่ยวกับการจัดการฐานข้อมูล SQL Server 2005 รวมการสนับสนุนการบริหารมากขึ้นด้วย

ต้นกำเนิดคำสั่ง SQL Server ได้รับการพัฒนาโดย Sybase ในปลายทศวรรษ 1980 Microsoft, Sybase และ Ashton-Tate รวมมือในการผลิตเวอร์ชันแรกของผลิตภัณฑ์นี้เวอร์ชันแรก SQL Server 4.2 สำหรับ OS/2 นอกจากนี้ ทั้ง Sybase และ Microsoft เสนอผลิตภัณฑ์ SQL Server โดย Sybase เปลี่ยนชื่อผลิตภัณฑ์ของพวกเขาเป็น Adaptive Server Enterprise

Microsoft SQL Server เป็นโปรแกรมในการบริหารจัดการฐานข้อมูลที่ดีที่สุดของ Microsoft โดยเป็นในรูปแบบของ Relational Database Management System หรือที่เรียกว่า RDBMS ซึ่งจะบริหารข้อมูลให้กับผู้ใช้บริการต่าง ๆ รองรับการทำงานได้จำนวนมาก และมีความสามารถมากมาย เทียบเท่ากับระบบฐานข้อมูลอื่น ๆ เช่น Oracle, DB2, Informix เป็นต้น มีคุณสมบัติเด่นเรื่องของ User Interface ที่ใช้งานได้ง่าย

ภายใต้การจัดการของ Microsoft SQL Server นั้น จะมีฐานข้อมูลที่ Microsoft SQL Server จะต้องใช้ และมีมาตั้งแต่ต้น นับตั้งแต่ Install กันเลย โดยฐานข้อมูลนี้จะอยู่ในกลุ่มของ System Database ซึ่งแต่ละฐานข้อมูลจะมีหน้าที่ในการทำงานต่าง ๆ กันดังนี้

- **Master Database** มีความสำคัญมากที่สุด ใช้สำหรับเก็บข้อมูลที่สำคัญของระบบ เช่น Meta Data พาก User, Login Information, Error Message, Linkedserver รวมถึงบอกตำแหน่งใน Primary File ในแต่ละข้อมูลอีกด้วย โดยหากฐานข้อมูลนี้มีปัญหาก็จะส่งผลกระทบต่อการทำงานของ Microsoft SQL Server เลยทีเดียว
- **MSDB Database** สำคัญ รองจาก Master เก็บข้อมูลเกี่ยวกับ Alert, Job, Schedule ซึ่งถูกใช้โดยบริการของ SQL Server Agent ซึ่งเป็นบริการที่ทำงานอัตโนมัติ เช่น Backup ข้อมูลอัตโนมัติแต่ละวัน การแจ้งเตือนเมื่อระบบมีปัญหา โดยจะทำการส่ง Email ให้กับผู้ดูแลระบบ เป็นต้น
- **Distribution Database** ใช้ในการ Replication ของฐานข้อมูล เพื่อให้ข้อมูลของ Microsoft SQL Server แต่ละที่มีความตรงกัน
- **Model Database** เป็นฐานข้อมูลที่เป็นต้นแบบ (Database Template) กรณีที่เราสร้าง Database ใหม่ Microsoft SQL Server จะเอา Model Database นี้มาเป็นตัวตั้งต้น
- **TempDB Database** ไว้ในการเก็บข้อมูลที่เป็นชั่วคราว Temporary สำหรับ process ที่จำเป็นต้องการนำ data มาพักไว้ก่อนแล้วค่อยนำไปทำอย่างอื่นต่อ ลบทุกครั้งที่มีการ Restart Service หรือ Shutdown ระบบ

ลำดับขั้นตอนการสร้างโปรแกรมบน .NET FRAMEWORK

คอมพิวเตอร์ที่ใช้ทดสอบการสร้างโปรแกรมต้องติดตั้ง .NET Framework หรือติดตั้ง Visual Studio มาแล้ว คู่มือการติดตั้งได้ในบล็อกรายการ: การติดตั้งโปรแกรม Visual Studio 2010 Express edition

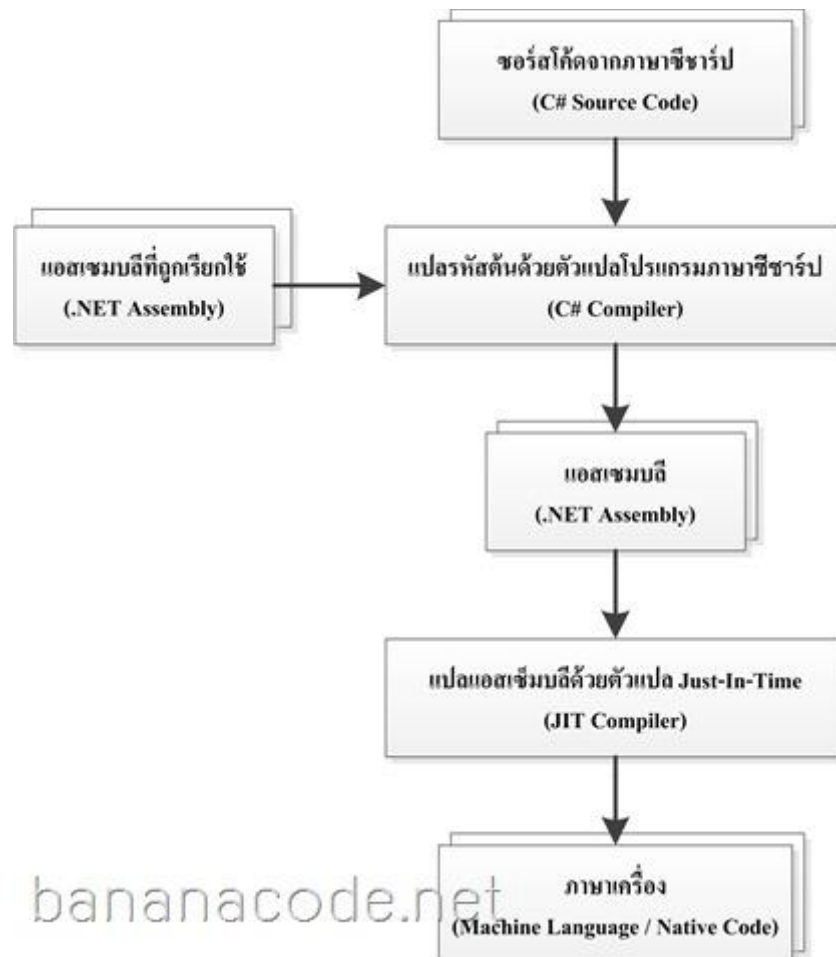
- **ขั้นตอนที่ 1** ทำการสร้างรหัสต้นทางภาษาซีชาร์ป (C# Source Code) บางครั้งถูกเรียกว่าซอร์สโค้ด ในขั้นตอนนี้เราใช้โปรแกรมประเภทเท็กซ์เอดิเตอร์ (Text Editor) เช่น Notepad ที่มีอยู่ในวินโดวส์ หรือจะใช้เครื่องมือที่สนับสนุนการพัฒนาโปรแกรมบนคอทเน็ตเฟรมเวิร์กโดยเฉพาะอย่าง Visual Studio (Visual Studio) ช่วยสร้างรหัสต้นทาง โดยชื่อของแฟ้มรหัสต้นทางนั้นต้องมีส่วนขยาย cs
- **ขั้นตอนที่ 2** ทำการแปลรหัสต้นทางด้วยตัวแปลโปรแกรมภาษาซีชาร์ป (C# Compiler) หรือเรียกอีกชื่อหนึ่งว่าคอมไพเลอร์ซึ่งเป็นองค์ประกอบหนึ่งในคอทเน็ตเฟรมเวิร์ก ทำหน้าที่แปลรหัสต้นทางให้เป็นแอสเซมบลี (Assembly) โดยในกระบวนการแปลนี้จะมีการตรวจสอบความถูกต้องในกฎระเบียบการเขียนโปรแกรม หรือวากยสัมพันธ์ (Syntax) ตามหลักภาษาซีชาร์ป ซึ่งการแปลรหัสต้นทางเป็นแอสเซมบลีจะสำเร็จก็ต่อเมื่อรหัสต้นทางนั้นไม่มีข้อผิดพลาดทางวากยสัมพันธ์ (Syntax error) แล้วนั่นเอง

เราสามารถกำหนดให้มีการแปลรหัสต้นทางหลายๆแฟ้มให้เป็นแอสเซมบลีแฟ้มเดียวกันได้ และแอสเซมบลีที่ได้จากการแปลนั้น จะมี 2 ประเภทด้วยกัน คือ

ประเภทที่ถูกเรียกใช้ประมวลผลได้โดยตรง เรียกว่า Process Assembly บ่อยครั้งจะเรียกแอสเซมบลีประเภทนี้ว่า “แอปพลิเคชัน” ซึ่งแอสเซมบลีประเภทนี้จะมีส่วนขยายในชื่อแฟ้มว่า EXE (Executable) ตัวอย่างของแอสเซมบลีประเภทนี้ที่พบได้บ่อย คือ โปรแกรมเครื่องคิดเลขที่แถมมากับระบบปฏิบัติการวินโดวส์ เราเรียกใช้โปรแกรมเครื่องคิดเลขได้โดย พิมพ์คำสั่ง calc.exe หรือเรียกผ่านสตาร์ทเมนูของวินโดวส์นั่นเอง

ประเภทที่ถูกเรียกใช้โดยแอสเซมบลีอื่นๆ เสมือนเป็นตัวช่วยงานแอสเซมบลีอื่นนั่นเอง เรียกว่า Library Assembly ในหนังสือเล่มนี้จะเรียกแอสเซมบลีประเภทนี้อย่างสั้นว่า “ไลบรารี” ซึ่งไลบรารีนี้จะมีส่วนขยายในชื่อแฟ้มว่า DLL (Dynamic Link Library) ในคอทเน็ตเฟรมเวิร์กนั้นจะมีคลังสำหรับเก็บแอสเซมบลี เรียกว่า Base Class Library หรือ BCL ซึ่งมีความสามารถแตกต่างกัน เช่น บางแอสเซมบลีจะมีความสามารถในการกราฟฟิก หรือบางแอสเซมบลีจะมีความสามารถในการเชื่อมต่อระบบฐานข้อมูล เป็นต้น ข้อดีของแอสเซมบลีประเภทนี้คือ ทำให้เราได้ประโยชน์จากหลักการนำโปรแกรมที่เคยเขียนไว้แล้วกลับมาใช้ในโปรแกรมใหม่ (Software Reuse) ซึ่งทำได้โดยการเขียนคำสั่งในรหัสต้นทางเพื่อเรียกใช้งานแอสเซมบลีนั่นเอง

- **ขั้นตอนที่ 3** เรียกใช้งานแอสเซมบลี หลังจากที่เราได้แอสเซมบลีใหม่จากการแปลซอร์สโค้ดแล้ว เมื่อมีการเรียกใช้แอสเซมบลีจะถูกแปลครั้งที่ 2 ด้วย JIT compiler เพื่อแปลแอสเซมบลีที่ถูกเรียกใช้เป็นภาษาเครื่อง (Machine Language) และทำงาน



รูปที่ 8 : แสดงขั้นตอนการสร้างโปรแกรมบน .NET Framework

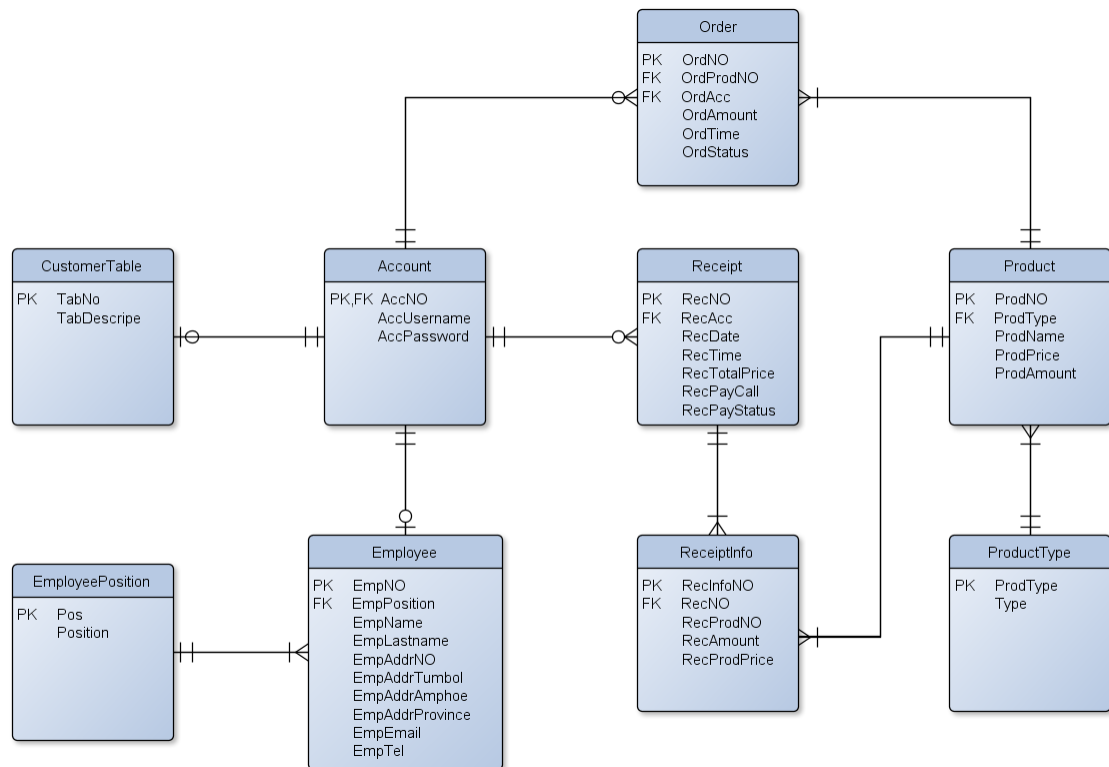
สรุปขั้นตอนในการสร้างโปรแกรมบน .NET Framework โดยเริ่มจาก

1. การเขียนโปรแกรมด้วยภาษา C#
2. นำโปรแกรมที่เขียนไปแปลด้วยตัวแปลภาษา C# ทำให้เราได้แอสเซมบลี และ
3. เรียกใช้แอสเซมบลี

บทที่ 3

วิเคราะห์และออกแบบระบบ

Entity-Relationship Model



รูปที่ 9 : แสดง Entity-Relationship Model ของระบบจัดการร้านกาแฟ

Data Dictionary

Product						
Column Name	Type	Size	Null	Default	Description	Link to...
<u>ProdNO</u>	varchar	10	NO		รหัสสินค้า	
ProdName	varchar	64	NO		ชื่อสินค้า	
ProdPrice	decimal	4,2	NO	0	ราคาสินค้า	
ProdAmount	integer	4	NO	0	จำนวนสินค้า	

ตารางที่ 3 : แสดง Data Dictionary ของตาราง “Product”

Employee						
Column Name	Type	Size	Null	Default	Description	Link to...
<u>EmpNO</u>	varchar	5	NO		รหัสพนักงาน	
EmpName	varchar	32	NO		ชื่อพนักงาน	
EmpLastname	varchar	32	NO		นามสกุลพนักงาน	
EmpHouseNO	varchar	128	NO		ที่อยู่พนักงาน	
EmpTel	varchar	10	NO		เบอร์โทรศัพท์ พนักงาน	

ตารางที่ 4 : แสดง Data Dictionary ของตาราง “Employee”

Data Dictionary(2)

CustomerTable						
Column Name	Type	Size	Null	Default	Description	Link to...
<u>TabNO</u>	varchar	5	NO		หมายเลขโต๊ะ	
TabDescribe	varchar	128	YES		รายละเอียด โต๊ะ (ตำแหน่ง)	

ตารางที่ 5 : แสดง Data Dictionary ของตาราง “CustomerTable”

Account						
Column Name	Type	Size	Null	Default	Description	Link to...
<u>AccNO</u>	varchar	5	NO		รหัสบัญชีผู้ใช้	EmpNO, TabNO
AccUsername	varchar	16	NO		บัญชีผู้ใช้	
AccPassword	varchar	16	NO		รหัสบัญชีผู้ใช้	
AccPosition	varchar	1	NO		ตำแหน่ง	

ตารางที่ 6 : แสดง Data Dictionary ของตาราง “Account”

Data Dictionary(3)

Receipt						
Column Name	Type	Size	Null	Default	Description	Link to..
<u>RecNO</u>	varchar	10	NO		เลขที่ใบเสร็จ	
RecAcc	varchar	5	NO		รหัสบัญชีผู้ใช้	AccNO
RecDate	date	YYYY-MM-DD	NO		วันที่บันทึก	
RecTime	time	hh:mm:ss	NO		เวลาที่บันทึก	
RecTotalPrice	decimal	6,2	NO		ราคาทั้งหมด	
RecPayCall	integer	1	NO	0	สถานะเรียกชำระเงิน	
RecPayStatus	integer	1	NO	0	สถานะการชำระเงิน	

ตารางที่ 7 : แสดง Data Dictionary ของตาราง “Receipt”

ReceiptInfo						
Column Name	Type	Size	Null	Default	Description	Link to..
<u>RecInfoNO</u>	integer	10	NO		ลำดับรายการ	
RecNO	varchar	10	NO		เลขที่ใบเสร็จ	RecNO
RecProdNO	varchar	10	NO		รหัสสินค้า	ProdNO
RecAmount	integer	4	NO		จำนวนสินค้า	
RecProdPrice	decimal	4,2	NO		ราคาสินค้าทั้งหมด	

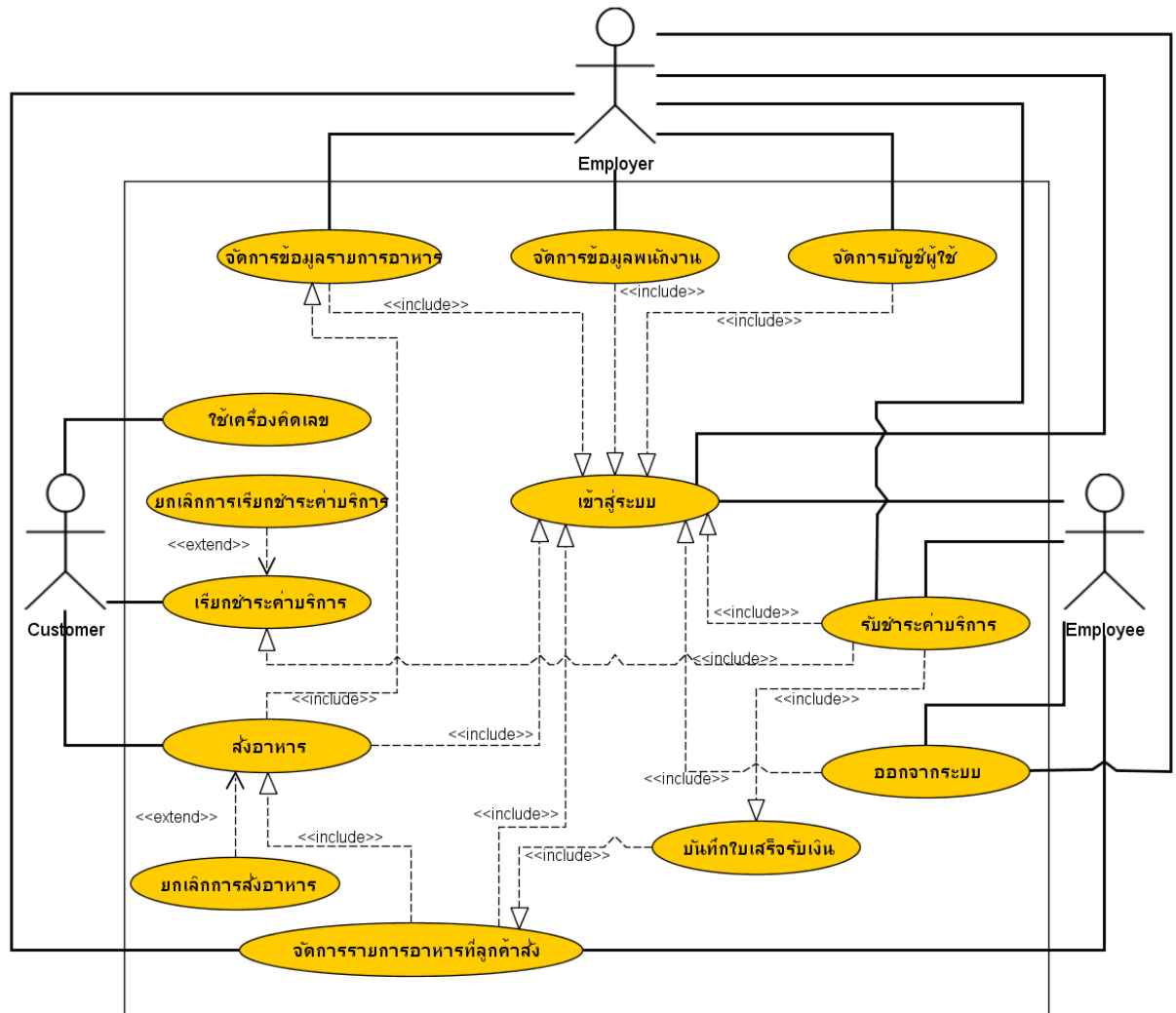
ตารางที่ 8 : แสดง Data Dictionary ของตาราง “ReceiptInfo”

Data Dictionary(4)

Order						
Column Name	Type	Size	Null	Default	Description	Link to..
<u>OrdNO</u>	integer	10	NO		ลำดับการสั่ง	
OrdProdNO	varchar	10	NO		รหัสสินค้า	ProdNO
OrdAcc	varchar	5	NO		รหัสบัญชีผู้ใช้	
OrdAmount	integer	4	NO			
OrdTime	time	hh:mm:ss	NO			
OrdStatus	integer	1	NO	0		

ตารางที่ 9 : แสดง Data Dictionary ของตาราง “Order”

Unified Modeling Language(UML)



รูปที่ 10 : แสดงความสัมพันธ์ Use Case Diagram

Use Case Scenario

Use case Name	เข้าสู่ระบบ	
Area	ระบบการจัดการร้านกาแฟ	
Actor(s)	เจ้าของร้าน (Employer) พนักงาน (Employee)	
Description	ยืนยันตัวตนในการเข้าสู่ระบบ เพื่อจัดการสิทธิ์ในการใช้งานส่วนต่างๆ	
Triggering Event	เจ้าของร้านหรือพนักงานต้องการใช้งานระบบ	
Step Performed (main path)		Information for steps
เปิดใช้งานระบบ		Show UI : LoginForm
กรอก Username และ Password และกดปุ่ม “Login”		Show UI : LoginForm, {Call checkLogin} Check from Account Table : AccUserName+AccPassword+AccPosition
Preconditions	เจ้าของร้านหรือพนักงานต้องการใช้งานระบบ	
Postconditions	ใช้งานฟังก์ชันต่างๆของระบบได้ตามสิทธิ์ที่ได้รับ	
Assumptions	ไม่สามารถใช้งานฟังก์ชันต่างๆได้ เพราะไม่ได้ยืนยันตัวตน	

ตารางที่ 10 : แสดง Use Case Scenario ของงาน “เข้าสู่ระบบ”

Use Case Scenario(2)

Use case Name	จัดการข้อมูลอาหาร	
Area	ระบบการจัดการร้านกาแฟ	
Actor(s)	เจ้าของร้าน (Employer)	
Description	เพิ่ม ลบ หรือปรับปรุง ข้อมูลของอาหาร	
Triggering Event	เจ้าของร้านต้องการเพิ่ม ลบ หรือปรับปรุงข้อมูลของอาหาร	
Step Performed (main path)	Information for steps	
เปิดใช้งานระบบ	Show UI : LoginForm	
กรอก Username และ Password และกดปุ่ม “Login”	Show UI : LoginForm, {Call checkLogin} Check from Account Table : AccUserName+AccPassword+AccPosition	
แสดงหน้าจอสำหรับเจ้าของร้าน	Show UI : OwnerForm	
เลือกงานที่ต้องการ (เพิ่ม ลบ ปรับปรุง ข้อมูลรายการอาหาร)	ShowUI : OwnerForm If(เพิ่มรายการอาหาร) {call addProduct} Else if(ลบรายการอาหาร) {call delProduct} Else if(ปรับปรุงรายการอาหาร) {call updateProduct}	
จัดการข้อมูลรายการอาหาร	Manage Product Table : ProdNO+ProdName+ProdPrice+ProdAmount	
Preconditions	เจ้าของร้านสั่งสินค้าใหม่, ยกเลิกการผลิต หรือมีการเปลี่ยนแปลงรายละเอียดของสินค้า	
Postconditions	ข้อมูลรายการอาหารมีการเปลี่ยนแปลง	
Assumptions	ไม่มีข้อมูลของอาหารในระบบหรือข้อมูลไม่ตรงกับความเป็นจริง	

ตารางที่ 11 : แสดง Use Case Scenario ของงาน “จัดการข้อมูลอาหาร”

Use Case Scenario(3)

Use case Name	จัดการข้อมูลพนักงาน	
Area	ระบบการจัดการร้านค้าแฟ	
Actor(s)	เจ้าของร้าน (Employer)	
Description	เพิ่ม ลบ หรือปรับปรุง ข้อมูลของพนักงาน	
Triggering Event	เจ้าของร้านต้องการเพิ่ม ลบ หรือปรับปรุงข้อมูลของพนักงาน	
Step Performed (main path)	Information for steps	
เปิดใช้งานระบบ	Show UI : LoginForm	
กรอก Username และ Password และกดปุ่ม “Login”	Show UI : LoginForm, {Call checkLogin} Check from Account Table : AccUserName+AccPassword+AccPosition	
แสดงหน้าจอสำหรับเจ้าของร้าน	Show UI : OwnerForm	
เลือกงานที่ต้องการ (เพิ่ม ลบ ปรับปรุง ข้อมูลพนักงาน)	ShowUI : OwnerForm If(เพิ่มข้อมูลพนักงาน) {call addEmployee} Else if(ลบข้อมูลพนักงาน) {call delEmployee} Else if(ปรับปรุงข้อมูลพนักงาน) {call updateEmployee}	
จัดการข้อมูลพนักงาน	Manage Employee Table : EmpNO+EmpName+EmpAddress+EmpTel	
Preconditions	มีรับสมัครพนักงานใหม่, ลาออก หรือพนักงานต้องการเปลี่ยนแปลงข้อมูล	
Postconditions	ข้อมูลพนักงานมีการเปลี่ยนแปลง	
Assumptions	ไม่มีข้อมูลของพนักงานในระบบหรือข้อมูลไม่ตรงกับความเป็นจริง	

ตารางที่ 12 : แสดง Use Case Scenario ของงาน “จัดการข้อมูลพนักงาน”

Use Case Scenario (4)

Use case Name	จัดการบัญชีผู้ใช้
Area	ระบบการจัดการร้านกาแฟ
Actor(s)	เจ้าของร้าน (Employer)
Description	เพิ่ม ลบ หรือปรับปรุงข้อมูลบัญชีผู้ใช้
Triggering Event	เจ้าของร้านต้องการเพิ่ม ลบ หรือปรับปรุงข้อมูลบัญชีผู้ใช้
Step Performed (main path)	Information for steps
เปิดใช้งานระบบ	Show UI : LoginForm
กรอก Username และ Password และกดปุ่ม “Login”	Show UI : LoginForm, {Call checkLogin} Check from Account Table : AccUserName+AccPassword+AccPosition
แสดงหน้าจอสำหรับเจ้าของร้าน	Show UI : OwnerForm
เลือกงานที่ต้องการ (เพิ่ม ลบ ปรับปรุง บัญชีผู้ใช้)	ShowUI : OwnerForm If(เพิ่มบัญชีผู้ใช้) {call addAccount} Else if(ลบบัญชีผู้ใช้) {call delAccount } Else if(ปรับปรุงบัญชีผู้ใช้) {call updateAccount }
จัดการข้อมูลบัญชีผู้ใช้	Manage Account Table : AccNO+AccUserName+ AccPassword+AccPosition
Preconditions	มีรับสมัครพนักงานใหม่, ลาออก, พนักงานต้องการเปลี่ยนแปลงข้อมูล, หรือจำนวนโต๊ะมีการเปลี่ยนแปลง
Postconditions	ข้อมูลบัญชีผู้ใช้มีการเปลี่ยนแปลง
Assumptions	ไม่มีข้อมูลบัญชีของพนักงานในระบบหรือพนักงานต้องการเปลี่ยนแปลงรหัสในการเข้าระบบ

ตารางที่ 13 : แสดง Use Case Scenario ของงาน “จัดการบัญชีผู้ใช้”

Use Case Scenario (5)

Use case Name	ใช้เครื่องคิดเลข
Area	ระบบการจัดการร้านกาแฟ
Actor(s)	ลูกค้า (Customer)
Description	ลูกค้าสามารถใช้เครื่องคิดเลขเพื่อคำนวณค่าใช้จ่ายหรืองานอื่นๆ
Triggering Event	ลูกค้าต้องการทราบค่าใช้จ่ายทั้งหมด, ลูกค้าต้องการหาค่าเฉลี่ยของค่าใช้จ่ายทั้งหมดตามจำนวนคนที่ต้องการ
Step Performed (main path)	Information for steps
เปิดใช้งานระบบ	Show UI : LoginForm
กรอก Username และ Password และกดปุ่ม “Login”	Show UI : LoginForm, {Call checkLogin} Check from Account Table : AccUserName+AccPassword+AccPosition
แสดงหน้าจอสำหรับลูกค้า	Show UI : CustomerForm
กดปุ่ม “เครื่องคิดเลข”	Show UI : CalculatorForm
กรอกตัวเลขเพื่อคำนวณ	Show UI : CalculatorForm {Call calculate}
Preconditions	ลูกค้าต้องการทราบค่าใช้จ่ายทั้งหมด, ลูกค้าต้องการหาค่าเฉลี่ยของค่าใช้จ่ายทั้งหมดตามจำนวนคนที่ต้องการ
Postconditions	ลูกค้าทราบผลการคำนวณ
Assumptions	มีการสั่งอาหารจำนวนมากและต้องการเฉลี่ยค่าใช้จ่ายตามจำนวนคน

ตารางที่ 14 : แสดง Use Case Scenario ของงาน “ใช้เครื่องคิดเลข”

Use Case Scenario (6)

Use case Name	เรียกชำระค่าบริการ	
Area	ระบบการจัดการร้านกาแฟ	
Actor(s)	ลูกค้า (Customer)	
Description	ลูกค้ากดปุ่มเรียกพนักงานเพื่อชำระค่าบริการ	
Triggering Event	ลูกค้าไม่ต้องการสั่งอาหารเพิ่มอีก	
Step Performed (main path)	Information for steps	
เปิดใช้งานระบบ	Show UI : LoginForm	
กรอก Username และ Password และกดปุ่ม “Login”	Show UI : LoginForm, {Call checkLogin} Check from Account Table : AccUserName+AccPassword+AccPosition	
แสดงหน้าจอสำหรับลูกค้า	Show UI : CustomerForm	
กดปุ่ม “ชำระค่าบริการ”	Show UI : PaymentForm	
แสดงข้อมูลค่าบริการทั้งหมดและกรอกจำนวนเงินที่ต้องการจะจ่าย	Show UI : PaymentForm	
กดยืนยันเพื่อดำเนินการต่อและจะแสดงรายละเอียดเกี่ยวกับค่าบริการทั้งหมดและเงินทอนหรือกดยกเลิกเพื่อยกเลิกการเรียกชำระค่าบริการ	Show UI : PayConfirmForm {Call paymentCall} Change RecPayCall in Receipt Table From 0 to 1	
แสดงหน้าจอขอบคุณลูกค้า	Show UI : ThankForm	
Preconditions	ลูกค้าต้องการทราบค่าใช้จ่ายทั้งหมด, ลูกค้าต้องการหาค่าเฉลี่ยของค่าใช้จ่ายทั้งหมดตามจำนวนคนที่ต้องการ	
Postconditions	ลูกค้าทราบผลการคำนวณ	
Assumptions	มีการสั่งอาหารจำนวนมากและต้องการเฉลี่ยค่าใช้จ่ายตามจำนวนคน	

ตารางที่ 15 : แสดง Use Case Scenario ของงาน “เรียกชำระค่าบริการ”

Use Case Scenario (7)

Use case Name	สั่งอาหาร
Area	ระบบการจัดการร้านกาแฟ
Actor(s)	ลูกค้า (Customer)
Description	ลูกค้าสั่งอาหารจากหน้าจอแสดงรายการอาหาร
Triggering Event	ลูกค้าต้องการสั่งอาหาร
Step Performed (main path)	Information for steps
เปิดใช้งานระบบ	Show UI : LoginForm
กรอก Username และ Password และกดปุ่ม “Login”	Show UI : LoginForm, {Call checkLogin} Check from Account Table : AccUserName+AccPassword+AccPosition
แสดงหน้าจอสำหรับลูกค้า	Show UI : CustomerForm
กดปุ่ม “ซื้อ” หลังรายการอาหาร	Show UI : OrderForm
เลือกตัวเลือกเสริมเช่น หวานน้อย ปกติ เป็นต้น และกรอกจำนวนที่จะสั่งซื้อ	Show UI : OrderForm
กดยืนยันเพื่อสั่งซื้อหรือกดยกเลิกเพื่อยกเลิกการสั่งซื้อ	Show UI : OrderForm {Call orderProduct}
กลับไปหน้าจอสำหรับลูกค้า	Show UI : CustomerForm
Preconditions	ลูกค้าต้องการนั่งที่โต๊ะและต้องการสั่งอาหาร
Postconditions	ลูกค้าสั่งซื้ออาหารสำเร็จและรอรับอาหาร
Assumptions	ลูกค้าใหม่กำลังเดินไปนั่งที่โต๊ะและจะสั่งอาหาร


ตารางที่ 16 : แสดง Use Case Scenario ของงาน “สั่งอาหาร”

Use Case Scenario (8)

Use case Name	จัดการรายการอาหารที่ลูกค้าสั่ง	
Area	ระบบการจัดการร้านกาแฟ	
Actor(s)	เจ้าของร้าน (Employer) พนักงาน (Employee)	
Description	แสดงรายการอาหารที่ลูกค้าสั่งโดยเรียงลำดับจากสั่งก่อนไปจนถึงสั่ง หลังสุด และสามารถจัดการหรือระบุว่ารายการแรกสุดทำเสร็จแล้ว รายการถัดไปก็จะขึ้นมาแทนที่ในตำแหน่งแรกทันที	
Triggering Event	ลูกค้าสั่งอาหาร	
Step Performed (main path)	Information for steps	
เปิดใช้งานระบบ	Show UI : LoginForm	
กรอก Username และ Password และกดปุ่ม “Login”	Show UI : LoginForm, {Call checkLogin} Check from Account Table : AccUserName+AccPassword+AccPosition	
แสดงหน้าจอสำหรับพนักงาน/เจ้าของร้าน	Show UI : EmployeeForm / OwnerForm	
กดปุ่มแสดงรายการอาหารที่ลูกค้าสั่ง	Show UI : OrderedForm,{Call orderedQueue} //ถ้า OrdStatus เป็น 0 คือรายการอาหารที่ยังไม่ ดำเนินการ จะนำมาแสดงในรายการอาหารที่ลูกค้าสั่ง ถ้าดำเนินการแล้วจะเปลี่ยนเป็น 1	
แสดงรายการอาหารที่ลูกค้าสั่งทั้งหมด	Show UI : OrderedForm	
กดปุ่ม “เสร็จสิ้น” ในแถวของรายการอาหาร ที่ทำเสร็จแล้ว	Show UI : OrderedForm {Call confirmOrdered},{Call orderedQueue}	
แสดงรายการอาหารที่ลูกค้าสั่งทั้งหมด	Show UI : OrderedForm,{Call orderedQueue}	
Preconditions	ลูกค้าสั่งอาหาร	
Postconditions	ลูกค้าได้รับอาหารที่สั่ง	
Assumptions	ลูกค้ายืนยันการสั่งซื้ออาหาร	

ตารางที่ 17 : แสดง Use Case Scenario ของงาน “จัดการรายการอาหารที่ลูกค้าสั่ง”

Use Case Scenario (9)

<u>Use case Name</u>	รับชำระค่าบริการ	
<u>Area</u>	ระบบการจัดการร้านกาแฟ	
<u>Actor(s)</u>	เจ้าของร้าน (Employer) พนักงาน (Employee)	
<u>Description</u>	แสดงหมายเลข โต๊ะที่ลูกค้าเรียกชำระค่าบริการ	
<u>Triggering Event</u>	ลูกค้ากดเรียกชำระค่าบริการ	
<u>Step Performed (main path)</u>		<u>Information for steps</u>
เปิดใช้งานระบบ		Show UI : LoginForm
กรอก Username และ Password และกดปุ่ม “Login”		Show UI : LoginForm, {Call checkLogin} Check from Account Table : AccUserName+AccPassword+AccPosition
แสดงหน้าจอสำหรับพนักงาน/เจ้าของร้าน		Show UI : EmployeeForm / OwnerForm
แจ้งเตือนเมื่อมีลูกค้าเรียกชำระค่าบริการ		Show UI : OrderedForm , callPaymentForm  {Call checkCallPayment}
//ฟังก์ชันทำงานอัตโนมัติและทำงานใหม่ทุกๆ 5 วินาที โดยตรวจสอบจาก RecPayCall และ RecPayStatus ถ้า RecPayCall มีเลขกำกับเป็น 0 คือยังไม่เรียกชำระค่าบริการ เลข1 คือเรียกชำระค่าบริการแล้ว ส่วน RecPayStatus ไว้ตรวจสอบการชำระค่าบริการ เลข 0 คือยังไม่ชำระค่าบริการ เลข 1 คือชำระค่าบริการแล้ว		
กดปุ่มจัดการการชำระเงิน		Show UI : managePayForm
แสดงการแจ้งเตือนก่อนหน้านี้ทั้งหมดและกดยืนยันในโต๊ะที่มีการชำระเงินแล้ว		Show UI : managePayForm {Call confirmPayment} //RecPayStatus=1
<u>Preconditions</u>	ลูกค้ากดปุ่มเรียกชำระเงิน	
<u>Postconditions</u>	ลูกค้าชำระค่าบริการและได้รับเงินทอน	
<u>Assumptions</u>	ลูกค้าไม่สั่งอาหารเพิ่มอีก	

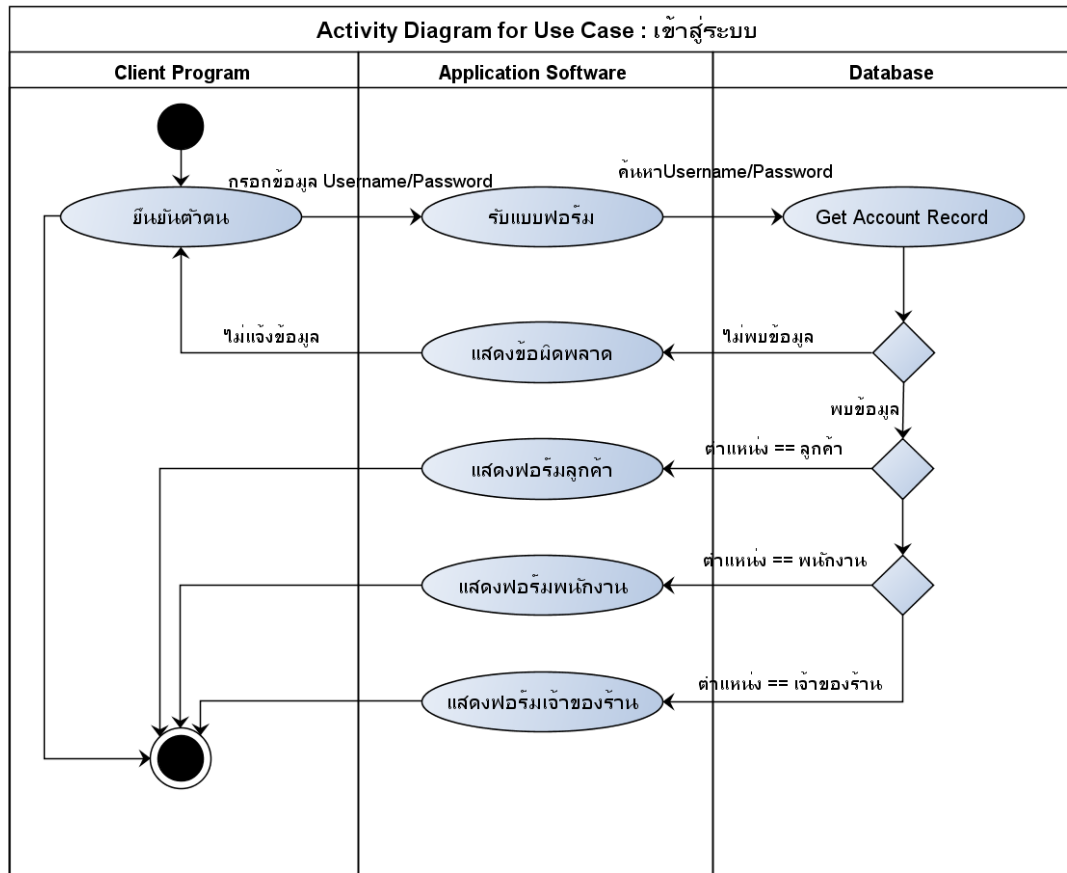
ตารางที่ 18 : แสดง Use Case Scenario ของงาน “รับชำระค่าบริการ”

Use Case Scenario (10)

Use case Name	ออกจากระบบ
Area	ระบบการจัดการร้านกาแฟ
Actor(s)	เจ้าของร้าน (Employer) พนักงาน (Employee)
Description	ออกจากระบบเมื่อใช้งานระบบเสร็จแล้ว
Triggering Event	หมดเวลาให้บริการหรือต้องการออกจากระบบ
Step Performed (main path)	Information for steps
เปิดใช้งานระบบ	Show UI : LoginForm
กรอก Username และ Password และกดปุ่ม “Login”	Show UI : LoginForm, {Call checkLogin} Check from Account Table : AccUserName+AccPassword+AccPosition
แสดงหน้าจอสำหรับพนักงานหรือเจ้าของร้าน	Show UI : EmployeeForm / OwnerForm
กดปุ่ม “ออกจากระบบ”	Show UI : EmployeeForm / OwnerForm {Call checkLogout}
แสดงหน้าจอสำหรับเข้าสู่ระบบ	Show UI : managePayForm
Preconditions	เข้าสู่ระบบและใช้งานระบบเสร็จสิ้นแล้ว
Postconditions	ออกจากระบบสำเร็จ
Assumptions	หมดเวลาให้บริการหรือต้องการออกจากระบบ

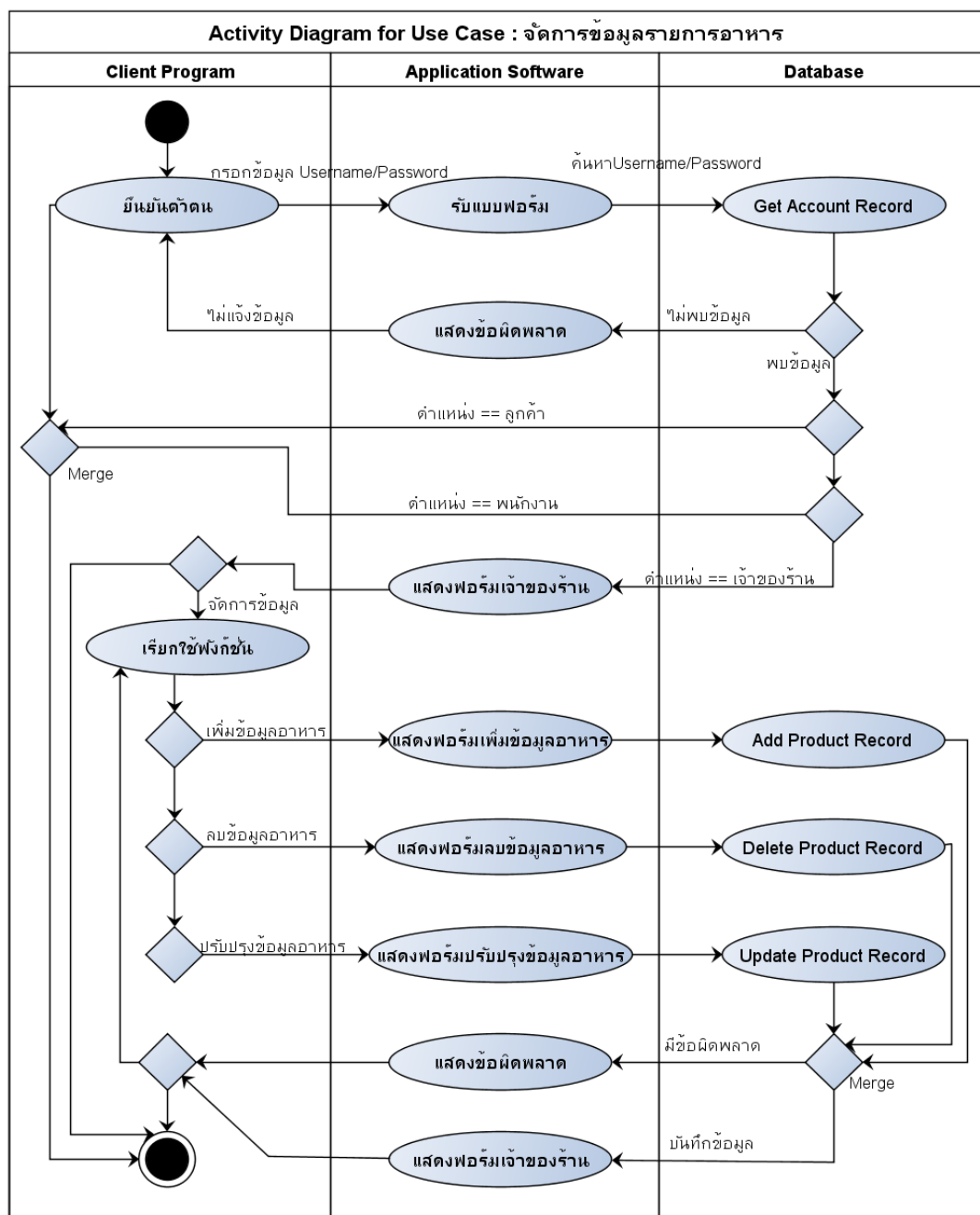
ตารางที่ 19 : แสดง Use Case Scenario ของงาน “ออกจากระบบ”

Activity Diagram for Use Case



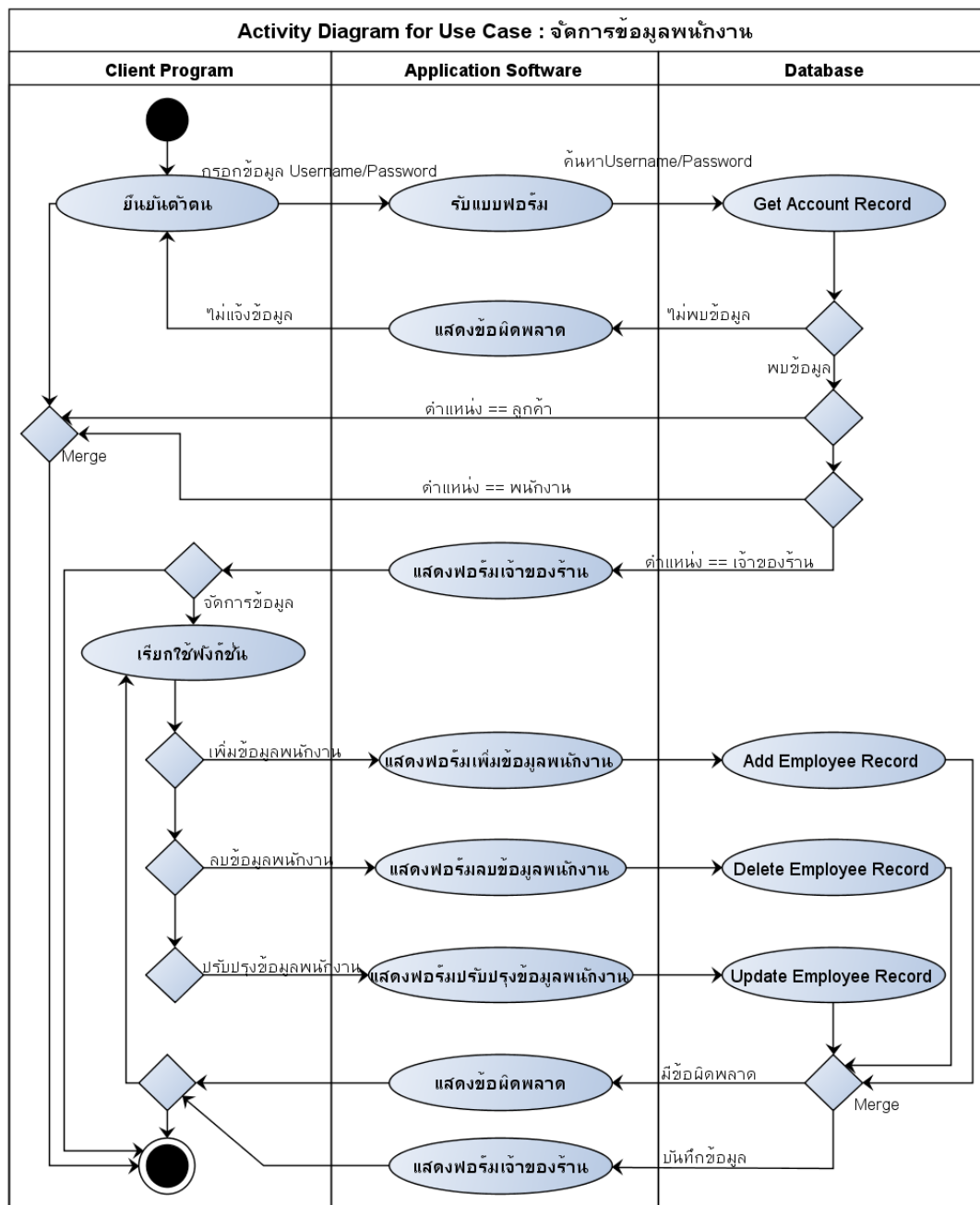
รูปที่ 11 : แสดง Activity Diagram สำหรับ Use Case เข้าสู่ระบบ

Activity Diagram for Use Case (2)



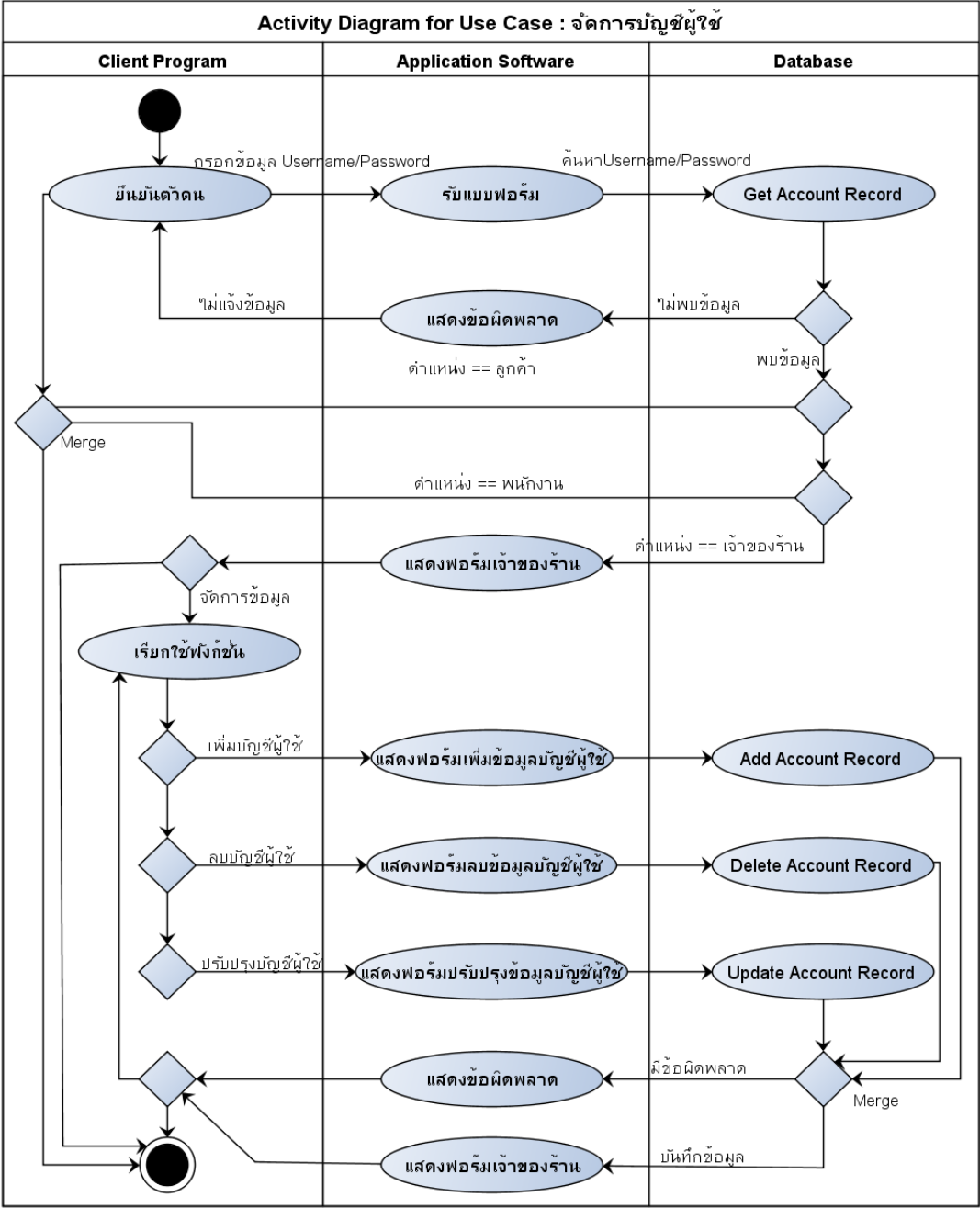
รูปที่ 12 : แสดง Activity Diagram สำหรับ Use Case จัดการข้อมูลรายการอาหาร

Activity Diagram for Use Case (3)



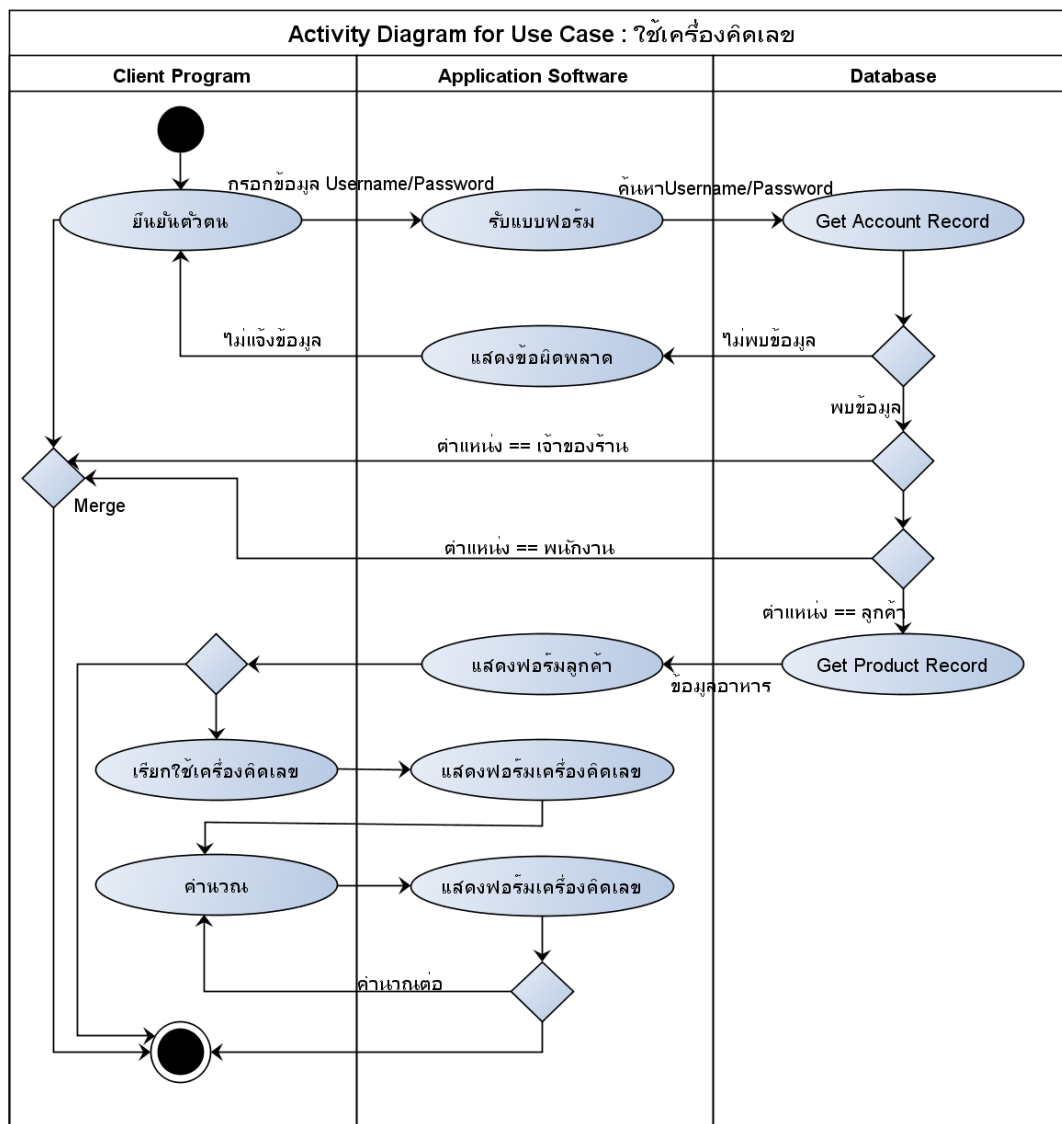
รูปที่ 13 : แสดง Activity Diagram สำหรับ Use Case จัดการข้อมูลพนักงาน

Activity Diagram for Use Case (4)



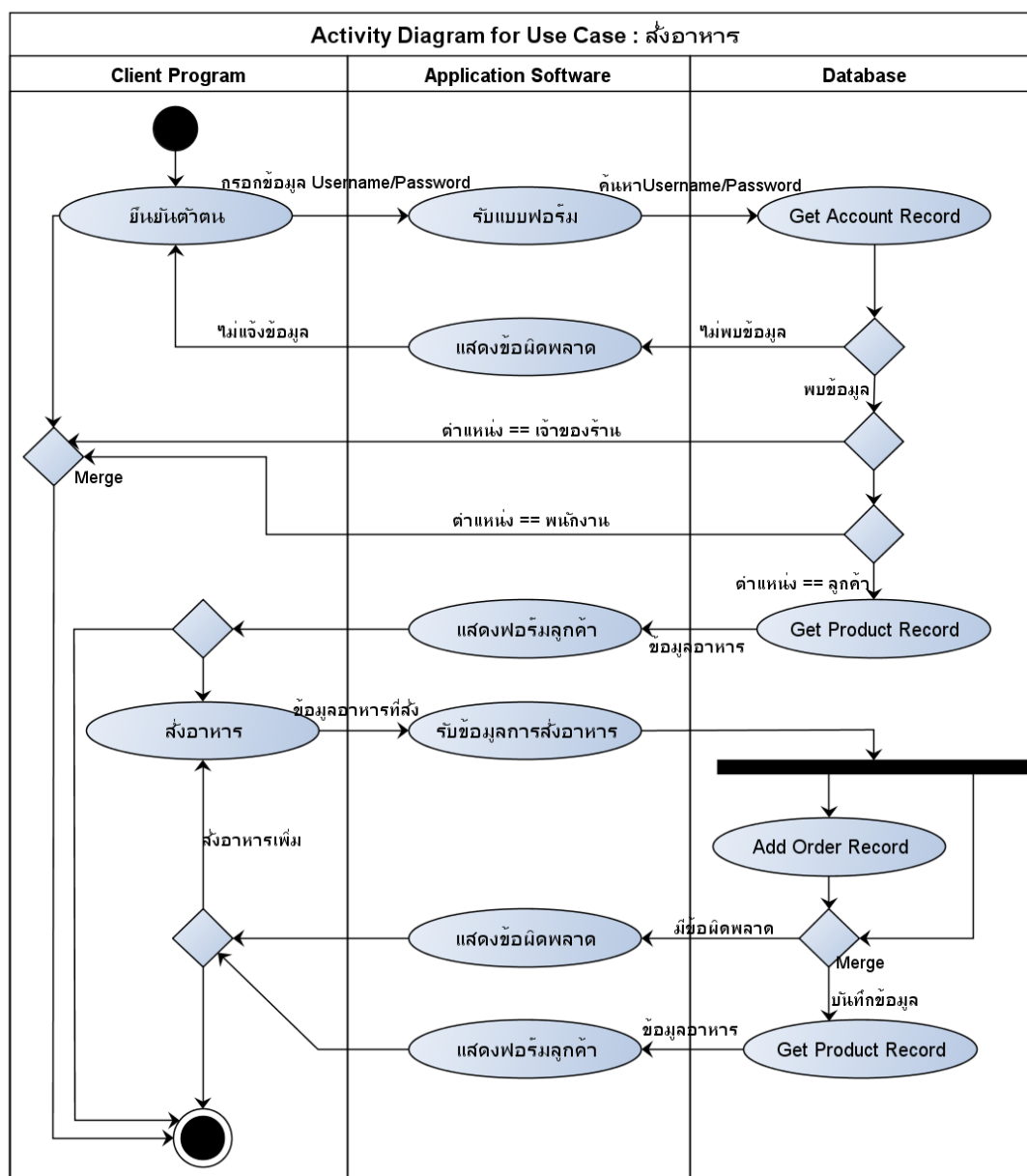
รูปที่ 14 : แสดง Activity Diagram สำหรับ Use Case จัดการบัญชีผู้เช่า

Activity Diagram for Use Case (5)



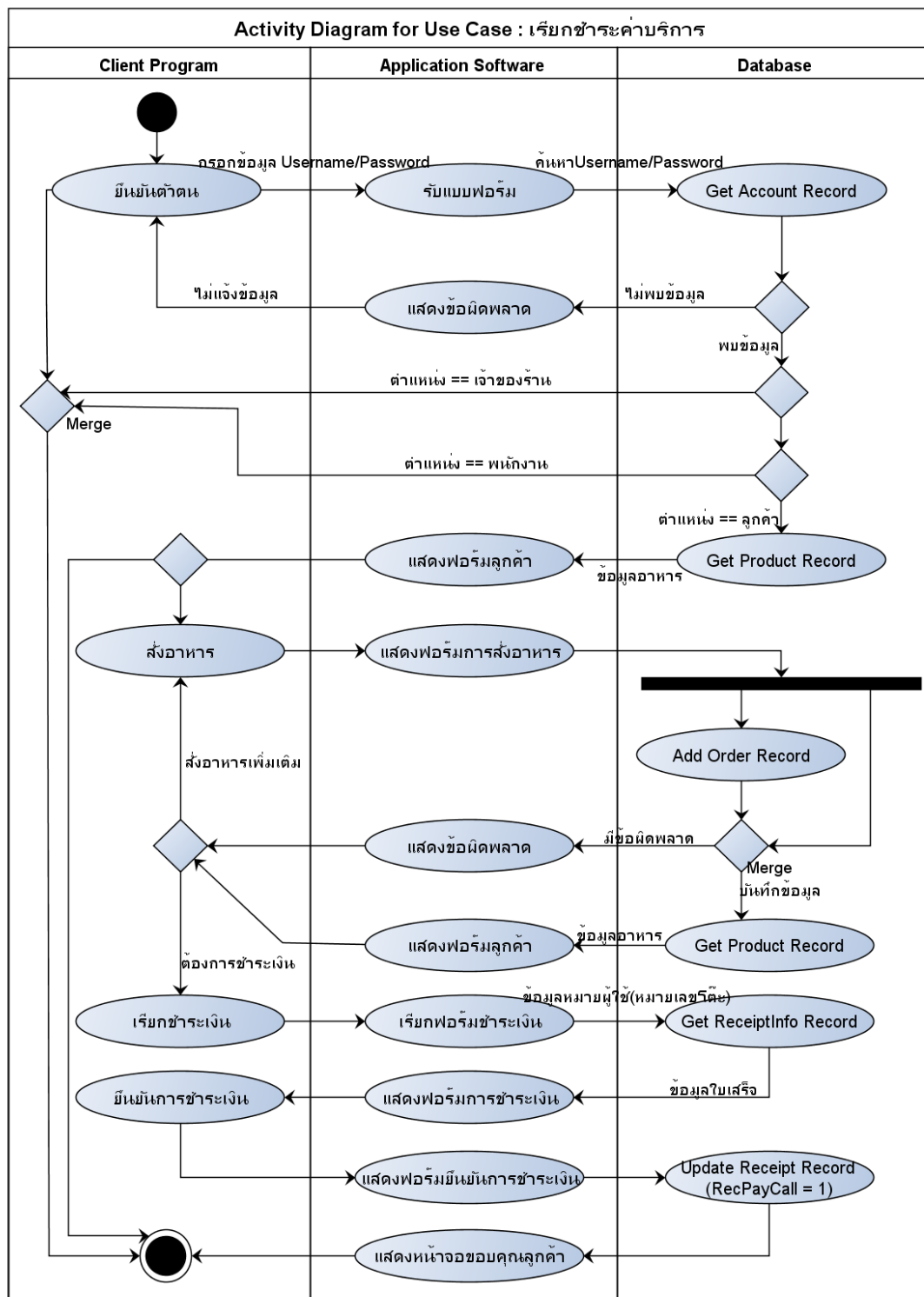
รูปที่ 15 : แสดง Activity Diagram สำหรับ Use Case ใช้เครื่องคิดเลข

Activity Diagram for Use Case (6)

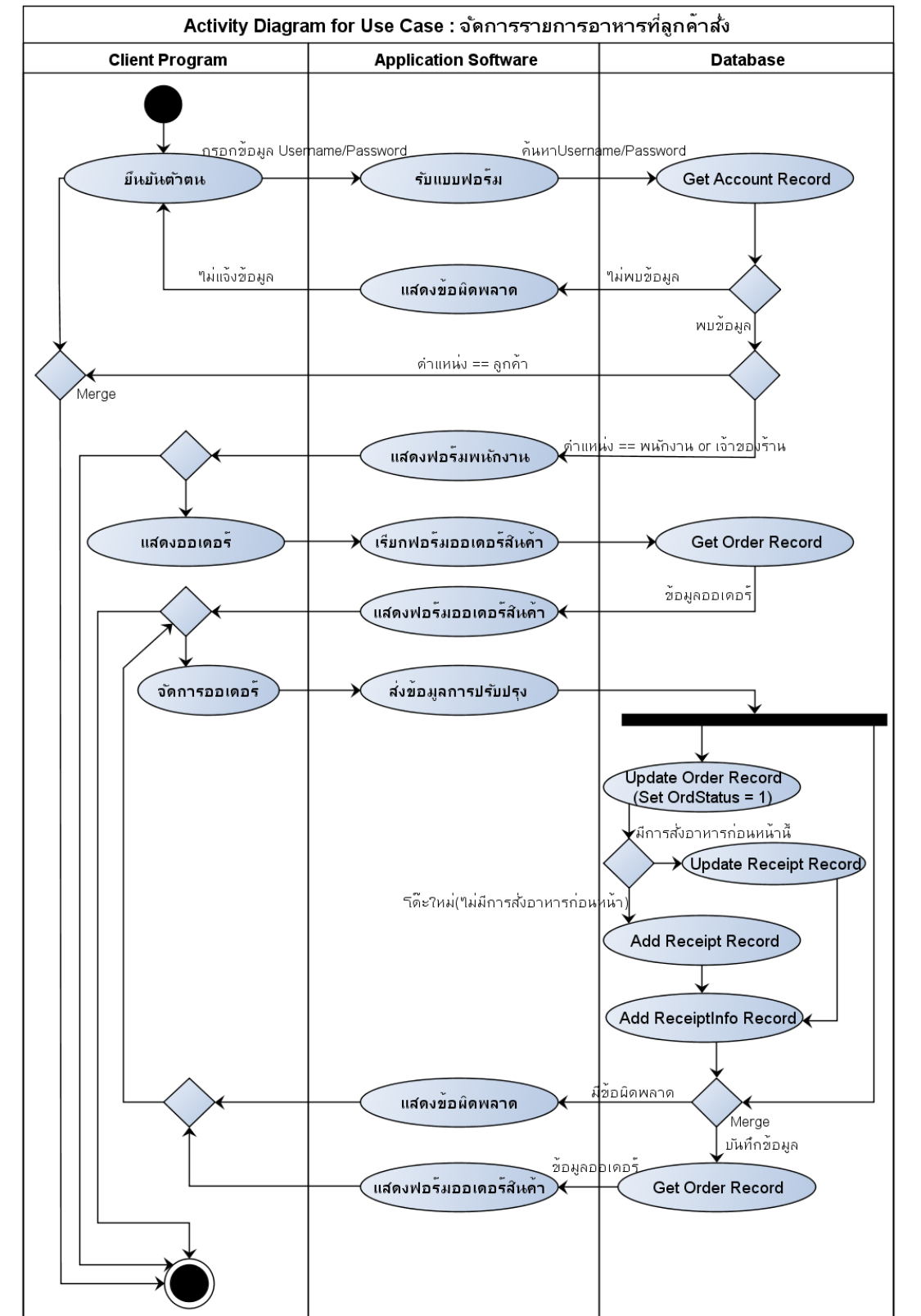


รูปที่ 16 : แสดง Activity Diagram สำหรับ Use Case สั่งอาหาร

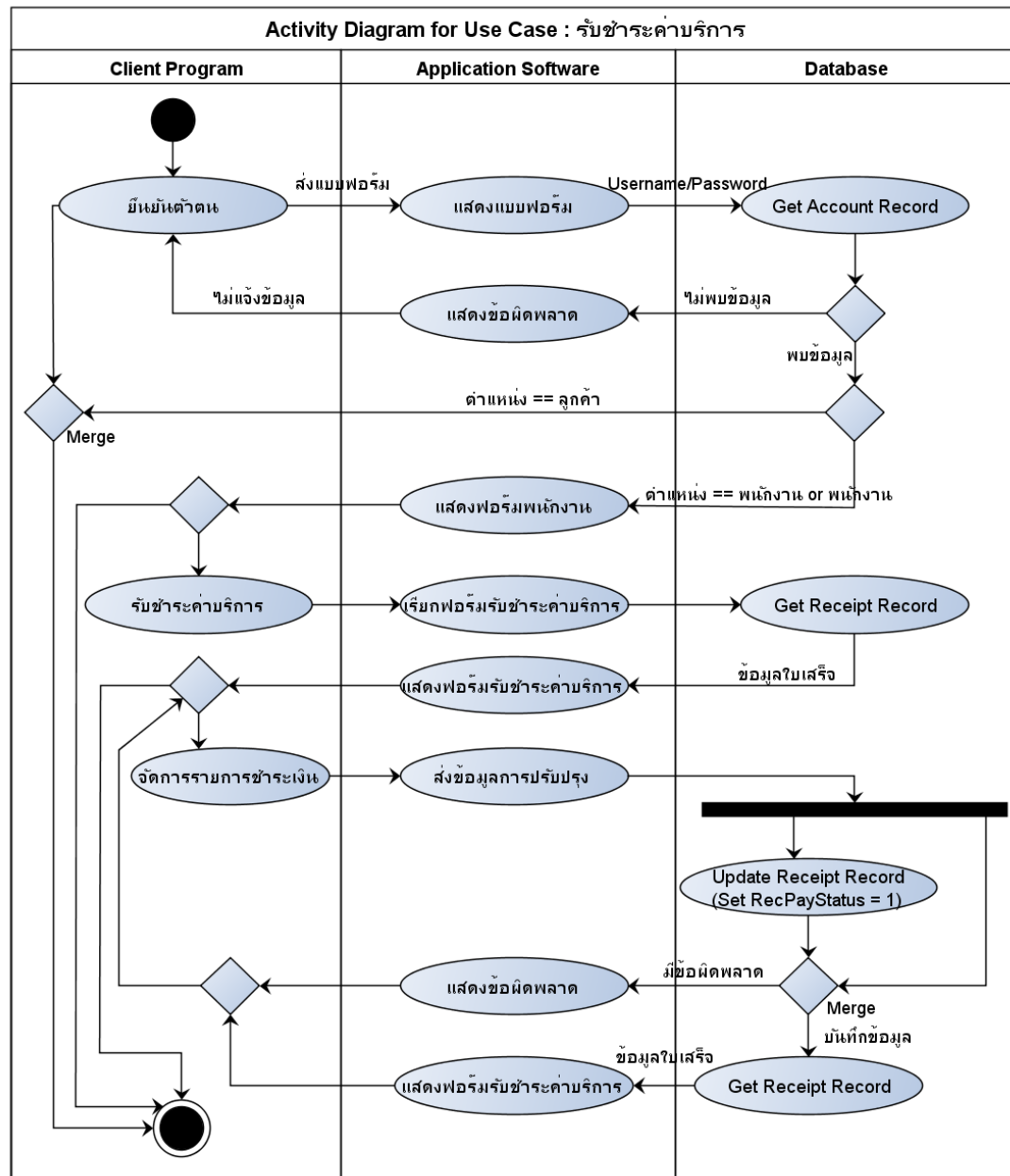
Activity Diagram for Use Case (7)



รูปที่ 17 : แสดง Activity Diagram สำหรับ Use Case เรียกชำระค่าบริการ

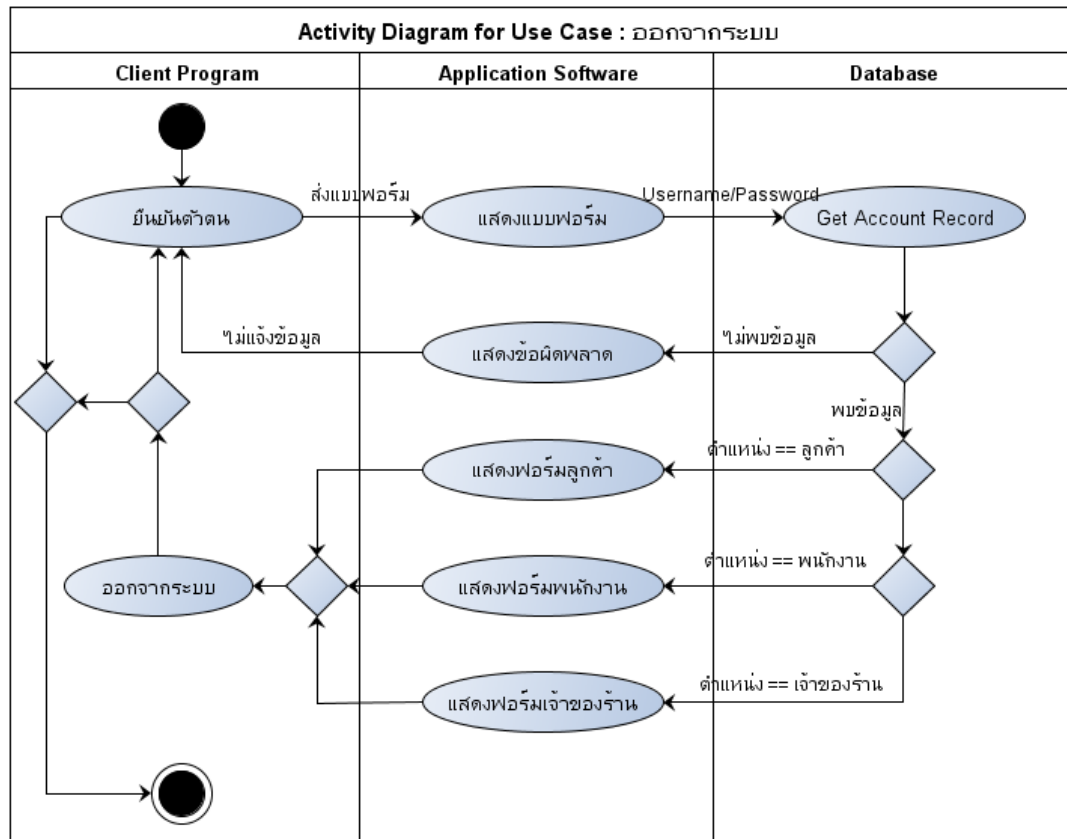


Activity Diagram for Use Case (9)



รูปที่ 19 : แสดง Activity Diagram สำหรับ Use Case รับชำระค่าบริการ

Activity Diagram for Use Case (10)



รูปที่ 20 : แสดง Activity Diagram สำหรับ Use Case ออกจากระบบ

Interface Design

ร้าน CuisineVision
Open 10.00 - 18.00 น.

Account :
Password :

Clear Login

Clear Login

Address : PSU, Hatyai. Tel 088-123-4567 Email : psucoffee@psu.ac.th

รูปที่ 21 : แสดงหน้า Login Form

Log out

ร้าน CuisineVision

ชื่อ : cus1
ขณะนี้กำลังดำเนินการในลำดับที่ 12

Product	Price	Amount
Coffee	45.00	50
Cookie	15.00	125

Order

รายการอาหารที่สั่งซื้อ

Product	Price	Amount
---------	-------	--------

Calculator

Payment

รูปที่ 22 : แสดงหน้า Customer Form



Product Name Coffee

Stock(s) 50 Cup(s)

Price 45.00 Baht

Order Amount Low Sugar < 0 > Cup(s)

 Medium < 0 > Cup(s)


 Sweet < 0 > Cup(s)

Total Amount 0 Cup(s)

Total Price 0 Baht

รูปที่ 23 : แสดงหน้า Order Form

AdminForm

 CUISNEVISION

Product Employee Account

Add

Delete

Edit

Searching **Select Row To Edit

Product Type All

Keyword Search

Product ID	Product Name	Type	Price	Amount
1	Coffee	Drink	45.00	50
2	Cookie	Food	15.00	125

Product Type Product Price

Product Name Product Amount Update

รูปที่ 24 : แสดงหน้าจอการจัดการสำหรับเจ้าของร้าน

อ้างอิง

ไมโครซอฟท์ วิชาการสตูดิโอ. 2557. [ออนไลน์]. สืบค้นจาก: http://th.wikipedia.org/wiki/ไมโครซอฟท์_วิชาการสตูดิโอ (วันที่สืบค้น 15 ธันวาคม 2557).

แนะนำโปรแกรม VISUAL STUDIO 2010 EXPRESS EDITION. 2553. [ออนไลน์]. สืบค้นจาก: <http://www.bananacode.net/archives/7> (วันที่สืบค้น 15 ธันวาคม 2557).

.NET Framework คืออะไร มีที่มาและความสำคัญอย่างไร. 2554. [ออนไลน์]. สืบค้นจาก: <http://notebookspec.com/net-framework-คืออะไร-มีที่มาและความสำคัญอย่างไร/88056/> (วันที่สืบค้น 15 ธันวาคม 2557).

ภาษาC#. [ออนไลน์]. สืบค้นจาก: <http://tonkung.ueuo.com/index.html> (วันที่สืบค้น 15 ธันวาคม 2557).

ลำดับขั้นตอนการสร้างโปรแกรมบน .NET FRAMEWORK. 2553. [ออนไลน์]. สืบค้นจาก: <http://www.bananacode.net/archives/74> (วันที่สืบค้น 15 ธันวาคม 2557).