



การพัฒนาแอปพลิเคชันแจ้งเตือนผู้ใช้
กรณีศึกษา บริษัท โกซอฟท์ (ประเทศไทย) จำกัด
THE DEVELOPMENT OF USER NOTIFICATION APPLICATION
ON iOS PLATFORM
CASE STUDY : GOSOFT (THAILAND) CO., LTD.

นายพงศร์พี อัครณัย

TNI

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีไทย-ญี่ปุ่น

พ.ศ.2554

การพัฒนาแอปพลิเคชันแจ้งเตือนผู้ใช้
กรณีศึกษา บริษัท โกซอฟท์ (ประเทศไทย) จำกัด
THE DEVELOPMENT OF USER NOTIFICATION APPLICATION
ON iOS PLATFORM
CASE STUDY : GOSOFT (THAILAND) CO., LTD.

นายพงศ์พี อัครธัญย์

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีไทย – ญี่ปุ่น

พ.ศ. 2554

คณะกรรมการสอบ

..... ประธานกรรมการสอบ

(อาจารย์ วรวิทย์ จิตขจรวานิช)

..... กรรมการสอบและอาจารย์ที่ปรึกษา

(อาจารย์ ดร. วิมล แสนอ้อม)

..... กรรมการสอบ

(อาจารย์ ปรีวัตร คงกำเนิด)

ลิขสิทธิ์ของสถาบันเทคโนโลยีไทยญี่ปุ่น

บทสรุป

ชื่อโครงการ	การพัฒนาแอปพลิเคชันแจ้งเตือนผู้ใช้ กรณีศึกษา บริษัท โกซอฟท์ (ประเทศไทย) จำกัด พ.ศ. 2554 THE DEVELOPMENT OF USER NOTIFICATION APPLICATION ON iOS PLATFORM Case study: gosoft (Thailand) Co., Ltd. D.C.2011
ผู้เขียน	นายพงศ์พี อัครธัญย์
คณะวิชา	วิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร.วิมล แสนอ้อม
พนักงานที่ปรึกษา	คุณอรุรรา โรจนวาส (Head, Business Development) คุณอรุณวรรณ นาคารย์ (Leader, Business Development) คุณเฉลิมชัย สถาบันรักษ์วงศ์ (Software Engineering)
ชื่อบริษัท	gosoft (Thailand) Co., Ltd.
ประเภทธุรกิจ/สินค้า	ให้คำปรึกษาและบริการด้านสารสนเทศ
งานที่ปฏิบัติ	

รูปแบบของการทำงานคืองานพัฒนาแอปพลิเคชันบน iOS โดยเน้นพัฒนาแอปพลิเคชัน บน iPhone/iPod มีหัวข้อในการทำงานคือ แอปพลิเคชัน ที่แจ้งเตือนผู้ใช้เกี่ยวกับ Missing Timesheet จาก ระบบ RPM (Rational Portfolio Manager) ที่มาจากฝ่าย PMO (Project Management Office), ข้อมูลการ ขาดงาน จากระบบ HR (Human Resource), และข้อมูล Aging SR(Non-Project) จากระบบ LNW (Lotus Note Workflow) โดยให้ผู้ใช้ทำการเข้าสู่ระบบและยืนยันตัวตนผ่าน แอปพลิเคชันเพื่อรับข้อมูล การแจ้งเตือนจากฐานข้อมูลในเซิร์ฟเวอร์

ในกระบวนการทำงานได้เริ่มต้นด้วยการประชุมโครงการนัดกรรมเพื่อวางแผนงานและออกแบบ แผนผังการทำงาน จัดทำใบเสนอโครงการเพื่อส่งเข้าประกวดโครงการนัดกรรมรอบที่ 1 และหน้าที่ที่ ได้รับมอบหมายมาคือออกแบบส่วนติดต่อกับผู้ใช้ (User Interface) ในการใช้งานตามความต้องการ (Requirement) ที่ได้รับมา จากนั้นแยกกันเขียนโปรแกรมในส่วนของอินเทอร์เน็ตเฟส และ ส่วนของ

ฐานข้อมูลและนำมาประกอบกัน แบ่งการทำงานออกเป็นเฟส (Phase) เพื่อเป็นการสำรองข้อมูลและพัฒนาแอปพลิเคชันให้มีความสมบูรณ์ขึ้นตามลำดับ

การทำงานแต่ละขั้นตอนจะมีการเพิ่มเติมและปรับเปลี่ยนความต้องการบางครั้งจึงต้องมีการปรับเปลี่ยนบางอย่างซึ่งจะต้องผ่านการประชุมวิเคราะห์ว่าหากปรับเปลี่ยนไปเป็นแบบที่กล่าวมาจะต้องปรับเปลี่ยนการดำเนินงานอย่างไรเพื่อให้การทำงานเป็นไปได้อย่างลุล่วง และ ผลของการปรับเปลี่ยนไปนั้นทำให้แอปพลิเคชันง่ายต่อการใช้งานมากขึ้น หรือ มีเสถียรภาพมากขึ้นเพียงใด

ผลที่ได้รับจากการดำเนินงานและประโยชน์ที่ได้รับ

จากการดำเนินงานภายในระยะเวลาช่วงที่ผ่านมา ทำให้ได้รับประสบการณ์ในการทำงานในชีวิตจริงเช่นการทำงานร่วมกับบุคคลอื่นที่ต้องเขียนโปรแกรมให้ไปในลักษณะเดียวกันเพื่อที่จะนำมาประกอบกันได้, การทำงานให้สำเร็จลุล่วงภายในเวลาที่กำหนดตามแผนงานที่ตั้งไว้ ซึ่งจะต้องมีความรับผิดชอบต่องานที่ได้รับ มอบหมาย, หากประสบปัญหาในการทำงานควรจะปรึกษาที่ปรึกษาโครงการอย่างไร หรือ นำเรื่องเสนอเข้าที่ประชุมเพื่อหาแนวทางแก้ไขอย่างไรให้ถูกต้องและเหมาะสมที่สุด รวมไปถึงการปฏิบัติตัวภายในองค์กรที่ต้องมีความเหมาะสม, การแต่งกาย, คำพูด ท่าทาง กริยา, การวางสิ่งของในขอบเขตปฏิบัติงานตามมาตรฐาน 7ส, กฎเกณฑ์และ กฎระเบียบในการใช้สถานที่และสิ่งของของบริษัท, มีความสัมมาคารวะกับผู้อาวุโสกว่า และประสบการณ์อื่น ๆ อีกมากมาย เป็นต้น

กิตติกรรมประกาศ

ในช่วงระยะเวลาสี่เดือนที่ข้าพเจ้าได้ปฏิบัติงานสหกิจศึกษา ณ บริษัทโกซอฟท์ (ประเทศไทย) จำกัด ข้าพเจ้าได้รับการดูแล แนะนำในเรื่องต่าง ๆ ทั้งที่เกี่ยวกับงานและไม่เกี่ยวกับงาน ทำให้การปฏิบัติสหกิจศึกษาของข้าพเจ้าครั้งนี้สำเร็จไปได้ด้วยดี อีกทั้ง ประสบการณ์ต่าง ๆ ที่ได้รับการแนะนำมา ถือเป็นประสบการณ์จริงอันหายากยิ่งที่หาไม่ได้จากการเรียนในห้องเรียน และประสบการณ์ต่าง ๆ เหล่านั้นสามารถนำไปประยุกต์ใช้ในชีวิตได้อย่างมาก

การจัดทำโครงการครั้งนี้จะสำเร็จลุล่วงไปไม่ได้ หากขาดการดูแลจากบุคคลที่เกี่ยวข้องทุกฝ่าย ขอขอบคุณ คุณอรุณ วิจารณ์วาศ (Head Business Development), คุณอรุณวรรณ นามารักษ์ (Leader Business Development) และ คุณเฉลิมชัย สถาปนิกษ์วงศ์ (Software Engineer) ที่ให้คอยให้คำปรึกษา และแนะนำแนวทางในการปฏิบัติงาน การเขียนแอปพลิเคชันเบื้องต้น, การวิเคราะห์กระบวนการทำงานเชิงอัลกอริทึม, การปฏิบัติตัวภายในองค์กร, การปฏิบัติงานพื้นฐาน อาทิเช่น การนำเสนองาน, การเขียนเอกสารทางการ, การทำงานอย่างมีมาตรฐาน ซึ่งทั้งหมดนี้ทำให้ตลอดการทำงาน ของข้าพเจ้า เป็นไปอย่างราบรื่นและเต็มไปด้วยประสบการณ์มากมาย จึงขอขอบคุณบริษัท โกซอฟท์ (ประเทศไทย) จำกัด ที่ให้โอกาสข้าพเจ้าได้มาปฏิบัติสหกิจศึกษา ณ ที่นี้ด้วย

พงศวีร์พี อัครธัญย์

ผู้จัดทำรายงาน

	สารบัญ	หน้า
บทสรุป		ข
กิตติกรรมประกาศ		ง
สารบัญ		จ
รายการตาราง		ช
รายการรูปประกอบ		ณ
บทที่		
1. บทนำ		1
1.1 ชื่อและที่ตั้งของสถานประกอบการ		1
1.2 ลักษณะธุรกิจของสถานประกอบการ หรือการให้บริการหลักขององค์กร		2
1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร		3
1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย		4
1.5 พนักงานที่ปรึกษา และ ตำแหน่งของพนักงานที่ปรึกษา		4
1.6 ระยะเวลาที่ปฏิบัติงาน		4
1.7 วัตถุประสงค์หรือจุดมุ่งหมายของการปฏิบัติงานหรือ โครงการที่ได้รับมอบหมายให้ปฏิบัติงานสหกิจศึกษา		5
1.8 ผลที่คาดว่าจะได้รับการปฏิบัติงานหรือ โครงการที่ได้รับมอบหมาย		5
2. ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน		6
2.1 ซอฟต์แวร์ที่ใช้ในการพัฒนาแอปพลิเคชัน		6
2.2 ฮาร์ดแวร์ที่ใช้ในการพัฒนาแอปพลิเคชัน		6
2.3 ภาษาอ็อบเจกทีฟ-ซี		6
2.3.1 @interface		7
2.3.2 @implementation		8

2.4 การเขียนโปรแกรมเชิงวัตถุ - Object-oriented programming (OOP)	10
--	----

สารบัญ(ต่อ)

บทที่	หน้า
2.5 Graphic User Interface (GUI)	10
2.6 การใช้โปรแกรม Xcode ในการเขียนโปรแกรม	12
2.6.1 การสร้างโปรเจกใหม่	12
2.6.2 การเพิ่มเฟรมเวิร์ค	13
2.6.3 พื้นที่ทำงานบน Xcode	14
2.6.4 การใช้ NeXT Interface Builder	15
2.6.5 การประกาศแบบ วิธี Drag & Drop	17
2.7 พื้นฐานการใช้งานภาษา ภาษาอ็อบเจกทีฟ-ซี ภายใน X-Code เบื้องต้น	21
2.7.1 การนำเข้าเฟรมเวิร์คเข้ามาในไฟล์	21
2.7.2 การสร้างคลาสและการประกาศตัวแปร, ฟังก์ชันของคลาส	21
2.7.3 การเรียกใช้ฟังก์ชัน เมธอด	23
2.7.4 การนำไฟล์ที่อยู่ภายในไลบรารีเดียวกันกับโปรเจกต์ เข้ามาในไฟล์ปัจจุบัน	23
2.7.5 การประกาศชื่อคลาส	23
2.7.6 การสร้างคลาสเป็นชนิดที่สร้างขึ้นเองภายในโปรเจกต์	23
2.7.7 การจองพื้นที่ในหน่วยความจำ และการคืนค่าหน่วยความจำ พื้นฐาน	24
2.7.8 การกำหนดคุณสมบัติต่าง ๆ ให้กับตัวแปรที่เราสร้างขึ้น	25
2.8 หลักการใช้งาน Delegate เพื่อเข้าถึง อินเทอร์เน็ต ทั้งหมดภายในโปรเจกต์	25
2.8.1 วิธีการใช้งาน Delegate	25
2.8.2 ตัวอย่างการใช้งาน	26
2.8.3 ขั้นตอนการทำงานของแอปพลิเคชันตัวอย่าง	30
2.9 UI Class (User Interface Class) พื้นฐานภายในโปรแกรม Xcode	30
2.9.1 UINavigationController & UINavigationController	31
2.9.2 UIViewController & UIView	33
2.9.3 UITableViewController & UITableView	36
2.10 การเขียนภาษา PHP เป็นตัวกลางในการสืบค้นฐานข้อมูล	40
2.11 การเขียนภาษา C#.net เพื่อให้เป็นตัวกลางแทน PHP ในการสืบค้นฐานข้อมูล	43

สารบัญ(ต่อ)

บทที่	หน้า
3. บทที่แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน	46
3.1 แผนการปฏิบัติงาน	46
3.2 รายละเอียดงานที่นักศึกษาปฏิบัติในงานสหกิจศึกษา หรือรายละเอียดโครงการที่ได้รับมอบหมาย	46
3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงานหรือโครงการ	47
3.3.1 วางแผนการดำเนินงาน	47
3.3.2 จัดทำใบเสนองาน (Proposal)	47
3.3.3 นำเสนอโครงการ (รอบแรก)	47
3.3.4 การออกแบบ (Design)	48
3.3.5 การพัฒนา (Develop)	48
3.3.6 การทดสอบระบบ (Testing)	53
4. ผลการดำเนินงาน การวิเคราะห์และสรุปผลต่าง ๆ	54
4.1 ขั้นตอนและผลการดำเนินงาน	54
4.2 ผลการวิเคราะห์ข้อมูล	67
4.3 วิเคราะห์และวิจารณ์ข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์ และจุดมุ่งหมายในการปฏิบัติงานหรือการจัดทำโครงการ	67
5. บทสรุปและข้อเสนอแนะ	68
5.1 สรุปผลการดำเนินงาน	68
5.2 แนวทางการแก้ไขปัญหา	68
5.3 ข้อเสนอแนะจากการดำเนินงาน	69
เอกสารอ้างอิง	70
ประวัติผู้วิจัย	72

รายการตาราง

ตาราง

หน้า

3.1 ตารางแสดงแผนการปฏิบัติงาน

46



รายการรูปประกอบ

รูป		หน้า
1.1	สถานที่ตั้ง บริษัท โกซอฟท์ (ประเทศไทย) จำกัด	1
2.1	การสร้างโปรเจกใหม่	12
2.2	การตั้งชื่อโปรเจก และชื่อผู้ผลิต	13
2.3	การเพิ่มเฟรมเวิร์ค เข้าไปในโปรเจก	14
2.4	พื้นที่การทำงานของโปรแกรม Xcode	14
2.5	หน้า Interface Builder	16
2.6	การแทรกตาราง	16
2.7	File's Owner Property	17
2.8	การกำหนดคุณสมบัติให้วัตถุ	17
2.9	File Owner Property	17
2.10	การเรียกหน้าต่างแยก	18
2.11	การลากโดยใช้ Assistance Editor	18
2.12	การลาก Object ไปที่ ไฟล์ เพื่อกำหนด Property	19
2.13	การตั้งชื่อตัวแปรของวัตถุ	19
2.14	การประกาศ Property แบบ drag & drop ของ Object หน้า .h	20
2.15	การประกาศ Property แบบ drag & drop ของ Object หน้า .m	20
2.16	หน้า AppDelegate.xib	27
2.17	หน้า RootViewController	28
2.18	หน้า OtherViewController	29
2.19	ขั้นตอนการทำงานของแอปพลิเคชันตัวอย่าง	30
2.20	พื้นฐานภายในโปรแกรม Xcode	30
2.21	หน้า UINavigationController	31
2.22	หน้า UIViewController	33
2.23	การแสดง UITableViewController	36
2.24	Custom Cell	38
3.1	ผังการทำงานของ Navigation Controller	48

3.2	Sequence Diagram ของระบบยืนยันตัวตนแบบมีการเชื่อมต่อ	49
3.3	การแสดงคอนโทรลเลอร์ที่เป็นแถบเครื่องมือ	51
3.4	XML ที่เซิร์ฟเวอร์ส่งกลับมา	52
3.5	ซอร์สโค้ด (Source Code) ของ XML ที่เซิร์ฟเวอร์ส่งกลับมา	52
4.1	ขั้นตอนผลการดำเนินงาน	54
4.2	การแจ้งเตือนเมื่อไม่มีการเชื่อมต่ออินเทอร์เน็ต	55
4.3	หน้าจอสำหรับเข้าสู่ระบบ	56
4.4	เมนูหลักของการทำงาน	57
4.5	การยืนยันตัวตนเมื่อมีการเชื่อมต่ออินเทอร์เน็ต	58
4.6	การเข้าสู่ระบบโดยไม่มีการเชื่อมต่ออินเทอร์เน็ต	59
4.7	รูปแสดงการแจ้งเตือนเมื่อทำการกดรีเฟรช	60
4.8	การแสดงผลข้อมูลของระบบ RPM	61
4.9	การแสดงผลข้อมูลของระบบ HR	62
4.10	การแสดงผลข้อมูลของระบบ LNW	63
4.11	การตั้งเวลาให้แจ้งเตือนบันทึกของผู้ใช้	64
4.12	การใส่บันทึกลงไปในแต่ละหมวด	65
4.13	การแสดงผลข้อมูลที่บันทึกไว้ผ่านการแจ้งเตือน	66



1.2 ลักษณะธุรกิจของสถานประกอบการ หรือการให้บริการหลักขององค์กร

1.2.1 บทย่อ

บริษัทโกซอฟท์ (ประเทศไทย) จำกัด หรือ gosoft (Thailand) Co., Ltd. เป็นกลุ่มบริษัทในกลุ่มธุรกิจของบริษัท ซี.พี. เซเวนอีเลฟเวน จำกัด (มหาชน) ซึ่งปัจจุบันคือบริษัท ซีพี ออลล์ จำกัด (มหาชน) จัดตั้งเป็นบริษัทเมื่อวันที่ 1 มกราคม 2546 ก่อตั้งขึ้นด้วยการนำองค์กรด้าน IT ที่มีประสบการณ์หลายสายงานภายในกลุ่มให้เป็นหนึ่งเดียวกันจึงทำให้บริษัทเป็นแหล่งความรู้ (Know How) และเทคโนโลยีที่หลากหลายด้าน Hardware , Software และ Solution ต่าง ๆ

วัตถุประสงค์ของบริษัทฯคือเป็นบริษัทชั้นนำในการให้คำปรึกษาและบริการด้านสารสนเทศ (Information System) ในประเทศไทย ซึ่งให้บริการครอบคลุมถึงการพัฒนาโปรแกรมใช้งานภายในธุรกิจ (In-house Application Development) , ติดตั้งระบบโปรแกรมชั้นนำ (System Integrator) , ระบบเครือข่าย (Networking) , ระบบรักษาความปลอดภัยข้อมูล (Security System) และอื่น ๆ

นอกจากนี้บริษัทฯ ยังมีศูนย์ข้อมูล Data Center และศูนย์ Call Service ให้บริการแก่ลูกค้าด้วย ส่วนของ Data Center เป็นศูนย์ที่มีมาตรฐานระดับสูง Tier IV ซึ่งมีระบบสำรองในจุดที่มีความเสี่ยงต่าง ๆ ไว้เพื่อให้ความมั่นใจถึงความปลอดภัย และความเสถียร ของข้อมูล พื้นที่ตั้งของศูนย์แยกอย่างอิสระจากพื้นที่ประกอบธุรกิจ (Isolated Area) เพื่อลดความเสี่ยงในการเกิดเหตุการณ์ต่าง ๆ ในส่วนของสำนักงาน Call Service เป็นศูนย์บริการลูกค้าในเรื่องต่าง ๆ (Helpdesk), บริการลูกค้าสัมพันธ์ (CRM), แจ้งเรื่อง (Call Dispatch) และอื่น ๆ ให้บริการลักษณะ 7 วัน หรือ 24 ชั่วโมง

1.2.2 วิสัยทัศน์ของบริษัท

“เราให้บริการเทคโนโลยีสารสนเทศ และ คอนแทคเซ็นเตอร์ อย่างมีคุณภาพ”

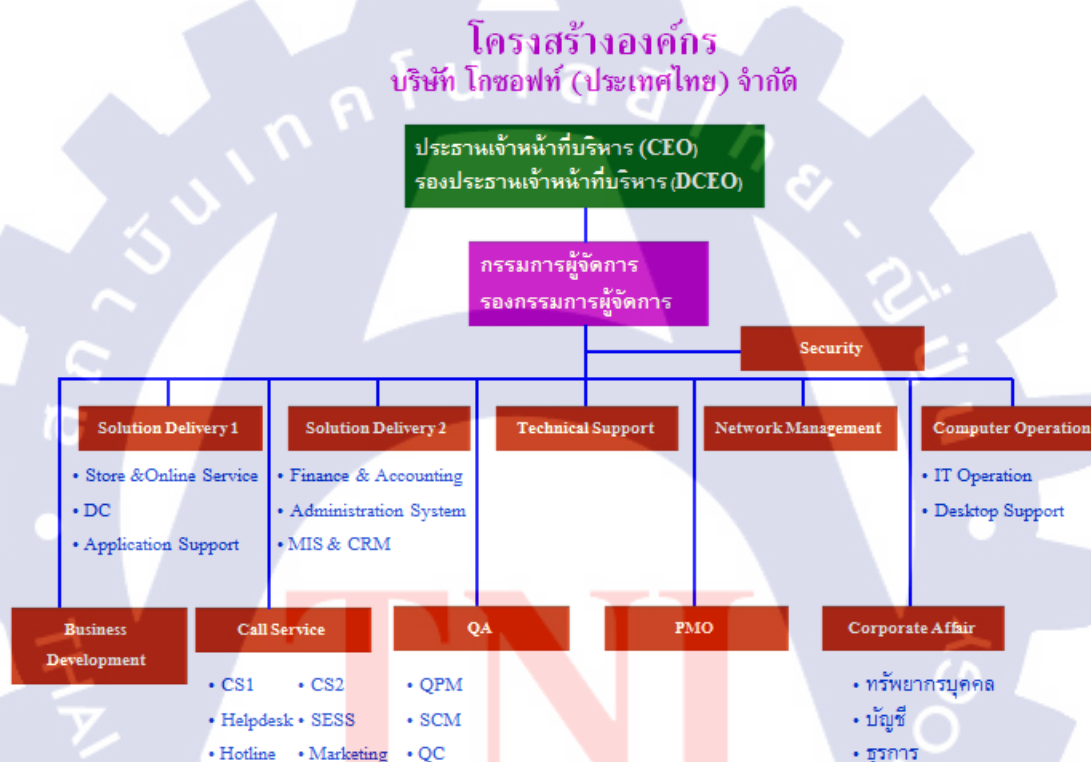
1.2.3 ภารกิจของบริษัท

- สร้างผลการดำเนินงานให้เติบโตและมีผลกำไรที่ยั่งยืน
- ส่งมอบและให้บริการระบบงานเทคโนโลยีสารสนเทศ และ คอนแทคเซ็นเตอร์ อย่างมีคุณภาพ
- ค้นคว้าวิจัย พัฒนาเพื่อนำเทคโนโลยีมาประยุกต์ใช้ในธุรกิจของลูกค้า
- มีส่วนร่วมในการช่วยเหลือชุมชนและสังคม

- พัฒนาระบบงาน การทำงานที่มีคุณภาพอย่างต่อเนื่องทั่วทั้งองค์กร
- พัฒนาขีดความสามารถของบุคลากรอย่างต่อเนื่อง เพื่อมุ่งสู่องค์กรแห่งการเรียนรู้
- ส่งเสริมให้บุคลากรมีคุณภาพชีวิตในการทำงานที่ดี และมีความผูกพันต่อองค์กร

1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร

1.3.1 แผนผังองค์กร



รูปที่ 1.2 แผนผังองค์กร

1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย

1.4.1 ตำแหน่ง Trainee

1.4.2 หน้าที่งานที่ได้รับมอบหมาย

1.ออกแบบส่วนติดต่อกับผู้ใช้ (Interface) และ สร้างแอปพลิเคชันที่ทำงานบน iOS โดยสามารถทำงานในเรื่องต่าง ๆ เช่นการแจ้งเตือน, การติดต่อฐานข้อมูลผ่านตัวกลางติดต่อที่เขียนขึ้นมาด้วย C#.net และมีระบบยืนยันตัวตนเป็นต้น ซึ่งโปรแกรมสามารถแจ้งเตือนผู้ใช้เมื่อมีเหตุการณ์ต่าง ๆ เช่น เวลางานขาด ก็จะแจ้งเตือนให้ผู้ใช้ได้รับทราบ

2.สร้าง XML Parser เพื่อทำการติดต่อฐานข้อมูลหลักของบริษัท เพื่อดึงเอาข้อมูลที่ระบบ Human Resource, ฝ่าย Project Manager Office, ระบบ Lotus Note Workflow จัดไว้มาแสดง และ แจ้งเตือนผู้ใช้งานผ่านทางแอปพลิเคชัน

1.5 พนักงานที่ปรึกษา และตำแหน่งของพนักงานที่ปรึกษา

ชื่อ : นางสาวอรรวรา โรจนวาส

ตำแหน่ง : Head Business Development

ชื่อ : นางอรุณวรรณ นาคารย์

ตำแหน่ง : Leader Business Development

ชื่อ : นายเฉลิมชัย สถานุรักษ์วงศ์

ตำแหน่ง : Software Engineering

1.6 ระยะเวลาที่ปฏิบัติงาน

เริ่มต้นการปฏิบัติงานสหกิจศึกษาวันที่ 3 มิถุนายน 2554 และสิ้นสุดการปฏิบัติงานสหกิจศึกษาวันที่ 30 กันยายน 2554

1.7 วัตถุประสงค์ของการปฏิบัติงาน

1. เพื่อเพิ่มความสะดวกให้กับพนักงานในการจัดการ Timesheet และ SR
2. เพื่อป้องกันปัญหาการลืม หรือ ไม่มีเวลาเช็ค E-Mail
3. เพื่อให้กระบวนการทำงานรวดเร็วขึ้น
4. สามารถนำไปต่อยอดเพื่อแจ้งเตือนสิ่งอื่น ๆ ได้

1.8 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย

1. ได้รับความรู้ความสามารถในการเขียนโปรแกรมด้วยภาษา Objective-C
2. ได้รับความรู้ความสามารถในการเขียนโปรแกรมด้วยภาษา C#
3. ได้รับความรู้ความสามารถในการเขียน Web service เพื่อติดต่อ Database
4. ได้รับความรู้ในการทดสอบการทำงานโดยระบบมาตรฐาน CMMI
5. ได้รับความรู้ในการจัดทำมาตรฐาน Thailand Quality Award (TQA)
6. ได้รับความรู้ในการจัดทำโครงการเกี่ยวกับ Corporate Social Responsibility (CSR)
7. ได้รับความรู้ในการเขียนวาระการประชุม Minute of Meeting (MOM)
8. ได้พัฒนากระบวนการคิดและวิเคราะห์เชิงการเขียนโปรแกรม
9. ได้พัฒนาทักษะการทำงานร่วมกันเป็นทีม
10. ได้พัฒนาทักษะการนำเสนองาน

บทที่ 2

ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

2.1 ซอฟต์แวร์ที่ใช้ในการพัฒนาแอปพลิเคชัน

- OS X (Snow Leopard, Lion), iOS(4.3), Window 7
- โปรแกรม Xcode (Version 4.0, 4.1)
- Web browser Firefox SQLite Manager Add-on
- โปรแกรม Visual Studio 2010

2.2 ฮาร์ดแวร์ที่ใช้ในการพัฒนาแอปพลิเคชัน

- iMac, iPhone, iPad
- Notebook

2.3 ภาษาอ็อบเจกทีฟ-ซี

เป็นภาษาโปรแกรมเชิงวัตถุ โดยแรกเริ่มภาษา ภาษาอ็อบเจกทีฟ-ซี พัฒนาขึ้นจากภาษาC โดยยังคงคุณลักษณะของภาษาC ไว้ครบทุกประการเพียงแต่เพิ่มระบบส่งข้อความ (messaging) แบบเดียวกับภาษา Small Talk เข้าไปเท่านั้น (ภาษาอ็อบเจกทีฟ-ซี runtime) ปัจจุบันภาษา ภาษาอ็อบเจกทีฟ-ซี มีคุณสมบัติอื่น ๆ เพิ่มเติมจากการพัฒนาภาษา ภาษาอ็อบเจกทีฟ-ซี 2.0 โดยบริษัท Apple Inc.

ภาษา ภาษาอ็อบเจกทีฟ-ซี เป็นชั้นบาง ๆ บน C และเป็น สตริคต์ซูเปอร์เซตของ C ดังนั้นคอมไพเลอร์ภาษา ภาษาอ็อบเจกทีฟ-ซี จึงสามารถคอมไพล์ โปรแกรมภาษา C ได้ ๆ ก็ได้ ภาษา ภาษาอ็อบเจกทีฟ-ซี ได้รับรูปแบบการเขียนมาจากภาษาC และภาษาSmall Talk โดยรูปแบบส่วนใหญ่ (preprocessing, expressions, การประกาศฟังก์ชัน และการเรียกฟังก์ชัน) มาจากภาษาC ขณะที่ส่วนที่เป็นการจัดการเชิงวัตถุมาจากSmall Talk

ภาษาภาษาอ็อบเจกทีฟ-ซี ได้เพิ่มเติมรูปแบบการเขียนโปรแกรมเพื่อรองรับการออกแบบโปรแกรมเชิงวัตถุ โดยจะใช้การส่งเมสเสจไปยังอ็อบเจกต์ต่าง ๆ เช่นเดียวกับ Small Talk ซึ่งแตกต่างจากภาษาในตระกูล Simula (เช่น C++) ข้อแตกต่างนี้มีความสำคัญ เพราะภาษาภาษาอ็อบเจกทีฟ-ซี จะไม่เรียกเมธอด แต่จะส่ง เมสเสจ

ในภาษาอ็อบเจกทีฟ-ซี ถ้ามีอ็อบเจกต์ หนึ่งชื่อ obj โดยคลาส มีเมธอด ชื่อ doSomething หมายความว่า obj respond หรือตอบสนองต่อ message doSomething และถ้าเราต้องการจะส่งเมสเสจ doSomething ไปยัง obj เราจะเขียนคำสั่งดังนี้

```
[obj doSomething];
```

ขณะที่ถ้าเป็น C++ เราจะเขียนว่า

```
obj.doSomething ();
```

ในภาษาภาษาอ็อบเจกทีฟ-ซี ส่วนอินเทอร์เฟซ และอิมพลีเมนต์เทชั่นจะถูกแยกออกจากกัน ในทางปฏิบัติ เรามักเก็บส่วน อินเทอร์เฟซ ไว้ในแฟ้ม .h และส่วนอิมพลีเมนต์เทชั่นใน .m

2.3.1 @interface

เรามักนิยามส่วนอินเทอร์เฟซของคลาสในแฟ้ม .h โดยทั่วไปเรามักตั้งชื่อแฟ้มนี้ให้ตรงหรือสอดคล้องกับชื่อของคลาส เช่นถ้าคลาสเราชื่อ RootViewController เราก็มักจะประกาศอินเทอร์เฟซของคลาส RootViewController ในแฟ้ม RootViewController.h โดยรูปแบบของการประกาศอินเทอร์เฟซมีลักษณะดังนี้:

```
@interface className : superclass name {
    instance variables
}
+ classMethod1;
```

```

+ (return_type) classMethod2;
+ (return_type) classMethod3: (param1_type) parameter_varName;
- (return_type) instanceMethod1: (param1_type) param1_varName : (param2_type)
  param2_varName;
- (return_type) instanceMethod2WithParameter: (param1_type) param1_varName
  andOtherParameter: (param2_type) param2_varName;
@end

```

เมธอดแบบอินสแตนต์จะถูกนำหน้าด้วยเครื่องหมายลบ "-" ส่วนเมธอด แบบคลาส จะถูกนำหน้าโดยเครื่องหมายบวก "+" ตรงนี้จะแตกต่างจาก UML diagrams ซึ่งใช้ในแสดงว่าเมธอด เป็นแบบ private หรือ public

ค่าที่ถูกริเทิร์น จากเมธอด มีลักษณะเช่นเดียวกับในภาษาซี เช่น void, int, ฯลฯ โดยจะมีการประกาศประเภท ชื่อ id ไว้แทนอินสแตนต์อ็อบเจกต์ อะไรก็ได้

เราประกาศพารามิเตอร์ของเมธอด ด้วยเครื่องหมายทวิภาคหรือโคลอน ":" ตามด้วยประเภทของพารามิเตอร์ในวงเล็บ แล้วตามด้วยชื่อของพารามิเตอร์ เรามักใส่ชื่อที่มีความสอดคล้องกับพารามิเตอร์หน้าเครื่องหมายทวิภาค เพื่อบอกว่าพารามิเตอร์แต่ละตัวมีบทบาทอย่างไร

```

-(void) setRange: (int) start : (int) end;
-(void) importDocumentWithName: (NSString *) name withSpecifiedPreferences: (Preferences *)
  prefs beforePage: (int) insertPage;

```

2.3.2 @implementation

ส่วนอินเทอร์เฟซ จะประกาศแต่ต้นแบบของเมธอดเท่านั้น โดยไม่รวมถึงการกำหนดว่า เมธอดนั้นจะต้องทำอะไรบ้าง การกำหนดว่าเมธอด นั้นจะต้องทำอะไรจะทำในส่วนอิมพลีเมนต์เทชั่น ตามปกติส่วนอิมพลีเมนต์เทชั่น จะอยู่ในไฟล์นามสกุล .m ตัวอย่างเช่น Thing.m ที่มีการกำหนดส่วนอิมพลีเมนต์เทชั่น ไว้ดังนี้

```

@implementation classname
+ classMethod {
    // implementation
}
- instanceMethod {
    // implementation
}
@end

```

คอมไพเลอร์ของภาษา ภาษาอ็อบเจกทีฟ-ซี แต่ละตัวก็จะกำหนดชื่อเมธอดเป็นการภายในแตกต่างกันไป เช่นถ้าเมธอด -changeColorWithRed:green:blue: เป็นเมธอดแบบอินสแตนต์ของ คลาสColor คอมไพเลอร์ก็อาจจำเมธอด นี้ในชื่อ _i_Color_changeColorWithRed_green_blue เป็นต้น โดยตัว i จะบอกว่าจุดนี้คืออิมพลิเมนต์เทชั่นเมธอดแบบอินสแตนต์ และตามด้วยชื่อคลาสและชื่อเมธอดและมีการเปลี่ยนเครื่องหมาย : เป็น _ ดังนี้แล้ว เมื่อพารามิเตอร์เป็นส่วนหนึ่งของชื่อเมธอด ลำดับของพารามิเตอร์ในภาษา ภาษาอ็อบเจกทีฟ-ซี จึงสลับที่กันไม่ได้

อย่างไรก็ตาม เราแทบไม่ได้ใช้ชื่อภายในของฟังก์ชันโดยตรง โดยทั่วไป การส่งเมสเสจ จะถูกเปลี่ยนให้อยู่ในรูปแบบของการเรียกฟังก์ชันที่กำหนดไว้ในไลบรารี (Library) ส่วน run-time และปรกติเวลาเราส่งเมสเสจ เราก็จะไม่คำนึงถึงคลาส ที่แท้จริงของส่วนรีซิฟเวอร์(myColor) อยู่แล้ว นั่นคือการจัดการหา เมธอดจะเป็นหน้าที่ของ run-time

เป็นส่วนเพิ่มเติม syntax จากภาษา ภาษาอ็อบเจกทีฟ-ซี โดยบริษัท Apple เป็นผู้พัฒนาเพิ่มเช่น

- Garbage collection (computer science) (ทั้งนี้ ใน ภาษาอ็อบเจกทีฟ-ซี runtime ของ GNU สามารถใช้งาน Boehm-Demers-Weiser conservative garbage collector ได้ก่อนหน้านี้ ภาษาอ็อบเจกทีฟ-ซี 2.0 แล้ว)
- Properties (@property) ช่วยจัดการการประกาศตัวแปรแบบอินสแตนต์
- Fast enumeration การเพิ่มประสิทธิภาพในส่วน runtime

2.4 การเขียนโปรแกรมเชิงวัตถุ - Object-oriented programming (OOP)

คือวิธีการหนึ่งในการเขียนโปรแกรมในเชิงวัตถุ ซึ่งสามารถทำงานร่วมกัน หรือนำมารวมกันได้ โดย การแลกเปลี่ยนข่าวสาร เพื่อใช้ในการประมวลผลและส่งข่าวสารที่ได้รับไปให้วัตถุอื่น ๆ ที่เกี่ยวข้อง เพื่อทำงานต่อ ซึ่งจะให้ความสำคัญ กับ ข้อมูล (Data) และ พฤติกรรม (Behavior) ของวัตถุ และความสัมพันธ์กันระหว่างวัตถุ

2.5 Graphic User Interface (GUI)

User Interface(UI) สิ่งที่มีไว้ให้ผู้ใช้ใช้ในการกระทำกับระบบหรือสิ่งของต่าง ๆ ซึ่งอาจจะเป็น คอมพิวเตอร์ เครื่องจักร เครื่องกล อุปกรณ์ใช้ไฟฟ้าใด ๆ หรือระบบที่มีความซับซ้อนอื่น ๆ เพื่อให้สิ่ง ๆ นั้นทำงานตามความต้องการของผู้ใช้

ส่วนต่อประสานกับผู้ใช้สามารถจัดได้เป็น 2 ประเภทใหญ่ได้แก่

1. ส่วนที่นำข้อมูลเข้า หรือส่วนส่งงาน เรียกว่าอินพุต
2. ส่วนที่ใช้แสดงผลลัพธ์ หรือส่วนที่ไว้รอคำสั่งจากผู้ใช้ เรียกว่าเอาต์พุต

ตัวอย่างของส่วนต่อประสานกับผู้ใช้ เช่น เครื่องคิดเลข จะมีส่วนอินพุตคือเป็นตัวเลข 0-9 และ เครื่องหมายทางคณิตศาสตร์ ผู้ใช้จะต้องกดหมายเลขที่ต้องการคำนวณผลผ่านเป็นตัวเลขนั้น และเมื่อ เครื่องคิดเลขทำการประมวลผลเสร็จสิ้น ก็จะแสดงผลลัพธ์ออกมาบนหน้าจอ LED ซึ่งเป็นส่วนของ เอาต์พุต

แต่ในส่วนของ GUI จะเน้นไปทางกราฟฟิก ที่แสดงออกมาจาก iPhone ให้กับผู้ใช้ผ่านทาง สัญลักษณ์หรือภาพนอกเหนือจากทางตัวอักษร โดยมีหลักการเหมือนกับ UI คือมีทั้งอินพุต และ เอาต์พุต เช่น สัญลักษณ์ ภาพ และส่วนที่เพิ่มขึ้นมา คือ ไอคอน หน้าต่างการใช้งาน เมนู ปุ่มเลือก และการใช้เมาส์ หรือแม้แต่ในระบบสัมผัสหน้าจอ

ในเรื่องของส่วนต่อประสานกับผู้ใช้ มีศาสตร์หนึ่งที่เป็นส่วนระบุถึงคุณภาพของส่วนติดต่อผู้ใช้ เรียกว่า ความเหมาะสมต่อการใช้งาน หรือ Usability

เป็นศาสตร์ที่ว่าด้วยการออกแบบส่วนต่อประสานกับผู้ใช้ให้มีความเหมาะสมต่อการใช้งาน กล่าวคือสามารถใช้งานได้ง่าย สะดวก รวดเร็ว โดยศาสตร์ของความเหมาะสมต่อการใช้งาน จะมีความเกี่ยวข้องกับด้าน จิตวิทยา และ สรีรวิทยา เป็นหลัก ซึ่งจิตวิทยาจะช่วยให้นักออกแบบส่วนติดต่อผู้ใช้ สามารถคิดระบบระเบียบขั้นตอนวิธีการใช้งานที่ทำให้ผู้ใช้สามารถใช้งานอุปกรณ์ได้สะดวก และง่าย

ต่อการทำความเข้าใจ ส่วนสรีรวิทยาจะช่วยในการออกแบบให้อุปกรณ์นั้นเหมาะสมต่อกับใช้ในด้าน สรีระ เช่น ความสะดวกในการสัมผัสหน้าจอ เป็นต้น

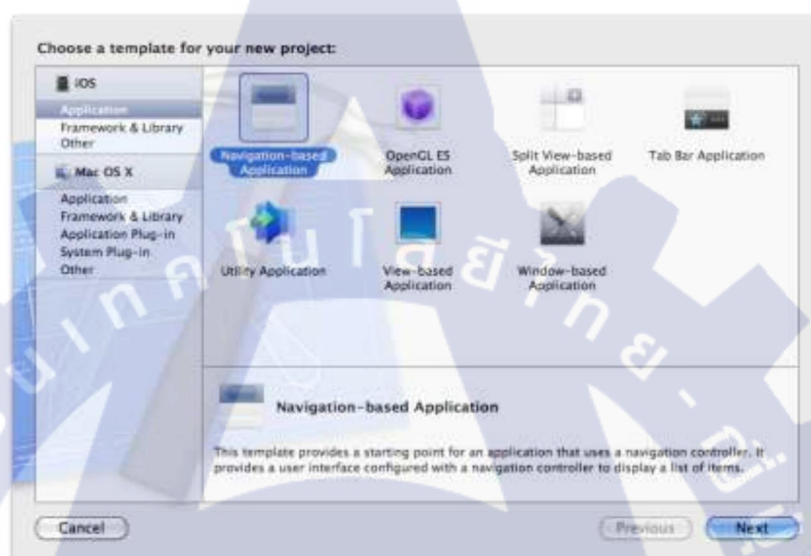
ส่วนต่อประสานกราฟิกกับผู้ใช้ ช่วยให้การเรียนรู้การใช้งานคอมพิวเตอร์ทำได้ง่ายและเร็วขึ้น เปรียบเทียบกับระบบเก่า

ที่ต้องพิมพ์ชุดคำสั่งที่ใช้ในระบบคอมพิวเตอร์หรือยูนิคซ์ ซึ่งเป็นส่วนต่อประสานผู้ใช้แบบข้อความ (Text user interface) ในบางครั้งจะเรียกว่า ส่วนต่อประสานแบบชุดคำสั่ง (Command Line Interfaces, CLI) ในการใช้งานผู้ใช้ต้องพิมพ์คำสั่งผ่านทางคีย์บอร์ด โดยคำสั่งเฉพาะต่าง ๆ ที่พิมพ์จำเป็นต้องใช้เวลา ระยะเวลาหนึ่งในการเรียนรู้ อย่างไรก็ตามในบางระบบเช่น Linux ก็ยังใช้ GUI เป็นฟรอนเอนด์เพื่อที่ทำงาน กับส่วนต่อประสานแบบชุดคำสั่ง รวมถึงการพัฒนาของ GUI ได้มีการพัฒนาไปอย่างรวดเร็ว โดยมีทั้งในส่วนของคอมเมอร์เชียล และ แบบโอเพ่นซอร์ส ซึ่งในส่วนของคอมเมอร์เชียล ได้มีบริษัท ยักษ์ใหญ่บางบริษัทเริ่มเข้าจับตลาดทางด้านนี้แล้วเช่น Sun ก็มีในส่วนของ Java Desktop เป็นต้น โดย กล่าวถึงรูปแบบการติดต่อกับผู้ใช้แบบนี้ว่า Desktop ระบบของ GUI เช่นในส่วนของการปฏิบัติการ วินโดวส์จะเป็นในลักษณะถูกสร้างขึ้นมาให้เป็นส่วนหนึ่งกับตัวระบบปฏิบัติการเลย โดยมีการเรียก ใช้ ส่วนการติดต่อกับผู้ใช้ผ่านระบบ API โดยไปเรียกส่วนประมวลผลที่ชื่อว่า GDI และมีโหมดที่เป็น ส่วนติดต่อกับผู้ใช้แบบชุดคำสั่ง แยกเป็นส่วนหนึ่งอีกต่างหากแต่ยังคงอยู่ภายใต้ระบบปฏิบัติการ (shell) ระบบ GUI ในส่วนของโอเพ่นซอร์ส ได้มีการจัดตั้งองค์กรขึ้นมาเพื่อทำการกำหนดมาตรฐานกลางที่ ใช้ในการทำงานในส่วนการพัฒนาของ Desktop ร่วมกันเช่น การกำหนดมาตรฐานของเมนู, ลักษณะส่วน ติดต่อย่อยอื่น ๆ เป็นต้น โดยองค์กรนี้มีชื่อว่า freedesktop Desktop ที่ร่วมใช้มาตรฐานเดียวกันกับ freedesktop เช่น Gnome, KDE, XFCE เป็นต้น รวมถึงมีการพัฒนาลูกเล่นต่าง ๆ ออกไปมากมาย รวมถึงการใช้งานที่หลากหลายมากขึ้นโดยลดการติดต่อระหว่าง ผู้ใช้และส่วนที่เป็นการประสาน ชุดคำสั่งลงให้มากที่สุด ในระบบปฏิบัติการใหม่ ๆ ในปัจจุบันมีการรองรับการใช้งานทั้งแบบกราฟิก และแบบข้อความ โดยแสดงผลผ่านทางกราฟิกเป็นหลักสำหรับผู้ทั่วไป และต้องใช้คำสั่งพิเศษเพื่อ เรียกใช้คำสั่งของการใช้ส่วนต่อประสานแบบข้อความ โดยแบ่งแยกออกเป็น โหมดการทำงานได้ด้วย

2.6 การใช้โปรแกรม Xcode ในการเขียนโปรแกรม

2.6.1 การสร้างโปรเจกใหม่

เปิดโปรแกรม Xcode เลือก File > New > New Project จะมีหน้าต่างดังรูปปรากฏขึ้น



รูปที่ 2.1 การสร้างโปรเจกใหม่

โดยมีประเภทต่าง ๆ ดังนี้

Navigation-based Application

คือการเริ่มโปรเจกโดย xcode จะสร้าง Navigation Controller และ TableView ให้

OpenGL ES Application

คือการเริ่มโปรเจก โดย xcode จะเพิ่ม Framework ที่จำเป็นต่อการใช้ OpenGL ให้

Split View-based Application

คือการเริ่มโปรเจก โดย xcode จะสร้าง View ให้ 2 view โดย View ทางซ้ายเป็น Master pane ซึ่งเป็น Table View และ ทางขวาเป็น Detail pane ซึ่งเป็น View

Tab Bar Application

คือการเริ่มโปรเจก โดย xcode จะสร้าง Tab Bar Controller ให้

Utility Application

คือการเริ่มโปรเจกต์ โดย xcode จะสร้าง View ที่มี info Button ที่สามารถเปลี่ยนหน้า ใน Present Modal View ได้

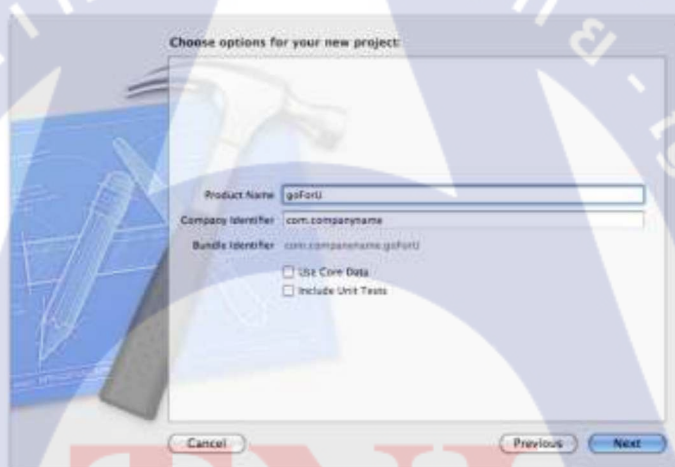
View-based Application

คือการเริ่มโปรเจกต์ โดย xcode จะสร้าง View ที่มี View Controller ให้

Window-based Application

คือการเริ่มโปรเจกต์ โดย xcode จะสร้างเฉพาะ AppDelegate ขึ้นมาให้เท่านั้น

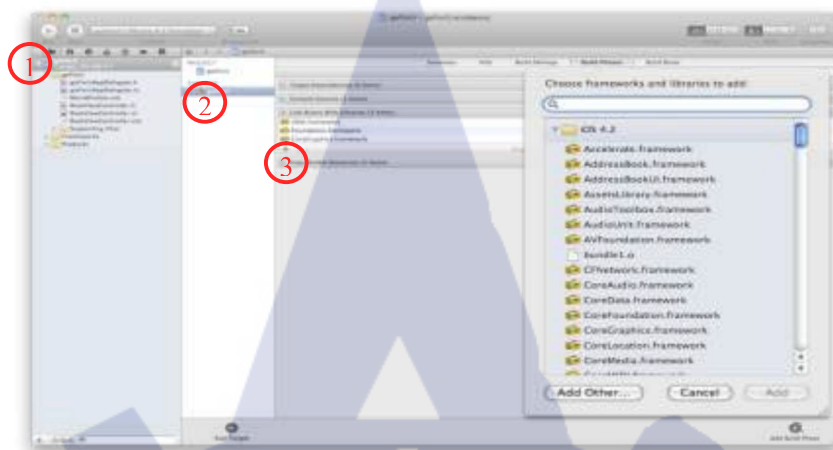
จากนั้นกด Next > ตั้งชื่อใน Product Name > กด Next > เลือก Location > กด Create



รูปที่ 2.2 การตั้งชื่อโปรเจกต์ และชื่อผู้ผลิต

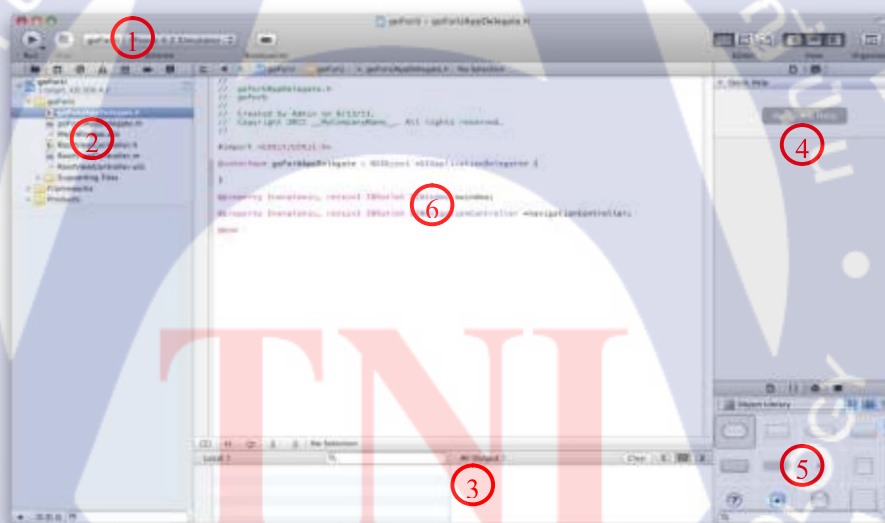
2.6.2 การเพิ่มเฟรมเวิร์ค

หากต้องการเพิ่มเฟรมเวิร์คที่เกี่ยวข้องเข้าไป จะต้องกดที่ ชื่อโปรเจกต์ในเฟลนด้านซ้าย > ชื่อโปรเจกต์ > Target > Build Phases > Link Binary With Libraries > เครื่องหมาย +



รูปที่ 2.3 การเพิ่มเฟรมเวิร์ก เข้าไปในโปรเจก

2.6.3 พื้นที่ทำงานบน Xcode



รูปที่ 2.4 พื้นที่การทำงานของโปรแกรม Xcode

1. **Toolbar** Run คือ Run Program คลิกค้างไว้ จะมี Run, Test, Profile, Analyze

Stop คือ ปุ่มหยุดการรัน

Scheme คือ Target ของการรัน เช่น Simulator, iOS Device

Breakpoint คือ การยกเลิก Breakpoint ทุกจุดในหน้านั้น ๆ

Editor คือ การเลือก Page Control สำหรับ Workspace

View คือ การแสดง/ซ่อน view อื่น ๆ

Organizer คือ การจัดการ Apple Certification Authorization ของ Device ที่เป็น Target

2. **Navigator** คือส่วนที่แสดงไฟล์และการDebugทั้งหมดในโปรเจก เช่นเฮดเดอร์, อิมพลิเมนต์ เทชั่น, เฟรมเวิร์ค, ฟังก์ชัน, คลาส, เมธอด, วอร์นนิ่ง, เออเรอร์, เบรกพอยต์ และอื่น ๆ
3. **Debug Area** คือส่วนที่ใช้สำหรับ Debug แสดงว่าตัวแปรไหนเก็บค่าเท่าไรไว้ หรือถ้าในโปรแกรมเขียนสั่งแสดง Log (NSLog) ไว้ ก็จะนำมาแสดงในส่วนนี้
4. **Utilities** คือส่วนที่แสดง Quick Help, Property ต่าง ๆ
5. **Libraries** คือส่วนที่แสดงส่วนใช้งาน ต่าง ๆ เช่น Object, Snippet, Media, Template Libraries
6. **Coding Area** คือส่วนที่เขียนโค้ดหลักการทำงานของโปรแกรม

2.6.4 การใช้ NeXT Interface Builder

การใช้ IB (Interface Builder) นั้น คือการใช้ฟังก์ชัน ที่เป็นลักษณะ drag & drop ของ อินเทอร์เฟซ ของ xcode ตัวอย่างการใช้งานมีดังนี้

ประกาศตัวแปร

```
IBOutlet UITableView *TableView;
```

หรือ

```
@property (nonatomic, retain) IBOutlet UITableView *TableView;
```

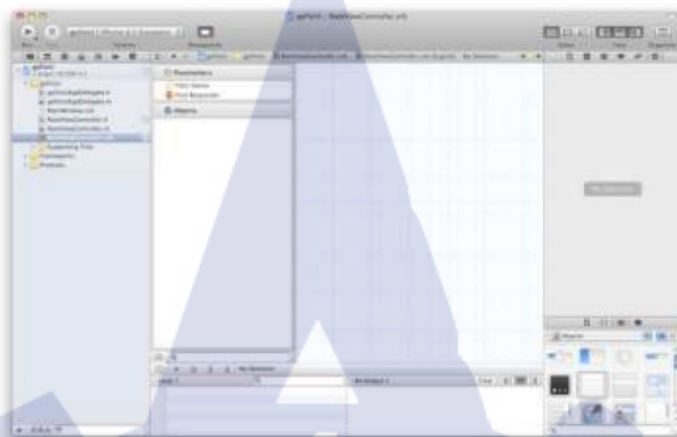
หากทำวิธีหลังต้องประกาศ @synthesize ในหน้าอิมพลิเมนต์เทชั่น ด้วย

```
@synthesize TableView;
```

ประกาศ IBOutlet เพื่อให้ IB รู้จักตัวแปร

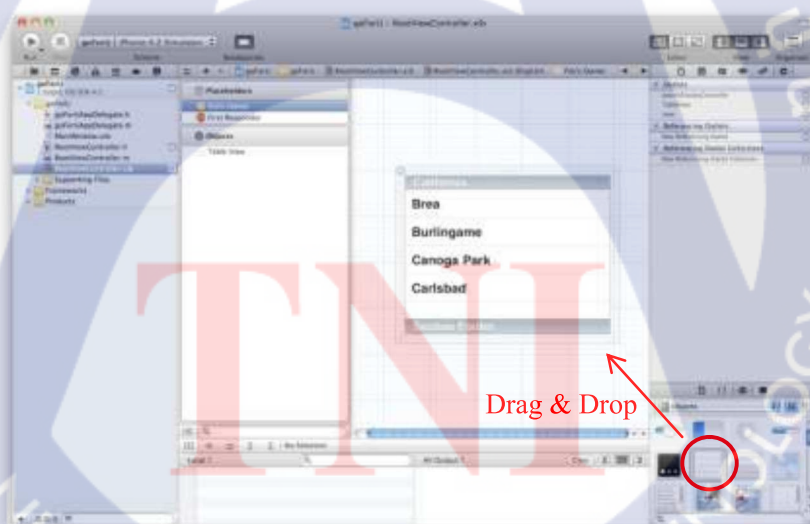
UITableView คือชนิดของตัวแปร

*TableView คือชื่อของตัวแปร



รูปที่ 2.5 หน้า Interface Builder

จากนั้นลาก TableView มาวางบนพื้นที่ว่างจะปรากฏตารางดังรูปที่



รูปที่ 2.6 การแทรกตาราง

จากนั้นต้องกำหนด Property ให้ตารางดังนี้

1.คลิกขวาที่ File's Owner จะปรากฏหน้าต่างดังรูป



รูปที่ 2.7 File's Owner Property

2.คลิกกลาง TableView ไปที่ Table ที่ลากเข้ามา



รูปที่ 2.8 การกำหนดคุณสมบัติให้วัตถุ

3.จะปรากฏหน้าต่างดังรูป

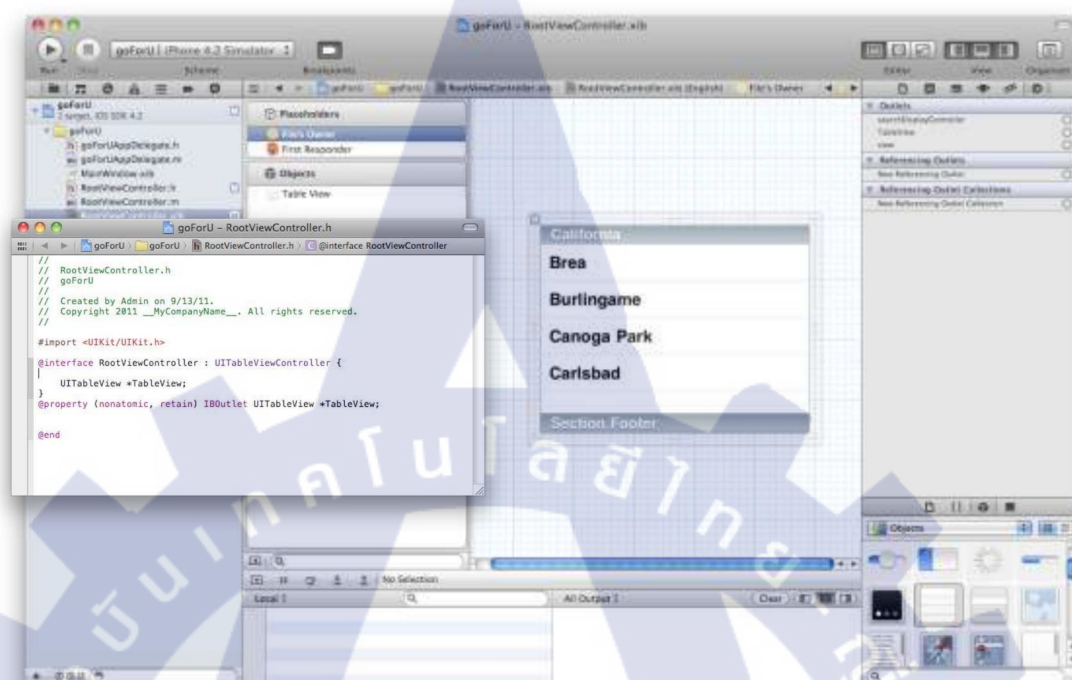


รูปที่ 2.9 File Owner Property

2.6.5 การประกาศแบบ วิธี Drag & Drop

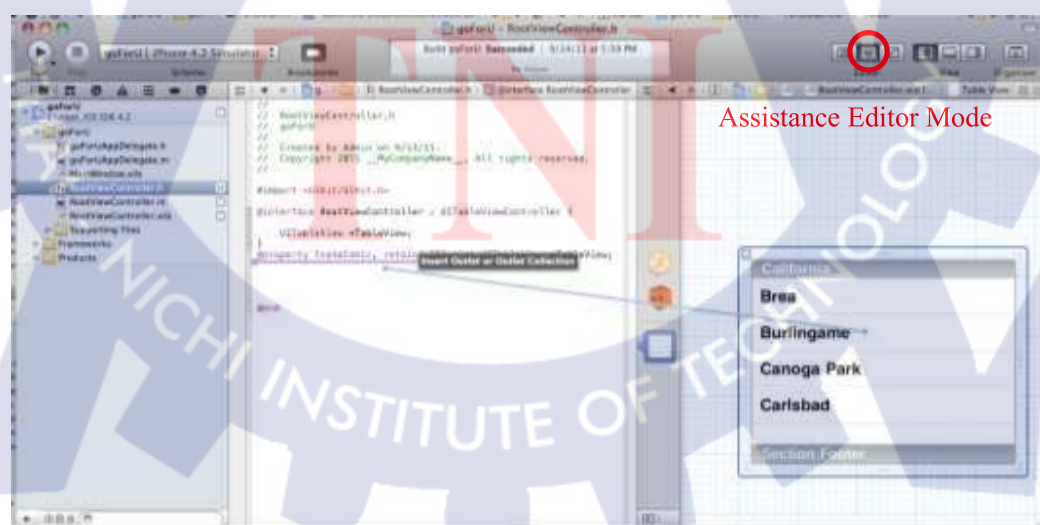
1.ลาก TableView มาวางบนพื้นที่ว่างจะปรากฏตารางดังรูปที่

2.ดับเบิลคลิกที่ .h ของ File จะปรากฏหน้าต่างดังรูป



รูปที่ 2.10 การเรียกหน้าต่างแยก

หรือเรียกใช้ Assistance Editor ดังรูป



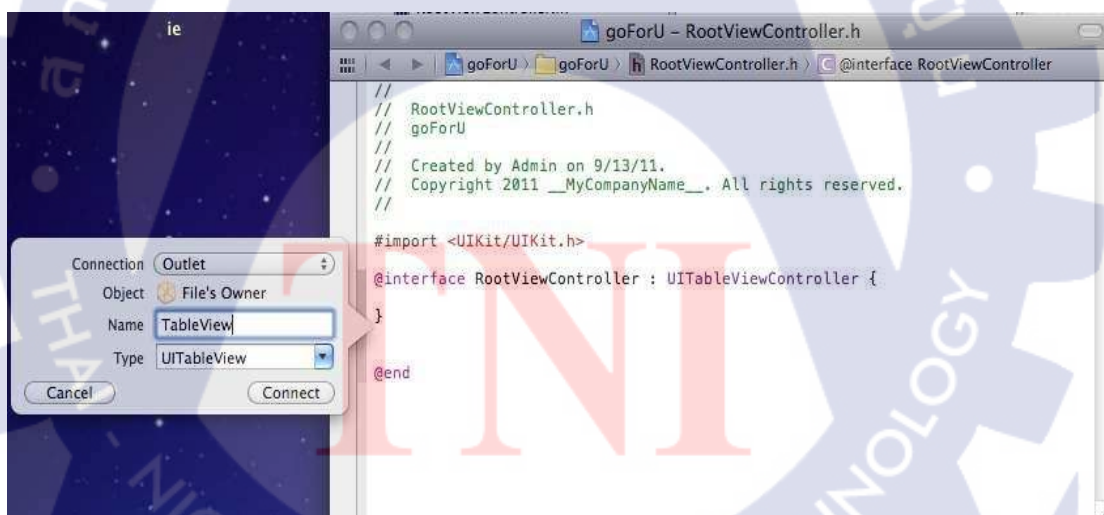
รูปที่ 2.11 การลากโดยใช้ Assistance Editor

3.คลิกขวาที่ TableView ที่ แล้ว ลากไปที่ หน้าต่าง.h ที่เปิดขึ้นมาดังรูป



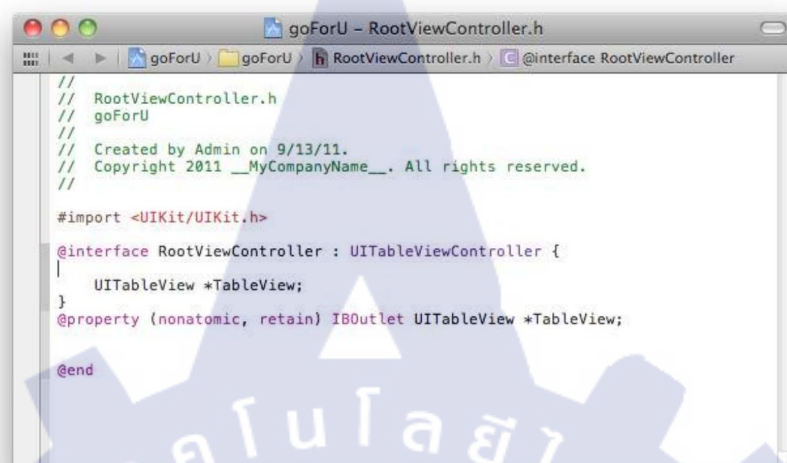
รูปที่ 2.12 การลาก Object ไปที่ ไฟล์ เพื่อกำหนด Property

4.จะปรากฏหน้าต่างดังรูป จากนั้นให้ตั้งชื่อ ตัวแปรที่ใช้ อ้างอิงถึง Object

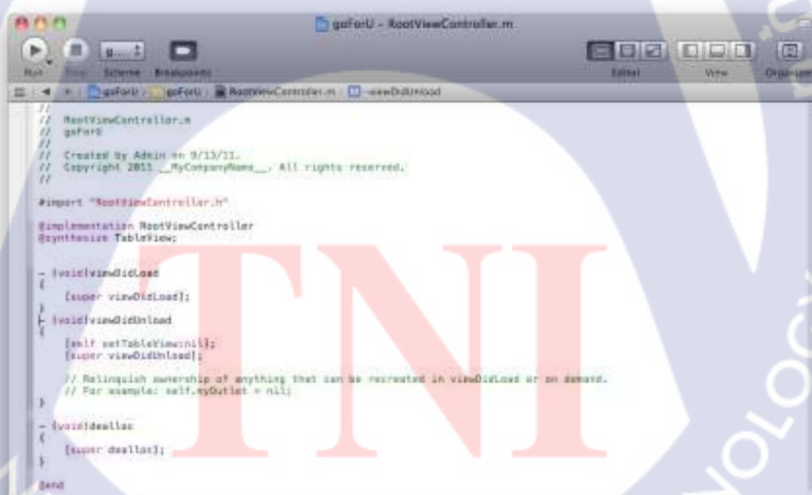


รูปที่ 2.13 การตั้งชื่อตัวแปรของวัตถุ

จากนั้นโปรแกรมจะทำการสร้าง @property, ตัวแปร Local, @synthesize, release, delegate ให้โดยอัตโนมัติดังรูป



รูปที่ 2.14 การประกาศ Property แบบ drag & drop ของ Object หน้า .h



รูปที่ 2.15 การประกาศ Property แบบ drag & drop ของ Object หน้า .m

2.7 พื้นฐานการใช้งานภาษา ภาษาอ็อบเจกทีฟ-ซี ภายใน Xcode เบื้องต้น

2.7.1 การนำเฟรมเวิร์กเข้ามาในไฟล์

```
#import <FrameworkName/FrameworkName.h>
```

โดยในเฟรมเวิร์ก ก็จะมีคลาสที่สร้างมาให้สำเร็จสามารถนำไปใช้งานได้ทันที แต่จำเป็นที่จะต้องรู้ว่าเฟรมเวิร์กไหนมีคลาสอะไรให้ใช้งานได้บ้าง ในส่วนนี้อาจจะต้องค้นหา จาก API Reference ที่มีอยู่ในโปรแกรม Xcode ยกตัวอย่างเช่น `<UIKit/UIKit.h>` จะสามารถใช้งาน ตัวแปรที่เกี่ยวกับการทำแอปพลิเคชัน ที่ใช้ UIView ตัวอย่างเช่น ตัวแปรประเภท `UIView` ที่จะ สร้าง UIView ขึ้นมาโดยไม่จำเป็นต้องใช้ Interface Builder ของตัวโปรแกรม

2.7.2 การสร้างคลาสและการประกาศตัวแปร, ฟังก์ชันของคลาส

ClassName.h (header)

```
@interface ClassName : ClassType <ProtocolReferenceName> {
    [1] VariableType *VariableName;
}

@property (nonatomic, retain) InterfaceBuilderType VariableType *VariableName;

-(void)FunctionName:(VariableType *)Parameter1 and:(VariableType *)Parameter2;
+(void)FunctionName:(VariableType *)Parameter1 and:(VariableType *)Parameter2;

@end
```

ClassName.m (implementation)

```

@implementation ClassName

-(void)FunctionName:(VariableType *)Parameter1 and:(VariableType *)Parameter2{
    //function Method
};

+(void)FunctionName:(VariableType *)Parameter1 and:(VariableType *)Parameter2{
    //function Method
};

@end

```

การสร้างคลาสสามารถกระทำได้โดยประกาศ ชื่อคลาส ชนิดของคลาส (หากต้องการให้เป็นคลาสที่ไม่ใช่ชนิดของคลาสใดๆ ให้เราใส่ชนิดว่าเป็น `NSObject`) และเราสามารถเพิ่มโปรโตคอล ที่ต้องการ ใช้งานได้ตรงหลังชื่อชนิดของคลาภายในสัญลักษณ์ < > เพิ่มให้คลาส ที่สร้างขึ้น มีการตอบสนองต่อโปรโตคอล เหล่านั้นเช่นเมื่อต้องการใช้งาน `UITableView` ภายในคลาสที่ไม่ใช่ ประเภท `UITableViewController` จึงจำเป็นที่จะต้องกำหนดโปรโตคอลให้กับคลาส ตาม คำสั่งเช่น

```
@interface classSample : NSObject <UITableViewDelegate, UITableViewDataSource>
```

การประกาศตัวแปรที่ตำแหน่ง [1] คล้ายกับการประกาศ `private` ของคลาส ภายในภาษาซี จะใช้งานได้เพียงแคภายในอิมพลีเมนต์ ของคลาสตัวเองเท่านั้นถ้าจะประกาศเป็น `public` จะสามารถ ประกาศตัวแปรเป็น `property` ดังในตัวอย่างข้างบนและต้องทำการ `synthesize` ในไฟล์อิมพลีเมนต์เท่านั้น

การสร้างฟังก์ชัน ของคลาส จะมีการสร้างเป็นลักษณะเดียวกันกับตัวแปร ที่มีทั้ง `private` และ `public` โดยจะถูกบอกที่ก่อนหน้าชื่อฟังก์ชัน ชนิด `private` จะเป็น (-) และ `public` จะเป็น (+) จากนั้นจะสามารถสร้างตัวแปรพารามิเตอร์ เพื่อรับค่าจากนอกฟังก์ชันได้เช่นเดียวกับภาษาซีดังเช่นตัวอย่างข้างต้นที่แสดงอยู่

หมายเหตุ : การประกาศตัวแปรภายใน ภาษาอ็อบเจกทีฟ-ซี นั้นส่วนใหญ่ จะใช้เป็นพอยน์เตอร์ยกเว้นการประกาศ

ตัวแปรชนิดที่ไม่ใช่ NSxxx หรือ UIxxx

2.7.3 การเรียกใช้งานฟังก์ชัน เมธอด

ตัวอย่าง

```
[self myFunction];

[self viewDidLoad];

[NSString stringWithFormat:@"%@", StringVariable];
```

สามารถเรียกใช้งานฟังก์ชันได้ดังตัวอย่างข้างต้นซึ่งในแต่ละคลาสที่นำมาใช้งานมักจะมี ฟังก์ชันที่ทำมาให้อยู่แล้วเพื่อความสะดวกในการใช้งาน โดยถ้าต้องการเรียกฟังก์ชัน ภายในคลาสตัวเอง จะสามารถทำได้โดยใช้คำว่า `self` เพื่อบอกว่าเป็นฟังก์ชันของมันเอง

2.7.4 การนำไฟล์ที่อยู่ในไลบรารีเดียวกันกับโปรเจกต์ เพิ่มเข้ามาในไฟล์ปัจจุบัน

```
#import "ImplementName.m" or "HeaderName.h"
```

โดยสามารถทำการอิมพอร์ต ได้ทั้งในไฟล์เฮดเดอร์และไฟล์อิมพลีเมนต์

2.7.5 การประกาศชื่อคลาส

```
@class ClassName;
```

2.7.6 การสร้างคลาสเป็นชนิดที่สร้างขึ้นเองภายในโปรเจกต์

2.7.6.1 สร้างโดย import ไฟล์จาก header อื่นมา

```
#import "OtherClassName.h"

@interface ClassName : ClassType <ProtocolReferenceName> {
```

```
OtherClassName *VariableName;
}
```

2.7.6.2 สร้างโดยประกาศชื่อ class

```
@class OtherClassName;

@interface ClassName : ClassType <ProtocolReferenceName> {
    OtherClassName *VariableName;
}
```

ภายในไฟล์ ClassName.m

```
#import "OtherClassName.h"
@implementation ClassName

@end
```

การประกาศแบบนี้มีข้อแตกต่างกันตรงที่ถ้าไม่ทำการอิมพอร์ต คลาสเข้ามาจะทำให้ตัวแปร ชนิด OtherClassName ไม่รู้จักโครงสร้างภายในของคลาประเภท OtherClassName จะรู้แค่เพียงว่า OtherClassName คือชื่อคลาสหนึ่งเท่านั้น

หากจำเป็นต้องใช้วิธีประกาศชื่อคลาส แต่อยากให้มัน รู้ว่าโครงสร้างภายในคลาส เป็นอย่างไร สามารถทำได้โดยการเพิ่มไฟล์ .h ของคลาสนั้นไปยังไฟล์ .m ที่เราต้องการใช้งาน ดังเช่นตัวอย่างข้างต้น (โดยปกติที่นิยมกันจะประกาศตัวแปรด้วยวิธี สร้างโดยประกาศชื่อคลาส)

2.7.7 การจองพื้นที่ในหน่วยความจำ และการคืนค่าหน่วยความจำ พื้นฐาน

การจองหน่วยความจำ

```
VariableType *VariableName = [[VariableType alloc] init];
```

การคืนหน่วยความจำ

```
[VariableName release];
```

ในการจองพื้นที่ให้กับตัวแปรเราจำเป็นต้องจองเมมโมรี่ กับตัวแปรที่เป็นพอยน์เตอร์ และเมื่อมีการใช้งานเสร็จไม่ควรที่จะลืมคืนค่ามันทิ้งไป เพราะอาจทำให้เป็นสาเหตุของการเกิดหน่วยความจำรั่ว (memory leak) ได้

2.7.8 การกำหนดคุณสมบัติต่าง ๆ ให้กับตัวแปรที่เราสร้างขึ้น

```
VariableName = 13.727775;
ClassName.PropertyName = NO;
VariableName = [UIColor colorWithWhite:Value alpha:Value];
[UIViewVariableName setFrame:CGRectMake(yPoint, xPoint, ySize, xSize)];
```

การกำหนดค่าให้กับตัวแปรต่าง ๆ สามารถทำได้ในหลาย ๆ วิธีในภาษา ภาษาอ็อบเจกทีฟ-ซี เช่นเดียวกับ ภาษาซี ก็สามารถใช้ Expression (=) ในการแอดค่าได้ทันที หรือจะใช้ฟังก์ชัน ที่สร้างขึ้นมาใน ตัวแปรชนิดนั้น ๆ กำหนดค่าก็ได้ดังตัวอย่างที่แสดงข้างต้น

2.8 หลักการใช้งาน Delegate เพื่อเข้าถึง อินเทอร์เฟซ ทั้งหมดภายในโปรเจก

ในการเขียนแอปพลิเคชัน บางครั้งเราต้องการที่จะเรียกถึงฟังก์ชันในหน้าอื่น ๆ แต่ไม่สามารถเรียกได้ ซึ่งอาจมีเหตุผลมาจากคลาสที่มีฟังก์ชันที่ต้องการเรียกเป็นคลาสแม่ของคลาสที่ทำงานอยู่สามารถประกาศ AppDelegate ขึ้นมาเพื่อใช้ในการเข้าถึงโดยมีวิธีการดังนี้

2.8.1 วิธีการใช้งาน Delegate

1. ทำการอิมพอร์ต ไฟล์ AppDelegate.h ของโปรเจกเข้ามา
2. ประกาศตัวแปรขึ้นมาเป็นประเภท AppDelegate เพื่อใช้ในการ Share Delegate ตาม syntax นี้

```
AppDelegate *delegate = [[UIApplication sharedApplication] delegate];
```

3. เรียกจับคลาส ผ่าน delegate ไปจนถึงไฟล์ที่ต้องการ (จำเป็นต้องอิมพอร์ต ไฟล์คลาสที่จะเข้าถึง มาด้วยเพื่อให้คลาส ปัจจุบันรู้จักว่ามีตัวแปรนั้น ๆ อยู่ในคลาสที่จะเข้าถึง)

2.8.2. ตัวอย่างการใช้งาน

TestAppDelegate.h

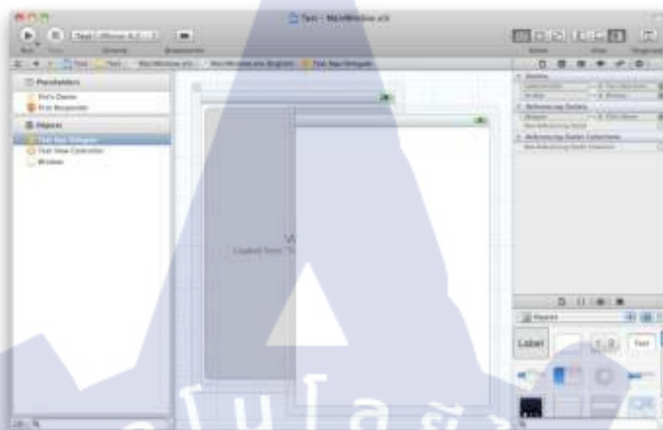
```
#import <UIKit/UIKit.h>
@class TestViewController;
@interface TestAppDelegate : NSObject <UIApplicationDelegate> {
}
@property (nonatomic, retain) IBOutlet UIWindow *window;
@property (nonatomic, retain) IBOutlet TestViewController *viewController;
@end
```

หมายเหตุ : **IBOutlet** คือการประกาศเพื่อให้ทาง Interface Builder รู้ว่ามี Outlet ซ่อนอยู่เพื่อนำไป โยง เข้ากับ UI นั้น ๆ

TestAppDelegate.m

```
#import "TestAppDelegate.h"
#import "TestViewController.h"
@implementation TestAppDelegate
@synthesize window=_window;
@synthesize viewController=_viewController;
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions{
    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    return YES;
}
- (void)dealloc{
    [_window release];
    [_viewController release];
    [super dealloc];
}
@end
```

TestAppDelegate.xib



รูปที่ 2.16 หน้า TestAppDelegate.xib

RootViewController.h

```
#import <UIKit/UIKit.h>
@class OtherViewController;
@interface TestViewController : UIViewController
@property (nonatomic,retain) OtherViewController *otherViewController;
@property (nonatomic,retain) IBOutlet UIButton *button;
- (IBAction)GoFunction;
- (void)CallFunction;
@end
```

RootViewController.m

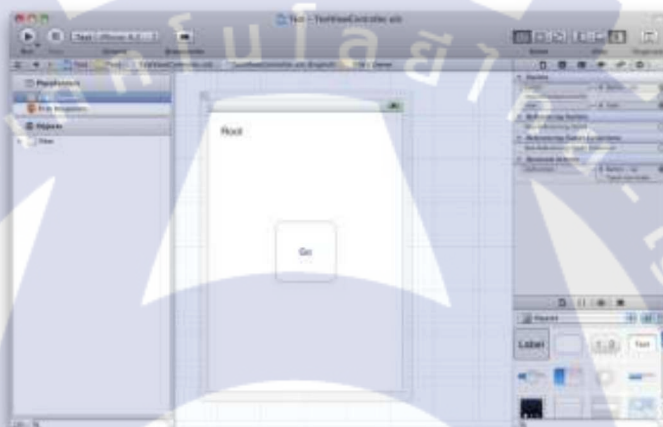
```
#import "TestViewController.h"
#import "OtherViewController.h"
@implementation TestViewController
@synthesize otherViewController;
@synthesize button;
- (void)dealloc {
    [super dealloc];
    [otherViewController release];
    [button release];
}
```

```

- (IBAction)GoFunction {
    otherViewController = [[OtherViewController alloc] initWithNibName:
        @"OtherViewController" bundle:nil];
    [self presentViewController:otherViewController animated:YES];
}
- (void)CallFunction {
    otherViewController.label.text = @"Hello World!";
}
@end

```

RootViewController.xib



รูปที่ 2.17 หน้า RootViewController

OtherViewController.h

```

#import <UIKit/UIKit.h>

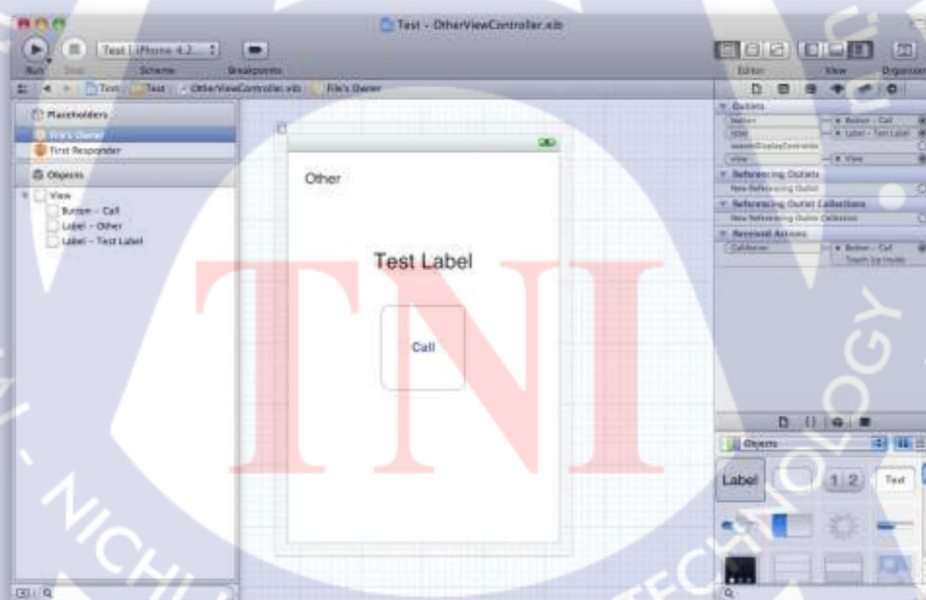
@interface OtherViewController : UIViewController
@property (nonatomic,retain) IBOutlet UIButton *button;
@property (nonatomic,retain) IBOutlet UILabel *label;
- (IBAction)CallAction;
@end

```

OtherViewController.m

```
#import "OtherViewController.h"
#import "TestAppDelegate.h"
#import "TestViewController.h"
@implementation OtherViewController
@synthesize button,label;
- (void)dealloc {
    [super dealloc];
    [button release];
}
-(IBAction)CallAction {
    TestAppDelegate *appDelegate = [[UIApplication sharedApplication]delegate];
    [appDelegate.viewController CallFunction];
}
@end
```

OtherViewController.xib



รูปที่ 2.18 หน้า OtherViewController

2.8.3 ขั้นตอนการทำงานของแอปพลิเคชันตัวอย่าง

(เริ่มที่กดปุ่มหน้า Root > กดปุ่มที่หน้า Other > เรียกฟังก์ชันใน Root)



รูปที่ 2.19 ขั้นตอนการทำงานของแอปพลิเคชันตัวอย่าง

2.9 UI Class (User Interface Class) พื้นฐานภายในโปรแกรม Xcode



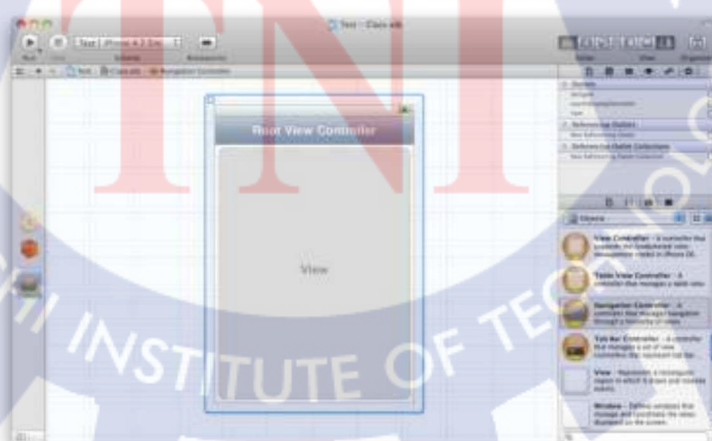
รูปที่ 2.20 พื้นฐานภายในโปรแกรม Xcode

ในการเขียนแอปพลิเคชัน นั้นการสร้าง อินเทอร์เฟซ ของแอปพลิเคชันแบ่งออก เป็น 2 วิธี คือ การเขียนด้วยโค้ด (programmatically) หรือการใช้อุปกรณ์สำเร็จรูปที่สามารถใส่อินเตอร์เฟซ ได้เลย (NeXT Interface Builder หรือไฟล์ .xib) โดยการสร้างทั้ง 2 แบบจะให้ผลลัพธ์ที่เหมือนกันกับ การใช้โค้ดเขียน อินเทอร์เฟซขึ้นมา จะมีความยุ่งยากมากกว่าการใช้ NIB (NeXT Interface Builder) แต่ จะมีข้อดีตรงที่มีความยืดหยุ่นมากกว่า สามารถปรับแต่งในรูปแบบที่มีความเฉพาะตัวได้

การสร้างอินเทอร์เฟซนั้นจะมีคอนโทรลเลอร์ที่เปรียบเสมือนการวางแผนรองลงไปบนพื้นที่ก่อนที่จะเริ่มทำการวาง อินเทอร์เฟซ อื่น ๆ ลงไปยังหน้าแอปพลิเคชันโดยหลัก ๆ ที่ใช้ก็จะมี `UIViewController`, `UITableViewController`, `UITabBarController`, `UISearchBar`, `UINavigationController` หากต้องการเพิ่มการทำงานของคอนโทรลเลอร์ นั้นเราสามารถเพิ่ม Protocol อื่น ๆ ลงไปได้อีกด้วย เช่นเมื่อเพิ่ม `<UIAlertViewDelegate>` จะทำให้คอนโทรลเลอร์ นั้นสามารถใช้งาน `UIAlertView` โปรโตคอลได้ ลำดับของการใส่คอนโทรลเลอร์เป็นชั้น ลำดับได้ดังนี้ `TabBarController > UINavigationController > UITableViewController = ViewController` และใน `UITableViewController > UISearchBar Controller`

ในโปรแกรม Xcode จะมีอินเทอร์เฟซ ต่าง ๆ ให้เราใช้งานซึ่งหากต้องการใช้อินเทอร์เฟซ ใด ๆ ก็ สามารถเข้าไปที่ API Reference เพื่อศึกษาวิธีการใช้งานได้ด้วยตนเอง โดยเราจะยกตัวอย่าง อินเทอร์เฟซ ที่เป็นคอนโทรลเลอร์ พื้นฐานที่ใช้งานภายในโปรเจกที่มีหน้าที่สำคัญ ๆ ดังนี้

2.9.1 UINavigationController & UINavigationController



รูปที่ 2.21 หน้า UINavigationController

การทำงานของ UINavigationController เป็นคอนโทรลเลอร์ ที่มีหน้าที่หลักในการพาผู้ใช้งาน ไปยังหน้า (View) อื่น ๆ และสามารถทำการกลับไปเป็นลำดับเพื่อกลับไปยังหน้า (View) แรกได้ โดยตัว UINavigationController นั้นสามารถที่จะกำหนดค่าข้อความของหัวข้อเพื่อให้ทราบว่าอยู่ที่หน้าใดหรือเพื่อเป็นการบอกว่าแอปพลิเคชันนี้มีชื่อว่าอะไร รวมไปถึงการเซตค่าของชนิด UINavigationController และอื่น ๆ

ตัว UINavigationController สามารถทำการใส่ UINavigationControllerItem ยกตัวอย่างเช่นการใส่ปุ่ม Setting, Cancel, Done เป็นต้นโดยปกติแล้วหากเราไม่ได้ทำการใส่ปุ่มลงใน UINavigationController ตัวมันเองจะมีปุ่ม Back อยู่ทางซ้ายหากหน้า (View) นั้นทำการเปลี่ยนหน้า (Push) มาจากหน้าใด ๆ

การ Push/Pop Navigation เพื่อไปยังหน้าต่อไปหรือกลับไปยังหน้าก่อนหน้า

```
[navController pushViewController:otherViewController animated:YES];
[otherViewController.navigationController popViewControllerAnimated:YES];
```

ตัวอย่างการตั้งค่าเบื้องต้นของ UINavigationController

```
navController.title = @"TitleName";
navController.navigationBar.tintColor = [UIColor blackColor];
```

การสร้าง UINavigationController

```
UIViewController *rootViewController = [[UIViewController alloc] init];
UINavigationController *navController = [[UINavigationController alloc]
initWithRootViewController:rootViewController];
```

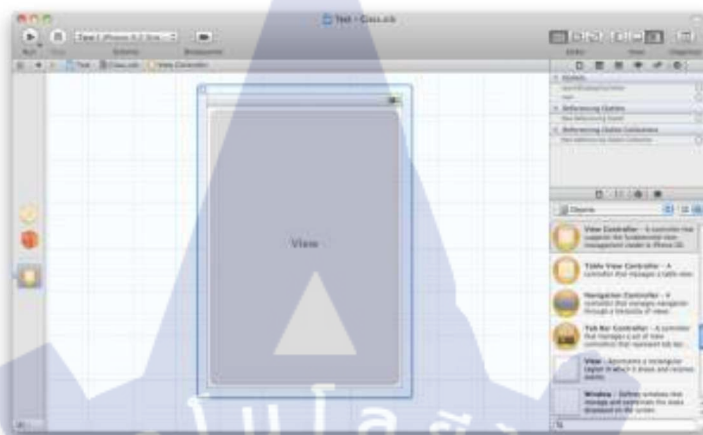
หมายเหตุ : การสร้าง UINavigationController ปกติเราจะใส่ ViewController หรือ TableViewController อย่างใดอย่างหนึ่งลงเป็น RootViewController เพื่อแสดงผลในรูปแบบที่ต้องการ

การสร้างปุ่มบน UINavigationController

```
UIBarButtonItem *rightButton = [[UIBarButtonItem alloc]
initWithTitle:@"Edit"
style:UIBarButtonSystemItemEdit
target:self
action:@selector(EditFunction)];
navController.navigationItem.rightBarButtonItem = rightButton;
[rightButton release];
```

ในการสร้างปุ่มนั้นสามารถสร้างปุ่มบน UINavigationController โดยจะสามารถใส่ทางซ้ายและขวาข้างละปุ่ม

2.9.2 UIViewController & UIView



รูปที่ 2.22 หน้า UIViewController

เป็น UI ที่มีหน้าที่ในการควบคุมการแสดงผลของ **UIView** โดยสามารถวางอินเทอร์เฟซ ต่าง ๆ ลงไปบน **UIViewController** ได้ ซึ่งการสร้างคอนโทรลเลอร์ โดยโค้ดจะสามารถทำได้โดย syntax ดังนี้

สร้าง ViewController ปกติ

```
UIViewController *viewController = [[UIViewController alloc] init];
```

สร้าง Class ประเภท ViewController

```
@interface myViewController : UIViewController
```

หากสร้างคลาส ที่เป็นชนิด ViewController ภายในของ ViewController จะมีเมธอด ที่จะทำงานในแต่ละ State ของการทำงานหลัก ๆ ดังนี้

ViewController View Life Cycle

- (void)viewDidLoad
- (void)viewWillAppear
- (void)viewDidAppear
- (void)viewWillDisappear
- (void)viewDidDisappear
- (void)viewDidUnload
- etc.

โดยในแต่ละฟังก์ชัน จะถูกทำงานเป็นลำดับแตกต่างกัน ซึ่งในจุดนี้จะเป็นจุดที่ใส่โค้ด ของการทำงานลงไปเช่นหากต้องการให้โค้ด ใด ๆ เริ่มทำงานก่อนหน้าที่ ViewController นี้จะแสดงผล ก็จะใส่โค้ดนั้นไปที่ viewWillAppear ที่ เป็นฟังก์ชัน ที่ถูกเรียกทุกครั้งเมื่อจะมีการแสดงผลหน้าคอนโทรลเลอร์ซึ่งต่างจาก ViewDidLoad ซึ่งจะรันเพียงครั้งเดียวหลังจากที่รันแอปพลิเคชัน และหากต้องการให้เมื่อ ViewController นั้นแสดงผลแล้วให้ทำการเริ่มการทำงานโค้ด ก็ให้ใส่ลงในฟังก์ชัน viewDidAppear เป็นต้น

การจัดการคืนค่าหน่วยความจำ

- (void)dealloc

การคืนค่าหน่วยความจำ ของโปรแกรมสามารถกระทำได้โดยการเขียนโค้ดคืนค่า ภายในฟังก์ชัน dealloc เพื่อให้ เมื่อจบการทำงานของแอปพลิเคชัน แล้ว ส่วนของหน่วยความจำที่ถูกจองค่าไว้จะมีการปล่อยคืนเพื่อป้องกันไม่ให้เกิดปัญหาหน่วยความจำรั่ว (Memory Leak)

การนำ View ไปวางบน ViewController.view

```
[1] [viewController.view addSubview:MyView];
[2] [viewController.view insertSubview:MyView aboveSubview:MyViewBG];
[3] [viewController.view insertSubview:MyView belowSubview:MyViewBG];
```

เมื่อต้องการใส่ Object อื่น ๆ ลงไปบน UINavigationController ที่สร้างขึ้นสามารถใส่คำสั่งเพื่อให้ Object เหล่านั้นไปวางอยู่บน Controller ได้ด้วยคำสั่งข้างต้นซึ่งแต่ละคำสั่งมีการทำงานดังนี้

- [1] คือการเพิ่ม MyView ลงไปบน view โดยจะทับลงไปบน view เรื่อย ๆ หากมีการเพิ่ม ซ้ำอีกครั้ง
- [2] คือการเพิ่ม MyView ลงไปบน view แต่จะอยู่เหนือ view ที่เรากำหนด (ในที่นี้คือ MyViewBG)
- [3] คือการเพิ่ม MyView ลงไปบน view แต่จะอยู่หลัง view ที่เรากำหนด (ในที่นี้คือ MyViewBG)

การกำหนดขนาดและตำแหน่ง UIView

```
[MyView setFrame:CGRectMake(float x, float y, float width, float height)];
MyView.frame = CGRectMake(float x, float y, float width, float height);
```

การกำหนดขนาด, ตำแหน่งของ UIView ได้ เพื่อให้หน้า (View) ของเรามีขนาด, ตำแหน่ง ตามที่ต้องการได้โดยจะมีความสำคัญมากในเวลาที่เราต้องการสร้างหน้า อินเทอร์เฟซ ที่มีความเฉพาะตัว ซึ่งอาจไม่สามารถกำหนดได้ในหน้า NIB

การกำหนด ModalPresentationStyle

```
[1] rootViewController.modalTransitionStyle =
UIModalTransitionStyleCoverVertical;
[2] rootViewController.modalTransitionStyle =
UIModalTransitionStyleCrossDissolve;
[3] rootViewController.modalTransitionStyle =
UIModalTransitionStyleFlipHorizontal;
[4] rootViewController.modalTransitionStyle =
UIModalTransitionStylePartialCurl;
```

โดยแต่ละคำสั่งมีลักษณะการทำงานดังนี้

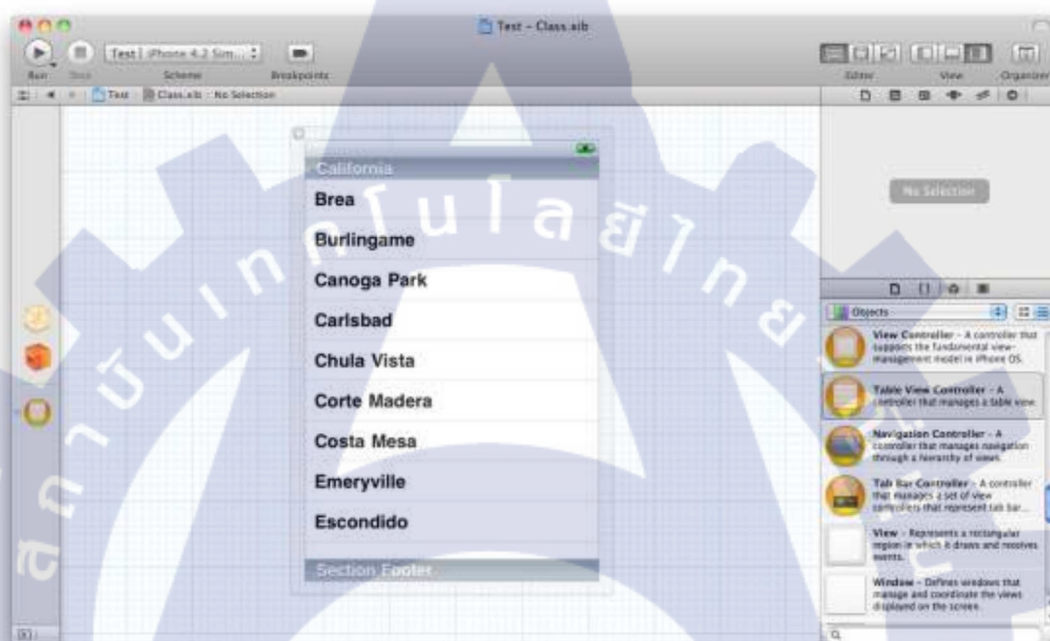
- [1] จะแสดงหน้าที่เรียกด้วยการแทรกจากด้านล่างของจอแสดงผล
- [2] จะแสดงหน้าที่เรียกด้วยการค่อย ๆ เข้มขึ้นมาทับหน้าปัจจุบัน (Fade)
- [3] จะแสดงหน้าที่เรียกด้วยการพลิกหน้าปัจจุบันมาเป็นหน้าที่เรียก
- [4] จะแสดงหน้าที่เรียกด้วยการพลิกในลักษณะการเปิดหนังสือ

การ presentModalViewController และการ dismissModalViewController

```
[rootViewController presentModalViewController:otherViewController
animated:YES];
[otherViewController dismissModalViewControllerAnimated:YES];
```

การ presentModalViewController เปรียบเสมือนการแทรกหน้าของ ViewController อื่นขึ้นมาทับบน หน้า Controller ปัจจุบันด้วยการ addSubview ส่วนการ dismissModalViewController ก็เสมือนกับการ removeFromSuperview นั่นเอง

2.9.3 UITableViewController & UITableView



รูปที่ 2.23 การแสดง UITableViewController

เป็นคอนโทรลเลอร์ ที่จะแสดง UITableView ออกมาเป็นหน้าหลักและมีเมธอดในการทำงาน เช่นเดียวกับ ViewController ที่จะมีเมธอด ในการเช็คสถานะจำพวก viewDidLoad, viewDidLoad, viewWillAppear ที่เหมือนกันแต่จะมีเพิ่มขึ้นมาในส่วนของ UITableViewController โดยเมธอดที่ใช้ งานหลัก ๆ มีดังนี้

จำนวนเซกชั่น ของ TableView ภายใน UITableViewController

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {
    return 0;
}
```

หมายเหตุ : เซลล์คือหัวข้อภายใน TableView ดังภาพข้างต้นคือส่วนแถบที่ชื่อว่า California ซึ่งเราสามารถกำหนดค่าของชื่อหัวข้อและขนาดของหัวข้อได้โดยการใส่ฟังก์ชันลงไปเพิ่มเติม

ตัวอย่าง - การกำหนดค่าความกว้างและชื่อของหัวข้อ (Section)

```
-(CGFloat)tableView:(UITableView *)tableView
heightForHeaderInSection:(NSInteger)section {
    return 10.0;
}
-(NSString *)tableView:(UITableView *)tableView
titleForHeaderInSection:(NSInteger)section {
    return @"TableTitleText";
}
```

จำนวนแถวในแต่ละหัวข้อ ของ TableView ภายใน TableViewController

```
-(NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section {
    return 0;
}
```

ตั้งค่าเซลล์ ของ TableView ภายใน TableViewController

```
-(UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath {
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView
    dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc]
        initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier]
        autorelease];
    }
    return cell;
}
```

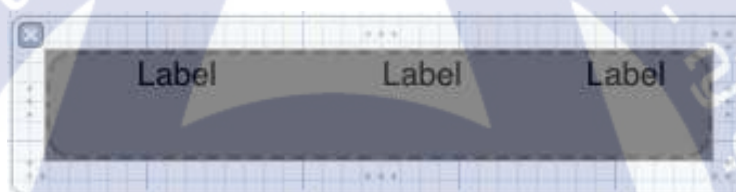
เป็นการสร้างเซลล์ ภายใน UITableView โดยหลักการทำงานคือจะมีการเรียกใช้งานฟังก์ชันนี้ ใน ทุก ๆ ครั้งที่มีการขยับหน้าจอจนทำให้มองเห็น Cell ที่ก่อนหน้านี้ไม่แสดงผลในหน้าจอเพื่อสร้างเซลล์ ขึ้นมา ให้ผู้ใช้งานมองเห็นได้นั่นเอง

การใช้งาน Custom Cell

```
#import <UIKit/UIKit.h>
@interface MyCell : UITableViewCell{
    IBOutlet UILabel *label1;
    IBOutlet UILabel *label2;
    IBOutlet UILabel *label3;
}
-(void)setLabel1:(NSString *)l1;
-(void)setLabel2:(NSString *)l2;
-(void)setLabel3:(NSString *)l3;
@end
```

ในหน้า Custom Cell.h ให้ประกาศอ็อบเจกต์ ที่ต้องการจะใช้ในเซลล์ โดยมี IBOutlet ด้วยเมื่อต้องการใช้ Interface Builder จากนั้นจึงประกาศ Function เพื่อตั้งค่าให้อ็อบเจกต์

เมื่อประกาศอ็อบเจกต์ หน้า .h เสร็จแล้ว ต้องโยง ตัวแปรที่ประกาศกับอ็อบเจกต์ในหน้า xib ด้วย



รูปที่ 2.24 Custom Cell

จากนั้นก็มาเขียนฟังก์ชันตั้งค่า เพื่อใส่ค่าให้อ็อบเจกต์ ดังนี้

```
#import "MyCell.h"
@implementation MyCell
-(void)setLabel1:(NSString *)text1{
    label1.text = text1;
}
-(void)setLabel2:(NSString *)text2{
    label2.text = text2;
}
-(void)setLabel3:(NSString *)text3{
    label3.text = text3;
}
@end
```

การตั้งค่าให้สามารถทำการปรับเปลี่ยน/แก้ไขเซลล์ได้

```
- (BOOL)tableView:(UITableView *)tableView canEditRowAtIndexPath:(NSIndexPath *)indexPath {
    return YES;
}
```

โดยปกติแล้วเราสามารถสไลด์นิ้วไปบนเซลล์ เพื่อทำการแก้ไข (ลบ) เซลล์ โดยสามารถ กำหนดค่าให้เป็น NO ในกรณีที่ไม่ต้องการให้มีการแก้ไข หรือ YES เพื่อให้สามารถทำการแก้ไขได้

เมื่อมีการแก้ไข TableView ให้แสดงผลรูปแบบการแก้ไข เป็นตามที่ตั้งค่าไว้

```
- (void)tableView:(UITableView *)tableView
commitEditingStyle:(UITableViewCellEditingStyle)editingStyle
forRowAtIndexPath:(NSIndexPath *)indexPath
{
    if (editingStyle == UITableViewCellEditingStyleDelete) {
        [tableView deleteRowsAtIndexPaths:[NSArray arrayWithObject:indexPath]
        withRowAnimation:UITableViewRowAnimationFade];
    }
    else if (editingStyle == UITableViewCellEditingStyleInsert) {
    }
}
```

ในการแก้ไข แต่ละครั้งของเซลล์ คอนโทรลเลอร์จะรู้ว่ามีแก้ไข แต่จะยังไม่ทราบว่าให้ทำอะไรต่อไป ซึ่งเราสามารถกำหนดคำสั่งที่อยากให้งานได้เองด้วยการนำโค้ด ที่เราต้องการใส่ลงมาภายในฟังก์ชัน นี้โดยกำหนดได้ว่าหากจะทำการลบให้ทำอะไร หากทำการเพิ่ม ให้ทำอะไรที่เซลล์ ไหน เป็นต้น

เมื่อกดเลือกเซลล์ ของ TableView ใน TableViewController

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {
    [tableView deselectRowAtIndexPath:indexPath animated:YES];
}
```

ฟังก์ชันนี้ใช้ในการตรวจจับการสัมผัสที่ TableViewCell ภายใน TableView ว่าเลือกที่เซลล์ไหนของ TableView และ เราสามารถที่จะกำหนดให้มีการทำงานใดต่อไปภายในฟังก์ชันนี้นั่นเอง เช่นเมื่อเลือกแถวที่ 1 ของหัวข้อ (Section) ที่ 2 ให้ทำการเปลี่ยนหน้า(Push)ไปยังหน้าแสดงรายละเอียดสินค้า เป็นต้น

2.10 ภาษา PHP เป็นตัวกลางในการสืบค้นข้อมูลในฐานข้อมูล

ใช้เพื่อทดสอบการทำงานของโปรแกรมวัดกรรมโดยส่ง Username และ Password จากหน้าจอเข้าสู่ระบบของแอปพลิเคชันใน iPhone เข้าไปใน PHP เพื่อให้ PHP เป็นตัวกลางในการติดต่อกับฐานข้อมูล และ สั่งให้ PHP เขียนลงไฟล์ XML เพื่อที่จะนำ XML ไปตัด Tag และนำข้อมูลไปใช้ได้

```
<?php
$u_lan=$_GET["u_name"];
$p_lan=$_GET["p_word"];
```

ส่วนนี้เป็นส่วนที่ใช้รับข้อมูลซึ่งรับมาจากที่ผู้ใช้กรอกในหน้ายืนยันตัวตนของแอปพลิเคชัน โดยที่ผู้ใช้จะไม่เห็นลิงค์ (Link) รวมไปถึงชื่อตัวแปรที่ใช้

```
$DB="goForUDB";
mysql_connect("localhost","root","") or die ("Cannot connect to Server");
mysql_select_db($DB);
```

ส่วนติดต่อกับโฮส และฐานข้อมูล เนื่องด้วยในตอนนี้เป็นารเขียนโปรแกรมจำลองเพื่อให้ xcode สามารถนำ Tag ไปตัดเพื่อรับข้อมูลได้ จึงทดสอบบน local host

```
$filename="data.xml";
$contents=file($filename);
```

ส่วนที่ใช้สร้าง File ซึ่งตั้งชื่อให้เป็น data.xml และ ตั้ง Contents ให้เป็นไฟล์ ของ data.xml

```
$sql="select *
      from mapping mp
      inner join table_rpm rpm
      inner join table_hr hr
      inner join table_lnw lnw
      inner join master_user ma
      on (mp.mp_rpm_employee_id = rpm.rpm_employee_id)
      and (mp.mp_hr_id = hr.hr_id)
      and (mp.mp_lnw_notes_name = lnw.lnw_notes_name)
      and (mp.mp_user_lan = ma.user_lan)
      where ma.user_lan like '$u_lan' and ma.password_lan like '$p_lan'";
$test=mysql_query($sql);
$result=mysql_fetch_array($test);
```

ส่วนของภาษา sql เพื่อสืบค้นข้อมูล 4 ตาราง แล้วนำมาจอยน์กัน และ ส่วนของการคิวรี (query) ใน
ในภาษา MySQL

```
if($result)
{
    $head("<?xml version='1.0' encoding='utf-8'?'>\n";
    $data("<channel>\n";
    $data.="<item>\n";
    $data.="<lnw_notes_name>$result[24]</lnw_notes_name>\n";
    $data.="<lnw_sr_number>$result[25]</lnw_sr_number>\n";
    $data.="<lnw_status>$result[26]</lnw_status>\n";
    $data.="<lnw_date_cut_off>$result[27]</lnw_date_cut_off>\n";
    $data.="<lnw_aging>$result[28]</lnw_aging>\n";
```

```

        $data."<lnw_expire_date>$result[29]</lnw_expire_date>\n";
        $data."</item>\n";
        $data."</channel>";
    }
    else

        $data="0";

```

ส่วนของการสร้าง tag ให้เป็นรูปแบบของภาษา xml เพื่อให้ xcode สามารถนำไปตัดได้

```

$fp=fopen($filename,'w');
fwrite($fp,$head.$data);
?>

```

ส่วนของการส่งเขียนข้อมูลลงไฟล์ ทุกครั้งที่มีการ Execute หน้า PHP นี้

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

2.11 การเขียน C#.net เพื่อให้เป็นตัวกลางแทน PHP ในการสืบค้นฐานข้อมูล

```
using System;
using System.Data;
using System.IO;
using System.Configuration;
```

การ Using คือการเพิ่มเฟรมเวิร์คเข้ามาในโปรเจค

```
public partial class _Default : System.Web.UI.Page
{
    SqlConnection con;

    string constr = ConfigurationManager.ConnectionStrings["SmartAlertConnectionString"]
        .ToString();
    string[] temp;
```

• ส่วนของการเขียนโค้ด และ การสร้างตัวแปรแบบโอบอล (Global Variable)

```
protected void Page_Load(object sender, EventArgs e)
{
    string user;
    string pass;
```

ฟังก์ชัน Page Load เป็น ฟังก์ชันที่จะรันทุกครั้งที่มีการทำงาน

```
user = Request.QueryString["u_name"];
pass = Request.QueryString["p_word"];
```

การใช้ Request คือการร้องขอโดยตัวโปรแกรมจะตัดสินใจเองว่าตัวแปรที่ส่งมานั้นเป็นเมธอด Get หรือเมธอด Post

```
string comstr = "select * from v_Get_Data where User_Lan like '";
comstr += user;
comstr += "' and Password_Lan like '";
comstr += pass;
comstr += "'";
con = new SqlConnection(constr);
SqlDataAdapter da = new SqlDataAdapter(comstr, con);
DataSet ds = new DataSet();
```

ส่วนของการสืบค้นฐานข้อมูล โดยรับพารามิเตอร์ จากหน้าอื่น

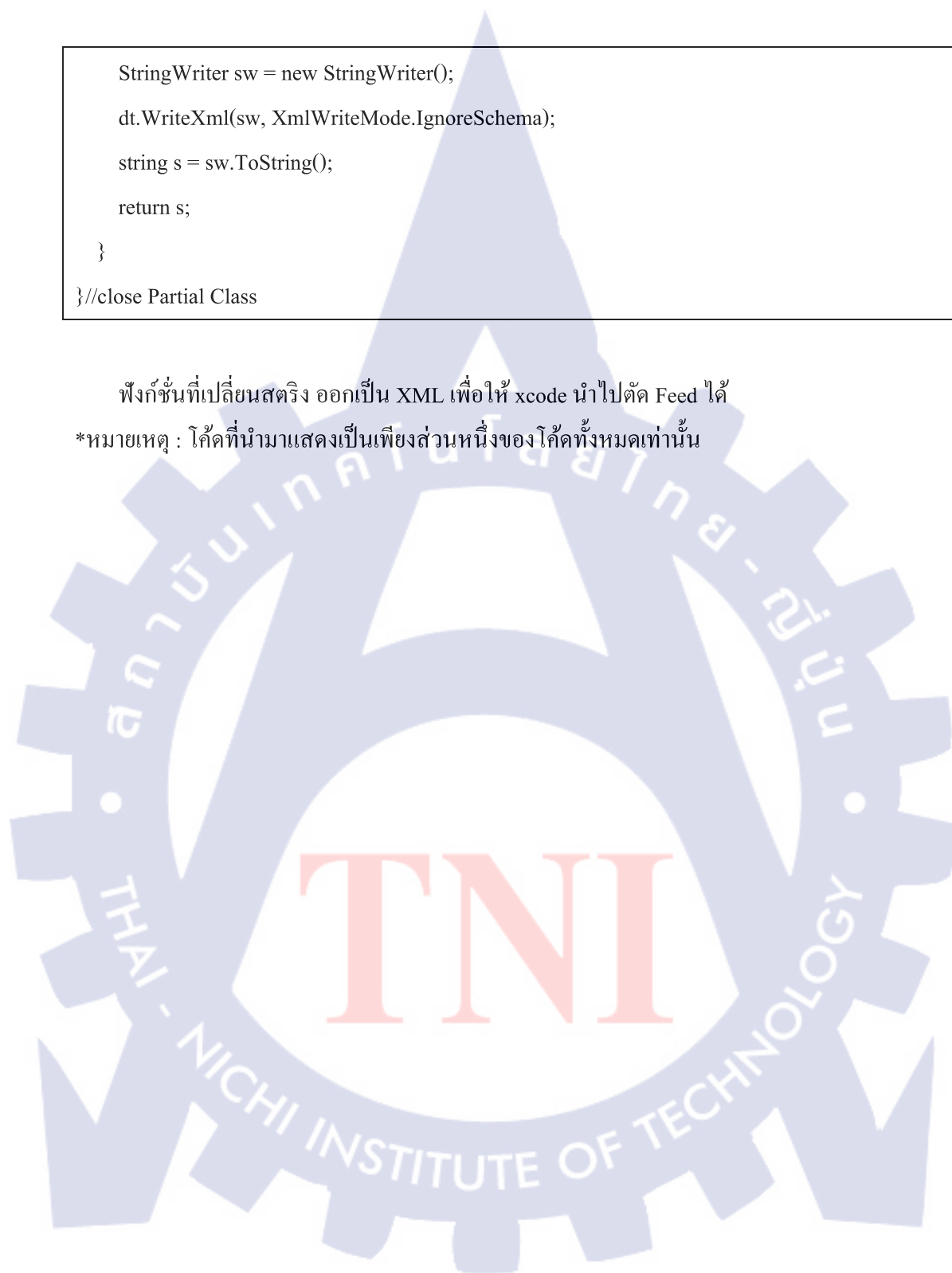
```
try {
    da.Fill(ds, "mapping");
    temp = new string[31];
    string ss = ToStringAsXml(ds.Tables["mapping"]);
    Response.Write(ss);
}
Catch {
    Response.Write("\t\t<invalid>Invalid Username and/or Password</invalid>\n");
}
} //close Page_Load
```

ส่วนของการเก็บค่าที่สืบค้นออกมาได้และ เขียนออกไป ถ้าสืบค้น ไม่เจอจะส่งค่า Invalid ออกไป

```
public static string ToStringAsXml(DataTable dt) {
```

```
StringWriter sw = new StringWriter();  
dt.WriteXml(sw, XmlWriteMode.IgnoreSchema);  
  
string s = sw.ToString();  
  
return s;  
  
}  
} //close Partial Class
```

ฟังก์ชันที่เปลี่ยนสตริง ออกเป็น XML เพื่อให้ xcode นำไปตัด Feed ได้
*หมายเหตุ : โค้ดที่นำมาแสดงเป็นเพียงส่วนหนึ่งของโค้ดทั้งหมดเท่านั้น



บทที่ 3

แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

3.1 แผนงานการปฏิบัติงาน

ตารางที่ 3.1 ตารางแสดงแผนการปฏิบัติงาน

แผนการดำเนินการ	ผู้รับผิดชอบ	Manday	ม.ย.		ก.ย.		ธ.ค.		ก.ย.	
			1-15	16-30	1-15	16-31	1-15	16-31	1-15	16-30
วางแผนการดำเนินงาน	ทุกคน	2.5								
จัดทำ proposal	รองพรณ	0.5								
นำเสนอโครงการ (รอบคัดเลือก)	รองพรณ/ เจลิ้มชัย	3								
Design	ทุกคน	12								
Develop	ทุกคน	20								
Testing	ทุกคน	5.5								
สรุปผลการดำเนินงาน	ทุกคน	2.5								

3.2 รายละเอียดงานที่ได้รับมอบหมาย

ข้อโปรเจก : GO for U Smart Alert (iAlert)

เนื่องจากปัจจุบันมีระบบแจ้งเตือนงานต่าง ๆ ผ่าน E-mail จำนวนมาก ซึ่งอาจจะทำให้การเตือนงานหรือข้อมูลที่ต้องทำการแก้ไขปรับปรุง ไม่ได้รับการตรวจสอบและดำเนินการภายในระยะเวลาที่กำหนด ซึ่งนอกจากจะมีผลโดยตรงต่อผู้รับผิดชอบแล้วนั้น ยังส่งผลกระทบต่อหน่วยงานในลำดับถัดไป ที่จะได้ข้อมูลไม่ครบถ้วนอีกด้วย

โปรเจกนี้จึงเป็นการสร้างแอปพลิเคชัน ที่ใช้ในการแจ้งเตือนผู้ใช้ โดยข้อมูลที่เตือนมาจากทางหลาย ๆ ฝ่าย เช่น ระบบ Human Resource, หน่วยงาน Project Management Office, ระบบ Lotus Note Workflow เพื่อเตือนให้ผู้ใช้ทราบว่ายังมีเรื่องที่ยังไม่ได้ทำเช่น การลงTimesheetไม่ครบ, ลืมขออนุมัติลา, รวมทั้ง แจ้งให้ทราบถึงเอกสารที่รออนุมัติด้วย

หน้าที่ที่ได้รับมอบหมายให้ทำการออกแบบระบบ และอินเทอร์เฟซ ทั้งหมดของแอปพลิเคชัน ตัวนี้ให้มีความเหมาะสมกับบริษัท และใช้งานได้ง่าย โดยศึกษาจากแอปพลิเคชัน ที่บริษัทเคยสร้างขึ้นมา และทำการวิเคราะห์ปรับแต่งให้มีความเหมาะสมเพื่อให้ผู้ใช้งานมีความสนใจในเวลาที่มีการแจ้งเตือนจากทางระบบ

3.3 ขั้นตอนการดำเนินงาน

ขั้นตอนการปฏิบัติงานจะดำเนินการเป็นขั้นตอนตามตารางที่ 3.1 โดยจะอธิบายโดยละเอียดได้ดังนี้

3.3.1 วางแผนการดำเนินงาน

ขั้นตอนการวางแผนเริ่มจากการสำรวจจากพนักงานว่าต้องมีการลง Time Sheet และ การส่งใบลา โดยจะมีการแจ้งเตือนว่า จำเป็นต้องลง Time Sheet และ การลาที่ยังไม่ได้ส่งใบลา ผ่านมาทางอีเมลล์ ของพนักงานทุกคน เพื่อให้จัดการให้เรียบร้อย แต่พบปัญหาบางอย่างเนื่องจากอีเมลล์ ที่มีเข้ามาใหม่ในแต่ละวันมีจำนวนมากทำให้พนักงานไม่ได้สังเกตอีเมลล์ และมีการวิเคราะห์ว่าปัญหาที่เกิดขึ้นส่งผลกระทบต่ออย่างไร ซึ่งจากการวิเคราะห์จะเห็นได้ว่า ส่งผลให้กระบวนการทำงานของบริษัทเกิดความล่าช้า เพราะหน่วยงานถัดไปไม่สามารถทำงานต่อได้เนื่องจากรอข้อมูลจากพนักงาน วิธีการแก้ปัญหา จึงได้มีการคิดว่า ถ้าเปลี่ยนจากอีเมลล์ มาเป็นการแจ้งเตือนบน iPhone แทนจะทำให้พนักงานเจ้าของเครื่องสามารถรับรู้ได้รวดเร็วและง่ายยิ่งขึ้น จึงมีการเขียนแผนผังการทำงานและออกแบบแอปพลิเคชัน ที่จะมารองรับในส่วนนี้ และกำหนดระยะเวลาในการพัฒนาแอปพลิเคชัน

3.3.2 จัดทำใบเสนองาน (Proposal)

ขั้นตอนในการจัดทำใบเสนองานจะต้องมีการระบุข้อมูลของโครงการให้ครบถ้วน โดยมีหัวข้อ รูปแบบของนวัตกรรม วัตถุประสงค์ เป้าหมาย และระยะเวลาของโครงการ เพื่อส่งเข้าประกวดโครงการนวัตกรรม ของบริษัท

3.3.3 นำเสนอโครงการ (รอบแรก)

จัดทำ Power Point และนำไปนำเสนอให้กับคณะกรรมการ เพื่อคัดเลือกเข้าแข่งขัน

3.3.4 การออกแบบ (Design)

เนื่องด้วยโปรเจกต์นี้เป็นโปรเจกต์ที่ทำคู่กับนายณัฐพล เลขวิชกุล ซึ่งมีหน้าที่รับผิดชอบในส่วนของการออกแบบ Database ก่อน จากนั้นผู้เขียนจึงเริ่มในส่วนของการออกแบบอินเทอร์เฟซ ซึ่งจะเน้นการใช้ธีมของบริษัทเป็นหลัก และจัดวางปุ่ม ช่องใส่ข้อความ ตารางแสดงข้อความ ตามความเหมาะสม โดยมีการทดลองใช้เองเพื่อคาดคะเนความง่ายในการใช้งานของผู้ใช้

3.3.5 การพัฒนา (Develop)

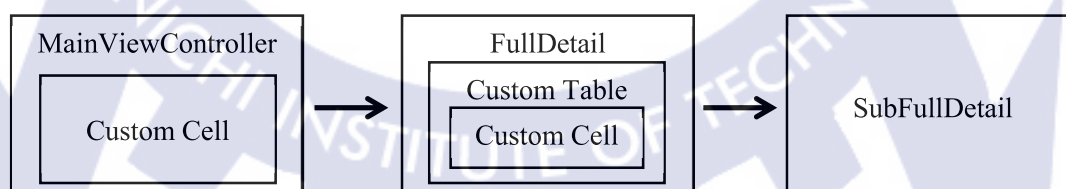
ต่อจากขั้นตอนการออกแบบอินเทอร์เฟซ สำเร็จแล้ว จึงมาเริ่มการเขียนโปรแกรมในส่วนของฟังก์ชันการทำงานของโปรแกรม โดยเขียนเป็น ภาษาอ็อบเจกทีฟ-ซีซึ่งใช้หลักการ Object Oriented Programming (OOP) ในการเขียน จากนั้นจึงนำมาทำการรวมกับระบบฐานข้อมูลที่พัฒนาแยกไว้

ขั้นตอนต่อไป จึงเป็นส่วนของการสร้างตัวกลางในการติดต่อกับระบบฐานข้อมูล ซึ่งฐานข้อมูลที่จะใช้นั้นเป็นฐานข้อมูลที่สร้างขึ้นใหม่ เพื่อรองรับการทำงานของระบบนี้ โดยเป็นการรวมข้อมูลทั้งหมดจากฐานข้อมูลของฝ่ายและระบบต่าง ๆ ดังนี้ ระบบ HR, ระบบ LNW และฝ่าย PMO

แต่เนื่องด้วยมีความจำเป็น ต้องขอการอนุมัติใช้เครื่องเซิร์ฟเวอร์ของทางบริษัท จึงเขียนตัวกลางในการติดต่อฐานข้อมูลด้วยภาษา PHP ใน Local host ก่อนเพื่อนำไปใช้ในการทดลองตัด Tag XML จากนั้นจึงเปลี่ยนมาใช้ C#.net หลังจากที่ได้รับอนุมัติใช้เครื่องเซิร์ฟเวอร์เพื่อเพิ่มความเสถียรให้กับระบบและเนื่องด้วยแอปพลิเคชันเป็นระบบที่ใช้ XML ในการรับส่งข้อมูลอยู่แล้ว จึงเป็นภาษากลางที่สามารถนำไปใช้งานได้ในทุกแพลตฟอร์ม โดยมีการเขียนและแบ่งฟังก์ชันการทำงานออกเป็นดังนี้

3.3.5.1 เนวิเกชัน คอนโทรลเลอร์ (Navigation Controller)

แอปพลิเคชันมีโครงสร้างหลักเป็นการทำงานแบบ Navigation Controller คือมีลักษณะเป็นตารางที่สามารถเลือกไปดูหน้าถัดไปเพื่อดูข้อมูลได้โดยมีหน้าหลักดังนี้



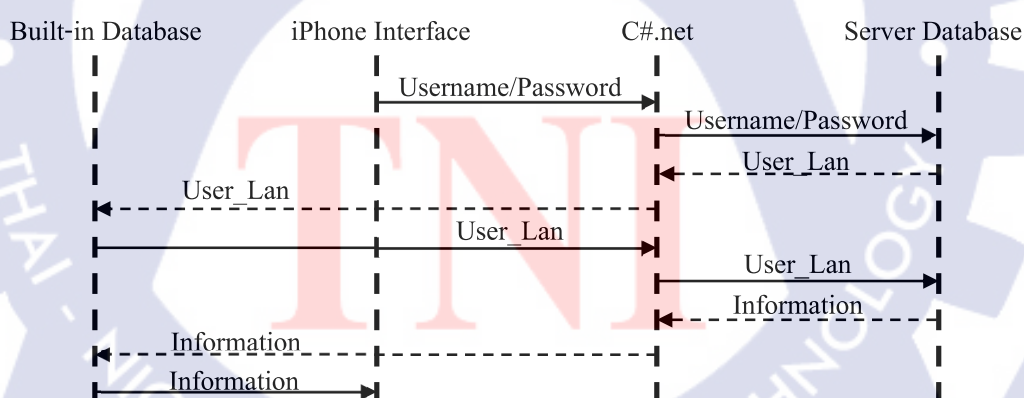
รูปที่ 3.1 ฟังก์ชันการทำงานของ Navigation Controller

โดยในหน้า MainViewController นั้นจะประกอบด้วย Navigation Controller ซึ่งใช้ Custom Cell เมื่อเลือกแถวของ TableView แล้ว จะเปลี่ยนหน้าไปยังหน้า FullDetail และในหลักการเดียวกันสามารถเลือกแถวและเปลี่ยนหน้าต่อไปยังหน้า SubFullDetail ได้

3.3.5.2 การยืนยันตัวตน

ก่อนที่จะใช้งานหน้า MainViewController ได้นั้นจะต้องผ่านการเข้าสู่ระบบ เพื่อยืนยันตัวตนของผู้ใช้ก่อน โดยการนำ Username และ Password ของผู้ใช้ไปตรวจสอบในระบบฐานข้อมูลพนักงาน ของบริษัท เพื่อเป็นการรักษาความลับของแต่ละบุคคล จากนั้น ก็จะนำ User ที่ได้จากการตอบกลับจาก Server ไป Query หาข้อมูลทำงานที่ยังไม่ได้ทำของแต่ละบุคคล ใน User Table ของเซิร์ฟเวอร์และส่งกลับมาให้ผู้ใช้ในลักษณะของ Tag XML เพื่อให้แอปพลิเคชัน นำไปตัด tag และนำข้อมูลเก็บลงฐานข้อมูลภายในของเครื่อง ก่อนจะนำมาแสดง

โดย Username ที่ใช้ยืนยันตัวตนครั้งแรก และผ่านการยืนยันจากเซิร์ฟเวอร์แล้ว จะถูกบันทึกลงฐานข้อมูลภายในของเครื่อง เพื่อที่จะจำว่า Username ที่ใช้ยืนยันตัวตนครั้งแรกนั้นเป็นเจ้าของเครื่อง เพื่อในกรณีที่ไม่มีกรเชื่อมต่ออินเทอร์เน็ตก็ยังสามารที่จะยืนยันตัวตนผ่านหน้าจอเข้าสู่ระบบของแอปพลิเคชันเพื่อดูข้อมูลได้



รูปที่ 3.2 Sequence Diagram ของระบบยืนยันตัวตนแบบมีการเชื่อมต่อ

3.3.5.3 การแจ้งเตือนเมื่อไม่มีการเชื่อมต่ออินเทอร์เน็ต

เมื่อเปิดแอปพลิเคชันขึ้นมา แอปพลิเคชันจะทำการตรวจสอบว่ามีการเชื่อมต่ออินเทอร์เน็ตอยู่หรือไม่เพื่อที่จะทำข้อมูลให้เป็นปัจจุบันที่สุด ดังนั้นการทำงานจึงแบ่งออกเป็น 2 กรณีคือ การใช้งานแอปพลิเคชันขณะที่มีการเชื่อมต่ออินเทอร์เน็ต และ การใช้งานแอปพลิเคชัน ขณะที่ไม่มี การเชื่อมต่ออินเทอร์เน็ต ดังนี้

การทำงานขณะที่มีการเชื่อมต่ออินเทอร์เน็ตนั้นจะทำการส่ง Username และ Password ที่ผู้ใช้กรอกไปที่เครื่องเซิร์ฟเวอร์ส่ง User_Lan กลับมาที่เครื่องผ่าน C#.net ก่อนจะส่งกลับไปให้เซิร์ฟเวอร์ ผ่าน C#.net อีกครั้ง เพื่อค้นหาข้อมูลของตัวบุคคลที่เข้าสู่ระบบและส่งข้อมูลกลับมาผ่าน C#.net

การทำงานขณะที่ไม่มีการเชื่อมต่ออินเทอร์เน็ตนั้นจะทำการตรวจสอบในฐานข้อมูลภายในของเครื่อง ว่า Username และ Password ตรงกับ Username ในเครื่องหรือไม่ โดย Username ที่เก็บในฐานข้อมูลภายในนั้นจะมีเพียง 1 User เท่านั้นคือ Username ที่ใช้ยืนยันตัวตน ถูกต้องเมื่อตรวจสอบกับเซิร์ฟเวอร์ครั้งแรก เนื่องจากการกำหนดให้แอปพลิเคชัน มีการทำงานแบบ 1 user : 1 device และสามารถดูข้อมูลได้แม้ไม่มีการเชื่อมต่ออินเทอร์เน็ต

3.3.5.4 ภาพเคลื่อนไหว (Animation)

การใช้งานภาพเคลื่อนไหว (Animation) มีการใช้งานหลัก ๆ ออกเป็น 2 แบบ ดังนี้

1. แบบเป็นพารามิเตอร์ ออฟชั่นของฟังก์ชัน

```
[self presentViewController:auth2 animated:YES];
```

เป็นการใช้เพื่อให้การทำงานในส่วนของอินเทอร์เฟซราบรื่นขึ้นโดยผู้ใช้งานจะมองเห็นการเคลื่อนไหวในช่วงเวลาที่เกิดการเปลี่ยนแปลงก่อนการเปลี่ยนแปลงจะเสร็จสิ้น หากตั้งค่าเป็น animated:NO ผู้ใช้จะเห็นโปรแกรมเปลี่ยนแปลงโดยไม่เห็นเป็นภาพเคลื่อนไหว

2. แบบสั่งให้อ็อบเจกต์ ทำงานให้เสร็จสิ้นเป็นช่วงเวลา

```
[UIView *view animateWithDuration:4.0f animations:^(
    tempImage. tranform = CGAffineTransformMakeRotation(-22/7);
)];
```

เป็นการสั่งให้ ทำบรรทัดที่อยู่ภายใน { } ให้เสร็จสิ้นภายในเวลา 4.0 วินาที หรือตามเวลาที่กำหนด มักใช้ร่วมกับ CGAffineTransform ซึ่งเป็นการสั่งให้อ็อบเจกต์เคลื่อนไหว ตามหลักการของ Animation โดยประกอบด้วยการเคลื่อนที่ (Translate), การย่อ/ขยาย (Scale), การหมุนภาพ (Rotation)

3.3.5.5 การตั้งเวลา (Timer)

เป็นการสร้างดีเลย์ของฟังก์ชันการทำงานคือเมื่อสั่ง Timer แล้วจะเกิดดีเลย์ เป็นจำนวนวินาทีที่ตั้งค่าไว้ก่อนจะไปเรียกฟังก์ชัน ที่ตั้งค่าไว้ตามลำดับ โดยใช้ในการเรียก ฟังก์ชันต่าง ๆ ระหว่างรอการทำงานที่ยังทำไม่เสร็จ

```
progressTimer = [NSTimer scheduledTimerWithTimeInterval:1f target:self
selector:@selector(progressRun) userInfo:nil repeats:YES];
```

ดังโค้ดการทำงานที่เห็นคือ เมื่อ บรรทัดนี้ถูก execute จะทำการเรียกฟังก์ชันprogressRun ในอีก 1 วินาทีข้างหน้า และหากต้องการส่งพารามิเตอร์เข้าไปในฟังก์ชัน ต้องส่งผ่าน userInfo:(parameter) และต้องไปเขียนรับค่า userInfo ในฟังก์ชัน progressRun ด้วย

3.3.5.5 คอนโทรลเลอร์แถบเครื่องมือ (Toolbar Controller)

ใช้ในการเพิ่มพื้นที่ในการวางปุ่มออกพชั่นต่าง ๆ ของโปรแกรมลงไปแอพพลิเคชั่น นี้เพิ่มขึ้นเพื่อใช้เป็นพื้นที่ในการวางปุ่ม Refresh และ ปุ่ม Log out การสร้างสามารถสร้างได้ด้วยโค้ดและ IBOutlet



รูปที่ 3.3 การแสดงคอนโทรลเลอร์ที่เป็นแถบเครื่องมือ

3.3.5.6 การแจ้งเตือน (Notification)

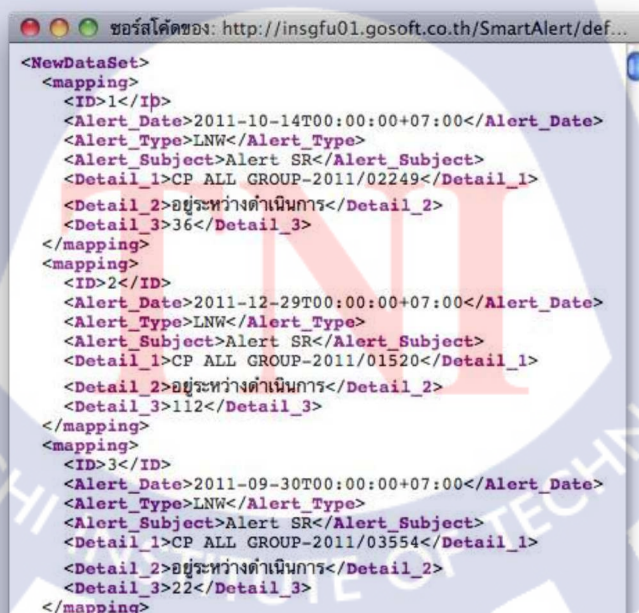
สำหรับแอปพลิเคชันนี้จะใช้ Local Notification ซึ่งเป็นการแจ้งเตือนผู้ใช้ตามเวลาที่ถูกต้องไว้โดยอ้างอิงจากเวลาจากเครื่องโดยการตั้งเวลานั้นมาจากการตั้งค่า ด้วย UIPickerView และ เก็บเวลาลงฐานข้อมูล โดยเมื่อถึงเวลาที่ตั้งไว้ แอปพลิเคชันจะเตือนขึ้นมาเป็นบันทึกที่สร้างเอาไว้ของแต่ละหมวด

3.3.5.7 การส่ง XML Parser

สำหรับการติดต่อกับฐานข้อมูลบนเซิร์ฟเวอร์นั้นจะติดต่อผ่าน C#.net เพื่อจำกัดการติดต่อกับฐานข้อมูลโดย C#.net จะทำการแปลงข้อมูลให้ออกมาในแพ็คเกจ XML เพื่อที่จะทำให้เป็นภาษากลางและสามารถนำไปใช้กับแพลตฟอร์มใดก็ได้ที่มาติดต่อเพื่อขอข้อมูล

```
1 2011-10-14T00:00:00+07:00 LNW Alert SR CP ALL GROUP-2011/02249 อยู่ระหว่างดำเนินการ 36
2 2011-12-29T00:00:00+07:00 LNW Alert SR CP ALL GROUP-2011/01520 อยู่ระหว่างดำเนินการ 112
3 2011-09-30T00:00:00+07:00 LNW Alert SR CP ALL GROUP-2011/03554 อยู่ระหว่างดำเนินการ 22
4 2011-04-29T00:00:00+07:00 LNW Alert SR CP ALL GROUP-2011/01894 รอสรูป Requirement 131
5 2011-08-18T00:00:00+07:00 LNW Alert SR GOS-2011/00862 รอกาประมาณเวลาและค่าใช้จ่าย 20
```

รูปที่ 3.4 XML ที่เซิร์ฟเวอร์ส่งกลับมา



รูปที่ 3.5 ซอร์สโค้ด (Source Code) ของ XML ที่เซิร์ฟเวอร์ส่งกลับมา

3.3.6 การทดสอบระบบ (Testing)

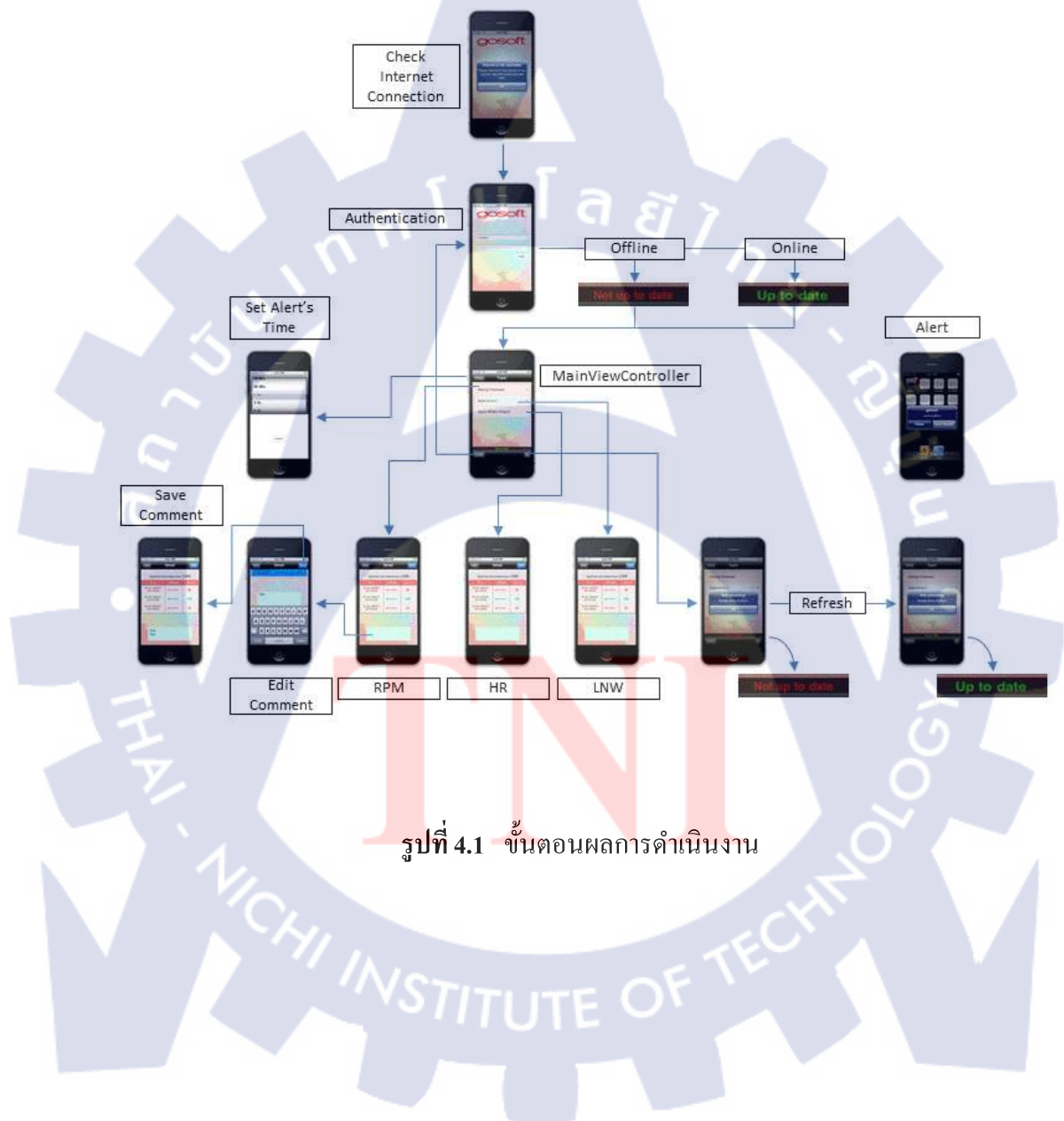
กระบวนการทดสอบระบบ เนื่องจากเป็นโครงการนวัตกรรมขนาดเล็ก หน้าที่ทดสอบระบบจึงเป็นการตกลงกันระหว่างผู้คุมโครงการ (Project Manager) และผู้เขียนโปรแกรม (Programmer) และเนื่องจากการประชุมโครงการเป็นการประชุมนอกเวลางานสัปดาห์ละ 1 ครั้ง โดยในแต่ละครั้ง PM จะบอกจุดที่ต้องแก้ไข ดังนั้น เบลไลน์จึงมีหลายเวอร์ชันมาก และเนื่องด้วย PM เองก็เป็นผู้ใช้แอปพลิเคชัน เพราะเป็นโครงการภายใน ดังนั้น Testing Phase, UAT (User Acceptance Test) Phase จึงรวมเป็นการทดสอบครั้งเดียว



บทที่ 4

ผลการดำเนินงาน การวิเคราะห์และสรุปผลต่าง ๆ

4.1 ขั้นตอนและผลการดำเนินงาน



รูปที่ 4.1 ขั้นตอนผลการดำเนินงาน



รูปที่ 4.2 การแจ้งเตือนเมื่อไม่มีการเชื่อมต่ออินเทอร์เน็ต

ถ้าหากเปิดแอปพลิเคชัน ขึ้นมาแล้วไม่มีการเชื่อมต่ออินเทอร์เน็ตจะมีการแจ้งเตือนขึ้นมาว่า Internet is not reachable เพื่อเตือนให้ผู้ใช้รู้ว่า ไม่สามารถตรวจสอบได้ว่าข้อมูลในเครื่องนั้นล่าสุดเท่าในเซิร์ฟเวอร์หรือไม่

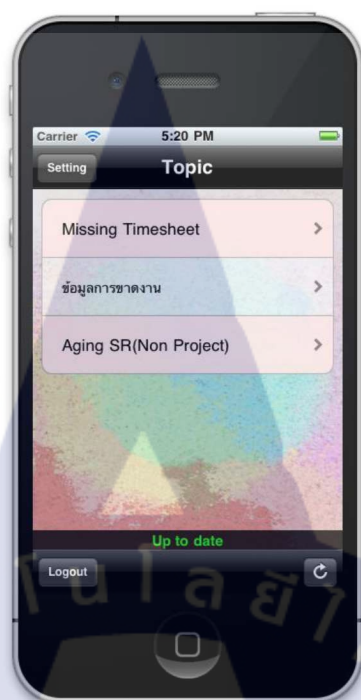
TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY



รูปที่ 4.3 หน้าจอสำหรับเข้าสู่ระบบ

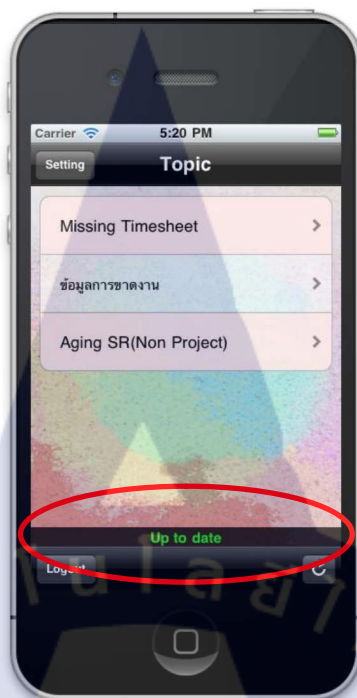
เป็นหน้าจอที่ประกอบด้วย ช่องสำหรับใส่ Username และ Password เพื่อส่งไปที่ C#.net ที่เป็นตัวกลางในการเชื่อมต่อฐานข้อมูล เพื่อตรวจสอบ Username กับ User_Lan ซึ่งมีลักษณะเป็น Single-Sign on โดยเมื่อผ่านการตรวจสอบแล้วและสามารถยืนยันได้ก็จะส่งค่า User_Lan กลับมา เพื่อนำไปค้นข้อมูลของ User_Lan ดังกล่าวในระบบฐานข้อมูลและส่งกลับมา



รูปที่ 4.4 เมนูหลักของการทำงาน

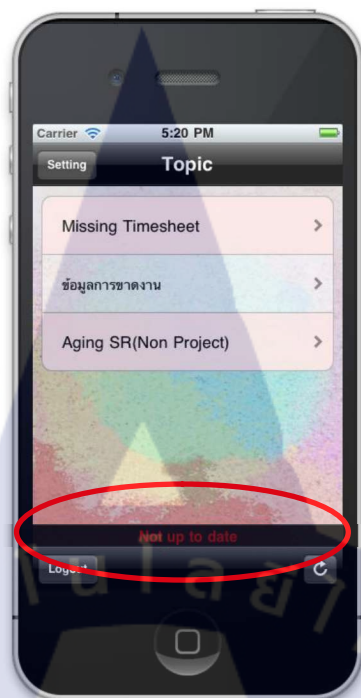
หน้าจอการทำงานหลักของโปรแกรม ประกอบด้วยข้อมูล Missing Timesheet จากฝ่าย Project Management Office, ข้อมูลการขาดงานจากระบบ Human Resource และ ข้อมูล Aging SR(Non Project) จากฝ่าย Lotus Note Workflow เพื่อแจ้งให้ผู้ใช้ทราบ

ด้านบนเป็น Navigation Bar ประกอบด้วยปุ่ม Setting เพื่อไว้ใช้ตั้งเวลาของการแจ้งเตือน ด้านล่างเป็นแถบเครื่องมือ (Tool Bar) ประกอบด้วย ปุ่ม Log Out, ปุ่ม Refresh



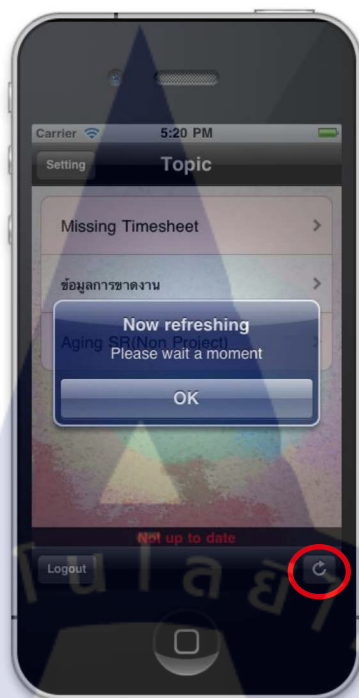
รูปที่ 4.5 การยืนยันตัวตนเมื่อมีการเชื่อมต่ออินเทอร์เน็ต

หากยื่นเข้าสู่ระบบและยืนยันตัวตนผ่านอินเทอร์เน็ต จะเห็นคำว่า Up to date ดังรูป แสดงว่าแอปพลิเคชัน ที่ส่ง User_Lan ไปที่ เซิร์ฟเวอร์ และ รับข้อมูลที่เป็นข้อมูลล่าสุด แล้วนั่นเอง



รูปที่ 4.6 การเข้าสู่ระบบโดยไม่มีการเชื่อมต่ออินเทอร์เน็ต

หากเข้าสู่ระบบโดยไม่เชื่อมต่ออินเทอร์เน็ต จะเห็นคำว่า Not up to date ดังรูป เนื่องจากไม่สามารถเช็คได้ว่า ข้อมูลที่มีอยู่ในเครื่องนั้นเป็นข้อมูลที่ล่าสุดแล้วหรือไม่



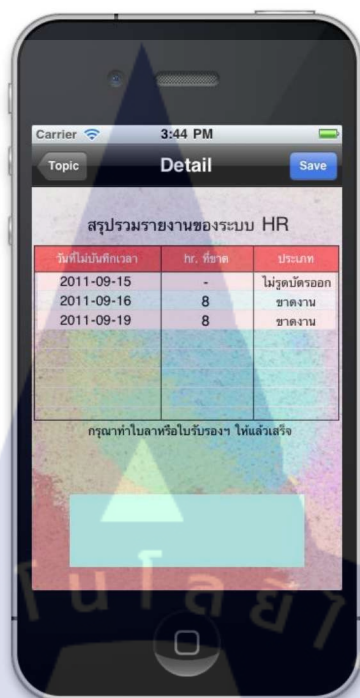
รูปที่ 4.7 รูปแสดงการแจ้งเตือนเมื่อทำการกดรีเฟรช

การรีเฟรช(Refresh) สามารถทำได้ด้วยการกดปุ่ม  บนแถบเครื่องมือด้านล่างของหน้าจอการทำงาน ระบบจะส่งคำร้องขอ ไปรับข้อมูลจากเซิร์ฟเวอร์ ใหม่อีกครั้ง และมีแจ้งเตือนว่า Now Refreshing เนื่อง จากกำลังส่งค่าไปที่เซิร์ฟเวอร์



รูปที่ 4.8 การแสดงข้อมูลของระบบ RPM

ระบบ RPM (Rational Portfolio Manager) คือ Missing Timesheet ที่ใช้ภายในเพื่อบอกว่า วันไหนทำงานขาดไปกี่ชั่วโมง ในสัปดาห์ที่เท่าไร



รูปที่ 4.9 การข้อมูลของระบบ HR

ระบบ HR (Human Resource) คือระบบที่ใช้ภายในบริษัทเพื่อแจ้งข้อมูลการขาดงานของพนักงานว่า วันไหนที่ลงเวลาไม่ครบ หรือ ขาดงาน หรือ ไม่รู้บัตรเข้า-ออก โดยหากไม่แก้ไขให้เรียบร้อยอาจเกิดผลต่อเงินเดือนของพนักงานคนนั้นจึงมีระบบที่คอยแจ้งเตือนให้พนักงานทราบ

SR No.	วันที่อนุมัติ	Aging (วัน)
CP ALL GROUP-2011/02249	อนุมัติแล้ว...	36
CP ALL GROUP-2011/01520	อนุมัติแล้ว...	112
CP ALL GROUP-2011/03554	อนุมัติแล้ว...	22

รูปที่ 4.10 การแสดงข้อมูลของระบบ LNW

ระบบ LNW (Lotus Note Workflow) คือระบบที่แจ้งอายุของ SR (Service Requirement) ที่ไม่ได้เป็นโปรเจกต์ โดยจะแสดงหมายเลขของ SR และ วันที่อนุมัติหากได้รับการอนุมัติแล้ว และถ้าหากยังไม่ได้อนุมัติจะแสดงว่าอยู่ระหว่างการดำเนินการ รวมถึงแสดงอายุของ SR นั้น ๆ ด้วย



รูปที่ 4.11 การตั้งเวลาให้แจ้งเตือนบันทึกของผู้ใช้

แอปพลิเคชันสามารถตั้งเวลาให้ระบบแจ้งเตือนบันทึกของผู้ใช้ได้ และจะเตือนวนรอบเตือน
ผู้ใช้เรื่อย ๆ จนกว่าจะลบข้อความในบันทึกทิ้ง



รูปที่ 4.12 การใส่บันทึกลงไปในแต่ละหมวด

แอปพลิเคชันสามารถใส่บันทึกลงไปในแต่ละหมวดสำหรับจดข้อความที่ต้องการได้ และสามารถตั้งให้แจ้งเตือนบันทึกที่ใส่ลงไป ตามเวลาที่ต้องการได้



รูปที่ 4.13 การแสดงข้อความที่บันทึกไว้ผ่านการแจ้งเตือน

เมื่อครบกำหนดเวลาที่ตั้งไว้ แอปพลิเคชันจะทำการเตือนข้อความที่ผู้ใช้บันทึกลงไปในช่วงบันทึกของแต่ละหมวด

4.2 ผลการวิเคราะห์ข้อมูล

จากการสำรวจผู้ทดลองใช้แอปพลิเคชัน พบว่าสามารถช่วยให้รับทราบข่าวสารได้ดีขึ้นในระดับหนึ่ง โดยข้อมูลถูกส่งมาที่อีเมลแล้ว แต่ไม่มีเวลาว่างที่จะไปเปิดคอมพิวเตอร์เพื่อดูอีเมลนั้น จึงทำให้ลืมน แต่เมื่อใช้แอปพลิเคชันนี้แล้วผู้ใช้สามารถเข้าไปดูในช่วงเวลาที่ว่างได้โดยไม่ต้องเสียเวลาเปิดคอมพิวเตอร์ เช่น ช่วงเวลาออฟฟิศ เป็นต้น

เมื่อข้อมูลไปถึงผู้ใช้งานเร็วขึ้น จะส่งผลให้กระบวนการถัดไปที่รอข้อมูลจากผู้ใช้ หรือ รออนุมัติจากผู้ใช้งานสามารถดำเนินงานในขั้นตอนต่อไปได้รวดเร็วยิ่งขึ้น

4.3 วิเคราะห์และวิจารณ์ข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมายในการปฏิบัติงานหรือการจัดทำโครงการ

เมื่อทำการเปรียบเทียบกับวัตถุประสงค์, จุดมุ่งหมายในการจัดทำโครงการ พบว่าในการทำงานจริงของแอปพลิเคชัน (Application) นั้นสามารถทำให้ผู้ใช้งานรับรู้ข่าวสารได้สะดวกและรวดเร็วยิ่งขึ้น ส่งผลให้การทำงานในแต่ละกระบวนการรวดเร็วยิ่งขึ้น และ โอกาสในการที่ไม่ได้รับข่าวสารจนถูกตัดเงินเดือนมีน้อยลง

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลดำเนินงาน

การดำเนินงานในโครงการที่ได้รับมอบหมายให้ปฏิบัติงาน ช่วยสร้างประสบการณ์ในการทำงานจริงที่จะเกิดขึ้นในอนาคต และสามารถจำแนกเป็นข้อหลักได้ดังต่อไปนี้

- สร้างความรับผิดชอบในการทำงานให้ลุล่วงภายในระยะเวลาที่กำหนด
- สร้างวินัย และการปฏิบัติตัวอย่างเหมาะสมกับการทำงาน
- สร้างการปฏิบัติงานร่วมกับผู้อื่นภายในองค์กร
- สร้างประสบการณ์ในการนำเสนองานและ การแสดงความคิดเห็นในที่ประชุม

และในระหว่างปฏิบัติงานก็ได้ประสบปัญหาต่าง ๆ มากมาย เช่น

- ขาดความสามารถในการเขียนโปรแกรมโดยใช้ภาษา Objective-C
- ขาดบุคลากรในการให้ความรู้เกี่ยวกับการเขียนแอปพลิเคชันบน iOS
- ในบางครั้งมีความจำเป็นต้องรื้อฐานข้อมูลจากระบบ/ฝ่ายต่าง ๆ

5.2 แนวทางการแก้ไขปัญหา

จากปัญหาทั้งหมดที่ได้กล่าวถึงภายในหัวข้อสรุปผลการดำเนินงาน เราสามารถที่จะทำการหาวิธีการแก้ไขปัญหาเหล่านั้นได้ดังนี้

ปัญหา : ขาดความสามารถในการเขียนโปรแกรมโดยใช้ภาษา Objective-C

แนวทางการแก้ไขปัญหา : ทำการค้นคว้าบนอินเทอร์เน็ต เพื่อหาความรู้และวิธีการในการเขียนโปรแกรม ถึงแม้ว่าจะมีพื้นฐานในการเขียนโปรแกรมด้วยภาษา C, พื้นฐานการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ด้วยภาษา C++ แล้วก็ตามแต่บางอย่างก็จำเป็นต้องศึกษาเพิ่มเติมเช่น

การใช้ระบบฐานข้อมูล, การจองหน่วยความจำ และการคืนหน่วยความจำ เพราะการทำงานบนเครื่อง iPhone ที่มีหน่วยความจำจำนวนจำกัดนั้น ต้องควบคุมการจองหน่วยความจำให้ดีเพื่อป้องกันการเกิดหน่วยความจำรั่ว (Memory Leak)

ปัญหา : ขาดบุคลากรในการให้ความรู้เกี่ยวกับการเขียนแอปพลิเคชันบน iOS

แนวทางการแก้ไขปัญหา : ทำการค้นหาคำความรู้เพิ่มเติมจากทางอินเทอร์เน็ตเช่นเดียวกับการเรียนรู้ในภาษาอ็อบเจกทีฟ-ซี ซึ่งจะเน้นการไปโหลดแอปพลิเคชันตัวอย่างมาเพิ่มศึกษาแนวทางการเขียนแอปพลิเคชันต่าง ๆ

ปัญหา : ในบางครั้งมีความจำเป็นต้องรื้อฐานข้อมูลจากระบบ/ฝ่ายต่าง ๆ

แนวทางการแก้ไขปัญหา: รบกวนพี่ที่เป็นคนดูแล ประสานงานไปที่ฝ่ายพัฒนา และประสานต่อไปยังผู้ควบคุมดูแลระบบ เพื่อขอใช้งานฐานข้อมูลบนเซิร์ฟเวอร์ และ ขอข้อมูลมาเก็บในฐานข้อมูลเพื่อที่จะนำมาแสดงบน iPhone ผ่าน C#.net

5.3 ข้อเสนอแนะจากการดำเนินงาน

จากการดำเนินงานทำให้รู้จักอ่อนหลายอย่างของการเป็นผู้เริ่มต้นเขียนโปรแกรมบน iOS ที่บางฟังก์ชันการทำงานสามารถเปลี่ยนไปใช้วิธีอื่นที่ทำให้ฟังก์ชันนั้นมีประสิทธิภาพมากขึ้น แต่การเขียนในฐานข้อมูลผู้เริ่มต้นและมีระยะเวลาจำกัดจึงไม่ทราบว่าวิธีที่การนั้นในขณะที่เขียนแอปพลิเคชันนี้ขึ้นมา ดังนั้นควรมีผู้เชี่ยวชาญในแพลตฟอร์มที่เป็นเป้าหมายการเขียนเข้าร่วมการประชุมออกแบบผังการทำงานด้วย เพื่อให้แอปพลิเคชันเกิดประสิทธิภาพสูงสุด และนำไปก่อให้เกิดประโยชน์ต่อองค์กรได้

ถึงแม้ว่าการเขียน C#.net จะทำให้ได้ข้อมูลออกมาเป็นภาษา XML ที่เป็นภาษากลางที่ไม่ว่าแพลตฟอร์มไหนก็สามารถนำไปตัดข้อมูลได้ แต่การเขียนแอปพลิเคชันที่ทำงานบน iOS นั้นมีข้อจำกัดตรงที่ว่า หากต้องการให้พนักงานในองค์กรใช้แอปพลิเคชันที่พัฒนาขึ้นต้องมีเครื่อง iPhone/iPod/iPad เท่านั้นจึงจะสามารถใช้งานแอปพลิเคชันนี้ได้ ดังนั้นควรมีการเขียนการทำงานในแพลตฟอร์มอื่นเช่นแอนดรอยด์ประกอบด้วย เพื่อลดต้นทุนในการให้เครื่องมือกับพนักงานเนื่องจากแอนดรอยด์ที่สเปกไม่สูงนั้นมีราคาตลาดต่ำกว่าเครื่องที่ใช้ระบบ iOS

เอกสารอ้างอิง

1. SQL Tutorial [Online], Avariable : <http://www.thaicreate.com/tutorial/sql.html> [2011, September 12].
2. Objective-C [Online], Avariable : <http://en.wikipedia.org/wiki/Objective-C> [2011, September 12].
3. Xcode 4 [Online], Avariable : <http://www.idevthai.com/xcode4/> [2011, September 12].
4. Stackoverflow [Online], Avariable <http://www.stackoverflow.com> [2011, September 12].
5. iPhone Dev SDK [Online], Avariable <http://www.iphonedevsdk.com> [2011, September 12].
6. iPhone SDK Article [Online], Avariable <http://www.iphonesdkarticles.com> [2011, September 12].
7. Khomkrit Article [Online], Avariable <http://khomkrit.blogspot.com> [2011, September 12].
8. Xcode Tutorials [Online], Avariable <http://www.xcode-tutorials.com/parsing-xml-files> [2011, September 12].
9. Apple Developer Forums [Online], Avariable https://discussions.apple.com/community/developer_forums [2011, September 12].
10. Youtube [Online], Avariable <http://www.youtube.com> [2011, September 12].
11. Mark Dalrymple and Scott Knaster, 2009, Learn Objective-C on the Mac, พิมพ์ครั้งที่ 1, สำนักพิมพ์ Apress.
12. Dave Mark and Jeff Lamarche, 2009 , Beginning iPhone Development Exploring the iPhone SDK, พิมพ์ครั้งที่ 1, สำนักพิมพ์ Apress
13. Maher Ali, 2009, iPhone SDK Programming, พิมพ์ครั้งที่ 1, สำนักพิมพ์ WHLEY
14. Erica Sadun, 2009, The iPhone Developer's Cookbook, พิมพ์ครั้งที่ 1, สำนักพิมพ์ Addison-Wesley

15. ชาญชัย ศุภอรธกร, 2553, สร้างเว็บอีคอมเมิร์ซด้วย PHP+MySQL ฉบับสมบูรณ์,
สำนักพิมพ์ ชิงค์ ปียอนด์ บุ๊คส์, บจก.
16. ศุภชัย สมพานิช, 2553, Basic ASP.NET 4.0, สำนักพิมพ์ ชิมพลีฟาย
17. รวิทัต ภู่อล้า, 2554, คู่มือเขียน iPhone Apps, พิมพ์ครั้งที่ 1, สำนักพิมพ์ Provision



ประวัติผู้วิจัย**ชื่อ - นามสกุล**

นายพงษ์พี อัครธัญย์

วัน เดือน ปีเกิด

22 กุมภาพันธ์ พ.ศ. 2533

ประวัติการศึกษา**ระดับมัธยมศึกษา**

มัธยมศึกษาตอนต้น-ปลาย พ.ศ. 2545

โรงเรียนอัสสัมชัญ สมุทรปราการ

ระดับอุดมศึกษา

คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์ พ.ศ. 2551

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ทุนการศึกษา -

ไม่มี-

ประวัติการฝึกอบรม -

ไม่มี-

ผลงานที่ได้รับการตีพิมพ์ -

ไม่มี-

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY