

ระบบช่วยจัดการนัดหมายเพื่อการจัดประชุม
(Meeting scheduling system)



นายบุญยະสิทธิ์ ขอบคุณ รหัส 50360326
นายณภสินธุ์ เข้มวัตร รหัส 50361347
นางสาวสิริกุล ชวงษ์ รหัส 50362641

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 19/๓.ค. 2555
เลขทะเบียน..... 15755578
เลขเรียกหนังสือ..... ๗๖.
มหาวิทยาลัยเกษตรศาสตร์ ๗534 8

2553

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
ปีการศึกษา 2553



ใบรับรองปริญญาโท

ชื่อหัวข้อโครงการ ระบบช่วยจัดการนัดหมายเพื่อการจัดประชุม
ผู้ดำเนินโครงการ นายบุญยະสิทธิ์ ขอบคุณ รหัส 50360326
นายณภิตินธุ์ เข้มวัตร รหัส 50361347
นางสาวศิริกุล ชูวงษ์ รหัส 50362641
ที่ปรึกษาโครงการ อาจารย์ภาณุพงศ์ สอนคม
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2553

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยราชภัฏบรจรัม อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะกรรมการสอบโครงการวิศวกรรม

.....ประธานกรรมการ
(อาจารย์ภาณุพงศ์ สอนคม)

.....กรรมการ
(ดร.วรลักษณ์ คงเด่นฟ้า)

.....กรรมการ
(อาจารย์เศรษฐา ตั้งค้ำวานิช)

ชื่อหัวข้อโครงการ	ระบบช่วยจัดการนัดหมายเพื่อการจัดประชุม	
ผู้ดำเนินโครงการ	นายบุญยะสิทธิ์ ขอบคุณ	รหัส 50360326
	นายณภสินธุ์ เข้มวัตร	รหัส 50361347
	นางสาวศิริกุล ชวงษ์	รหัส 50362641
ที่ปรึกษาโครงการ	อาจารย์ภาณุพงศ์ สอนคม	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2553	

บทคัดย่อ

ในปัจจุบันระบบสังคมออนไลน์ (Social Network) กำลังได้รับความนิยมเป็นอย่างมาก เนื่องจากช่วยให้การติดต่อสื่อสารสามารถทำได้อย่างรวดเร็วและทันต่อเหตุการณ์มากกว่าเทคโนโลยีในรูปแบบเดิม ๆ เมื่อมีคนมาใช้ในระบบมากขึ้นย่อมจะต้องมีการประชุมปรึกษาหรือพบปะหารือระหว่างกัน ดังนั้นจึงต้องมีการจัดการทางด้านการประชุม ผู้เข้าร่วมประชุม เวลาในการประชุม ตลอดจนการประชาสัมพันธ์ให้ผู้เข้าร่วมประชุมรับทราบ ถ้าสมาชิกทุกคนของระบบจำเป็นต้องออกจากระบบเพื่อไปใช้โปรแกรมเฉพาะสำหรับการจัดการประชุมก็จะทำให้เกิดความไม่สะดวกในการทำงาน คณะผู้จัดทำจึงได้เล็งเห็นว่าควรมีการพัฒนาช่วยจัดการนัดหมายเพื่อการจัดประชุมมาเป็นส่วนเสริมในระบบสังคมออนไลน์ เพื่อช่วยให้สมาชิกของระบบสามารถจัดการประชุมและกิจกรรมต่าง ๆ ได้ภายในระบบ ซึ่งจะช่วยให้เกิดความสะดวกต่อผู้ใช้งานระบบเป็นอย่างมาก

วิทยานิพนธ์นี้ได้ทำการพัฒนาระบบช่วยจัดการนัดหมายเพื่อการประชุมสำหรับสมาชิกภายในระบบสังคมออนไลน์ โดยผู้ใช้สามารถเลือกใช้โปรแกรมเฉพาะใด ๆ สำหรับการจัดการปฏิทินส่วนตัวของตนไม่ว่าจะเป็นโปรแกรม outlook หรือ google calendar โดยระบบที่พัฒนาขึ้นจะทำหน้าที่เชื่อมต่อกับโปรแกรมที่ใช้ในการจัดการปฏิทินส่วนตัวภายนอกเหล่านั้นผ่านทางส่วนต่อประสานโปรแกรมประยุกต์ (APIs : Application Programming Interface) ต่าง ๆ เพื่อนำข้อมูลตารางเวลาของสมาชิกแต่ละคนที่เข้าร่วมประชุมมาคำนวณหาเวลาที่เหมาะสมที่สุดในการจัดการประชุม และเมื่อผู้ใช้จัดการประชุมแล้วระบบยังทำการติดต่อกลับไปยังโปรแกรมการจัดการปฏิทินส่วนตัวภายนอกเหล่านั้นเพื่อทำการแก้ไขเพิ่มเติมข้อมูลเกี่ยวกับการประชุมไปในปฏิทินส่วนตัวของสมาชิกแต่ละคน ดังนั้นจะเห็นได้ว่าผู้ใช้ไม่จำเป็นต้องเปลี่ยนการทำงานไปมา

ระหว่างระบบสังคมออนไลน์ และโปรแกรมเฉพาะสำหรับการจัดการปฏิทินส่วนตัว นอกจากนี้ระบบที่พัฒนาขึ้นยังช่วยให้คำแนะนำเกี่ยวกับเวลาที่เหมาะสมสำหรับการจัดการประชุมและกิจกรรมต่าง ๆ โดยวิธีการหาเวลาที่เหมาะสมนั้นคณะผู้จัดทำได้ทำการดัดแปลงมาจาก N* Algorithm และจากผลการทดสอบการทำงานของระบบโดยผู้ใช้งานหนึ่งพบว่าส่วนใหญ่มีความพอใจในระดับมากถึงมากที่สุด เว้นแต่ในส่วนของการออกแบบในส่วนติดต่อกับผู้ใช้งานสามารถใช้งานได้สะดวกและเข้าใจง่ายซึ่งอยู่ในเกณฑ์ปานกลางจนถึงมากเป็นส่วนใหญ่ แต่ยังคงถือว่าอยู่ในระดับที่รับได้ และมีความพอใจในระดับมากถึงมากที่สุดในด้านผลลัพธ์ที่ได้จากระบบอีกด้วย



Project Title Meeting scheduling system

Name Mr.Boonyasith Khobkhun ID.50360326
Mr.Napasin Yamwat ID.50361347
Miss.Sirikul Choowong ID.50362641

Project Advisor Mr.Panupong Sornkhom

Major Computer Engineering.

Department Electrical and Computer Engineering.

Academic Year 2553

Abstract

Today, social network systems are being popular because it helps communication easier and up to date more than old technology. After most people use this system that meeting or scheduling between people is have more time to be create. Course, this system just can willing be can manage about meeting or scheduling including people those involved meeting, time and announcement. But if all people those involved meeting just out social network system for used specific meeting program for create meeting that will be inconvenient. Manipulator that should be developed system for help manage about meeting and used to plug-in or additional function of social network system. For help people who use social network system can create and manage meeting and some activity easier and comfortable.

This thesis developed meeting management and scheduling for create and manage meeting for people who used social network system. By that people would use another specific program such as "Outlook" or "Google Calendar". Then this system can connect with specific program than people choose with "APIs: Application Programming Interface" for get some data of each users those involved meeting and process and find appropriate. After meeting had created system can contract back to specific program for update each user calendar about that meeting data. Therefore, user will not change system between social network and specific program for manage about meeting. In addition, this system can suggest appropriate time for organizer or meeting maker with modified algorithm from N* algorithm. From a result of testing of suggestion

by some people seen that, most people are satisfy in high level except in interface design part is have satisfy in medium level and result of suggestion with modified N* algorithm is have satisfy in high level.



กิตติกรรมประกาศ

กิตติกรรมประกาศนี้จัดทำขึ้นเพื่อแสดงการขอบคุณต่อ อาจารย์ภาณุพงศ์ สอนคม ที่ช่วยสนับสนุนในการเสนอแนวทางการศึกษา เพื่อให้การศึกษาเป็นไปตามแนวทางที่ถูกต้อง และอาจารย์ ดร.วรลักษณ์ คงเด่นฟ้า ที่ให้คำปรึกษาด้านการทำเว็บเซอร์วิส เพื่อให้ระบบที่จัดทำขึ้นเกิดประสิทธิภาพ และยังช่วยกระตุ้นให้ผู้พัฒนาโครงการมีความกระตือรือร้นในการทำงานจนเสร็จสมบูรณ์ นอกจากนี้ขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่ได้ให้เงินสนับสนุนการทำโครงการนี้



นายบุญยะสิทธิ์ ขอบคุณ
นายนภสินธุ์ เข้มวัตร
นางสาวศิริกุล ชูวงษ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ค
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญตาราง.....	ช
สารบัญรูป.....	ฌ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบข่ายของโครงการ.....	2
1.4 ขั้นตอนการดำเนินงาน.....	2
1.5 ผลที่คาดว่าจะได้รับ.....	3
1.6 งบประมาณของโครงการ.....	4
บทที่ 2 หลักการและทฤษฎี	5
2.1 ฐานข้อมูล (Database).....	5
2.2 ความรู้เบื้องต้นเกี่ยวกับเว็บแอปพลิเคชัน (Web Applications).....	9
2.3 การพัฒนาเว็บด้วย ASP.NET.....	9
2.4 เจควีรี่ (jQuery).....	12
2.5 ไคลเอนต์/เซิร์ฟเวอร์ (Client / Server).....	13
2.6 ความรู้เบื้องต้นเกี่ยวกับเว็บเซอร์วิส (Web Services).....	15
2.7 Google Application Programming Interface : Google API.....	16
2.8 N* Algorithm.....	19
บทที่ 3 วิธีการดำเนินงาน	22
3.1 Use case Diagram.....	23
3.2 แบบจำลองการทำงานของระบบ.....	23
3.3 การพัฒนาโปรแกรม อัลกอริทึม และฐานข้อมูลของระบบ.....	29

สารบัญ (ต่อ)

	หน้า
3.4 รูปแบบของระบบ.....	31
3.5 พัฒนาโปรแกรมต้นแบบ.....	33
บทที่ 4 การทำงานของระบบ.....	35
4.1 ส่วนแสดงการเข้าสู่ระบบ.....	35
4.2 ส่วนแสดงกิจกรรมรวมทั้งหมด.....	35
4.3 ส่วนแสดงกิจกรรมเฉพาะบุคคล (My Event).....	37
4.4 ส่วนแสดงแบบฟอร์มในการจัดกิจกรรม.....	43
บทที่ 5 ผลการทดลอง.....	50
5.1 การประเมินความพึงพอใจของผู้ใช้งานระบบช่วยจัดการนัดหมายเพื่อการจัดประชุม.....	50
5.2 การวิเคราะห์เวลาที่ใช้ในการประมวลผล (time complexity).....	52
5.3 ผลการดำเนินการ.....	53
บทที่ 6 บทสรุปและข้อเสนอแนะ.....	54
6.1 บทสรุปการดำเนินการ.....	54
6.2 ปัญหาที่พบในการดำเนินการและแนวทางในการแก้ไข.....	54
6.3 ข้อเสนอแนะโครงการ.....	54
ภาคผนวก.....	55
เอกสารอ้างอิง.....	60
ประวัติผู้ดำเนินโครงการ.....	63

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงาน โครงการ.....	2
2.3.1 เปรียบเทียบการใช้งาน HTML Server Control กับอีลีเมนต์ของแท็ก HTML.....	10
2.7.1 แสดงรายละเอียดของ query parameters.....	16
2.7.2 แสดงรายละเอียดของ rel and type value.....	19
2.8.1 แสดงข้อเสนอต่าง ๆ ของการเข้าร่วมกิจกรรม.....	21
5.1 ตารางสรุปผลการดำเนินการ.....	53



สารบัญรูป

รูปที่	หน้า
2.1 แสดงรูปแบบแท็กคำสั่งควบคุมการทำงานทางฝั่งเซิร์ฟเวอร์	10
2.2 แสดงตัวอย่างการเขียนแท็ก <asp>	11
2.3 Activity Diagram ของ N* algorithm (an agent-based negotiation algorithm for dynamic scheduling and rescheduling Hon Wai Chun*, Rebecca Y.M. Wong)	20
3.1 Use Case Diagram แสดงการใช้งานระบบของ User	23
3.2 แบบจำลองการทำงานของระบบ	23
3.3 โค้ดตัวอย่างการดึงตารางปฏิทินวันเกิด	24
3.4 โค้ดตัวอย่างการการสร้งเหตุการณ์ลงปฏิทินวันเกิด	25
3.5 ตัวอย่างการดึงข้อมูล Username จากตาราง Account โดยการใช้ Linq to Sql	26
3.6 ตัวอย่างการเพิ่มข้อมูลลง Database	27
3.7 ตัวอย่างโค้ดการลบข้อมูลจากฐานข้อมูล	28
3.8 ตารางความสัมพันธ์ของฐานข้อมูลที่เกี่ยวข้องกับการจัดเหตุการณ์	29
3.9 แผนผังการทำงานของระบบ	29
3.10 ตัวอย่างโค้ดในส่วนการพัฒนาอัลกอริทึม	31
3.11 Class Diagram ของระบบ	32
3.12 โปรแกรมต้นแบบการติดต่อเซอร์วิสของปฏิทินวันเกิด (Google Calendar)	34
4.1 หน้าเว็บเข้าสู่ระบบ (Log In)	35
4.2 แสดงหน้าตาเพื่อแสดงกิจกรรมที่มีอยู่ปฏิทินตลอดทั้งเดือน	36
4.3.1 แสดงหน้าตาของกิจกรรมเฉพาะบุคคลของผู้ใช้คนนี้	37
4.3.2 แสดงหน้าตาการอัปเดตกิจกรรมที่จัด	38
4.3.3 แสดงการเปลี่ยนแปลงของกิจกรรมหลังการอัปเดตในหน้า Events	39
4.3.4 แสดงการเปลี่ยนแปลงของกิจกรรมหลังการอัปเดตในปฏิทินวันเกิดของผู้เข้าร่วมคนที่ 1	40
4.3.5 แสดงการเปลี่ยนแปลงของกิจกรรมหลังการอัปเดตในปฏิทินวันเกิดของผู้เข้าร่วมคนที่ 2	40
4.3.6 แสดงรายละเอียดของกิจกรรมที่ต้องการลบ	41
4.3.7 แสดงการเปลี่ยนแปลงหลังจากการลบกิจกรรมที่เลือก	41
4.3.8 แสดงกิจกรรมที่เหลือหลังการลบในหน้า Events	42
4.3.9 แสดงกิจกรรมที่เหลือหลังการลบในปฏิทินวันเกิดของผู้เข้าร่วมคนที่ 1	42
4.3.10 แสดงกิจกรรมที่เหลือหลังการลบในปฏิทินวันเกิดของผู้เข้าร่วมคนที่ 2	43

สารบัญญรูป (ต่อ)

รูปที่	หน้า
4.4.1 แสดงหน้า قالبแบบฟอร์มการสร้างกิจกรรม.....	43
4.4.2 แสดงหน้าต่างการเลือกผู้เข้าร่วมกิจกรรม.....	44
4.4.3 แสดงส่วนของการเลือกเวลาจากการคำนวณของระบบ.....	45
4.4.4 แสดงส่วนของการเลือกเวลาโดยผู้จัดกิจกรรมเป็นผู้กำหนดเอง.....	46
4.4.5 แสดงส่วนของการเพิ่มรูปภาพเพื่อใช้ประกอบรายละเอียดในการจัดกิจกรรม.....	47
4.4.6 แสดงกิจกรรมที่ถูกจัดขึ้น โดยใช้ระบบการจัดกิจกรรมในหน้า Events.....	48
4.4.7 แสดงกิจกรรมที่ถูกจัดขึ้น ในปฏิทินวันเกิดของผู้เข้าร่วมคนที่ 1.....	48
4.4.8 แสดงกิจกรรมที่ถูกจัดขึ้น ในปฏิทินวันเกิดของผู้เข้าร่วมคนที่ 2.....	49
5.1 แสดงกราฟสรุปผลการสำรวจความพึงพอใจด้านการใช้งานและความสามารถของระบบ.....	51
5.2 แสดงกราฟสรุปผลการสำรวจความพึงพอใจด้านผลลัพธ์ที่ได้จากระบบ.....	52

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ปัจจุบันเทคโนโลยีคอมพิวเตอร์ได้เข้ามามีบทบาทสำคัญในการปฏิบัติงาน การจัดการประชุมสัมมนา ก็เป็นอีกสิ่งหนึ่งที่สามารถนำเทคโนโลยีทางด้านคอมพิวเตอร์เข้ามามีส่วนร่วมในการจัดการได้ ซึ่งการประชุมในแต่ละครั้งจะต้องทำการเลือกสถานที่ในการจัดการประชุมสัมมนา การติดต่อผู้เข้าร่วม รวมถึงแจ้งรายละเอียดต่าง ๆ ที่จำเป็นต่อการประชุมสัมมนา การดำเนินการแบบเก่านั้นก่อให้เกิดความล่าช้าในการดำเนินการ ความไม่พร้อมของผู้เข้าร่วมประชุม หรือปัญหาการเปลี่ยนแปลงสถานที่ประชุมอย่างกะทันหัน ซึ่งอาจก่อให้เกิดความเสียหายขึ้นได้

ด้วยเหตุนี้จึงได้มีการจัดทำโครงการนี้ขึ้นมาเพื่อช่วยอำนวยความสะดวกในการจัดการประชุมให้กับผู้จัดการประชุม ซึ่งจะสามารถช่วยลดปัญหาต่าง ๆ ที่อาจเกิดขึ้น โดยที่สามารถทำการจองสถานที่ ที่ต้องใช้ในการจัดประชุมสัมมนา ผ่านทางอินเทอร์เน็ตโดยมีการนำความรู้ที่เรียนมาทางด้านระบบฐานข้อมูล เข้ามาผสมผสานกับเว็บแอปพลิเคชัน (Web Application) ซึ่งมีความสะดวกต่อผู้จัดการประชุม โดยสามารถนำตารางเวลาของผู้เข้าร่วมการประชุมมาหาช่วงเวลาที่เหมาะสมที่จะจัดการประชุม ซึ่งสามารถทำให้เกิดความสะดวกรวดเร็วในการจัดการประชุม และสามารถแจ้งเตือนให้ผู้เข้าร่วมการประชุมทราบถึงกำหนดการต่าง ๆ ในการประชุมได้ ถ้าหากไม่สะดวกในช่วงเวลานั้นก็สามารถติดต่อผู้จัดการประชุมให้ทำการหาช่วงเวลาใหม่ที่เหมาะสมได้ ซึ่งจะเป็นการดีกว่าที่จะขาดการประชุมเนื่องจากติดธุระจึงไม่สามารถมาประชุมได้

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อสร้าง โปรแกรมที่ช่วยให้ผู้ใช้ระบบสังคมออนไลน์สามารถเข้าถึงข้อมูลจากปฏิทินวันเกิดได้

1.2.2 เพื่อช่วยให้จัดการประชุมได้โดยใช้ปฏิทินวันเกิด

1.2.3 เพื่อแนะนำเวลาที่เหมาะสมในการจัดการประชุมได้

1.2.4 เพื่อให้ระบบเชิงโครโนซ์เหตุการณ์การจัดการประชุมกับปฏิทินวันเกิดได้

1.2.5 เพื่อดำเนินการเพิ่ม ลบ และอัปเดตการประชุมที่เป็นคนจัดขึ้นได้

1.3 ขอบเขตของโครงการ

1.3.1 เพื่อสามารถประมวลผลหาเวลาที่เหมาะสมแก่ผู้จัดประชุมได้

1.3.2 ผู้ใช้สามารถตรวจสอบเวลาที่เหมาะสมในการประชุมได้

1.3.3 ระบบสามารถดึงตารางเวลาได้อัตโนมัติจากตารางเวลาของผู้ใช้

1.3.4 สามารถแจ้งรายละเอียดต่าง ๆ เกี่ยวกับการประชุมให้ผู้เข้าร่วมประชุมทางหน้าเว็บ, e-mail และ sms ได้

1.3.5 จัดระดับความสำคัญของผู้ใช้ในการเข้าถึงข้อมูลได้

1.3.6 สามารถจัดการเกี่ยวกับระบบฐานข้อมูลของผู้ใช้ได้

1.3.7 สามารถจัดการกับเอกสารประกอบการประชุมได้

1.3.8 สามารถแบ่งชนิดของการจัดการประชุมได้

1.4 ขั้นตอนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงานโครงการ

เดือน	ปีการศึกษา 2553							
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.
สัปดาห์	1-4	1-4	1-4	1-4	1-4	1-4	1-4	1-4
กิจกรรม								
1) ศึกษาและเลือกหัวข้อโครงการ พร้อมส่งใบเสนอโครงการ รวมทั้งอาจารย์ที่ปรึกษาโครงการ	→							
2) ศึกษาข้อมูลของโครงการ ตัวอย่าง และกำหนดขอบเขตของโครงการ พร้อมทั้งจัดทำบทนำของโครงการ		→						
3) ทำการกำหนดความสามารถของโปรแกรม ออกแบบโปรแกรมที่จะต้องนำมาใช้			→					

เดือน	ปีการศึกษา 2553							
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.
สัปดาห์	1-4	1-4	1-4	1-4	1-4	1-4	1-4	1-4
กิจกรรม								
4) ศึกษาข้อมูลที่มีความเหมาะสมที่จะนำมาใช้อ้างอิง				→				
5) ดำเนินการเขียนโปรแกรม พร้อมกับบันทึกข้อบกพร่องระหว่างการดำเนินงาน					→			
6) ตรวจสอบและแก้ไขการทำงานของโปรแกรม						→		
7) จัดทำสรุปผลการทำงานคู่มือการใช้งาน โปรแกรม						→		
8) รวบรวมรูปเล่มโครงการให้อาจารย์ที่ปรึกษาพิจารณาแก้ไข							→	
9) จัดทำ Presentation เพื่อประกอบการนำเสนอโครงการ							→	
10) จัดทำรูปเล่มจริง พร้อมกับบทความที่เกี่ยวกับโครงการ								→

1.5 ผลที่คาดว่าจะได้รับ

- 1.5.1 โปรแกรมที่ช่วยให้ผู้ใช้ระบบสังคมออนไลน์สามารถเข้าถึงข้อมูลจากปฏิทินกูเกิลได้
- 1.5.2 ช่วยให้จัดการประชุมได้โดยใช้ปฏิทินกูเกิล
- 1.5.3 สามารถแนะนำเวลาที่เหมาะสมในการจัดการประชุมได้
- 1.5.4 สามารถชิงโครไนซ์เหตุการณ์การจัดประชุมกับปฏิทินกูเกิลได้
- 1.5.5 สามารถดำเนินการเพิ่ม ลบ และอัปเดตการประชุมที่เป็นคนจัดขึ้นได้

1.6 งบประมาณ

1.6.1	ค่าถ่ายเอกสารและค่าเช่าเล่มรายงานฉบับสมบูรณ์	เป็นเงิน	2,000 บาท
1.6.2	ค่าวัสดุสำนักงาน	เป็นเงิน	<u>1,000 บาท</u>
	รวมเป็นเงินทั้งสิ้น		<u>3,000 บาท</u>
			(สามพันบาทถ้วน)

หมายเหตุ ตัวเฉลี่ยทุกรายการ



บทที่ 2

หลักการและทฤษฎี

ในบทนี้จะกล่าวถึงหลักการ ทฤษฎีพื้นฐาน หรือความรู้ต่าง ๆ ที่จะนำมาใช้ในการสร้างระบบการจัดการประชุม โดยจะกล่าวถึงเรื่องต่าง ๆ ดังนี้

1. ฐานข้อมูล (Database)
2. ความรู้เบื้องต้นเกี่ยวกับเว็บแอปพลิเคชัน (Web Applications)
3. การพัฒนาเว็บด้วย ASP.NET
4. การใช้งานเจควีรี่ (jQuery)
5. ไคลเอนต์ / เซิร์ฟเวอร์ (Client / Server)
6. ความรู้เบื้องต้นเกี่ยวกับเว็บเซอร์วิส (Web Services)
7. Google Application Programming Interface : Google API
8. N* Algorithm

2.1 ฐานข้อมูล (Database) ^{III}

ฐานข้อมูล (Database) หมายถึง ชุดของข้อมูลที่รวมเอาข้อมูลที่เกี่ยวข้องกันเป็นเรื่องราวเดียวกันรวมกันเป็นกลุ่มหรือเป็นชุดข้อมูล เช่น ฐานข้อมูลนิสิต ฐานข้อมูลลูกค้า และฐานข้อมูลวิชาเรียน เป็นต้น ซึ่งข้อมูลเหล่านี้ได้มาจากการบันทึกข้อมูลโดยผู้ใช้ หรือบางข้อมูลอาจจะได้มาจากการประมวลผลข้อมูลแล้วบันทึกข้อมูลกลับไปเก็บที่ตำแหน่งที่ต้องการ

2.1.1 ระบบจัดการฐานข้อมูล

ซอฟต์แวร์สำหรับจัดการฐานข้อมูลนั้น โดยทั่วไปเรียกว่า ระบบจัดการฐานข้อมูล หรือดีบีเอ็มเอส (DBMS - Database Management System) สถาปัตยกรรมซอฟต์แวร์ของดีบีเอ็มเอสอาจมีได้หลายแบบ เช่น สำหรับฐานข้อมูลขนาดเล็กที่มีผู้ใช้คนเดียว บ่อยครั้งที่หน้าที่ทั้งหมดจะจัดการด้วยโปรแกรมเพียงโปรแกรมเดียว ส่วนฐานข้อมูลขนาดใหญ่ที่มีผู้ใช้งานจำนวนมากนั้นปกติจะประกอบด้วยโปรแกรมหลายโปรแกรมด้วยกัน และโดยทั่วไปส่วนใหญ่จะใช้สถาปัตยกรรมแบบไคลเอนต์ / เซิร์ฟเวอร์

โปรแกรมส่วนหน้า (front - end) ของดีบีเอ็มเอส คือ โปรแกรมรับบริการ จะเกี่ยวข้องเฉพาะการนำเข้าข้อมูล, การตรวจสอบ, และการรายงานผลเป็นสำคัญ ในขณะที่

โปรแกรมส่วนหลัง (back - end) คือ โปรแกรมให้บริการ จะเป็นชุดของโปรแกรมที่ดำเนินการเกี่ยวกับการควบคุม, การเก็บข้อมูล, และการตอบสนองการร้องขอจากโปรแกรมส่วนหน้า โดยปกติแล้วการค้นหาและการเรียงลำดับจะดำเนินการโดยโปรแกรมให้บริการ รูปแบบของระบบฐานข้อมูลมีหลากหลายรูปแบบด้วยกัน นับตั้งแต่การใช้ตารางอย่างง่ายที่เก็บในแฟ้มข้อมูลแฟ้มเดียว ไปจนถึงฐานข้อมูลขนาดใหญ่ที่มีระเบียบหลายล้านระเบียบ ซึ่งเก็บในห้องที่เต็มไปด้วยดิสก์ไดรฟ์หรืออุปกรณ์หน่วยเก็บข้อมูลอิเล็กทรอนิกส์รอบข้าง (peripheral) อื่น ๆ

ระบบจัดการฐานข้อมูล (Database Management System) เป็นกลุ่มโปรแกรมที่ทำหน้าที่เป็นตัวกลางในระบบคิดต่อระหว่างผู้ใช้กับฐานข้อมูล เพื่อจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ภายในฐานข้อมูล ซึ่งต่างจากระบบแฟ้มข้อมูลทำหน้าที่เหล่านี้จะเป็นหน้าที่ของ โปรแกรมเมอร์ ในการติดต่อกับข้อมูลในฐานข้อมูลไม่ว่าจะด้วยการใช้คำสั่งในกลุ่มดีเอ็มแอล (DML) หรือดีดีแอล (DDL) หรือจะด้วยโปรแกรมต่าง ๆ ทุกคำสั่งที่ใช้กระทำกับข้อมูลจะถูกคอมไพล์ (compile) เป็นการปฏิบัติการ (operation) ต่าง ๆ ภายใต้คำสั่งนั้น ๆ เพื่อนำไปกระทำกับตัวข้อมูลภายในฐานข้อมูลต่อไป สำหรับส่วนการทำงานต่าง ๆ ภายในดีบีเอ็มเอสที่ทำหน้าที่แปลคำสั่งไปเป็นการปฏิบัติการต่าง ๆ กับข้อมูลนั้น

2.1.2 หน้าที่ของระบบจัดการฐานข้อมูล

- 1) แปลงคำสั่งที่ใช้จัดการกับข้อมูลภายในฐานข้อมูล ให้อยู่ในรูปแบบที่ฐานข้อมูลเข้าใจ
- 2) นำคำสั่งต่าง ๆ ซึ่งได้รับการแปลแล้วไปสั่งให้ฐานข้อมูลทำงาน เช่น การเรียกใช้ (Retrieve) จัดเก็บ (Update) ลบ (Delete) เพิ่มข้อมูล (Add) เป็นต้น
- 3) ป้องกันความเสียหายที่จะเกิดขึ้นกับข้อมูลภายในฐานข้อมูล โดยจะคอยตรวจสอบว่าคำสั่งใดที่สามารถทำงานได้ และคำสั่งใดที่ไม่สามารถทำงานได้
- 4) รักษาความสัมพันธ์ของข้อมูลภายในฐานข้อมูลให้มีความถูกต้องอยู่เสมอ
- 5) เก็บรายละเอียดต่าง ๆ ที่เกี่ยวข้องกับข้อมูลภายในฐานข้อมูลไว้ในพจนานุกรมข้อมูล (Data Dictionary) ซึ่งรายละเอียดเหล่านี้มักจะถูกเรียกว่า เมทาดาทา (MetaData) ซึ่งหมายถึง “ข้อมูลของข้อมูล”
- 6) ควบคุมให้ฐานข้อมูลทำงานได้อย่างถูกต้องและมีประสิทธิภาพ
- 7) ควบคุมสถานะภาพของคอมพิวเตอร์ในการแปลสภาพฐานข้อมูล

2.1.3 ระบบจัดการฐานข้อมูลที่นิยมใช้กันในปัจจุบัน

ในปัจจุบันบริษัทต่าง ๆ ได้มีการพัฒนาผลิตภัณฑ์เกี่ยวกับฐานข้อมูลออกมาเป็นจำนวนมาก โดยที่มีจุดเด่นต่าง ๆ กันไป ตัวอย่างของระบบฐานข้อมูลที่ใช้ในปัจจุบัน มีดังนี้

- ออราเคิล (Oracle)
- ไอบีเอ็ม ดีบีทู (IBM DB2)
- ไมโครซอฟท์ เอสคิวแอล เซิร์ฟเวอร์ (Microsoft SQL Server)
- ซายเบส (Sybase)
- แคเช่ (Cache')
- PostgreSQL
- Progress
- มาชเอสคิวแอล (MySQL)
- Interbase
- Firebird
- Pervasive SQL
- แซพ ดีบี (SAP DB)
- ไมโครซอฟท์ แอคเซส (Microsoft Office Access)

2.1.4 การออกแบบฐานข้อมูล

การออกแบบฐานข้อมูล (Designing Databases) มีความสำคัญต่อการจัดการระบบฐานข้อมูลทั้งนี้เนื่องจากข้อมูลที่อยู่ภายในฐานข้อมูลจะต้องศึกษาถึงความสัมพันธ์ของข้อมูล โครงสร้างของข้อมูล การเข้าถึงข้อมูล และกระบวนการที่โปรแกรมประยุกต์จะเรียกใช้ฐานข้อมูล ดังนั้น เราจึงสามารถแบ่งวิธีการสร้างฐานข้อมูลได้ 3 ประเภท

1) รูปแบบข้อมูลแบบลำดับขั้น หรือโครงสร้างแบบลำดับขั้น (Hierarchical data model) วิธีการสร้างฐานข้อมูลแบบลำดับขั้นถูกพัฒนาโดยบริษัท ไอบีเอ็ม จำกัด ในปี ค.ศ.1980 ได้รับความนิยมมาก ในการพัฒนาฐานข้อมูลบนเครื่องคอมพิวเตอร์ขนาดใหญ่และขนาดกลาง โดยที่โครงสร้างข้อมูลจะสร้างรูปแบบเหมือนต้นไม้ โดยความสัมพันธ์เป็นแบบหนึ่งต่อหลาย (One - to - Many)

2) รูปแบบข้อมูลแบบเครือข่าย (Network data Model) ฐานข้อมูลแบบเครือข่ายมีความคล้ายคลึงกับฐานข้อมูลแบบลำดับขั้นต่างกันที่โครงสร้างแบบเครือข่าย อาจจะมีการติดต่อ

หลายต่อหนึ่ง (Many - to - one) หรือหลายต่อหลาย (Many - to - many) กล่าวคือ ลูก (Child) อาจมีพ่อแม่ (Parent) มากกว่าหนึ่ง สำหรับตัวอย่างฐานข้อมูลแบบเครือข่ายให้ลองพิจารณาการจัดการข้อมูลของห้องสมุด ซึ่งรายการจะประกอบด้วย ชื่อเรื่อง ผู้แต่ง สำนักพิมพ์ ที่อยู่ ประเภท

3) รูปแบบความสัมพันธ์ข้อมูล (Relation data model) เป็นลักษณะการออกแบบฐานข้อมูลโดยจัดข้อมูลให้อยู่ในรูปของตารางที่มีระบบคล้ายแฟ้ม โดยที่ข้อมูลแต่ละแถว (Row) ของตารางจะแทนเรคอร์ด (Record) ส่วนข้อมูลในแนวดิ่งจะแทนคอลัมน์ (Column) ซึ่งเป็นขอบเขตของข้อมูล (Field) โดยที่ตารางแต่ละตารางที่สร้างขึ้นจะเป็นอิสระ ดังนั้น ผู้ออกแบบฐานข้อมูลจะต้องมีการวางแผนถึงตารางข้อมูลที่เป็นต้องใช้ เช่น ระบบฐานข้อมูลบริษัทแห่งหนึ่ง ประกอบด้วย ตารางประวัติพนักงาน ตารางแผนก และตารางข้อมูลโครงการ แสดงประวัติพนักงาน ตารางแผนก และตารางข้อมูลโครงการ

2.1.5 ความสำคัญของระบบฐานข้อมูล

1) สามารถลดความซ้ำซ้อนของข้อมูลได้ การเก็บข้อมูลชนิดเดียวกันไว้หลาย ๆ ที่ทำให้เกิดความซ้ำซ้อน ดังนั้นระบบฐานข้อมูลจะลดการซ้ำซ้อนของข้อมูลเหล่านี้ให้มากที่สุดโดยจัดเก็บในฐานข้อมูลไว้ที่เดียวกัน ผู้ใช้ทุกคนที่ต้องการใช้ข้อมูลชุดนี้จะใช้โดยผ่านระบบฐานข้อมูลทำให้ไม่เปลืองเนื้อที่ในการเก็บข้อมูลและลดความซ้ำซ้อนลงได้

2) รักษาความถูกต้องของข้อมูล กรณีที่มีข้อมูลชุดเดียวกันปรากฏอยู่หลายแห่งในฐานข้อมูล ข้อมูลเหล่านี้จะต้องตรงกัน ถ้ามีการแก้ไขข้อมูลนี้ทุก ๆ แห่งที่ข้อมูลปรากฏอยู่จะแก้ไขให้ถูกต้องตามกันหมด โดยอัตโนมัติด้วยระบบจัดการฐานข้อมูล

3) การป้องกันและรักษาความปลอดภัยให้กับข้อมูลทำได้อย่างสะดวก การป้องกันและรักษาความปลอดภัยกับข้อมูลระบบฐานข้อมูลจะให้เฉพาะผู้ที่เกี่ยวข้องเท่านั้นจึงจะมีสิทธิ์เข้าไปใช้ฐานข้อมูลได้เรียกว่ามีสิทธิ์ส่วนบุคคล (privacy) ซึ่งก่อให้เกิดความปลอดภัย (security) ของข้อมูลด้วย ฉะนั้นผู้ใดจะมีสิทธิ์ที่จะเข้าถึงข้อมูลได้จะต้องมีการกำหนดสิทธิ์กันไว้ก่อนและเมื่อเข้าไปใช้ข้อมูลนั้น ๆ ผู้ใช้จะเห็นข้อมูลที่ถูกเก็บไว้ในฐานข้อมูลในรูปแบบที่ผู้ใช้ออกแบบไว้

4) สามารถใช้ข้อมูลร่วมกันได้ เนื่องจากในระบบฐานข้อมูลจะเป็นที่เก็บรวบรวมข้อมูลทุกอย่างไว้ ผู้ใช้แต่ละคนจึงสามารถที่จะใช้ข้อมูลในระบบได้ทุกข้อมูล

5) มีความเป็นอิสระของข้อมูล เมื่อผู้ใช้ต้องการเปลี่ยนแปลงข้อมูลหรือนำข้อมูลมาประยุกต์ใช้ให้เหมาะสมกับโปรแกรมที่เขียนขึ้นมาจะสามารถสร้างข้อมูลนั้นขึ้นมาใช้ใหม่ได้

โดยไม่มีผลกระทบต่อระบบฐานข้อมูล เพราะข้อมูลที่ผู้ใช้นำมาประยุกต์ใช้ใหม่นั้นจะไม่กระทบต่อโครงสร้างที่แท้จริงของการจัดเก็บข้อมูล นั่นคือ การใช้ระบบฐานข้อมูลจะทำให้เกิดความเป็นอิสระระหว่างการจัดเก็บข้อมูลและการประยุกต์ใช้

6) สามารถขยายงานได้ง่าย เมื่อต้องการจัดเพิ่มเติมข้อมูลที่เกี่ยวข้องจะสามารถเพิ่มได้อย่างง่ายไม่ซับซ้อน เนื่องจากมีความเป็นอิสระของข้อมูล จึงไม่มีผลกระทบต่อข้อมูลเดิมที่มีอยู่

7) ทำให้ข้อมูลกลับสู่สภาพปกติได้เร็วและมีมาตรฐาน การเก็บข้อมูลร่วมกันไว้ในฐานข้อมูลจะทำให้สามารถกำหนดมาตรฐานของข้อมูลได้รวมทั้งมาตรฐานต่าง ๆ ในการจัดเก็บข้อมูลให้เป็นไปในลักษณะเดียวกันได้ เช่น การกำหนดรูปแบบการเขียนวันที่ ในลักษณะวัน/เดือน/ปี หรือ ปี/เดือน/วัน ทั้งนี้จะมีผู้ที่คอยบริหารฐานข้อมูลที่เราเรียกว่า ผู้บริหารฐานข้อมูล (Database Administrator: DBA) เป็นผู้กำหนดมาตรฐานต่าง ๆ

2.2 ความรู้เบื้องต้นเกี่ยวกับเว็บแอปพลิเคชัน (Web Applications)

เว็บแอปพลิเคชัน (Web Applications) คือ โปรแกรมประยุกต์ที่เข้าถึงด้วยการใช้งานผ่านเว็บเบราว์เซอร์ซึ่งเป็นที่นิยมในเรื่องความสามารถในการปรับปรุงและดูแลโดยไม่ต้องแจกจ่ายและติดตั้งซอฟต์แวร์บนเครื่องผู้ใช้

2.3 การพัฒนาเว็บด้วย ASP.NET ¹²⁾

ASP.NET เป็นเทคโนโลยีที่พัฒนาจาก ASP (Active Server Page) ดังนั้นแนวคิดและองค์ประกอบโดยทั่วไปของ ASP.NET จึงมีลักษณะคล้ายกับ ASP แต่ ASP.NET นำเอาเทคโนโลยีของ .NET Framework มาใช้ และเว็บเพจที่พัฒนาขึ้นจากเทคโนโลยี ASP.NET จะจัดเก็บในไฟล์ .aspx รวมทั้งนำเอา ASP.NET Server Control มาใช้แทนอีลีเมนต์ (element) ที่สร้างขึ้นจากแท็ก (tag) HTML เนื่องจาก ASP.NET Server Control สามารถจดจำค่าที่กำหนดให้กับคอนโทรลที่เกิดขึ้นก่อนหน้าี้ และสามารถตอบสนองต่อเหตุการณ์ต่าง ๆ ที่เกิดขึ้นได้

การสร้างเพจเพื่อติดต่อกับผู้ใช้ ใน ASP.NET มีโครงสร้างในการเขียนโค้ดคำสั่ง 2 แบบ คือ แบบ Code Inline ซึ่งประกอบด้วยคำสั่งสำหรับการออกแบบเพจ และแบบ Code Behind ซึ่งเป็นส่วนคำสั่งควบคุมการทำงาน โดยรวมอยู่ในไฟล์เดียวกันคือ .aspx ซึ่งจะแยกเป็นส่วนคำสั่งของการแสดงผลไว้ที่ไฟล์ .aspx และส่วนคำสั่งที่ใช้ควบคุมการทำงานไว้ที่ไฟล์ .aspx.cs (กรณีที่เขียนด้วยภาษา C#) เพื่อให้ง่ายในการจัดการโค้ดคำสั่ง

ASP.NET Server Control คือ กลุ่มคอลโทรลของ ASP.NET ที่สามารถทำงานบนฝั่งเซิร์ฟเวอร์ได้ เนื่องจากใช้แท็ก HTML สร้างจอภาพเพื่อติดต่อกับผู้ใช้งาน แต่มีข้อจำกัดในด้านการประมวลผล คือ จะถูกประมวลผลบนเว็บเบราว์เซอร์เท่านั้น ดังนั้นถ้าโปรแกรมต้องการใช้ทรัพยากรจากเว็บเซิร์ฟเวอร์จำเป็นที่จะต้องนำเอา ASP.NET Server Control เข้ามาใช้แทนในอีลีเมนต์ของแท็ก HTML ที่ไม่สามารถกระทำกับข้อมูลภายในฐานข้อมูล เนื่องจาก ASP.NET Server Control ถูกพัฒนาขึ้นมาจากพื้นฐานของแท็ก HTML ดังนั้นรูปแบบคำสั่งจึงคงอยู่ในรูปของแท็ก HTML และเนื่องจาก ASP.NET Server Control ถูกใช้เพื่อสร้างส่วนติดต่อกับผู้ใช้ ซึ่งจะแสดงผลบนเว็บเบราว์เซอร์ที่เข้าใจเฉพาะแท็ก HTML เมื่อนำไปใช้งานจึงต้องทำการแปลง ASP.NET Server Control ให้อยู่ในรูปของแท็ก HTML ก่อนที่จะส่งไปแสดงผล

1) HTML Server Control คือ อีลีเมนต์ของแท็ก HTML ที่ควบคุมการทำงานของโปรแกรมทางฝั่งเซิร์ฟเวอร์ มีความสามารถในการประมวลผลที่รวดเร็ว เหมาะสมกับการทำงานที่ไม่ซับซ้อน และต้องการควบคุมการทำงานทางฝั่งไคลเอนต์ได้โดยตรง รูปแบบคำสั่งจะเหมือนกับแท็ก HTML แต่ต้องมีการกำหนดคำสั่ง `runat="server"` ซึ่งอาจมีบางคำสั่งที่ไม่ต้องกำหนดคำสั่ง `runat="server"` เช่น `<table>`, `<tr>` และ `<td>` เนื่องจากเป็นส่วนที่ใช้ในการจัดรูปแบบหน้าจอจึงไม่ต้องกำหนดการทำงานในฝั่งของเซิร์ฟเวอร์ก็ได้ นอกจากนี้ต้องมีการกำหนดชื่อให้กับคอลโทรลใน Property "id" แทน Property "name" ซึ่งสามารถกำหนด Property อื่น ๆ ของแต่ละแท็กคำสั่งได้เช่นเดียวกับ Property ของ HTML ตัวอย่างเช่น

```
<table id = "table 1" runat= "server">
```

รูปที่ 2.1 แสดงรูปแบบแท็กคำสั่งควบคุมการทำงานของฝั่งเซิร์ฟเวอร์

จากคำสั่งด้านบนเป็นคำสั่งของ HTMLTable Control ใน HTML Server Control โดยทำหน้าที่เช่นเดียวกับแท็ก `<table>` ของแท็ก HTMLTable Control ซึ่งสามารถทำงานในฝั่งของเซิร์ฟเวอร์ได้

ตารางที่ 2.3.1 เปรียบเทียบการใช้งาน HTML Server Control กับอีลีเมนต์ของแท็ก HTML

HTML Server Control	อีลีเมนต์ของแท็ก HTML
HTMLForm	<code><form></code>
HTMLAnchor	<code><a></code>

ตารางที่ 2.3.1 เปรียบเทียบการใช้งาน HTML Server Control กับอีลีเมนต์ของแท็ก HTML (ต่อ)

HTML Server Control	อีลีเมนต์ของแท็ก HTML
HTMLButton	<button>
HTMLGenericControl	<div> หรือ
HTMLImage	
HTMLInputButton	<input type= "button">
HTMLInputCheckBox	<input type= "checkbox">
HTMLInputFile	<input type= "file">
HTMLInputHidden	<input type= "hidden">
HTMLInputImage	<input type= "image">
HTMLInputRadioButton	<input type= "radio">
HTMLInputText	<input type= "text"> หรือ <input type= "password">
HTMLSelect	<select>
HTMLTable	<table>
HTMLTableCell	<td>
HTMLTableRow	<tr>
HTMLTextArea	<textarea>

2) ASP.net Tag คือ กลุ่มคอดโทรลเฉพาะประเภทหนึ่งทำงานในฝั่งเซิร์ฟเวอร์เหมาะสำหรับการสร้างเพจที่มีฟังก์ชันการทำงานที่ซับซ้อน เพื่อควบคุมการทำงานบนเว็บเพจซึ่งมีความสามารถมากกว่า HTML Server Control เช่น มี Property ที่รองรับการทำงานมากกว่า รูปแบบในการเขียนคำสั่งจะเขียนภายในแท็ก <asp: ... > ... </asp: ... > หรืออาจใช้ <asp: ... /> ก็ได้ ตัวอย่างเช่น

```

<asp:TextBox ID= "TextBox1" runat= "server"><asp:TextBox>
หรือ
<asp:TextBox ID= "TextBox1" runat= "server" />
```

รูปที่ 2.2 แสดงตัวอย่างการเขียนแท็ก <asp>

จากคำสั่งด้านบนเป็นการสร้าง TextBox Control เพื่อใส่ค่าข้อมูลต่างๆ โดยทำการประกาศในแท็ก <asp:> และยังต้องมีการประกาศ runat= "server" ไว้ภายในเพื่อให้สามารถใช้งานในฝั่งของเซิร์ฟเวอร์ได้

2.4 เจควีรี่ (jQuery) ¹⁹⁾

เจควีรี่ (jQuery) คือ ไลบรารีของจาวาสคริปต์ (JavaScript Library) ซึ่งเป็นส่วนหนึ่งของการสร้างเว็บไซต์ เจควีรี่ที่เข้ามาช่วยอำนวยความสะดวกในการสร้างสีสัน ลูกเล่นต่างๆ ให้แก่เว็บไซต์ โดยที่ผู้พัฒนาไม่ต้องเสียเวลาในการพัฒนาโค้ดโปรแกรมให้ยุ่งยากซับซ้อนเหมือนแต่ก่อน และช่วยให้เราเรียก เขียน หรือใช้งานจาวาสคริปต์ และเอแจ็กซ์ (AJAX - Asynchronous JavaScript And XML) ให้ง่ายขึ้น หรือจะเขียนจาวาสคริปต์เพื่อคลิกจับเหตุการณ์ (Event) ต่างๆ ที่ต้องการ เช่น การคลิก (Click)

2.4.1 ความสามารถของเจควีรี่

- ความสามารถในการทำงานแบบเอแจ็กซ์ (AJAX)
- การสร้างแอนิเมชัน (animation) ได้แบบง่าย ๆ
- ความสามารถในการผูก (bind) หรือจับฟังก์ชันที่เขียนขึ้นให้ทำงานร่วมกับฟังก์ชันอื่นๆ
- สามารถจัดการกับแคสเคดดิ้งสไตล์ชีตส์ (Cascading style sheet : CSS) ของอีลีเมนต์ (element) นั้นๆ ได้
 - ค้นหาอีลีเมนต์ (element) ที่เราต้องการและจัดการเพิ่มหรือลบคุณสมบัติ (Attributes) ที่เราต้องการได้
 - ทำเอฟเฟกต์ (Effect) ต่างๆ กับอีลีเมนต์ (element) ที่เราต้องการ เช่น การซ่อนแท็ก DIV ที่ต้องการ
 - การคลิกเหตุการณ์ต่างๆ

2.4.2 ตัวอย่างโปรแกรมที่เจควีรี่สามารถใช้งานร่วมกันได้

- พีเอสพี (PHP)
- เจเอสพี (JSP)
- เพลิร์ล (PERL)
- เอเอสพีคอตเน็ต (ASP.NET)

- วิบีคอตเน็ต (VB.NET)
- เซททีเอ็มแอล (HTML)
- ซีเอสเอส (CSS)
- เอ็กซ์เอ็มแอล (XML)

2.5 ไคลเอนต์ / เซิร์ฟเวอร์ (Client / Server)

ไคลเอนต์ (Client) คือ เครื่องที่ไปขอใช้บริการอย่างใดอย่างหนึ่งจากเซิร์ฟเวอร์ (Server) ส่วนเซิร์ฟเวอร์ คือ เครื่องที่ทำหน้าที่ในการให้บริการแก่ไคลเอนต์ เป็นรูปแบบหนึ่งของเครือข่ายแบบพึ่งเครื่องบริการ (Server - based network) โดยจะมีคอมพิวเตอร์หลักเครื่องหนึ่งเป็นเซิร์ฟเวอร์ซึ่งจะไม่ได้ทำหน้าที่ประมวลผลทั้งหมดให้เครื่องลูกข่ายหรือไคลเอนต์ เซิร์ฟเวอร์ทำหน้าที่เสมือนเป็นที่เก็บข้อมูลระยะไกล (remote disk) และประมวลผลบางอย่างให้กับไคลเอนต์เท่านั้น โดยผู้ใช้ข้อมูลจะมีซอฟต์แวร์ในการติดต่อไปยังผู้ให้บริการเซิร์ฟเวอร์ เช่น การใช้อินเทอร์เน็ต เว็บเบราว์เซอร์ (Web browser) ของผู้ใช้จะเป็นโปรแกรมร้องขอการบริการจากเว็บเซิร์ฟเวอร์ (Web server) หรือที่เรียกว่า Hypertext Transfer Protocol Server ซึ่งเป็นคอมพิวเตอร์อีกเครื่องหนึ่งบนอินเทอร์เน็ต ถ้ายกคลึงกับคอมพิวเตอร์ที่ติดตั้ง TCP/IP โดยยินยอมให้ผู้ใช้สร้างคำขอเพื่อขอไฟล์จาก File Transfer Protocol Server ในคอมพิวเตอร์อีกเครื่องบนอินเทอร์เน็ตนั่นเอง

ระบบไคลเอนต์ / เซิร์ฟเวอร์ (Client / Server System) ได้ถูกพัฒนาขึ้นมาเพื่อตอบสนองแนวคิดการลดขนาดแรงงาน (Downsizing) เป็นการลดค่าใช้จ่ายของระบบแบ่งเวลา (Time Sharing) ของเครื่องเมนเฟรม (Mainframe) ซึ่งระบบไคลเอนต์ / เซิร์ฟเวอร์ เป็นระบบประมวลแบบกระจาย (Distributed Processing) โดยจะแบ่งการประมวลผลระหว่างไคลเอนต์และเซิร์ฟเวอร์ ซึ่งโปรแกรมประยุกต์ (Application Program) จะทำการประมวลผลบางส่วนที่ไคลเอนต์และบางส่วนที่เซิร์ฟเวอร์

2.5.1 ประเภทของเซิร์ฟเวอร์ (Type of Servers)

File Server เป็นเซิร์ฟเวอร์ที่มีหน้าที่เสมือนเป็นฮาร์ดดิสก์รวม มีหน้าที่จัดเก็บไฟล์ ซึ่งเป็นโมเดลที่ง่ายและราคาถูก ตัวอย่างเช่น PC LAN โดย Netware ไม่ได้มีหน้าที่ run program เป็นเพียงผู้ที่เก็บโปรแกรมและแจกจ่ายให้ผู้ใช้แต่ละคนนำไปใช้ Print Server หรือ Fax Server เป็นการส่งผ่านข้อมูลลักษณะ File เช่นเดียวกัน

Database Server เป็นเซิร์ฟเวอร์ที่ใช้เพื่อรันระบบที่เป็นฐานข้อมูล การทำงานของโมเดลนี้คือ ไคลเอนต์จะเรียกโดยการส่ง SQL request message ไปยังเซิร์ฟเวอร์ จากนั้นระบบจัดการฐานข้อมูล (Database Management System) จะทำการประมวลผลคำสั่งในการค้นหาหรือประมวลผลข้อมูลในฐานข้อมูลแล้วส่งผลลัพธ์กลับไปยังไคลเอนต์

Transaction Server โมเดลนี้สร้างขึ้นมาเพื่อแก้ปัญหาการชนกันของข้อมูลในเซิร์ฟเวอร์ขณะที่มีการเรียกใช้งานอย่างหนัก ลักษณะการทำงานของ Transaction Server คือรองรับการเรียกใช้งานทั้งหมดจากผู้ใช้ แล้วส่งไปประมวลผลที่เซิร์ฟเวอร์ในอัตราสูงสุดไม่เกินปริมาณที่เซิร์ฟเวอร์สามารถจะรับได้ในแต่ละช่วงเวลา

Groupware Server ใช้งานกับข้อมูลชนิดที่ไม่มีโครงสร้าง (Un - Structure) หรือกึ่งโครงสร้าง (Semi - Structure) รองรับงานจำพวก Email , Scheduling , Conferencing , Workflows หรือบางครั้งก็เป็น Multimedia หรืองานที่มีลำดับการทำงานไม่แน่นอน

Object Server ใช้วิธี Object Oriented Programming Language ในการเขียนและมี ORB (Object Request Broker) จะเป็นตัวเชื่อมต่อระหว่างผู้ใช้งานกับเซิร์ฟเวอร์ โดยเมื่อผู้ใช้งานจะติดต่อไปหาเซิร์ฟเวอร์ Object ก็จะติดต่อไปหา ORB เพื่อบอกว่าต้องการ Object อะไร ORB ก็จะจัดสรรให้ตามต้องการ

Web Application Server เป็น Application Level Protocol ซึ่งเดิมเป็นระบบ Web - Client เท่านั้น กล่าวคือ ผู้ใช้จะมีเบราว์เซอร์ที่จะร้องขอไปยัง host ที่ต้องการ แล้ว host ก็จะทำการส่ง html กลับมายังเครื่องของผู้ใช้เพื่อประมวลผลแล้วแสดงผลออกมาปัจจุบันพัฒนาเป็น Front - End Application Client

2.5.2 รูปแบบของไคลเอนต์/เซิร์ฟเวอร์

1) **Fat Client/ Server** คือ ระบบ Client / Server ที่มีภาระของการทำงานมากทางฝั่งของ Server หรือต้องพึ่งการคำนวณจาก Server เป็นหลัก

2) **Thin Client/ Server** คือ ระบบ Client / Server ที่ภาระการทำงานหรือการคำนวณข้อมูลต่างๆ จากฝั่ง Server มีน้อยกว่าแบบ Fat Client แต่ภาระการทำงานนั้นจะไปอยู่ที่ฝั่ง Client มากขึ้น

3) **Distributed Client / Server** คือ ระบบ Client / Server ที่ใช้การทำงานของ Server หลายๆ ตัวรวมกัน โดยที่แต่ละ Server จะทำหน้าที่ที่แตกต่างกัน ในการประมวลผลข้อมูลบางอย่างที่ Client ต้องการ

2.6 ความรู้เบื้องต้นเกี่ยวกับเว็บเซอร์วิส (Web Services) ¹⁴

เว็บเซอร์วิส (Web Services) เป็นเว็บแอปพลิเคชันยุคใหม่ที่ประกอบด้วยส่วนย่อยที่มีความสมบูรณ์ในตัวเอง มีความสามารถด้านการสนับสนุนการแลกเปลี่ยนข้อมูลกันระหว่างเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย ซึ่งจะเชื่อมบริการหลายๆ อย่างเข้าด้วยกัน โดยอาศัยการส่งข้อมูลผ่านอินเทอร์เน็ตโปรโตคอล และใช้ XML (Extensible Markup Language) เป็นมาตรฐานที่จะทำให้เว็บเซอร์วิสติดต่อกันได้อย่างมีประสิทธิภาพ โปรแกรมประยุกต์ที่เขียนโดยภาษาต่างๆ ที่ทำงานบนอุปกรณ์ที่ต่างกัน ซึ่งจะสามารถใช้เว็บเซอร์วิสเพื่อแลกเปลี่ยนข้อมูลกันได้ เช่น โปรแกรมที่พัฒนาโดยภาษา JAVA ก็ใช้ Apache Axis ในการพัฒนา โปรแกรมที่พัฒนาโดย ASP ก็ใช้ .NET (VB.NET, C#) ในการพัฒนา ส่วนโปรแกรมที่พัฒนาโดย PHP ก็ใช้ nuSOAP และ PHP SOAP ในการพัฒนา เป็นต้น นอกจากนี้เว็บเซอร์วิสยังสามารถทำงานได้บนทุกอุปกรณ์อีกด้วย

ส่วนประกอบหลักของเว็บเซอร์วิส

2.6.1 SOAP (Simple Object Access Protocol) เป็นโปรโตคอลที่ใช้ในการสื่อสารระหว่างแอปพลิเคชัน ซึ่งเป็นการเรียกใช้งานคอมพิวเตอร์ข้ามเครื่อง ข้ามแพลตฟอร์ม หรือข้ามภาษาที่ใช้ในการพัฒนา โดยจะทำงานร่วมกับโปรโตคอล HTTP และรูปแบบข้อความที่สื่อสารกันด้วยภาษา XML

2.6.2 WSDL (Web Services Description Language) เป็นสิ่งที่อธิบายรายละเอียดในการติดต่อกับของเว็บเซอร์วิส เพื่อให้แอปพลิเคชันทราบถึงลักษณะต่างๆ ของเว็บเซอร์วิสที่ทำการเรียกใช้งาน ซึ่งทำให้รู้ว่าเซอร์วิสนั้นมีบริการอะไรและจะติดต่อได้อย่างไร

2.6.3 UDDI (Universal Description Discovery and Integration) เป็นไดเรกทอรี (directory) ที่ใช้ในการเก็บรวบรวมเซอร์วิสต่างๆ ในอินเทอร์เน็ตไว้ ณ แหล่งเดียวกันเพื่อให้เกิดความสะดวกแก่ผู้ใช้คล้ายกับสมุดหน้าเหลืองที่ใช้ในการเปิดเพื่อหาเบอร์โทรศัพท์

2.7 Google Application Programming Interface : Google API ¹⁵⁾

Google Application Programming Interface : Google API สำหรับในโปรเจกต์นี้จะกล่าวถึงเฉพาะ Google Calendar API ซึ่งเกี่ยวข้องกับเซิร์ฟเวอร์ที่ได้พัฒนาขึ้น ซึ่งจะใช้ในการติดต่อเพื่อจัดการข้อมูลกับปฏิทินกูเกิล (Google Calendar)

2.7.1 การระบุตัวตนของ Google Calendar service (Authentication to the Calendar service)

การเพิ่มเติม แก้ไข หรือการลบข้อมูลนั้น จะต้องมีการระบุตัวตนเพื่อให้สามารถใช้ Google service ได้ ซึ่งใน Google Calendar API นี้ ได้จัดสรรระบบระบุตัวตนไว้แล้วซึ่งขึ้นอยู่กับชนิดของผู้ใช้ที่ผู้พัฒนา ซึ่งสามารถแบ่งออกได้เป็น 3 ชนิด คือ

1) การระบุตัวตนแบบ AuthSub proxy (AuthSub proxy authentication) คือ การระบุตัวตนที่ใช้สำหรับผู้ใช้งานที่ใช้ระบบผ่านเว็บแอปพลิเคชันที่ต้องการระบุตัวตนของตนเองเพื่อเข้าถึงบัญชีของกูเกิล (Google Account) โดยที่ไม่ต้องเข้าถึงระบบผ่านเว็บไซต์ที่ให้บริการ

2) การระบุตัวตนแบบ ClientLogin (ClientLogin username / password authentication) คือ การระบุตัวตนที่ต้องการ username และ password เพื่อเข้าสู่บัญชีของกูเกิล ซึ่งถ้าหากต้องการจะรับการติดต่อผ่านกลไกของ ClientLogin นั้น จะต้องใช้การส่งค่าข้อมูล (POST) ชื่อผู้ใช้ (username) และ รหัสผ่าน (password) ส่งไปที่ <https://www.google.com/accounts/ClientLogin> สำหรับรูปแบบของ POST นั้น จะประกอบไปด้วย ชุดของ query parameters หลาย ๆ ตัว ซึ่งมีรายละเอียดดังต่อไปนี้

ตารางที่ 2.7.1 แสดงรายละเอียดของ query parameters

Parameter	Description
Email	The user's email address.
Passwd	The user's password.
Source	Identifies your client application. Should take the form <i>companyName-applicationName-versionID</i> ; below, we'll use the name <i>exampleCo-exampleApp-1</i> .
Service	The string <i>cl</i> , which is the service name for Google Calendar.

ที่มา : http://code.google.com/intl/th-TH/apis/calendar/data/2.0/developers_guide_protocol.html

ถ้าหากการระบุตัวตนเกิดล้มเหลว จะได้รับข้อมูลสถานะโค้ดเป็น HTTP 403 Forbidden ถ้าหากว่าสามารถระบุตัวตนสำเร็จแล้วนั้น จะได้รับการตอบสนองจากเซิร์ฟเวอร์ในสถานะโค้ดเป็น HTTP 200 OK และโค้ดชุดตัวเลขเริ่มต้น 3 ชุดซึ่งอยู่ภายใน body ของข้อมูลสถานะที่ได้รับมา ซึ่งประกอบไปด้วย SID LSID และ Auth ซึ่งค่าของ Auth นั้นคือ authorization token ซึ่งใช้ในการร้องขอใช้งานเซิร์ฟเวอร์ต่าง ๆ ของปฏิทินกูเกิล

3) การระบุตัวตนโดยใช้ Magic cookie (Magic cookie authentication) เป็นวิธีการที่ง่ายที่สุดในการใช้ feed URL โดยจะเรียกว่า magic cookie ซึ่งใช้ได้เฉพาะการอ่านข้อมูลของปฏิทินกูเกิลเท่านั้น เนื่องจากว่าเป็น URL feed ที่ไม่ต้องใช้การระบุตัวตน ซึ่งโดยปกติแล้วขั้นตอนการเลือกใช้ URL นั้นจะเกี่ยวข้องกับการใช้ JavaScript GUI browser ในการรับ URL แต่หากว่าไม่ต้องการที่จะใช้เบราว์เซอร์เราสามารถใช่วิธีอื่นดังที่กล่าวไปข้างต้นในการระบุตัวตนเพื่อเข้าถึงปฏิทินกูเกิลได้ ซึ่งวิธีในการหา magic cookie ของปฏิทินกูเกิล สามารถทำได้ดังนี้

3.1) ในช่องรายการของปฏิทินกูเกิลด้านซ้ายมือ จะพบช่องชื่อของปฏิทินทั้งหมดที่มี ซึ่งเราสามารถแก้ไขทุกอย่างเกี่ยวกับปฏิทินของเราได้ดังนี้

3.2) กดที่ลูกศรสามเหลี่ยมปฏิทินที่ต้องการ แล้วเลือกการตั้งค่าปฏิทิน

3.3) เลือกลงมาที่หัวข้อ ที่อยู่ส่วนบุคคล แล้วเลือกกลุ่มที่เขียนว่า XML

3.4) จะได้ URL ที่เป็น magic cookie มา ซึ่งถ้าปฏิทินนี้ถูกตั้งให้สามารถเปิดเผยสู่สาธารณะได้ เราจะสามารถใช้ URL นี้ในการเข้าถึงเพื่ออ่านข้อมูลจากปฏิทินนี้ได้โดยไม่ต้องมีการระบุตัวตน

2.7.2 ชนิดของ feed ใน Google Calendar (Google Calendar feed types)

ปฏิทินกูเกิลได้จัดเตรียมวิธีการในการจัดการกับข้อมูลของปฏิทินมาหลายลักษณะซึ่งสามารถจำแนกได้ 2 แบบ นั่นคือ visibility และ projection

Visibility value แสดงถึงการแสดงปฏิทินว่าเป็นสาธารณะหรือไม่ ตัวอย่างเช่น visibility value ของ public แสดงถึง ความสามารถในการเข้าถึงหรืออ่านได้จากสาธารณะชน

Projection value แสดงถึงจำนวนข้อมูลหรือชนิดที่มีอยู่ในสิ่งนั้น ๆ ตัวอย่างเช่น value ของ basic จะแสดงถึงการเป็นตัวแทนของ Atom feed ซึ่งไม่มี extension element ประกอบอยู่

2.7.3 Calendar feeds

สำหรับ feed ของปฏิทินกูเกิลจะมีอยู่ทั้งหมด 3 ชนิดที่จะส่งคืนค่าของชุดของ Calendar entries ซึ่งได้แก่

1) Metafeed คือ feed ที่ใช้ในการอ่านข้อมูลเพียงอย่างเดียว ซึ่ง URL สำหรับ metafeed คือ <https://www.google.com/calendar/feeds/default>

2) allcalendars feed คือ feed ที่สามารถอ่านและเขียนได้ โดยใช้เพื่อการจัดการเกี่ยวกับ subscription และการตั้งค่าส่วนบุคคลของเจ้าของปฏิทินนั้น ซึ่งแตกต่างจาก metafeed ตรงที่ allcalendars feed สามารถแก้ไขได้ ซึ่งทำให้เราสามารถเพิ่มเติม หรือลบ Calendar entries แต่การ update ข้อมูลนั้น จะไม่สามารถใช้ allcalendars ในการ update ได้ แต่การ update ทำได้เพียงในส่วนของการตั้งค่าข้อมูลส่วนบุคคลเท่านั้น URL สำหรับ allcalendars feed คือ <https://www.google.com/calendar/feeds/default/allcalendars/full>

3) owncalendars feed คือ feed ที่สามารถแก้ไขข้อมูลในปฏิทินได้ ซึ่งใช้ในการจัดการกับปฏิทินโดยตรง ซึ่งสำหรับ owncalendars นั้นสามารถทำการเพิ่ม หรือลบ calendar entries ได้ URL ของ owncalendars feed คือ <http://www.google.com/calendar/feeds/default/owncalendars/full>

สำหรับ feed ทั้งสามนี้จะป็น private feed นั้นหมายถึงจะต้องมีการระบุตัวตนก่อนถึงจะสามารถใช้ feed เหล่านี้ได้

2.7.4 Google Calendar element reference

แต่ละ entry ใน Calendar feed คือ ชนิดของเหตุการณ์ (Event) ซึ่งแต่ละเหตุการณ์จะประกอบไปด้วยบางส่วนดังต่อไปนี้

1) One or more <gd:who> element ซึ่งแสดงถึงการเชื่อมโยงกันของคนด้วยเหตุการณ์ซึ่ง <gd:who> element ใน Calendar event นั้นคือส่วนขยายที่มาจาก gCal:additionalGuests

2) A <gd:comments> element นั่นคือจุดที่ feed บอกว่าส่วนนั้นคือ comment

3) One or more <gd:extendedProperty> element ใช้ในการเก็บ custom date ตัวอย่างเช่น เครื่องมือที่ใช้ในการ synchronization หรืออาจจะใช้ในการเก็บคุณสมบัติเพิ่มเติมของเหตุการณ์ของปฏิทินที่ไม่ได้ระบุไว้

เราจะพบความแตกต่างเล็กน้อยระหว่าง <feed> ของ Calendar Data API และ <feed> ของ Google Data APIs นั่นคือ

1) ใน Calendar <author> element เป็น feed ที่ไม่สามารถขาดได้ (สำหรับ Google Data APIs ไม่จำเป็น)

2) ใน Calendar <atom:linkrel="http://schemas.google.com/g/2005#feed" type="application/atom+xml" href="..." /> element จำเป็นที่จะต้องมีเพื่อให้สามารถ update feed ได้

3) ในอีกส่วนที่แตกต่างกันนั่นคือ ปฏิทินจะสนับสนุน rel และ type attribute value สำหรับ <atom:link> ซึ่งจะไม่สามารถใช้ได้ ใน Google Data APIs ดังตารางต่อไปนี้

ตารางที่ 2.7.2 แสดงรายละเอียดของ rel and type value

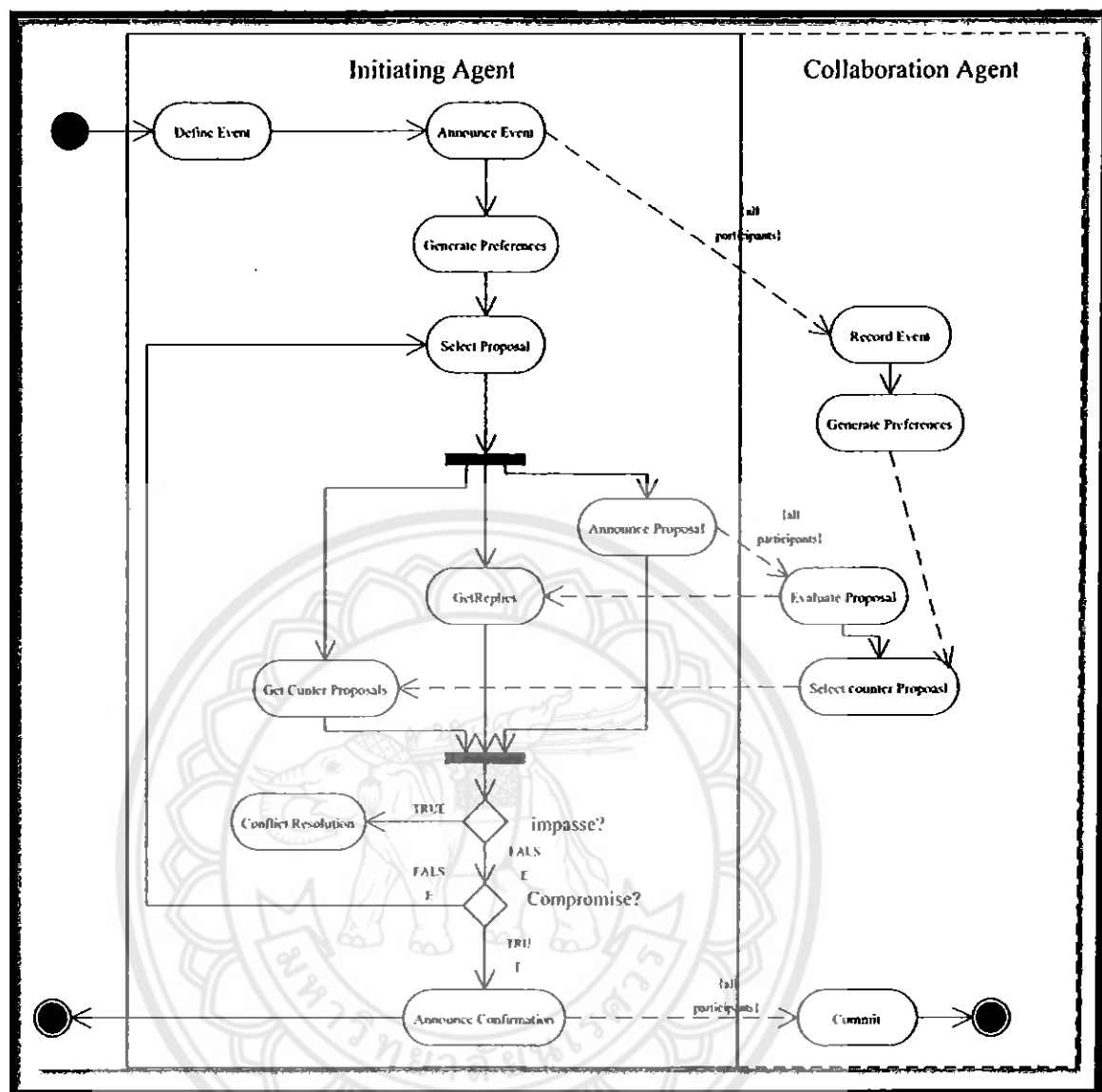
rel and type value	Description
atom:link[@rel='alternate'][@type='text/calendar']	Specifies the URL of the ICS source of the calendar, if it came from an ICS source.
atom:link[@rel='source'][@type='application/atom+xml']	Specifies the URL of the Data API source of the calendar, if it came from a Data API source.
atom:link[@rel='related']	Specifies a URL related to the feed.
atom:link[@rel='http://schemas.google.com/gCal/2005/webContent'] [@type='type']	Specifies "web content" for an event; for more information, see gCal:webContent.

ที่มา : <http://code.google.com/intl/th-TH/apis/calendar/data/2.0/reference.html#Elements>

4) ส่วนสุดท้ายคือ Calendar <entry> element นั้นอาจจะประกอบด้วย <atom:source> element ได้ซึ่งไม่สามารถใช้ entry นี้ใน Google Data APIs ตัวอื่นๆ ได้

2.8 N* Algorithm ¹⁶⁾

N* Algorithm เป็นอัลกอริทึม ชนิดหนึ่งของปัญญาประดิษฐ์แบบกระจาย (distributed AI : DAI) ใช้สำหรับการหาเวลาที่เหมาะสมในการจัดประชุมในเงื่อนไขที่หลากหลาย ซึ่งอัลกอริทึมชนิดนี้มีพื้นฐานมาจาก A* search



รูปที่ 2.3 Activity Diagram ของ N^* algorithm (an agent-based negotiation algorithm for dynamic scheduling and rescheduling Hon Wai Chun*, Rebecca Y.M. Wong)

อัลกอริทึมนี้จะช่วยในการหาข้อเสนอ (Proposal) ที่เหมาะสมและนำไปใช้ในการจัดเหตุการณ์ต่อไป ซึ่งการประยุกต์ใช้งานนั้นจะต้องทราบรายละเอียดของค่าต่างๆ ต่อไปนี้

Event (E_i) ประกอบด้วย

- Level ไว้บอกความสำคัญของเหตุการณ์ หากมีข้อขัดแย้ง (conflict) ในระหว่างการเจรจา (negotiate) เกิดขึ้น level เป็นตัวบอกว่าจะให้เจรจาเหตุการณ์ โดยการตั้งค่า (preference) จากผู้ใช้งานหรือยกเลิกเหตุการณ์นั้นไปเลย

- Fixed attribute (f) เป็นคุณลักษณะที่บอกถึงสิ่งที่ไม่สามารถเปลี่ยนแปลงได้ในการเจรจา เช่น สถานที่ ผู้จัดกิจกรรม (organizer) อาจจะเจาะจงว่าจะต้องเป็นที่ใดที่หนึ่งไปเลย

- Variable attribute (v) เป็นคุณลักษณะที่บอกถึงสิ่งที่เปลี่ยนแปลงได้ ซึ่งคำตอบ (solution) ของการเจรจาจะเป็นเซตของ variable attribute นี้ เช่น เวลา (8.00 , 9.00 , 10.00) โดยในแต่ละ variable attribute ต้องประกอบด้วย

o โดเมน (domain) เช่น เวลา อาจจะมีโดเมน (8.00 , 9.00 , ... , 17.00) หรือสถานที่ หากเป็น variable attribute อาจจะมีโดเมนเป็น (EN212, EN305, EN609)

o ลำดับความสำคัญ (attribute priority : ap) เพื่อบอกถึงความสำคัญของ variable attribute แต่ละตัว เช่น เวลา อาจจะมี priority = 10 และสถานที่อาจจะมี priority = 1 ซึ่งค่าสำหรับ priority นั้น อยู่ในช่วง $0 - ap_{max}$ ซึ่ง ap_{max} เป็น global constant ค่าหนึ่ง

o การกำหนดค่า (preference value : pv) เนื่องจาก variable attribute มีโดเมนเป็นของตัวเอง จึงต้องมี pv เป็นตัวกำหนด ค่าในโดเมนแต่ละตัวตามที่ผู้ใช้แต่ละคนตั้ง เช่น ผู้ใช้คนที่ 1 ตั้งค่าในโดเมนของเวลาว่า (8.00 = 10, 9.00 = 9, 10.00 = 2, ... , 17.00 = 1) โดยที่ pv มีค่าระหว่าง $0 - pv_{max}$ ซึ่งค่าของ pv นั้นหากมีค่ามากจะบอกถึงว่าผู้ใช้คนนั้นสนใจที่จะเข้าร่วมกิจกรรมนั้นมาก เช่น ผู้ใช้คนที่ 1 ต้องการเข้าร่วมประชุมเวลา 8.00 มากกว่าเวลา 9.00

- Preference Level (pl) บอกถึง ความสนใจหรือความต้องการในการเข้าร่วมเหตุการณ์ตามแต่ละข้อเสนอ ตามตารางที่ 2.8.1

ตารางที่ 2.8.1 แสดงข้อเสนอต่าง ๆ ของการเข้าร่วมกิจกรรม

Proposal#1	agent#1		agent #2	
	Priority	Preference	Priority	Preference
Location :: EN305	2	3	5	1
Time :: 09.00	7	9	9	0
pl	$2*3 + 7*9 = 69$		$5*1 + 9*0 = 5$	

ซึ่งอาจจะหมายความว่า ผู้ใช้คนที่ 1 มีความสนใจที่จะเข้าร่วมกิจกรรมนี้ (เนื่องจากค่า pl สูงถึง 69) หากข้อเสนอนี้ ได้รับการจัดขึ้น แต่ตรงกันข้ามผู้ใช้คนที่ 2 อาจจะไม่เห็นด้วย (pl มีค่าเพียง 5) และอาจจะเสนอข้อเสนอของตัวเอง (counter proposal) เพื่อให้เกิดการเจรจาอีกครั้ง

* หมายเหตุ สำหรับค่า pl ที่ใช้สำหรับบอกว่าผู้ใช้แต่ละคนจะยอมรับหรือไม่นั้น ขึ้นอยู่กับ การกำหนดค่าและสภาพแวดล้อมต่าง ๆ

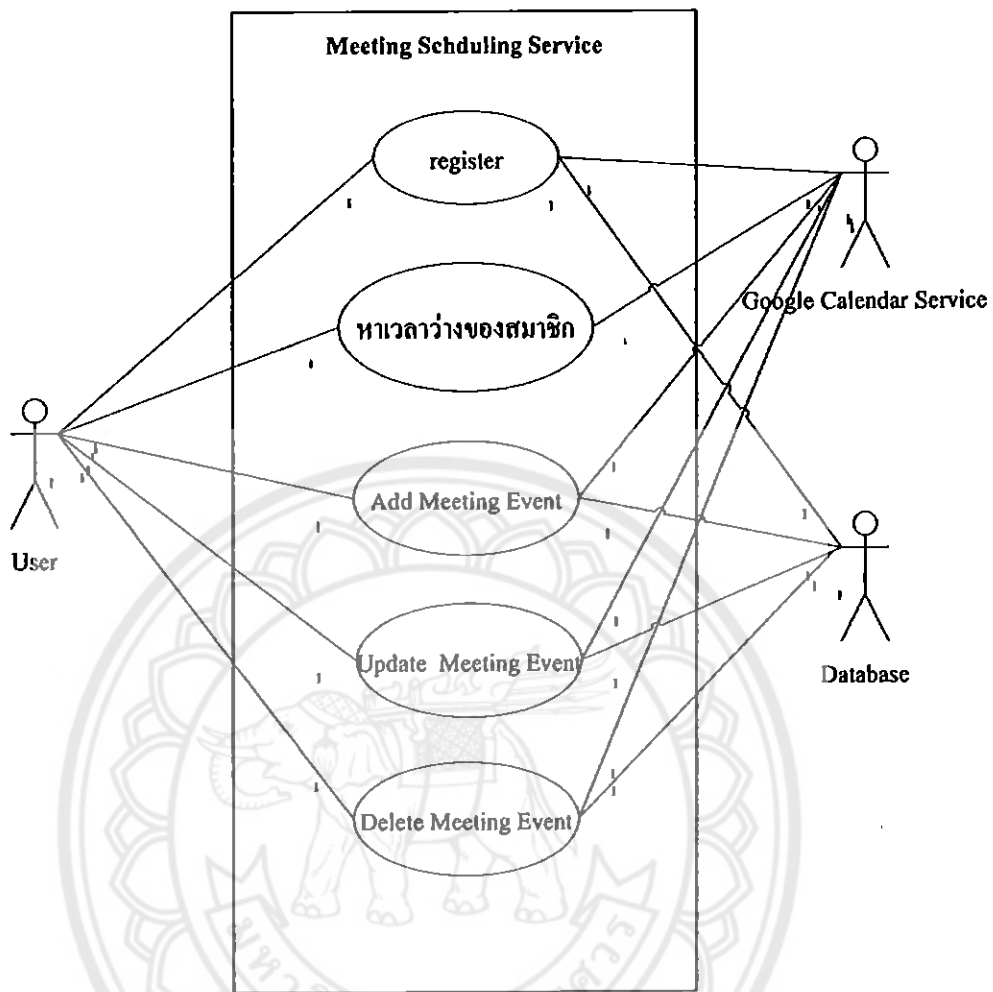
บทที่ 3

วิธีการดำเนินงาน

บทนี้จะกล่าวถึงวิธีการดำเนินการออกแบบและพัฒนาระบบ เพื่อสะดวกในการอธิบาย และการทำความเข้าใจ ซึ่งแบ่งออกเป็น ส่วน ๆ ดังนี้

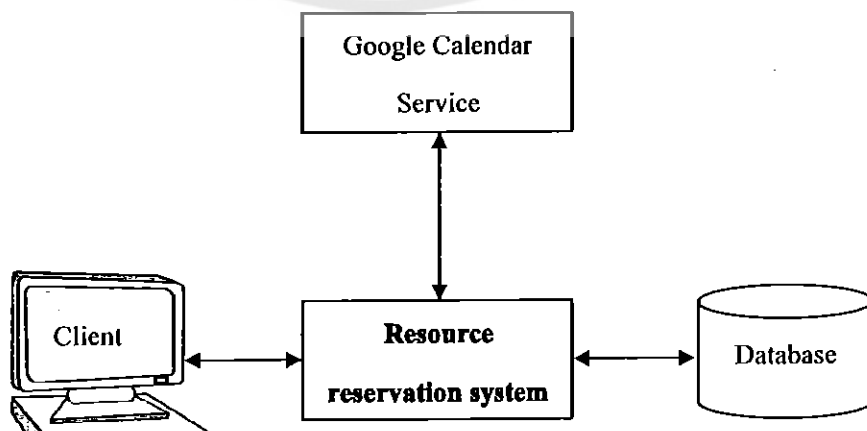
1. ออกแบบ แบบจำลองการทำงานของระบบ
2. รูปแบบของโปรแกรม
 - 2.1 ออกแบบเซอร์วิสและอัลกอริทึมสำหรับการหาเวลาว่างและการจัดประชุม
 - 2.2 ใช้งานเซอร์วิสของปฏิทินกูเกิล (Google Calendar)
 - 2.3 ออกแบบโปรแกรมในส่วนผู้ใช้บริการ (Client)
 - 2.4 ออกแบบระบบฐานข้อมูล
3. พัฒนาโปรแกรมต้นแบบ (Toy Application)
 - 3.1 โปรแกรมติดต่อเซอร์วิสของปฏิทินกูเกิล (Google Calendar)
 - 3.2 โปรแกรมติดต่อ Database
4. พัฒนาโปรแกรมและฐานข้อมูล
5. แก้ไขข้อผิดพลาดและทำการทดสอบซ้ำ
6. ทดสอบการใช้งานจริง

3.1 Use case Diagram



รูปที่ 3.1 Use Case Diagram แสดงการใช้งานระบบของ User

3.2 แบบจำลองการทำงานจากระบบ



รูปที่ 3.2 แบบจำลองการทำงานจากระบบ

สามารถแยกระบบการทำงานออกได้เป็น 4 ส่วน ซึ่งได้แก่ ส่วนการจัดการการจัดประชุม, ส่วนฐานข้อมูล, ส่วนเซิร์ฟวิจากปฏิทินกูเกิล (Google Calendar) และส่วนฝั่งผู้ใช้งานไคลเอนต์ (Client)

3.2.1 เซอร์วิซของปฏิทินกูเกิล (Google Calendar service)

เป็นเซิร์ฟวิซที่ระบบเรียกใช้เพื่อขอข้อมูลตารางเวลาของผู้ใช้งานและบุคคลที่เกี่ยวข้องมาใช้ในการหาเวลาว่างและจัดประชุม

ตัวอย่างการติดต่อ Google Calendar API

1) โค้ดตัวอย่างการดึงตารางปฏิทินกูเกิล

```
public List<CalendarData> getCalendar() {
    List<CalendarData> calendarDataList = new List<CalendarData>();
    CalendarData calResult = new CalendarData();
    try {
        CalendarService myService = new CalendarService("ContractCalendarService");
        myService.setUserCredentials(user, pass);
        CalendarQuery query = new CalendarQuery();
        query.Uri = new Uri("http://www.google.com/calendar/feeds/default/owncalendars/full");
        CalendarFeed cal_Feed = myService.Query(query);
        foreach (CalendarEntry centry in cal_Feed.Entries) {
            CalendarData calResultTemp = new CalendarData();
            calResultTemp.calendarName = centry.Title.Text;
            calResultTemp.calendarURI = "http://www.google.com/calendar/feeds/" +
            ((CalendarEntry)centry).SelfUri.ToString().Substring(((CalendarEntry)centry).SelfUri.ToString()
            .LastIndexOf("/") + 1) + "/private/full";
            calendarDataList.Add(calResultTemp);
        }
    } catch (Exception ex) {
        calResult.calendarName = ("Error cannot get event cause :: " + ex);
        calendarDataList.Add(calResult);
        return calendarDataList;
    }
    return calendarDataList;
}
```

รูปที่ 3.3 โค้ดตัวอย่างการดึงตารางปฏิทินกูเกิล

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์

โค้ดตัวอย่างนี้เป็นตัวอย่างร้องขอปฏิทินของผู้ใช้โดยการส่ง Username และ Password ของผู้ใช้นั้น ๆ ไป และสิ่งที่ได้กลับมานั้นคือ รายการปฏิทินทั้งหมดของผู้ใช้นั้น

2) โค้ดตัวอย่างการสร้างเหตุการณ์ลงปฏิทินฉุกเฉิน

```
public string createEvent(EventData eventData, ProxyData proxyData) {
    try {
        Google.GData.Calendar.EventEntry entry = new Google.GData.Calendar.EventEntry();
        entry.Title.Text = eventData.name;
        entry.Content.Content = eventData.description;
        Where eventLocation = new Where();
        eventLocation.ValueString = eventData.place;
        entry.Locations.Add(eventLocation);
        System.DateTime dstartdateime = eventData.startTime;
        System.DateTime dtenddatetime = eventData.endTime;
        When eventTime = new When(dstartdateime, dtenddatetime);
        entry.Times.Add(eventTime);
        string userName = user;
        string passWord = pass;
        CalendarService service = new CalendarService("CreateCalendarService");
        if (userName != null && userName.Length > 0) {
            service.setUserCredentials(userName, passWord);
        }
        Uri postUri;
        postUri = new Uri(calendarURI);
        GDataGAAuthRequestFactory requestFactory = (GDataGAAuthRequestFactory) -
service.RequestFactory;
    }
    requestFactory.CreateRequest(GDataRequestType.Insert, postUri);
    AtomEntry insertedEntry = service.Insert(postUri, entry);
    return "Event Successfully Added";
} catch (Exception ex) {
    return "Error :: " + ex.Message;
}
}
```

15755578

ร.ร.

รูปที่ 3.4 โค้ดตัวอย่างการสร้างเหตุการณ์ลงปฏิทินฉุกเฉิน

216345

2557

ในโค้ดตัวอย่างนี้เป็นการร้องขอปฏิทินวันเกิดให้มีการสร้างเหตุการณ์ลงในปฏิทินของผู้ใช้ที่เป็นเจ้าของปฏิทินวันเกิดนั้น ๆ โดยการส่ง Username Password รวมไปถึงรายละเอียดข้อมูลในการสร้างปฏิทิน นั่นคือ ชื่อเหตุการณ์ รายละเอียดเหตุการณ์ สถานที่ วัน เวลา เริ่มและจบ เพื่อให้ทางวันเกิดทำการสร้างปฏิทิน ลงในตารางปฏิทินของผู้ใช้คนนั้น

3.2.2 ส่วนฐานข้อมูล

เป็นส่วนที่ไว้จัดการเก็บข้อมูลของการจัดกิจกรรม เพื่อใช้ในการแสดงผลและการประมวลผลของระบบ

ตัวอย่างการติดต่อ Database โดยการใช้ Linq to Sql

1) ตัวอย่างการดึงข้อมูล Username จากตาราง Account โดยการใช้ Linq to Sql

```
public string getUser(int id)
{
    string user = "";
    dbEventClassesDataContext LDB = new dbEventClassesDataContext();
    var res = from r in LDB.Accounts
              where r.AccountID == id
              select r;
    foreach (var str in res) {
        user = (str.Username);
    }
    return user;
}
```

รูปที่ 3.5 ตัวอย่างการดึงข้อมูล Username จากตาราง Account โดยการใช้ Linq to Sql

โค้ดตัวอย่างนี้เป็นการดึงชื่อ Username จาก Database โดยการส่ง id หรือ AccountID ที่ต้องการไปให้และผลลัพธ์ที่ได้กลับมาคือ ชื่อ Username ที่เป็นของ AccountID นั้น ๆ

2) ตัวอย่างการเพิ่มข้อมูลลง Database

```

public string addEventTable(EventDataeventData, int organizer)
{
    try
    {
        dbEventClassesDataContext eventDB = new dbEventClassesDataContext();
        DateTime now = System.DateTime.Now;
        EventmyEvent = newEvent();
        myEvent.Organizer = organizer;
        myEvent.Title = eventData.name;
        myEvent.StartDate = eventData.startTime;
        myEvent.EndDate = eventData.endTime;
        myEvent.Location = eventData.place;
        myEvent.Description = eventData.description;
        myEvent.allDay = "false";
        myEvent.CreateDate = now;
        myEvent.LastUpdateDate = now;
        myEvent.PicResource = eventData.PicResource;
        myEvent.PicResourceMimeType = eventData.PicResourceType;
        eventDB.Events.InsertOnSubmit(myEvent);
        eventDB.SubmitChanges();
    } catch (Exception e) {
        return "Error :: " + e.Message;
    }
    return "AddComplete!!";
}

```

รูปที่ 3.6 ตัวอย่างการเพิ่มข้อมูลลง Database

โค้ดตัวอย่างการเพิ่มข้อมูลลงในดาต้าเบส โดยการสร้าง object ขึ้นมาเพื่อใส่ข้อมูลใน attribute ต่างๆ และทำการเพิ่ม โดยการใช้คำสั่ง InsertOnSubmit และ SubmitChanges

3) ตัวอย่างโค้ดการลบข้อมูลจากฐานข้อมูล

```

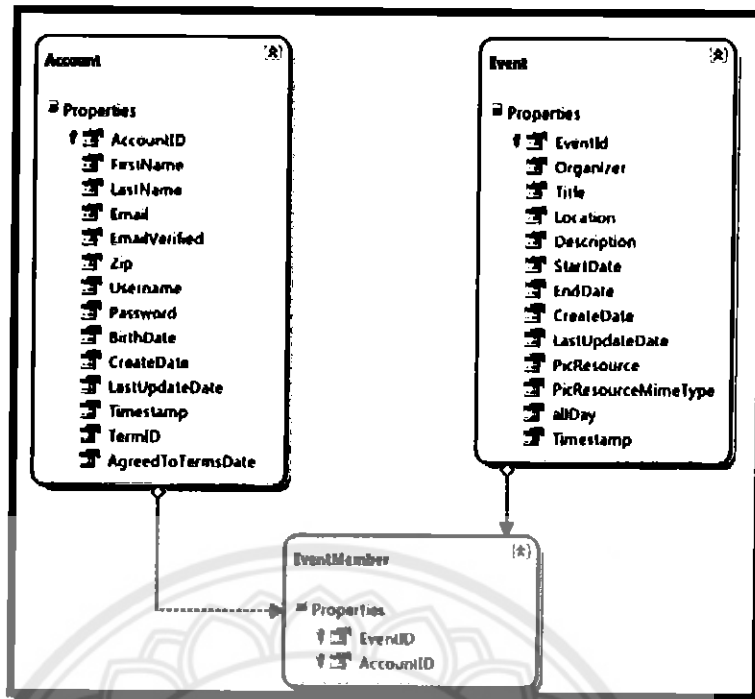
public string deleteEventTable(EventData eventData, int organizer)
{
    try
    {
        dbEventClassesDataContext eventDB = new dbEventClassesDataContext();
        var deleteEvent = from eventDeleted in eventDB.Events
        where eventDeleted.Title == eventData.name&&eventDeleted.Organizer == organizer
        select eventDeleted;
        foreach (var detail in deleteEvent) {
            eventDB.Events.DeleteOnSubmit(detail);
        }
        eventDB.SubmitChanges();
    } catch (Exception e) {
        return "Error :: " + e.Message;
    }
    return "Deleted Complete!!";
}

```

รูปที่ 3.7 ตัวอย่างโค้ดการลบข้อมูลจากฐานข้อมูล

โค้ดตัวอย่างการลบข้อมูลจากฐานข้อมูล โดยการเลือกข้อมูลที่ต้องการ จากการคิวรีข้อมูล เก็บไว้ใน object จากนั้น จึงทำการลบ object พวกนั้นออกจากฐานข้อมูล โดยการใช้คำสั่ง DeleteOnSubmit

การพัฒนาฐานข้อมูลของระบบได้พัฒนาฐานข้อมูลร่วมกับฐานข้อมูลหลักของ San Web ซึ่งได้เพิ่มตารางในส่วนของ Event ซึ่งทำหน้าที่เก็บรายละเอียดของการจัดการประชุม เช่น ชื่อหรือหัวข้อในการประชุม, รายละเอียด, ช่วงเวลาการจัด เป็นต้น และตาราง Event Member ซึ่งทำหน้าที่บอกเหตุการณ์ในแต่ละการประชุมว่ามีผู้เข้าร่วมเป็นใคร ซึ่งเป็นตารางความสัมพันธ์ระหว่างตาราง Account ที่เก็บรายละเอียดของผู้ใช้กับตาราง Event ที่ได้ออกแบบไว้ ซึ่งได้ใช้ Microsoft SQL Server ในการทำงาน



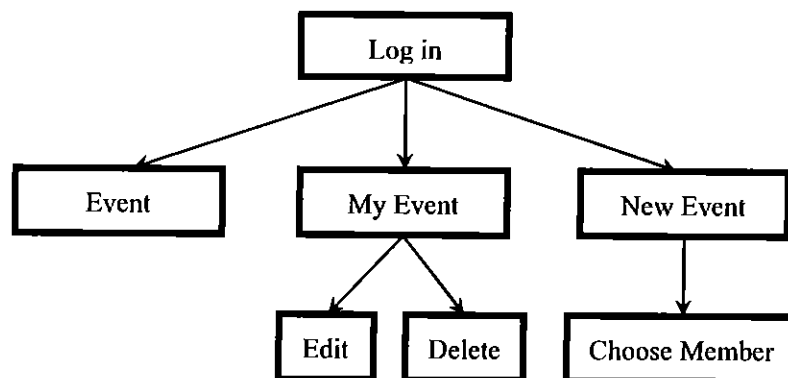
รูปที่ 3.8 ตารางความสัมพันธ์ของฐานข้อมูลที่เกี่ยวข้องกับการจัดเหตุการณ์

3.2.3 ส่วนไคลเอนต์

เป็นส่วนที่ใช้สำหรับติดต่อกับผู้ใช้งาน โดยตรงเพื่อให้ผู้ใช้งานสามารถเข้าถึงข้อมูลต่าง ๆ ได้โดยง่าย และเพื่อให้สามารถใช้งานระบบได้อย่างสะดวก ซึ่งแบ่งได้เป็น 2 ส่วน ดังนี้

- 1) ส่วนแสดงผลต่อผู้ใช้งาน ใช้สำหรับแสดงผล และรับข้อมูลต่าง ๆ จากผู้ใช้งานเพื่อนำไปใช้ในระบบต่อไป
- 2) ส่วนติดต่อกับเซิร์ฟวิส เป็นส่วนที่ไว้รับหรือส่งข้อมูลไปสู่ระบบเพื่อใช้ในการจัดประชุมหรือการจัดการต่าง ๆ ภายในระบบต่อไป

3.3 การพัฒนาโปรแกรม และอัลกอริทึมของระบบ



รูปที่ 3.9 แผนผังการทำงานของระบบ

3.3.1 การพัฒนาโปรแกรมของระบบ

พัฒนาโดยใช้ .NET Framework เวอร์ชัน 4.0 โดยเขียนโดย C# บน ASP.NET และพัฒนาในส่วนหนึ่งของ San Web ในส่วนของ Event ซึ่งได้พัฒนาระบบ interface ออกมาทั้งหมด 3 หน้าหลัก คือ

1. หน้า Event ซึ่งเป็นหน้าที่รวบรวมการประชุมที่เกิดขึ้นและกำลังจะเกิดขึ้นภายในสัปดาห์ หรือ เดือน

2. หน้า My Event เป็นหน้าที่แสดงการประชุมทั้งหมดที่เกี่ยวข้องกับผู้ใช้ใน 2 ลักษณะ นั่นคือ ด้านซ้ายมือจะแสดงการประชุมที่ผู้ใช้มีส่วนเกี่ยวข้อง โดยที่ไม่ใช่เจ้าของหรือคนจัดการประชุมนั้น และในด้านขวามือจะแสดงการประชุมที่ผู้ใช้คนนั้น ๆ เป็นคนสร้างขึ้น ซึ่งสามารถจะแก้ไขหรือลบการประชุมดังกล่าวได้

3. หน้า New Event เป็นหน้าที่ไว้สำหรับสร้างการประชุม โดยที่สามารถเลือกจำนวนคนที่จะให้เข้าร่วมเหตุการณ์ได้รวมไปถึงสามารถหาเวลาว่างที่ผู้เข้าร่วมทุกคนว่างได้ตามเงื่อนไขต่าง ๆ คือ การประชุมที่จะจัดต้องการจัดภายในกี่วัน หรือช่วงเวลาที่ต้องการ เช่น 08.00 - 12.00 น. และเงื่อนไขในการจัดการประชุมต้องการระยะเวลาติดต่อกันกี่ชั่วโมง

3.3.2 การพัฒนาอัลกอริทึมที่ใช้ในระบบ

ได้มีการประยุกต์ใช้ N* Algorithm โดยการปรับเปลี่ยนใหม่มีความเหมาะสมกับการจัดหาเวลาที่เหมาะสมกับการจัดการประชุม

ตัวอย่างโค้ดในส่วนของการพัฒนาอัลกอริทึม

```
public List<PreferenceVector> solution(List<AgentData> user, List<AgentData> checker) {
    Queue<PreferenceVector> open = new Queue<PreferenceVector>();
    List<PreferenceVector> close = new List<PreferenceVector>();
    List<PreferenceVector> sol = new List<PreferenceVector>();
    PreferenceVector node = new PreferenceVector();
    List<PreferenceVector> children = new List<PreferenceVector>();
    PreferenceVector first = new PreferenceVector();
    bool checkFindSeedPV = false;
    foreach (AgentData ad in user) {
        foreach (PreferenceVector pv in ad.PreferenceVector_Queue) {
            if (pv.Value == 0) {
                first = pv;
            }
        }
    }
}
```

```

        checkFindSeedPV = true;
        break;
    }
}
if (checkFindSeedPV) {
    break;
}
}
open.Enqueue(first);
int u = 0;
while (open.Count > 0) {
    node = open.Dequeue();
    if (announceProposal(checker, node)) {
        sol.Add(node);
    }
    close.Add(node);
    children.Clear();
    foreach (PreferenceVector PIV in getCounterProposal(user)) {
        if (PIV.PreferenceName != "") {
            children.Add(PIV);
        }
    }
    List<PreferenceVector> temp = new List<PreferenceVector>();
    //Check Infesible
    #region Check Infesible (เขตตัวที่ซ้ำกับ close)
    bool isSame = false;

```

รูปที่ 3.10 ตัวอย่างโค้ดในส่วนการพัฒนาอัลกอริทึม

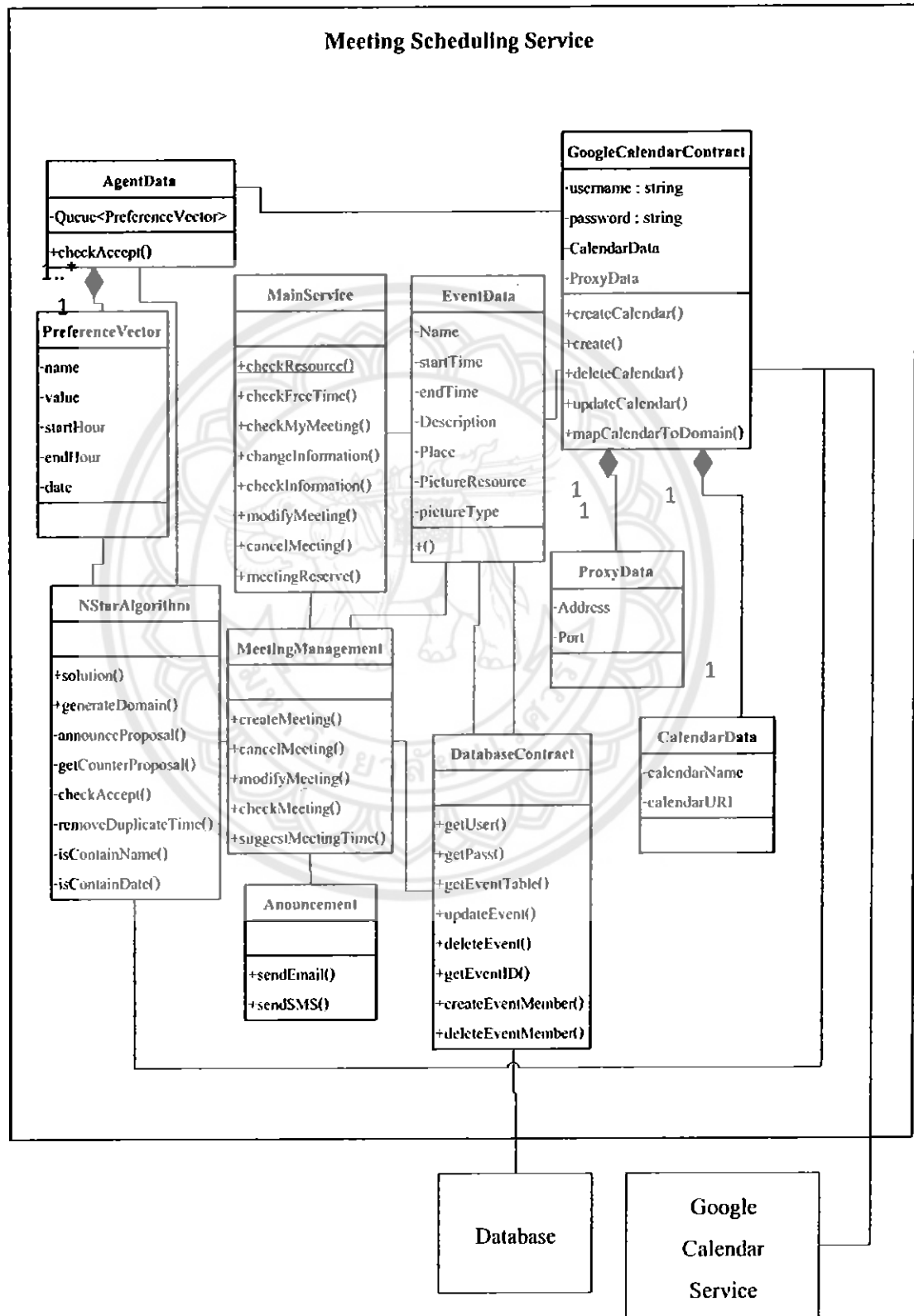
โค้ดนี้เป็นการเริ่มต้นค้นหาคำตอบที่เป็นเวลาของผู้ที่เข้าร่วมเหตุการณ์นั้น ๆ ว่าว่างตรงกัน และทำการวนเช็คต่อจนครบทุกคำตอบที่เป็นไปได้

3.4 รูปแบบของระบบ

การทำงานของโปรแกรมจะทำงานในลักษณะเว็บแอปพลิเคชัน (Web application) ที่จะให้บริการในการจัดประชุมและจองทรัพยากรที่ใช้ในการจัดประชุม โดยเว็บเซิร์ฟเวอร์นี้จะไปนำ

ข้อมูลตารางเวลาของผู้ใช้ที่เกี่ยวข้องจาก Google Calendar service โดยสามารถแบ่งการทำงานของระบบออกได้เป็น 4 ส่วนดังต่อไปนี้

3.4.1 Class diagram ของระบบ



รูปที่ 3.11 Class Diagram ของระบบ

ทำหน้าที่คอยให้บริการในการจัดประชุม และเป็นศูนย์กลางในการติดต่อกับส่วนอื่น ๆ ของระบบ ซึ่งสามารถแบ่งส่วนย่อยออกได้ ดังนี้

1) ส่วนหาเวลาว่างและการจัดประชุม (Meeting Management) เป็นหัวใจหลักของระบบซึ่งจะมีอัลกอริทึมเพื่อใช้ในการหาเวลาว่างตามเงื่อนไขที่ผู้ใช้ต้องการและสามารถใช้เวลาว่างนั้น ๆ มาจัดประชุมได้

2) ส่วนติดต่อกับเซอร์วิสของปฏิทินกูเกิล เป็นส่วนที่ไว้สำหรับดึงข้อมูลตารางเวลาของผู้ใช้และผู้ที่เกี่ยวข้องจากปฏิทินกูเกิลมาเพื่อใช้ในการหาเวลาว่างและจัดประชุมต่อไป

3) ส่วนการติดต่อกับระบบฐานข้อมูล เป็นส่วนที่ไว้สำหรับติดต่อกับระบบฐานข้อมูลเพื่อใช้เก็บข้อมูลการจัดประชุม รวมไปถึงสถานะของทรัพยากรต่าง ๆ และการใช้งานทรัพยากรเหล่านั้นเพื่อนำไปใช้ในการจัดประชุม

4) ส่วนการติดต่อกับไคลเอนต์ ซึ่งจะติดต่อกับระบบไคลเอนต์เพื่อใช้ในการรับหรือส่งข้อมูลระหว่างผู้ใช้งานและเซอร์วิส (service)

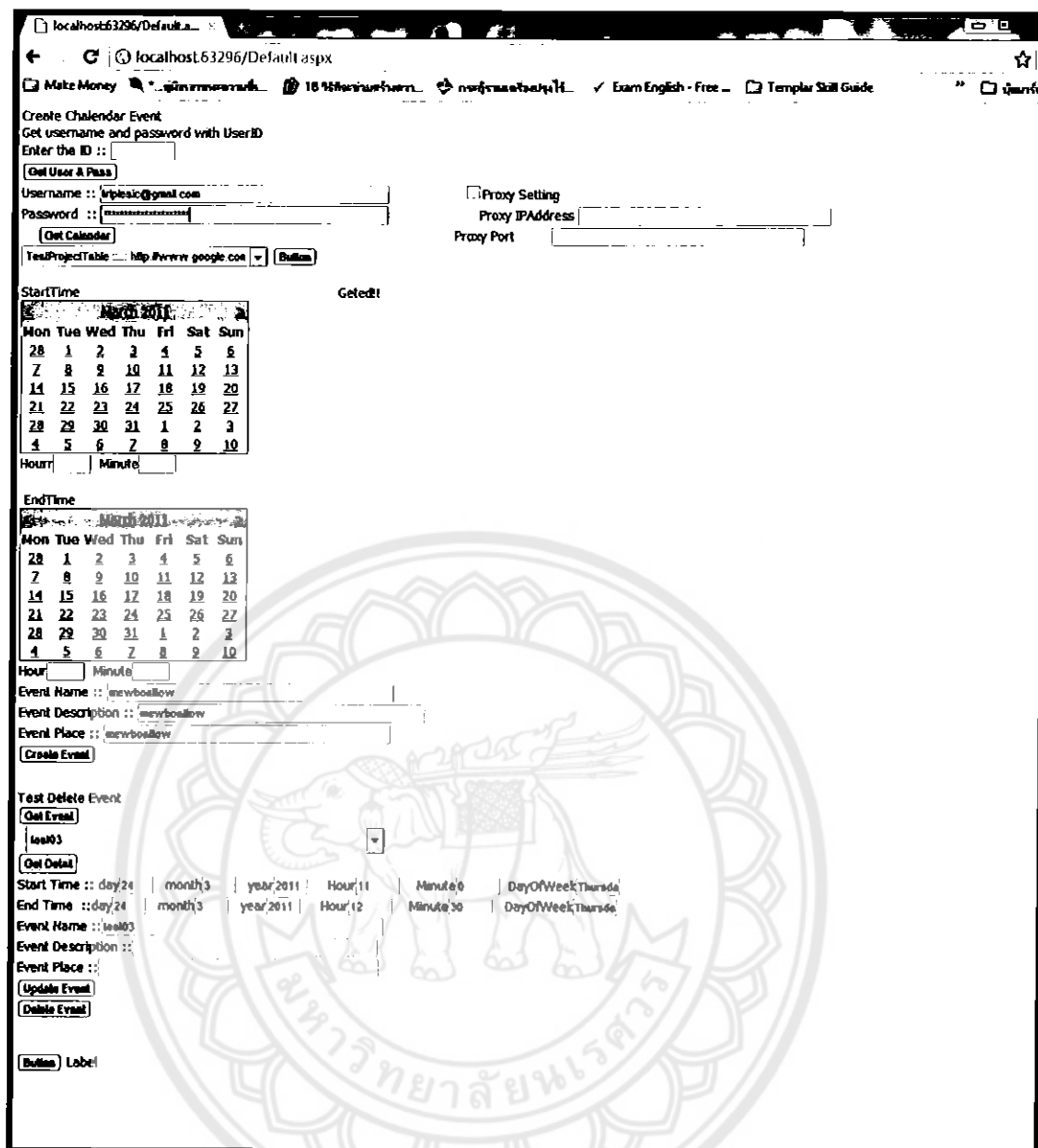
3.5 พัฒนาโปรแกรมต้นแบบ

3.5.1 โปรแกรมติดต่อเซอร์วิสของปฏิทินกูเกิล (Google Calendar)

ได้พัฒนาในส่วนของการดึงรายละเอียดของเหตุการณ์แต่ละเหตุการณ์ ในช่วงเวลาที่กำหนดได้ นอกจากนี้ ยังพัฒนาในส่วนการสร้าง แก๊จ และลบเหตุการณ์ต่าง ๆ ได้ ซึ่งต้องมีการตั้งค่าการใช้งานในปฏิทินกูเกิลตามภาคผนวก เมื่อตั้งค่าตามนั้นเสร็จเรียบร้อยแล้วจะทำให้สามารถใช้งานเซอร์วิสของส่วนปฏิทินของผู้เข้าร่วมแต่ละคนได้

3.5.2 โปรแกรมติดต่อ Database

พัฒนาการติดต่อกับ Database ในส่วนของการดึงข้อมูล รวมไปถึงการสร้าง ลบค่าต่าง ๆ ได้ ซึ่งได้ทำควบคู่ไปกับการพัฒนาโปรแกรมต้นแบบติดต่อเซอร์วิสของปฏิทินกูเกิล (Google Calendar)



รูปที่ 3.12 โปรแกรมต้นแบบการติดต่อเซิร์ฟเวอร์ของปฏิทินกูเกิล (Google Calendar)

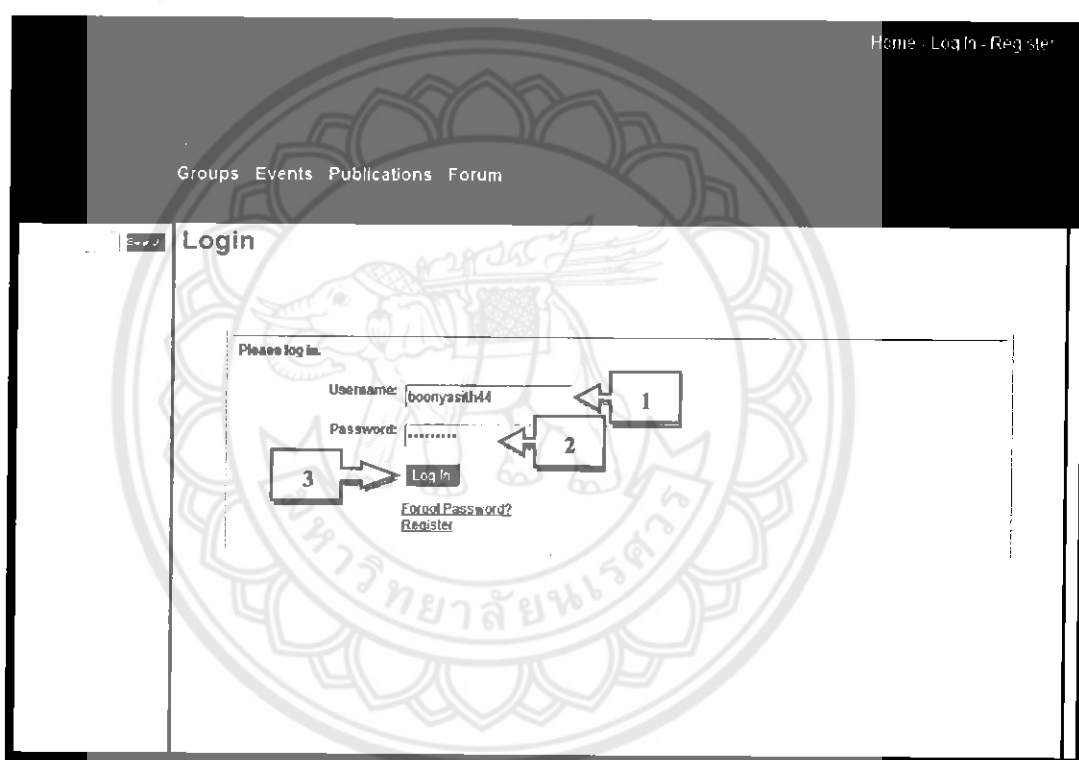
บทที่ 4

การทำงานของระบบ

ในบทนี้จะเป็นการแสดงผลการทำงานของโปรแกรม ซึ่งจะเป็นผลของการทดสอบระบบ และมีการอธิบายการใช้งานอย่างเป็นขั้นตอนซึ่งสามารถใช้เป็นคู่มือการใช้งานของระบบได้อีกด้วย

4.1 ส่วนแสดงผลการเข้าสู่ระบบ

ในการใช้งานเว็บไซต์ผู้ใช้ต้องเข้าสู่ระบบเพื่อแสดงตัวตนผ่านทางหน้าล็อกอิน (Log In)



รูปที่ 4.1 หน้าเว็บเข้าสู่ระบบ (Log In)

หมายเลข 1 ช่องใส่ Username

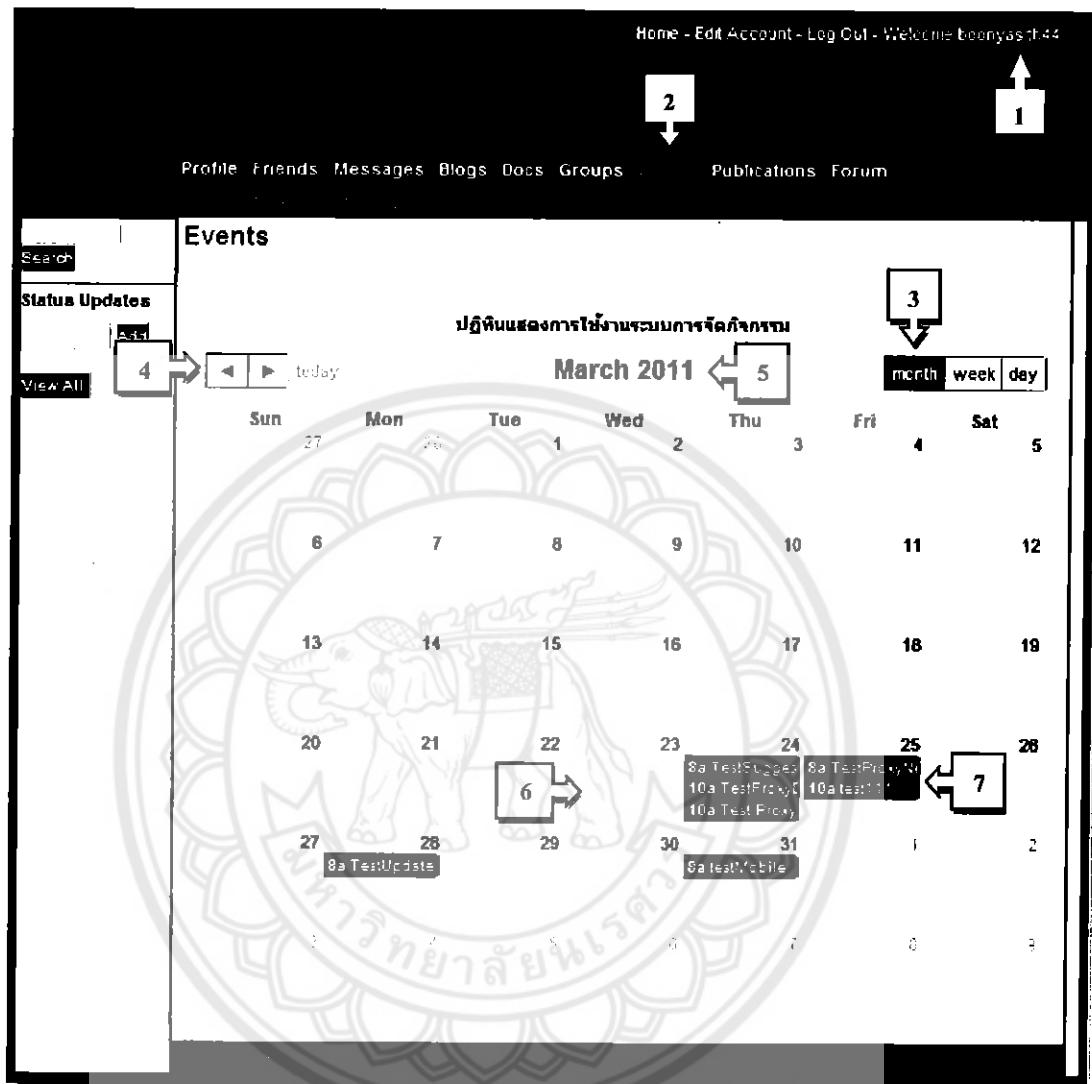
หมายเลข 2 ช่องใส่ Password

หมายเลข 3 ปุ่ม Log In เพื่อเข้าสู่ระบบ

4.2 ส่วนแสดงกิจกรรมรวมทั้งหมด

เมื่อเข้าสู่ระบบสำเร็จจะมีการแสดงชื่อผู้ใช้งาน และแสดงลิงค์เพิ่มเติมเพื่อไปสู่หน้าต่าง ๆ ของระบบ ในที่นี้จะไปที่หน้า Events เพื่อเข้าสู่ส่วนการทำงานหลักของระบบที่ได้จัดทำขึ้น ซึ่ง

หน้าที่ปรากฏขึ้นจะเป็นการส่วนของกิจกรรมรวมทั้งหมดซึ่งได้มีการออกแบบเพื่อให้สามารถเห็นกิจกรรมทั้งหมดที่เกิดขึ้นในรอบสัปดาห์ รอบเดือน หรือทั้งปี ซึ่งขึ้นอยู่กับทางเลือกของผู้ใช้



รูปที่ 4.2 แสดงหน้าต่างเพื่อแสดงกิจกรรมที่มีอยู่ปฏิทินตลอดทั้งเดือน

หมายเลข 1 แสดงชื่อผู้ใช้ที่เข้าสู่ระบบ

หมายเลข 2 ลิงค์เพื่อไปสู่หน้า Event

หมายเลข 3 ปุ่มแสดงปฏิทินในรูปแบบต่าง ๆ ในที่นี้แสดงในรูปแบบทั้งเดือน

หมายเลข 4 ส่วนการปรับเปลี่ยนเดือนที่แสดง

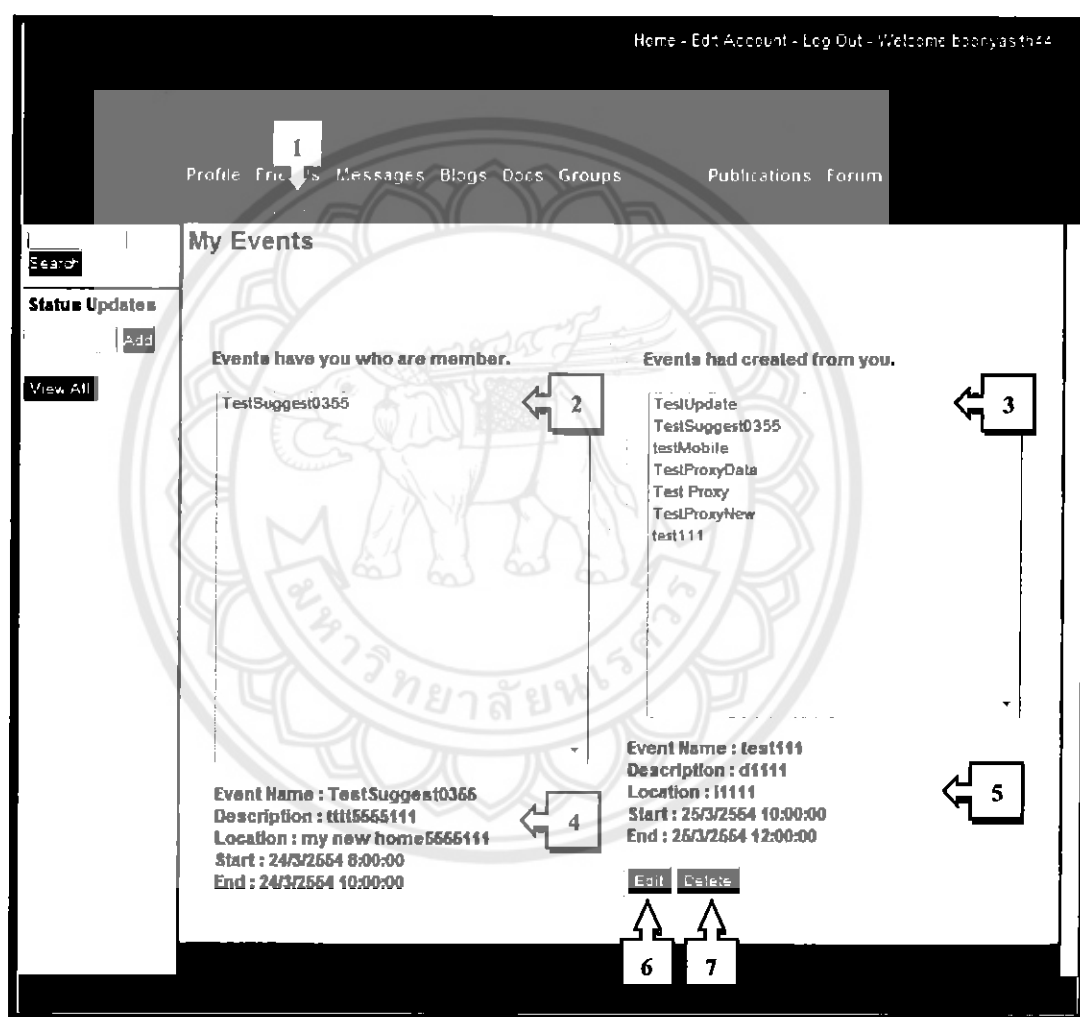
หมายเลข 5 ส่วนแสดงชื่อเดือนและปีของปฏิทินที่กำลังแสดงผล

หมายเลข 6 ส่วนแสดงวันปัจจุบัน

หมายเลข 7 ส่วนแสดงกิจกรรมต่าง ๆ ที่ถูกจัดในระบบ

4.3 ส่วนแสดงกิจกรรมเฉพาะบุคคล (My Event)

ส่วนการแสดงผลกิจกรรมเฉพาะบุคคล (My Event) ได้มีการออกแบบโดยเชื่อมต่อกับฐานข้อมูล (Database) และปฏิทินกูเกิล (Google Calendar) เพื่อให้สามารถทำการแสดงกำหนดการของผู้ใช้คนนั้น และยังสามารถทำการแก้ไขปรับปรุงการประชุมที่ผู้ใช้ได้ทำการจัดขึ้นได้อีกด้วย โดยจะทำการแบ่งเป็น 2 ส่วน คือ ส่วนที่เป็นกิจกรรมที่มีผู้ใช้นี้เข้าร่วมทางด้านซ้าย และส่วนที่เป็นกิจกรรมที่ถูกสร้างโดยผู้ใช้นี้ทางด้านขวา



รูปที่ 4.3.1 แสดงหน้าต่างกิจกรรมเฉพาะบุคคลของผู้ใช้นี้

หมายเลข 1 ส่วนแสดงหน้าย่อหลักทั้งหมดของระบบ

หมายเลข 2 แสดงชื่อกิจกรรมที่มีผู้ใช้นี้เข้าร่วม

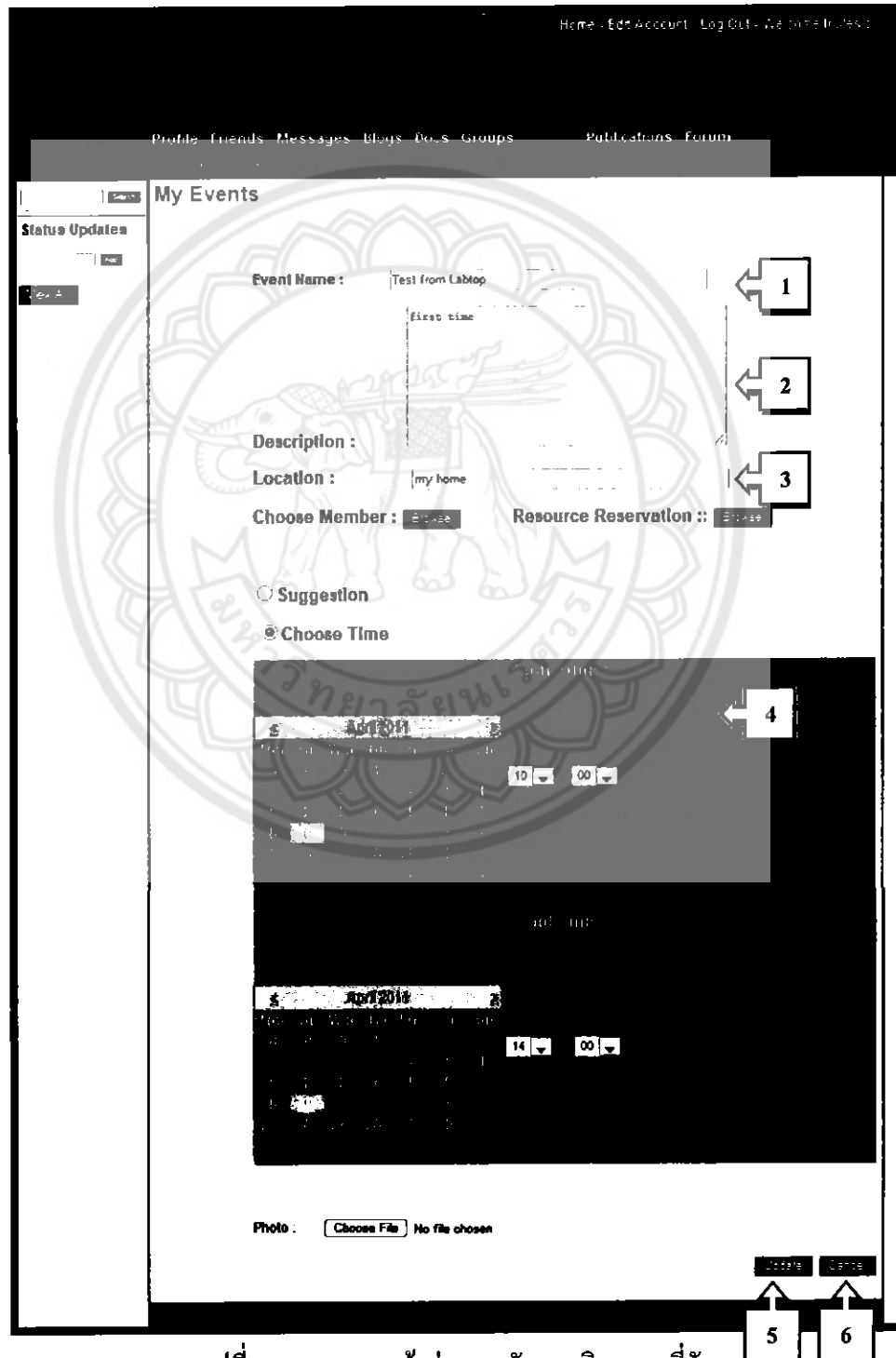
หมายเลข 3 แสดงชื่อกิจกรรมที่มีผู้ใช้นี้เป็นผู้จัด

หมายเลข 4 แสดงรายละเอียดของกิจกรรมที่เลือกในส่วนของกิจกรรมที่เข้าร่วม

หมายเลข 5 แสดงรายละเอียดของกิจกรรมที่เลือกในส่วนของกิจกรรมที่ได้จัด

หมายเลข 6 ปุ่ม Edit เพื่อใช้ในการอัปเดตกิจกรรมเพิ่มเติม (ใช้ได้เฉพาะกิจกรรมที่สร้างขึ้น)
 หมายเลข 7 ปุ่ม Delete เพื่อใช้ในการลบกิจกรรม (ใช้ได้เฉพาะกิจกรรมที่สร้างขึ้น)

นอกจากส่วนที่ไว้แสดงรายละเอียดของกิจกรรมแล้ว หน้าต่างของส่วนแสดงกิจกรรมสามารถทำการอัปเดต และลบในส่วนของกิจกรรมที่ได้สร้างขึ้นซึ่งเมื่อคลิกในปุ่ม Edit ระบบจะทำการเรียกหน้า editEvent ขึ้นมาเพื่อให้ผู้ใช้สามารถแก้ไขรายละเอียดของกิจกรรมที่สร้างขึ้นได้



รูปที่ 4.3.2 แสดงหน้าต่างการอัปเดตกิจกรรมที่จัด

หมายเลข 1 ส่วนการเปลี่ยนแปลงชื่อกิจกรรมที่จัด

หมายเลข 2 ส่วนการเปลี่ยนแปลงรายละเอียดกิจกรรมที่จัด

หมายเลข 3 ส่วนการเปลี่ยนแปลงสถานที่ของกิจกรรมที่จัด

หมายเลข 4 ส่วนการเปลี่ยนแปลงวันและเวลาของกิจกรรมที่จัด

หมายเลข 5 ปุ่ม Update เพื่อยืนยันการเปลี่ยนแปลงแก้ไขของกิจกรรม

หมายเลข 6 ปุ่ม Cancel เพื่อยกเลิกการเปลี่ยนแปลงกิจกรรมและกลับไปหน้าต่างแสดงกิจกรรม
เฉพาะบุคคล

Home - Edit Account - Log Out - Welcome beonyasth44

Profile Friends Messages Blogs Docs Groups Publications Forum

Search

Status Updates Add View All

Events

ปฏิทินแสดงการใช้งานระบบการจัดการกิจกรรม

March 2011

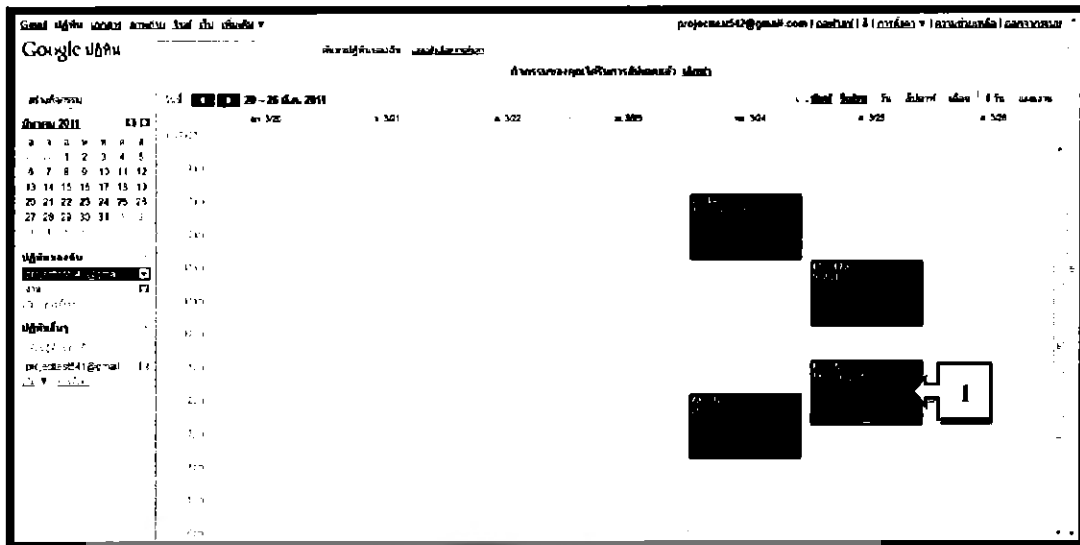
Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

8a TestSugges 8a TestProje, W
10a TestProje, U 10a test(11
10a Test Proje 1p Test Proje
8a TestUpdate 8a test/Mobile

1

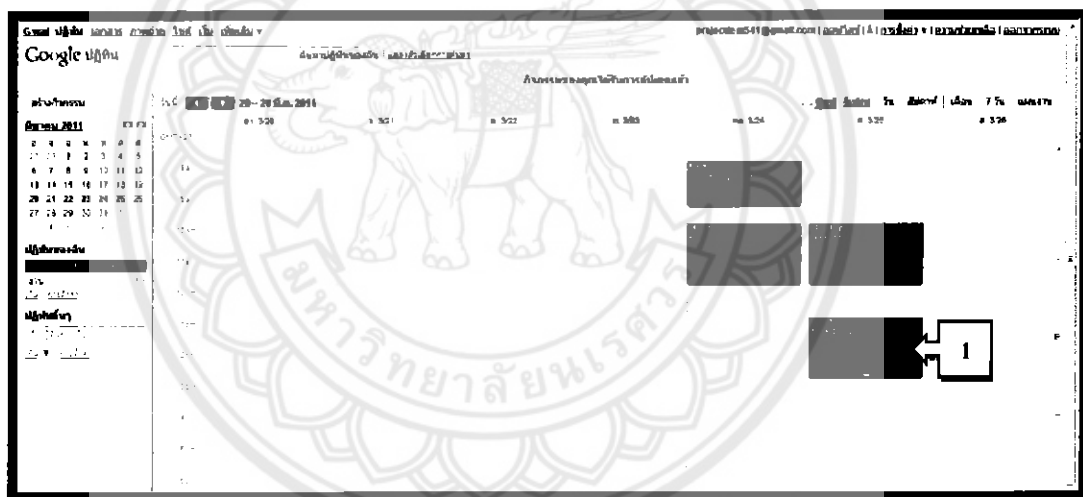
รูปที่ 4.3.3 แสดงการเปลี่ยนแปลงของกิจกรรมหลังการอัปเดตในหน้า Events

หมายเลข 1 กิจกรรมที่ถูกอัปเดต จะเห็นว่าเปลี่ยนจากเวลา 8 นาฬิกาเป็น 13 นาฬิกา



รูปที่ 4.3.4 แสดงการเปลี่ยนแปลงของกิจกรรมหลังการอัปเดตในปฏิทินกิลิซของผู้เข้าร่วมคนที่ 1

หมายเลข 1 กิจกรรมที่ถูกอัปเดต จะเห็นว่าเปลี่ยนจากเวลา 8 นาทีเป็น 13 นาที



รูปที่ 4.3.5 แสดงการเปลี่ยนแปลงของกิจกรรมหลังการอัปเดตในปฏิทินกิลิซของผู้เข้าร่วมคนที่ 2

หมายเลข 1 กิจกรรมที่ถูกอัปเดต จะเห็นว่าเปลี่ยนจากเวลา 8 นาทีเป็น 13 นาที

จากการแสดงรายละเอียดของกิจกรรมในหน้าต่างแสดงกิจกรรมเฉพาะบุคคล เมื่อคลิกที่ปุ่ม Delete จะเป็นการลบกิจกรรมที่จัดขึ้นออกจากระบบและปฏิทินกิลิซของผู้เข้าร่วมทุกคน

Home - Edit Account - Log Out - Welcome boonyas 11:44

Profile Friends Messages Blogs Docs Groups Publications Forum

My Events

Search

Status Updates [Add](#)

[View All](#)

Events have you who are member.

TestSuggest0355

Events had created from you.

TestUpdate
 TestSuggest0355
 testMobile
 TestProxyData
 Test Proxy
 TestProxyNew
 test111
 Test Projed

Event Name : Test Project
Description : Test Project Change
Location : Narasuan University Change
Start : 25/3/2554 13:00:00
End : 25/3/2554 15:00:00

[Edit](#) [Delete](#)

รูปที่ 4.3.6 แสดงรายละเอียดของกิจกรรมที่ต้องการลบ

หมายเลข 1 ส่วนแสดงรายละเอียดของกิจกรรมที่เลือก

หมายเลข 2 ปุ่ม Delete ซึ่งไว้ใช้ในการลบกิจกรรมที่เลือก

Status Updates [Add](#)

[View All](#)

Events have you who are member.

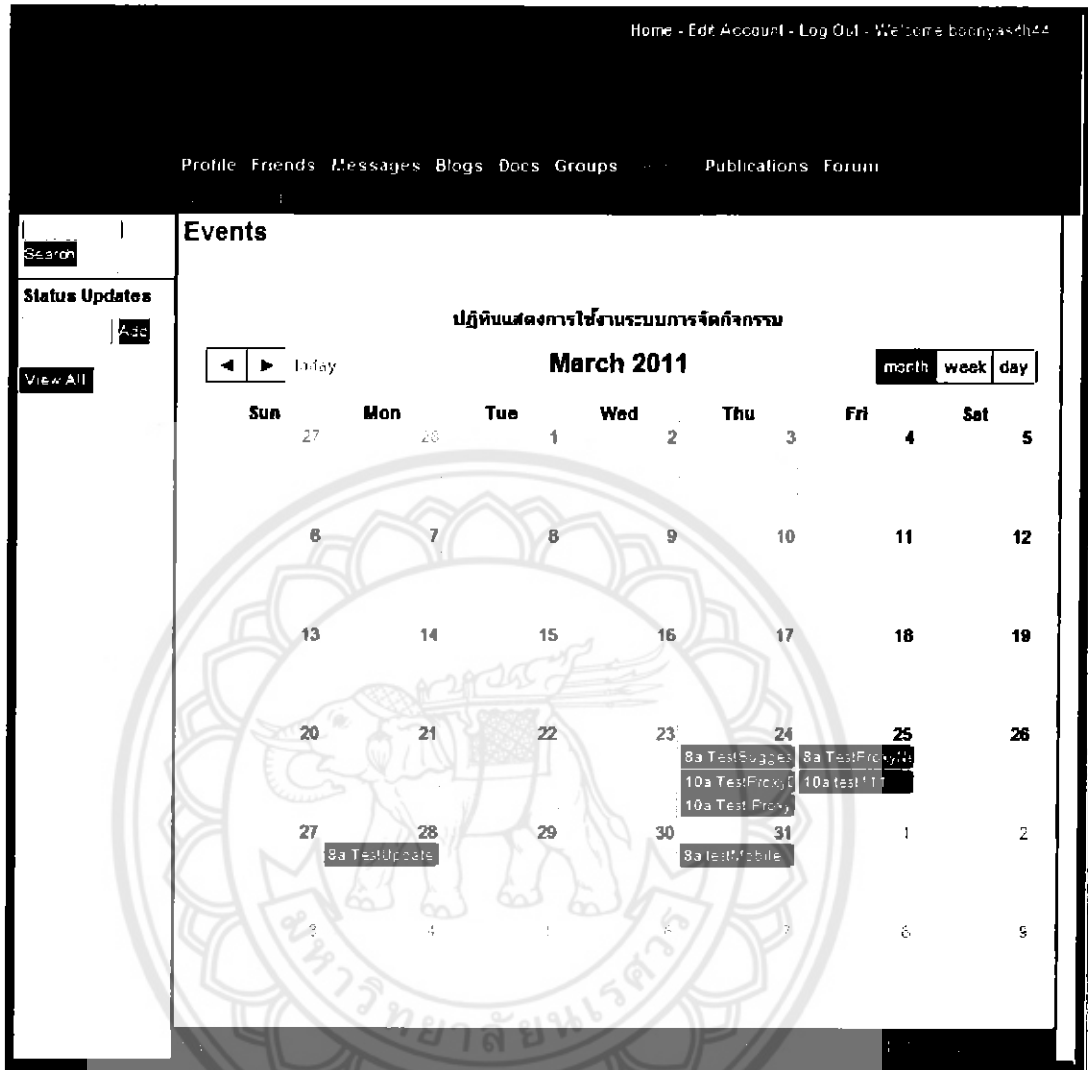
TestSuggest0355

Events had created from you.

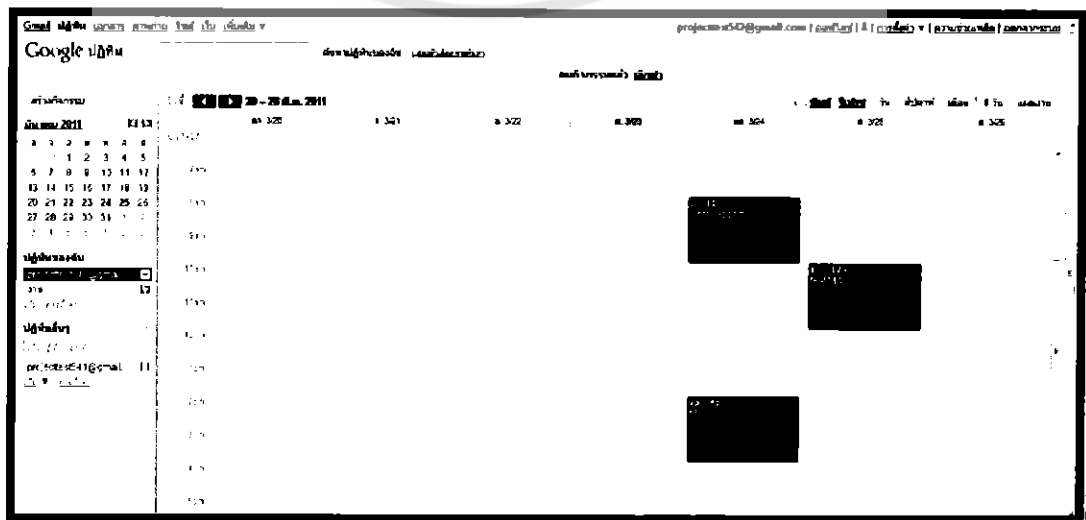
TestUpdate
 TestSuggest0355
 testMobile
 TestProxyData
 Test Proxy
 TestProxyNew
 test111

รูปที่ 4.3.7 แสดงการเปลี่ยนแปลงหลังจากการลบกิจกรรมที่เลือก

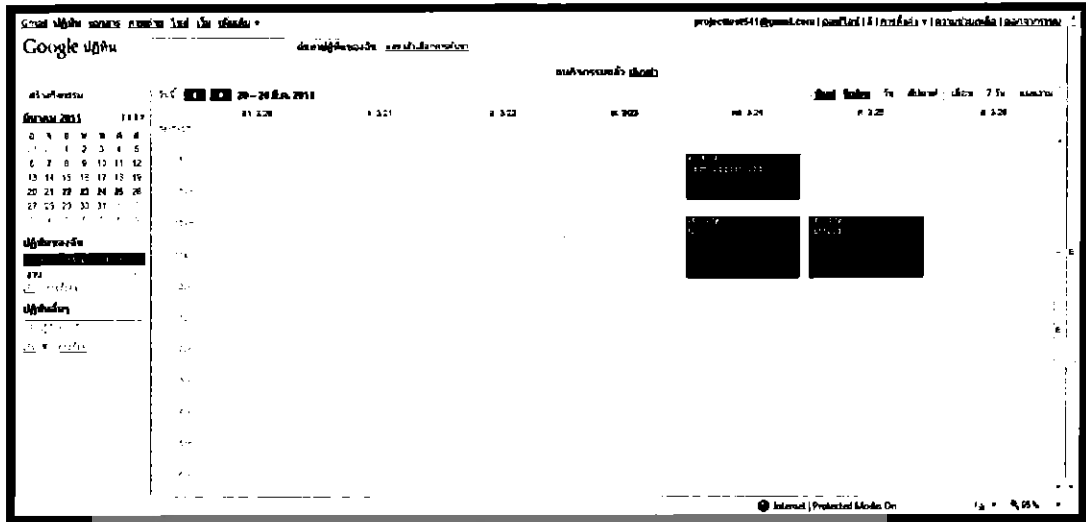
หมายเลข 1 ส่วนแสดงชื่อของกิจกรรมที่เหลือนอกจากการลบ



รูปที่ 4.3.8 แสดงกิจกรรมที่เหลือนหลังการลบในหน้า Events



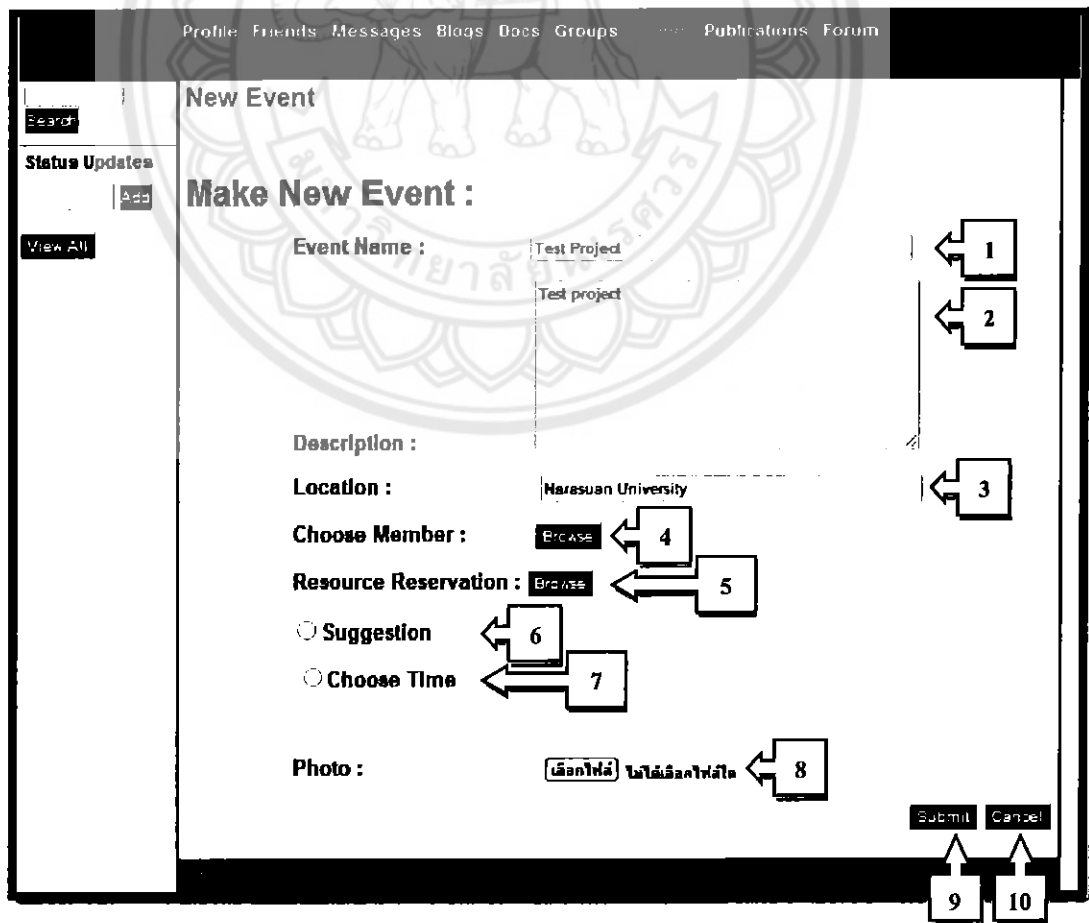
รูปที่ 4.3.9 แสดงกิจกรรมที่เหลือนหลังการลบในปฏิทินกูเกิลของผู้เข้าร่วมคนที่ 1



รูปที่ 4.3.10 แสดงกิจกรรมที่เหลือหลังการลบในปฏิทินวันเกิดของผู้เข้าร่วมคนที่ 2

4.4 ส่วนแสดงแบบฟอร์มในการจัดกิจกรรม

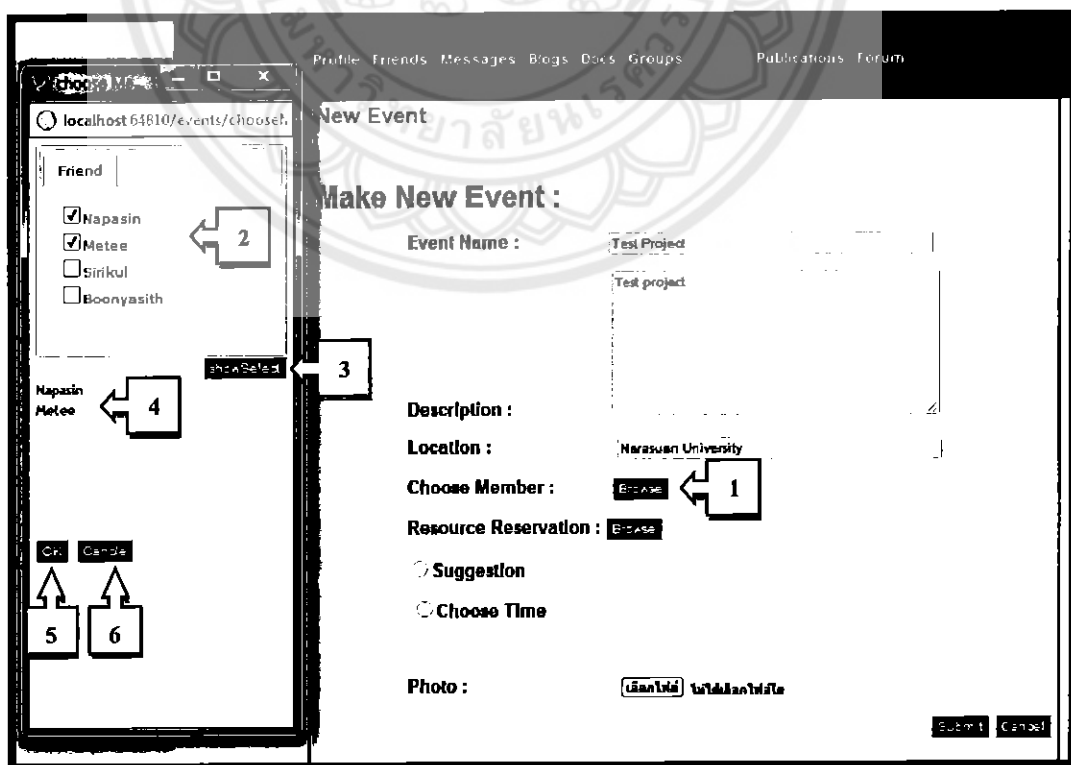
ส่วนแสดงแบบฟอร์มในการจัดการประชุมจะออกแบบให้มีการใส่รายละเอียดต่าง ๆ ที่มีความจำเป็นในการจัดการประชุม



รูปที่ 4.4.1 แสดงหน้าต่างแบบฟอร์มการสร้างกิจกรรม

- หมายเลข 1 ช่องไว้สำหรับใส่ชื่อของกิจกรรมที่จะจัด
- หมายเลข 2 ช่องไว้สำหรับใส่รายละเอียดของกิจกรรมที่จะจัด
- หมายเลข 3 ช่องไว้สำหรับใส่สถานที่ในการจัดกิจกรรม
- หมายเลข 4 ปุ่ม Choose Member เพื่อใช้ในการเลือกผู้ที่จะเข้าร่วมกิจกรรม
- หมายเลข 5 ปุ่ม Resource Reservation เพื่อใช้ในการเลือกทรัพยากรที่ใช้ในการจัดกิจกรรม
- หมายเลข 6 ส่วน Suggestion เพื่อใช้ระบบเลือกเวลาที่ว่างของแต่ละคนที่เข้าร่วมกิจกรรมตามช่วงเวลาที่กำหนด
- หมายเลข 7 ส่วน Choose Time เพื่อให้ผู้ใช้กำหนดช่วงเวลาในการจัดกิจกรรมด้วยตัวเองโดยไม่สนใจเวลาว่างของผู้เข้าร่วมกิจกรรม
- หมายเลข 8 ส่วนการเพิ่มรูป เช่น แผนที่ หรือรูปที่เกี่ยวข้องกับการจัดกิจกรรม อำนวยความสะดวกให้กับผู้เข้าร่วมกิจกรรม อาจใส่หรือไม่ก็ได้
- หมายเลข 9 ปุ่ม Submit เพื่อใช้ยืนยันการจัดกิจกรรม ซึ่งจะทำการส่งข้อมูลต่าง ๆ ที่กรอกเข้าสู่ระบบ
- หมายเลข 10 ปุ่ม Cancel เพื่อใช้ล้างค่าทั้งหมดที่กรอกลงไป

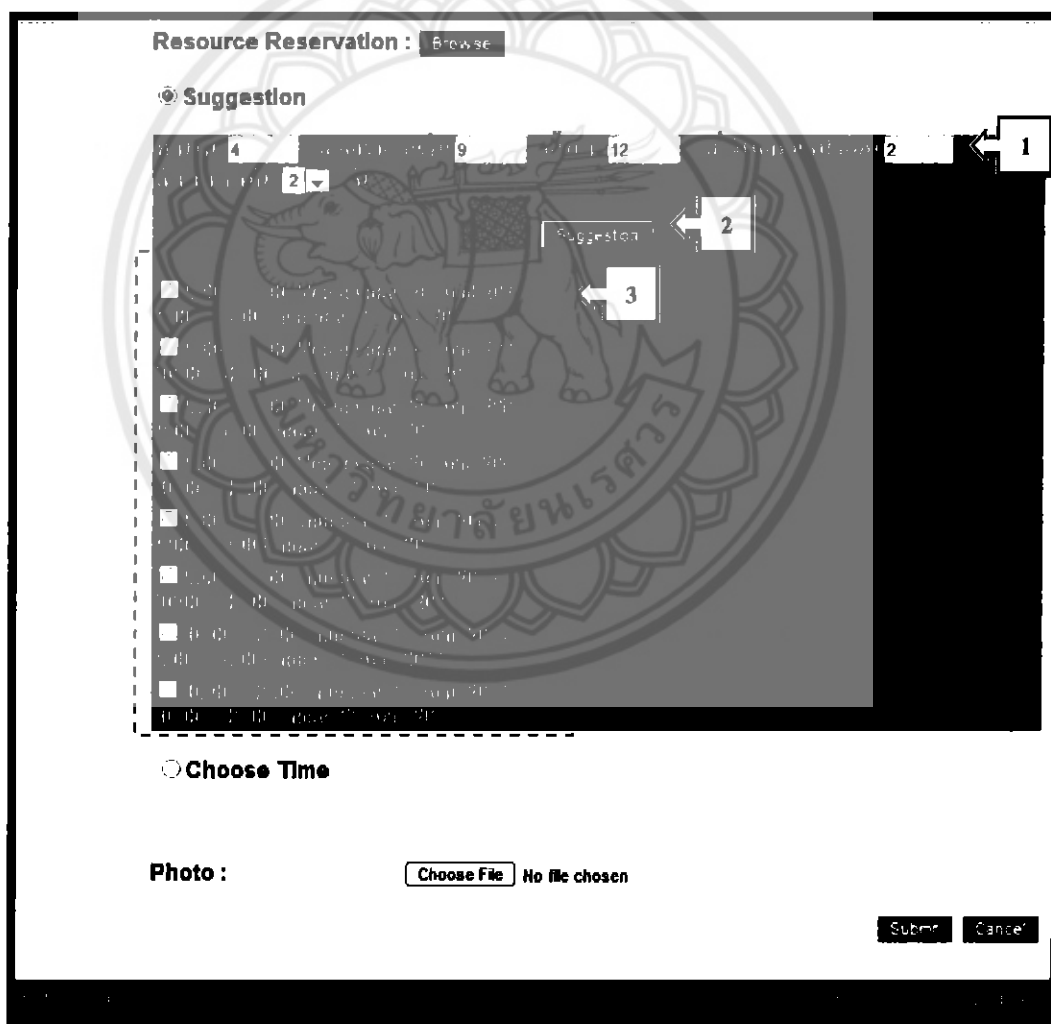
ในการเลือกผู้ที่จะเข้าร่วมกิจกรรมนั้น ได้มีการออกแบบหน้า Pop up เพื่อไว้สำหรับเลือกบุคคลเพื่อเข้าร่วมในกิจกรรมที่ได้จัดขึ้น



รูปที่ 4.4.2 แสดงหน้าต่างการเลือกผู้เข้าร่วมกิจกรรม

- หมายเลข 1 ปุ่ม Choose Member ไว้สำหรับเปิด Pop up ส่วนของการเลือกผู้เข้าร่วมกิจกรรม
 หมายเลข 2 ส่วนแสดงชื่อผู้ใช้ของผู้ที่เป็นสมาชิกของระบบ
 หมายเลข 3 ปุ่ม showSelect มีไว้เพื่อใช้ในการแสดงรายชื่อผู้ที่ถูกเลือกไว้เพื่อจะเข้าร่วมกิจกรรม
 หมายเลข 4 ส่วนแสดงผู้ที่ถูกเลือกให้เข้าร่วมกิจกรรม
 หมายเลข 5 ปุ่ม OK เพื่อใช้ในยืนยันผู้เข้าร่วมกิจกรรม
 หมายเลข 6 ปุ่ม Cancel เพื่อปิดหน้าการเลือกสมาชิกซึ่งจะไม่มีกรเลือกผู้เข้าร่วมกิจกรรม

เมื่อทำการเลือกผู้ที่จะเข้าร่วมในกิจกรรมที่กำลังจะจัดแล้ว ระบบได้มีการออกแบบให้ผู้ใช้สามารถเลือกเวลาว่างของผู้เข้าร่วมกิจกรรม เพื่อไว้ให้ผู้จัดกิจกรรมทำการกำหนดเงื่อนไขของช่วงเวลาในการจัดกิจกรรม ซึ่งจะช่วยให้ผู้จัดสามารถจัดกิจกรรมได้ง่ายขึ้น



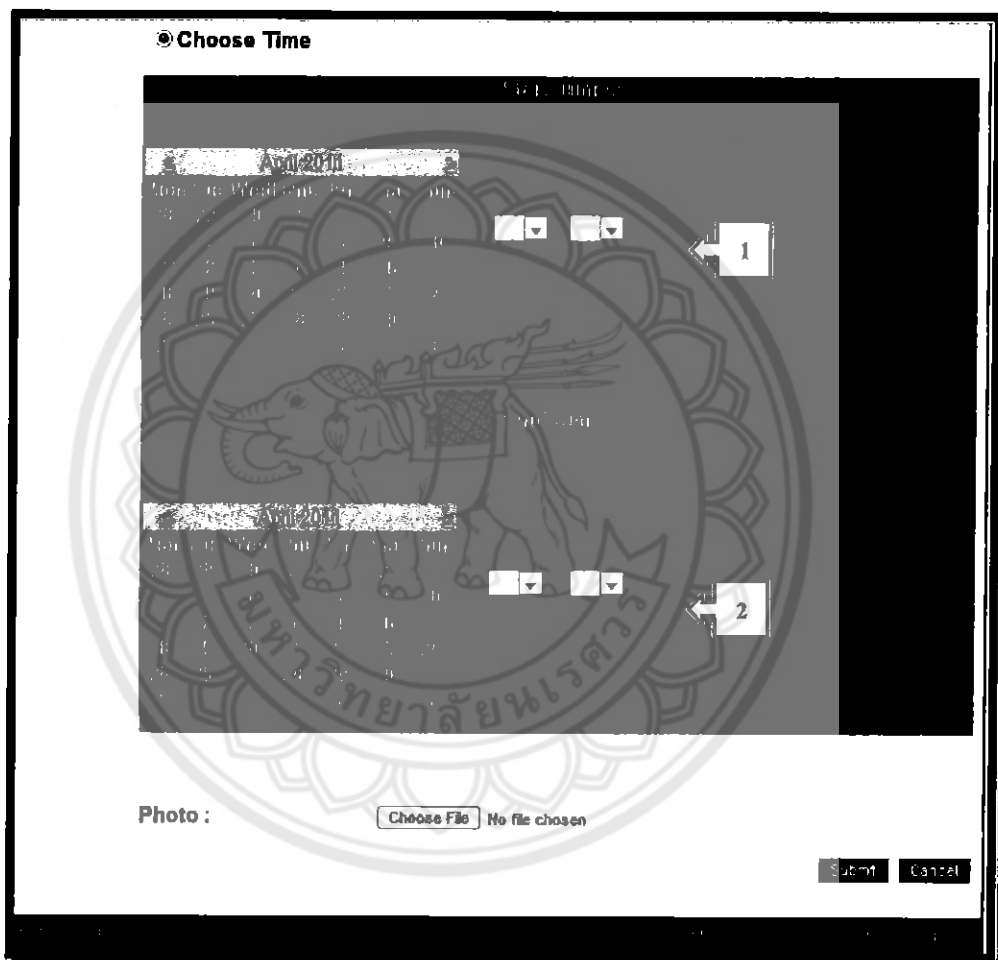
รูปที่ 4.4.3 แสดงส่วนของการเลือกเวลาจากการคำนวณของระบบ

หมายเลข 1 ส่วนเงื่อนไขที่ใช้ในการคำนวณหาเวลาว่าง

หมายเลข 2 ปุ่ม Suggestion มีไว้เพื่อให้ระบบทำการคำนวณหาเวลาที่ว่างตรงกันของแต่ละคนที่เข้าร่วมกิจกรรม

หมายเลข 3 ส่วนแสดงเวลาที่ทุกคนที่เข้าร่วมกิจกรรมว่างพร้อมกัน

หากผู้จัดกิจกรรมต้องการกำหนดเวลาในการจัดการประชุมเอง ผู้ใช้สามารถระบุเวลาที่ต้องการได้ โดยการเลือกที่ choose time จากนั้น โปรแกรมจะแสดงส่วนของการกำหนดวันและเวลาในการจัดกิจกรรมขึ้นมา ผู้ใช้สามารถเลือกวันและเวลาที่ต้องการได้เองทันที

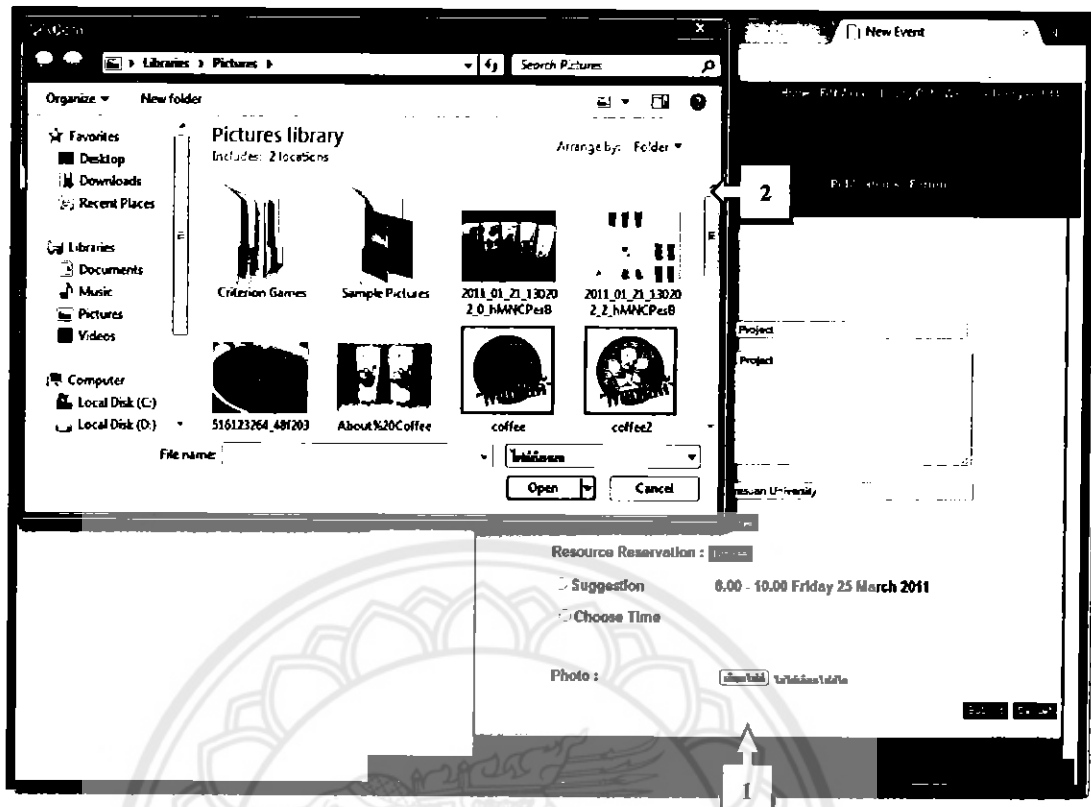


รูปที่ 4.4.4 แสดงส่วนของการเลือกเวลาโดยผู้จัดกิจกรรมเป็นผู้กำหนดเอง

หมายเลข 1 ส่วนการเลือกวันและเวลาเริ่มต้นในการจัดกิจกรรม

หมายเลข 2 ส่วนการเลือกวันและเวลาสิ้นสุดการจัดกิจกรรม

การจัดกิจกรรมบางครั้งต้องมีรูปภาพประกอบการจัดกิจกรรมหรือประกอบการเดินทาง ดังนั้นจึงมีการทำในส่วนของการเพิ่มรูปภาพเข้ามาเพื่อใช้เป็นรายละเอียดในทางการเข้าร่วมกิจกรรมหรือการเดินทางของผู้เข้าร่วมกิจกรรม

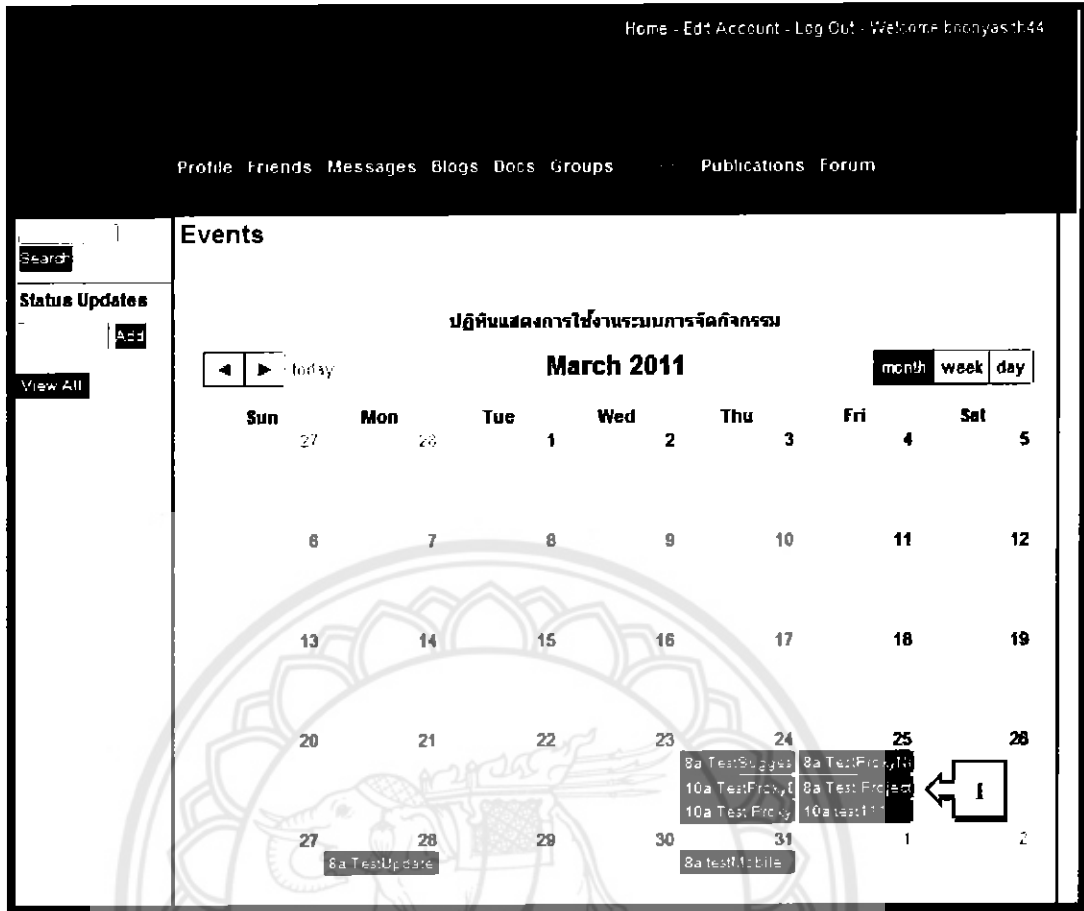


รูปที่ 4.4.5 แสดงส่วนของการเพิ่มรูปภาพเพื่อใช้ประกอบรายละเอียดในการจัดกิจกรรม

หมายเลข 1 ปุ่ม เลือกรูปภาพ ใช้สำหรับการเปิดส่วนการเลือกรูปภาพ

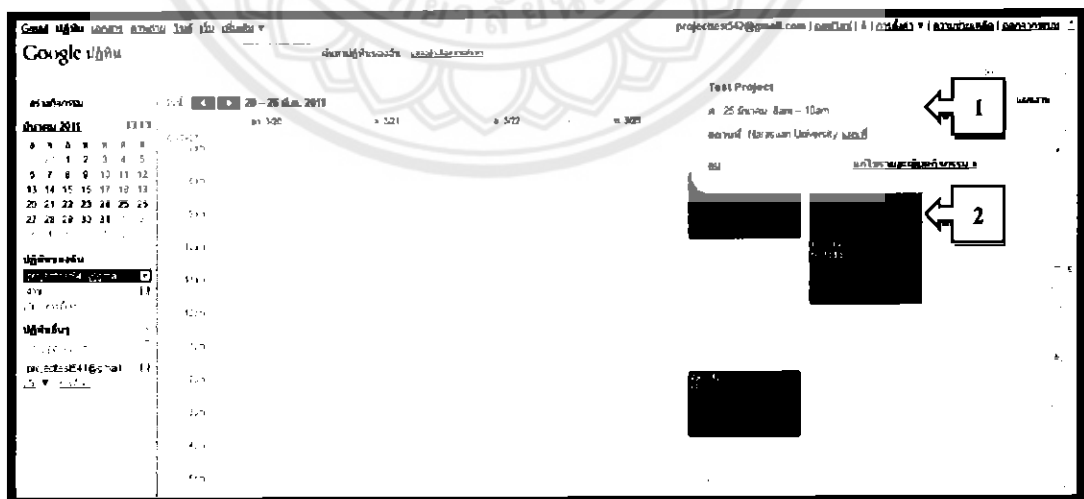
หมายเลข 2 ส่วนของหน้าต่างที่ใช้ในการเลือกรูปภาพประกอบการจัดกิจกรรม

เมื่อกรอกรายละเอียดทั้งหมด และคลิกที่ปุ่ม Submit แล้ว ระบบจะจัดการนำข้อมูลต่าง ๆ ที่กรอกลงไปเข้าสู่ฐานข้อมูล (Database) และปฏิทินกูเกิล (Google Calendar) ของผู้เข้าร่วมทุกคน



รูปที่ 4.4.6 แสดงกิจกรรมที่ถูกจัดขึ้น โดยใช้ระบบการจัดกิจกรรมในหน้า Events

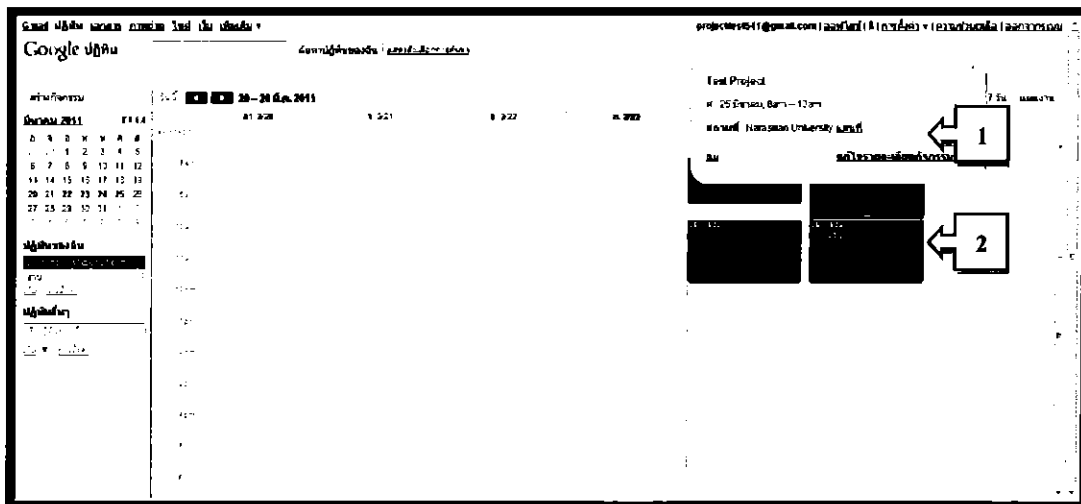
หมายเลข 1 กิจกรรมที่ถูกจัดขึ้น



รูปที่ 4.4.7 แสดงกิจกรรมที่ถูกจัดขึ้นในปฏิทินปฏิทินของผู้เข้าร่วมคนที่ 1

หมายเลข 1 รายละเอียดของกิจกรรมที่ถูกจัดขึ้นโดยใช้ระบบการจัดกิจกรรม

หมายเลข 2 กิจกรรมที่ถูกจัดขึ้น โดยใช้ระบบการจัดกิจกรรม



รูปที่ 4.4.8 แสดงกิจกรรมที่ถูกจัดขึ้นในปฏิทินปฏิทินของผู้เข้าร่วมคนที่ 2

หมายเลข 1 รายละเอียดของกิจกรรมที่ถูกจัดขึ้นโดยใช้ระบบการจัดกิจกรรม

หมายเลข 2 กิจกรรมที่ถูกจัดขึ้นโดยใช้ระบบการจัดกิจกรรม



บทที่ 5

ผลการทดลอง

5.1 การประเมินความพึงพอใจของผู้ใช้งานระบบช่วยจัดการนัดหมายเพื่อการจัดประชุม

ระบบช่วยจัดการนัดหมายเพื่อการจัดประชุม เป็นระบบที่สามารถให้ผู้ใช้สามารถใช้งานผ่านทางเว็บแอปพลิเคชัน ซึ่งเป็นระบบที่ออกแบบให้ง่ายต่อการใช้งาน จึงได้มีการสร้างแบบสอบถามเพื่อประเมินผลความพึงพอใจของผู้ใช้เมื่อได้ทดลองใช้งานระบบช่วยจัดการนัดหมายเพื่อการจัดประชุมว่ามีผลเป็นอย่างไร และมีการทำสอบถึงความเร็วในการเรียกใช้เว็บเอพีไอ

ตัวอย่าง

แบบสอบถามระดับความพึงพอใจของผู้ใช้ระบบช่วยจัดการนัดหมายเพื่อการจัดประชุม

คำชี้แจง

ทำเครื่องหมาย ✓ ลงในช่องในแบบสอบถามที่ตรงกับระดับความคิดเห็นของท่านมากที่สุด โดยตัวเลขของระดับความพึงพอใจแต่ละด้านมีความหมายดังนี้

- 5 หมายถึง ความเหมาะสม/ความพึงพอใจในระดับ มากที่สุด
- 4 หมายถึง ความเหมาะสม/ความพึงพอใจในระดับ มาก
- 3 หมายถึง ความเหมาะสม/ความพึงพอใจในระดับ ปานกลาง
- 2 หมายถึง ความเหมาะสม/ความพึงพอใจในระดับ น้อย
- 1 หมายถึง ความเหมาะสม/ความพึงพอใจในระดับ น้อยที่สุด

เพศ

ชาย

หญิง

รายการประเมิน	ระดับความพึงพอใจ				
	5	4	3	2	1
ด้านการใช้งานและความสามารถของระบบ					
1. ระบบสามารถแสดงผลข้อมูลได้ถูกต้อง					
2. ระบบมีความง่ายต่อการใช้งาน					
3. การออกแบบในส่วนติดต่อกับผู้ใช้สามารถใช้งาน ได้สะดวก และเข้าใจง่าย					
4. ความเหมาะสมในการวางตำแหน่งของส่วนประกอบต่างๆ ในการแสดงผล					

รายการประเมิน	ระดับความพึงพอใจ				
	5	4	3	2	1
ด้านผลลัพธ์ที่ได้จากระบบ					
5. ความถูกต้องของการเลือกกิจกรรมระหว่างระบบกับปฏิทินของผู้เข้าร่วมกิจกรรม					
6. ความรวดเร็วในการประมวลผลของระบบ					
7. ความน่าเชื่อถือได้ของระบบ					
8. ผลลัพธ์ที่ได้จากระบบสามารถนำไปใช้งานได้จริง					

ข้อเสนอแนะอื่น ๆ

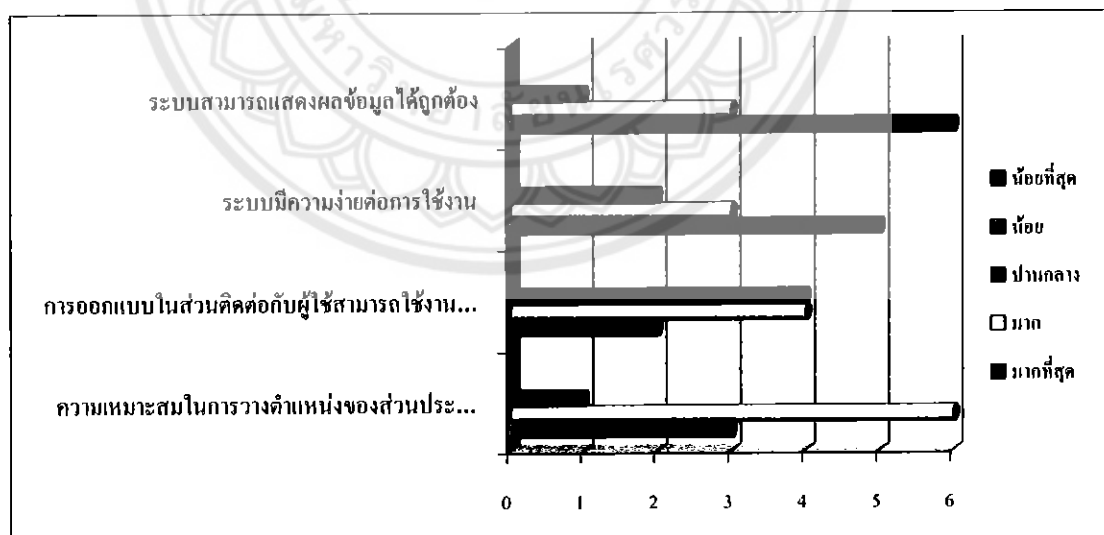
.....

.....

.....

จากการสำรวจระดับความพึงพอใจของผู้ใช้ระบบช่วยจัดการนัดหมายเพื่อการจัดประชุม จำนวน 10 คน เป็นเพศชาย 5 คน และเพศหญิง 5 คน พบว่าผู้ใช้มีความพึงพอใจ ดังนี้

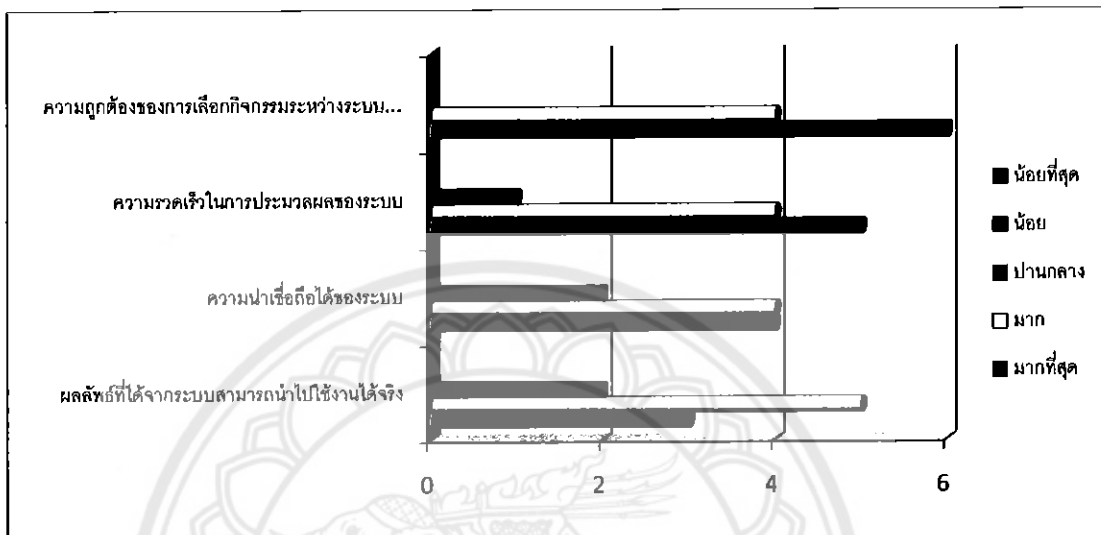
ด้านการใช้งานและความสามารถของระบบ



รูปที่ 5.1 แสดงกราฟสรุปผลการสำรวจความพึงพอใจด้านการใช้งานและความสามารถของระบบ

จากรูปที่ 5.1 จะพบว่าผู้ใช้งานส่วนใหญ่มีความพอใจในระดับมากถึงมากที่สุด เว้นแต่ในส่วนของการออกแบบในส่วนติดต่อกับผู้ใช้งานได้สะดวกและเข้าใจง่ายซึ่งอยู่ในเกณฑ์ปานกลางจนถึงมากเป็นส่วนใหญ่ แต่ยังถือว่าอยู่ในระดับที่รับได้

ด้านผลลัพธ์ที่ได้จากระบบ



รูปที่ 5.2 แสดงกราฟสรุปผลการสำรวจความพึงพอใจด้านผลลัพธ์ที่ได้จากระบบ

จากรูปที่ 5.2 จะพบว่าผู้ใช้งานส่วนใหญ่มีความพอใจในระดับมากถึงมากที่สุดในด้านผลลัพธ์ที่ได้จากระบบ

5.2 การวิเคราะห์เวลาที่ใช้ในการประมวลผล (time complexity)

5.2.1 การวิเคราะห์ในส่วนของการสร้างเหตุการณ์

- ส่วนของการสร้างเหตุการณ์นั้นมีการใช้เวลาในการคิดหาเวลาที่สามารถจัดได้ ใช้เวลา $\max(O(n^{x+1}), O(n^3))$
- ส่วนของการสร้างเหตุการณ์ซึ่งรวมการสร้างเหตุการณ์ลงปฏิทินของกุเกิลและข้อมูลในฐานข้อมูล ใช้เวลา $O(n^3)$

เพราะฉะนั้น ในส่วนของการสร้างเหตุการณ์ จึงใช้เวลา $\max(O(n^{x+1}), O(n^3))$

5.2.2 การวิเคราะห์ในส่วนของการแก้ไข และลบเหตุการณ์ แบ่งเป็น

- เวลาที่ใช้ในการโหลดข้อมูลเพื่อแสดงผลทางหน้าเว็บ ใช้เวลา $O(n^3)$
- เวลาที่ใช้ในการแก้ไข แบบมีการคิดหาเวลาที่เหมาะสมใหม่ ใช้เวลา $\max(O(n^{x+1}), O(n^3))$
- เวลาที่ใช้ในการแก้ไข แบบเลือกเวลาเอง ใช้เวลา $O(n^2)$

- เวลาที่ใช้ในการลบเหตุการณ์ซึ่งรวมการสร้างเหตุการณ์ในปฏิทินภูเกิดและข้อมูลในฐานข้อมูล ใช้เวลา $O(n^2)$

เพราะฉะนั้น เวลาในการแก้ไขและลบข้อมูล คิดเป็น $\max(O(n^{x+1}), O(n^3))$

5.3 ผลการดำเนินการ

ในโครงการนี้เป็นโครงการที่ศึกษาและพัฒนาาระบบช่วยจัดการนัดหมายเพื่อการจัดประชุม ซึ่งผลสรุปที่ได้สามารถอธิบายได้ดังตาราง

ตารางที่ 5.1 ตารางสรุปผลการดำเนินการ

ผลที่ได้	ทำได้	ทำไม่ได้
1. ระบบสามารถหาเวลาที่เหมาะสมในการจัดประชุมได้	✓	
2. ระบบสามารถจัดการประชุมได้	✓	
3. ระบบสามารถดำเนินการเพิ่ม ลบ และอัปเดตการประชุมที่เป็นคนจัดขึ้นได้	✓	
4. ระบบสามารถทำและแสดงผลไปที่ปฏิทินภูเกิด (Google Calendar) ของผู้เข้าร่วมทั้งหมด	✓	

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 บทสรุปการดำเนินการ

ในโครงการนี้เป็นโครงการที่ศึกษาและพัฒนาระบบช่วยจัดการนัดหมายเพื่อการจัดประชุม ซึ่งระบบที่ได้มีความสามารถในการทำงานดังนี้

1. ระบบสามารถหาเวลาที่เหมาะสมในการจัดประชุมได้
2. ระบบสามารถจัดการประชุมได้
3. ระบบสามารถดำเนินการเพิ่ม ลบ และอัปเดตการประชุมที่เป็นคนจัดขึ้นได้
4. ระบบสามารถทำและแสดงผลไปที่ปฏิทินกูเกิล (Google Calendar) ของผู้เข้าร่วม

ทั้งหมด

6.2 ปัญหาที่พบในการดำเนินการและแนวทางในการแก้ไข

1. ติดปัญหาในการทำงานผ่านพร็อกซี (Proxy) จึงจะต้องใช้การทำงานจากที่อื่น
2. เนื่องจากอัลกอริทึมหลักในการจัดการเวลาในระบบมีความยุ่งยากซับซ้อน จึงจำเป็นต้องทำการศึกษาให้เข้าใจก่อนแล้วค่อยดำเนินการพัฒนาโปรแกรม
3. ผู้พัฒนาไม่เคยเขียน ASP.net มาก่อนจึงต้องใช้เวลาในการศึกษาพอสมควร

6.3 ข้อเสนอแนะโครงการ

1. การทดสอบโปรแกรมควรหาทุกความเป็นไปได้ที่อาจเกิดขึ้น เมื่อพบปัญหาแล้วไม่ควรทิ้งไว้ให้รับหาวิธีแก้ไขในทันที
2. ในการพัฒนาต่อยอดโปรแกรมควรเริ่มจากการพัฒนาในส่วนของ Conflict Resolution เพื่อให้หาคำตอบในกรณีที่ผู้เข้าร่วมการประชุมไม่มีเวลาว่างตรงกัน
3. ในการพัฒนาโปรแกรมควรเพิ่มความสามารถในการดำเนินการบางส่วนให้เร็วขึ้น เช่น การอัปเดตและการลบการประชุมที่สร้างขึ้น ซึ่งในขณะนี้ระบบสามารถดำเนินการได้แต่ยังมีความหน่วงตามจำนวนผู้เข้าร่วมอยู่บ้าง หากจะพัฒนาต่อจึงควรพัฒนาส่วนนี้ด้วย

ภาคผนวก

การตั้งค่า Google Calendar

Google Calendar คือ บริการปฏิทินแบบออนไลน์ของ Google ซึ่งสามารถเก็บข้อมูลเหตุการณ์ต่าง ๆ รวมไปถึงที่เดียวกันได้ ไม่ว่าจะเป็นการสร้างกำหนดการนัดหมายและกำหนดเวลาเหตุการณ์ต่าง ๆ สามารถส่งข้อความเชิญ สามารถใช้ปฏิทินร่วมกับเพื่อนร่วมงาน และค้นหาเหตุการณ์ต่าง ๆ ได้ ซึ่ง Google Calendar มีข้อดีกว่าโปรแกรมที่อยู่ในคอมพิวเตอร์ดังนี้

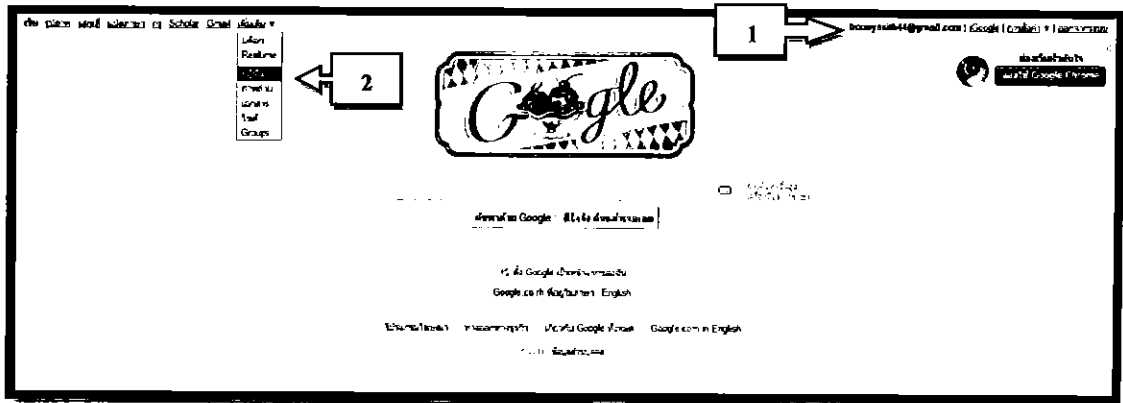
1. Google Calendar เป็นบริการออนไลน์และให้บริการฟรี ซึ่งต่างกับโปรแกรมในคอมพิวเตอร์ที่ต้องเสียค่าใช้จ่ายเพื่อซื้อโปรแกรมมาใช้
2. โปรแกรมในคอมพิวเตอร์มีการแสดงกิจกรรมในรูปแบบได้น้อย ซึ่งต่างจาก Google Calendar ที่แสดงตารางกิจกรรมได้หลายรูปแบบมากกว่า
3. Google Calendar มีการแจ้งเตือนผ่านทางโทรศัพท์เคลื่อนที่ ส่วนโปรแกรมที่คล้ายกับ Google Calendar ไม่มีการแจ้งเตือนแบบนี้
4. Google Calendar ใช้งานได้ง่ายและสะดวก จึงทำให้ผู้ที่เริ่มใช้งานเข้าใจได้ง่าย
5. Google Calendar สามารถใช้ทุกที่ที่มีอินเทอร์เน็ตจึงทำให้สะดวกกว่าโปรแกรมที่อยู่ในคอมพิวเตอร์ซึ่งถ้าคอมพิวเตอร์เครื่องไหนไม่มีโปรแกรมนั้นก็จะใช้ไม่ได้

การใช้งาน Google Calendar

1. การสมัครใช้งาน Google Calendar

กรณีที่ 1 ยังไม่ได้ทำการสมัครเป็นสมาชิกเพื่อใช้งานผลิตภัณฑ์ของ Google ต้องทำการสมัครเข้าเป็นสมาชิกเพื่อใช้งานผลิตภัณฑ์ของ Google ด้วย E-mail ก่อนแล้วจึงจะสามารถใช้งาน Google Calendar ได้

กรณีที่ 2 หากมีการสมัครเป็นสมาชิกแล้วให้ทำการ Login เข้าสู่ระบบ แล้วคลิกที่เพิ่มเติม > ปฏิทิน ดังรูป



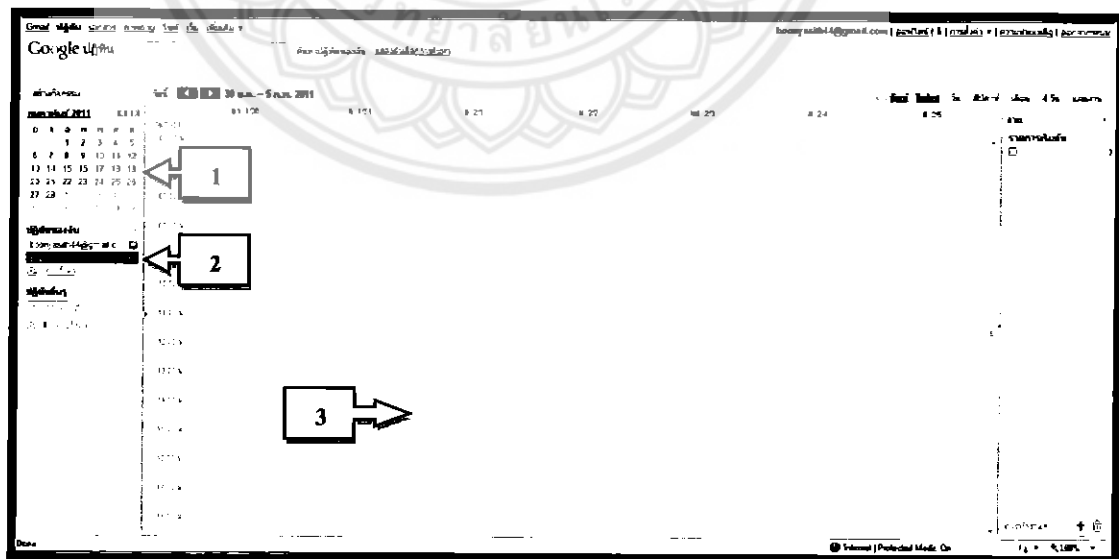
รูปที่ ผ-1 แสดงการเข้าสู่หน้าของ Google Calendar

หมายเลข 1 ผู้ใช้งานระบบ

หมายเลข 2 ลิงค์เพื่อเลือกไปหน้า Google Calendar

2. หน้าหลักของ Google Calendar

หน้าต่างหลักของ Google Calendar หลังจากที่ได้ Login เข้าสู่ระบบแล้ว จะพบว่าเป็นปฏิทินที่มีความสามารถในการแสดงผลที่หลากหลาย ซึ่งมีส่วนประกอบหลักอยู่ 3 ส่วน คือ ส่วนของปฏิทินเพื่อใช้เลือกดูกิจกรรมต่าง ๆ ในแต่ละวันที่ต้องการ ส่วนของรายการปฏิทินเพื่อใช้ตรวจสอบงานต่าง ๆ ซึ่งอาจมีได้หลากหลายตามต้องการ และส่วนตารางกิจกรรมเป็นส่วนที่แสดงกิจกรรมทั้งหมดซึ่งสามารถเลือกให้แสดงตามรายการปฏิทินต่าง ๆ ที่เลือกได้



รูปที่ ผ-2 แสดงหน้าต่างหลักของ Google Calendar

หมายเลข 1 ปฏิทิน

หมายเลข 2 รายการปฏิทิน

หมายเลข 3 ตารางกิจกรรม

3. การเพิ่มปฏิทิน

Google Calendar สามารถสร้างปฏิทินได้หลายปฏิทิน เพื่อการจัดการปฏิทินของตนเองได้ง่าย การเพิ่มปฏิทินสามารถจัดการแบ่งสีสรรเพื่อให้แยกประเภทของกิจกรรมต่าง ๆ ได้ โดยสามารถนำชื่อของปฏิทินที่สร้างขึ้นไปใช้กับระบบได้ ถ้าหากไม่ต้องการใช้ค่าเริ่มต้นของปฏิทิน

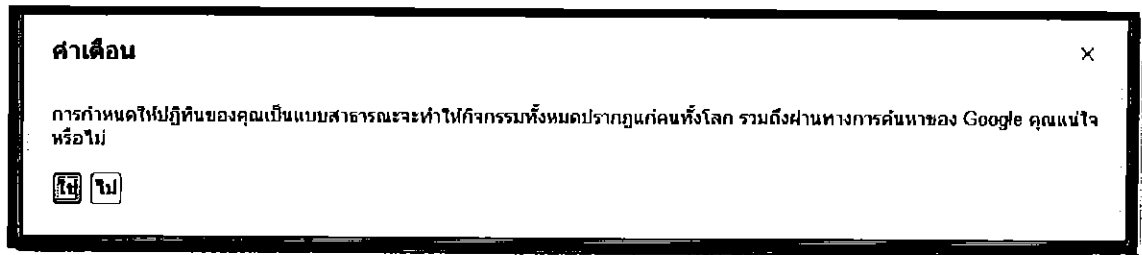


รูปที่ ผ-3 แสดงส่วนการเพิ่มปฏิทินใหม่

หมายเลข 1 ปุ่มเพิ่มปฏิทินใหม่

หมายเลข 2 ปุ่มตั้งค่าปฏิทินที่สร้างขึ้น

หลังจากคลิกปุ่มเพิ่มจะปรากฏหน้าต่างดังรูป ซึ่งต้องทำการกำหนดชื่อปฏิทินเพื่อให้ระบบสามารถอ้างอิงปฏิทินได้



รูปที่ ผ-5 แสดงคำเตือนเนื่องจากการสร้างปฏิทินในแบบสาธารณะ

หลังจากนั้นจะปรากฏหน้าต่างแสดงรายการปฏิทินที่สร้างขึ้นใหม่ดังรูป ซึ่งอาจนำชื่อของปฏิทินที่สร้างขึ้นมาใหม่นี้ไปใช้ในระบบได้



รูปที่ ผ-6 แสดงรายการปฏิทินที่เพิ่มเข้าไปใหม่

หมายเลข 1 ปฏิทินเริ่มต้นของผู้ใช้คนนี้

หมายเลข 2 ปฏิทินที่สร้างขึ้นใหม่

หมายเลข 3 ปฏิทินที่ถูกสร้างก่อนหน้า

เอกสารอ้างอิง

- [1] C# Key. **Data Binding : The SQL Data Reader**. สืบค้นเมื่อ 14 กุมภาพันธ์ 2554, จาก <http://www.csharpkey.com/aspnet/sqlserver/datareader.htm>
- [2] พิรพร หมุนสนิท และอัจจิมา เลี้ยงอยู่. (2552). **Visual Basic 2008 และ Visual C# 2008**. กรุงเทพฯ : บริษัท เททีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด.
- [3] โกศล โสฬสรุ่งเรือง. (2553). **ออกแบบและพัฒนาเว็บไซต์ด้วย jQuery**. กรุงเทพฯ : บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน).
- [4] imooopee. **เซิร์ฟเวอร์ (Server)**. สืบค้นเมื่อ 28 กุมภาพันธ์ 2554, จาก <http://www.vcharkarn.com/vblog/41000/1>
- [5] จูติรัตน์ ถิ่นคุชย์ และคณะ. **Google Calendar**. สืบค้นเมื่อ 1 เมษายน 2554, จาก http://www.pharm.chula.ac.th/webelaming/elearning2553/25_Google_Calendar/index.html
- [6] Hon Wai Chun และ Rebecca Y.M. Wong. **N*-an agent-based negotiation algorithm for dynamic scheduling and rescheduling**. China : Department of computer science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong.
- [7] N.R. Jennings และ A. J. Jackson. **Agent-based Meeting Scheduling: A Design and Implementation**. London : Dept. of Electronic Engineering, Queen Mary & Westfield College.
- [8] Henri Avancini และ Analia Amandi. **A Java Framework for Multi-agent Systems**. Argentina : Universidad Nacional del Comahue.
- [9] Jeffrey W. Herrmann และ Chung-Yee Lee. **Solving a Class Scheduling Problem with a Genetic Algorithm**. Florida : University of Florida.
- [10] ThaiCreate.Com Team. **ASP.NET & Visual Studio 2010 - File Structure**. สืบค้นเมื่อ 7 พฤศจิกายน 2553, จาก <http://www.thaicreate.com/asp.net/asp.net-visual-studio-2010-file-structure.html>
- [11] <http://code.google.com/apis/calendar/overview/>
- [12] Fuji Electric Co. Research and Development, Ltd., 1, **Fuji-machi, Hino**, Tokyo 191, Japan; and Faculty of Environmental Information, Keio University, 5322 Endoh, Fujisawa 252, Japan.

- [13] S. Balasubramanian and D. Norrie, **A multi-agent intelligent design system integrating manufacturing and shop-floor control**. In: Proceedings 1st International Conference on Multiagent Systems (1995).
- [14] A.H. Bond and L. Gasser, **An analysis of problems and research in DAI**. In: A.H. Bond and L. Gasser, Editors, Readings in Distributed Artificial Intelligence, Morgan Kaufmann, San Mateo, CA (1988).
- [15] A. Chavez and P. Maes, **Kasbah: an agent marketplace for buying and selling goods**. In: Proceedings 1st International Conference on the Practical Application of Intelligent Agents and Multi Agents Technology (1996).
- [16] N. Christofides and S. Korman, **A computational survey of methods for the set covering problem**. Math. Oper. Res. 21 (1975).
- [17] V. Chvatal, **A greedy heuristic for the set-covering problem**. Math. Oper. Res. 4 (1979)
- [18] A. Evenchik, **Inference system for argumentation in negotiation between automatic agents**. In: M.Sc. Thesis, Department of Mathematics and Computer Science, Bar-Ilan University, Ramat Gan, Israel (1995).
- [19] M. Fenster, S. Kraus and J. Rosenschein, **Coordination without communication: experimental validation of focal point techniques**. In: Proceedings 1st International Conference on Multiagent Systems (1995).
- [20] L. Glicoe, R. Staats and M. Huhns, **A multi-agent environment for department of defense distribution**. In: Proceedings IJCAI-95 Workshop on Intelligent Systems (1995).
- [21] B. Grosz and S. Kraus, **Collaborative plans for group activities**. In: Proceedings IJCAI-93 (1993).
- [22] B.J. Grosz and S. Kraus, **Collaborative plans for complex group activities**. Artificial Intelligence 86 (1996).
- [23] N.R. Jennings, **Controlling cooperative problem solving in industrial multi-agent systems using joint intentions**. Artificial Intelligence 75 (1995).
- [24] S. Kraus, **Planning and communication in a multi-agent environment**. In: Ph.D. Thesis, Hebrew University, Jerusalem (1988) (written largely in Hebrew).
- [25] S. Kraus, **Agents contracting tasks in non-collaborative environments**. In: Proceedings AAAI-93 (1993).

- [26] S. Kraus, **Beliefs, time and incomplete information in multiple encounter negotiations among autonomous agents**. *Ann. Math. Artif. Intell.* (1997).
- [27] S. Kraus and J.S. Rosenschein, **The role of representation in interaction: discovering focal points among alternative solutions**. In: E. Werner and Y. Demazeau, Editors, *Decentralized Artificial Intelligence Vol. 3*, Elsevier, Amsterdam (1992).



ประวัติผู้ดำเนินโครงการ



ชื่อ นายบุญยะสิทธิ์ ขอบคุณ

ภูมิลำเนา 74/9 หมู่ 8 ต.คลองตาล อ.ศรีสำโรง จ.สุโขทัย
ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนศรีสำโรง
ชนูปถัมภ์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยนเรศวร

E-mail: boat_boonyasith44@hotmail.com



ชื่อ นายณภสินธุ์ แซ่มวัตร

ภูมิลำเนา 45/1 ถ.พุทธบูชา ต.ในเมือง อ.เมือง จ.พิษณุโลก
ประวัติการศึกษา

- จบระดับมัธยมศึกษาจาก โรงเรียนพิษณุโลก
พิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยนเรศวร

E-mail: triplesic@hotmail.com



ชื่อ นางสาวสิริกุล ชวงษ์

ภูมิลำเนา 132-133 หมู่ 4 ต.ชัยสมอทอด อ.บึงสามพัน
จ.เพชรบูรณ์

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนบึงสามพัน
วิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยนเรศวร

E-mail: Sky_Fly_678@hotmail.com