



การพัฒนาโมบายแอปพลิเคชันที่ทำงานร่วมกับไฟร์เบสโดยมีกิตในการควบคุมเวอร์ชัน

**Develop Mobile Application Working on Firebase by using
Git for Version Control**

นาย ชนสิทธิ์ ยอดศรี

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยี ไทย-ญี่ปุ่น

พ.ศ. 2560

การพัฒนาโมบายแอปพลิเคชันที่ทำงานร่วมกับไฟร์เบสโดยมีโดยมีกิตในการควบคุมเวอร์ชัน

Develop Mobile Application Working on Firebase by using Git for Version Control

นาย ชนสิทธิ์ ยอดศรี

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต สาขาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย - ญี่ปุ่น

ปีการศึกษา 2560

คณะกรรมการสอบ

..... ประธานกรรมการสอบ

(ผศ. ตรีรัตน์ เมตต์การุณจิต)

..... กรรมการสอบ

(ดร. ปราณิสรา อิศรเสนา)

..... อาจารย์ที่ปรึกษา

(อาจารย์ สติลา ชีวกิตการ)

..... ประธานสหกิจศึกษาสาขาวิชา

(อาจารย์ อมรพันธ์ ชมกลิ่น)

ลิขสิทธิ์ของสถาบันเทคโนโลยีไทย - ญี่ปุ่น

หัวข้อ	การพัฒนาโมบายแอปพลิเคชันที่ทำงานร่วมกับไฟร์เบสโดยมีโดย มีกิตในการควบคุมเวอร์ชัน Develop Mobile Application Working on Firebase by using Git for Version Control		
ผู้เขียน	นายชนสิทธิ์ ยอดศรี		
คณะ	เทคโนโลยีสารสนเทศ	สาขาวิชา	เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษา	อาจารย์ สลิล ธีวักดาการ		
พนักงานที่ปรึกษา	นาย ณัฐวุฒิ พัทธกันสรเศรษฐ		
ชื่อบริษัท	A-HOST Company Limited		
ประเภทธุรกิจ	บริการทางด้าน Oracle Product และ Hosting Service		

บทสรุป

การปฏิบัติงานสหกิจศึกษา ข้าพเจ้าได้อยู่ตำแหน่ง Programmer ของแผน Business Solution จึงมีแผนการปฏิบัติงานทางด้านการศึกษาเทคโนโลยีใหม่ๆ เพื่อนำเทคโนโลยีเหล่านั้นมาใช้ในการช่วยเหลือหรือสนับสนุนพนักงานในแผนกพัฒนาระบบได้อย่างมีประสิทธิภาพ

การศึกษาเทคโนโลยีใหม่เพื่อนำมาประยุกต์ใช้การพัฒนาแอปพลิเคชันต่างๆ ทำให้ทีม นักพัฒนาสามารถช่วยพัฒนาร่วมกันได้อย่างมีประสิทธิภาพมากกว่าเดิม ทำให้มีความรับผิดชอบในหน้าที่และภาระงานที่ได้รับมอบหมาย สามารถปรับตัวเข้ากับสภาพแวดล้อมการทำงาน ซึ่งนับเป็น ประสบการณ์ที่ดีที่สามารถนำไปต่อยอดเพื่อใช้ในการปฏิบัติงานจริงภายในอนาคตได้

TNI

กิตติกรรมประกาศ

ในการที่ข้าพเจ้าได้มาศึกษา ณ บริษัท เอ-โฮสต์ จำกัด ตั้งแต่วันที่ 16 พฤษภาคม พ.ศ. 2560 ถึงวันที่ 30 กันยายน พ.ศ. 2560 ได้ทำให้ข้าพเจ้าได้เรียนรู้ประสบการณ์และความรู้จากการทำงานจริง ซึ่งมีค่าเป็นอย่างสูงและส่งผลให้ข้าพเจ้าสามารถนำสิ่งที่ได้รับจากการทำสหกิจ เหล่านั้น มาใช้พัฒนาทักษะของตนเอง

สุดท้ายนี้การที่จะทำรายงานการปฏิบัติงานสหกิจศึกษาในครั้งนี้ให้สำเร็จลุล่วงได้นั้นต้องมาจากความร่วมมือและการสนับสนุนจากทุกบุคคลภายในแผนกที่ทำงาน รวมไปถึงในบริษัท ขอขอบคุณไว้ ณ ที่นี้

ชนสิทธิ์ ยอดศรี
ผู้จัดทำ

TNI

สารบัญ

หน้า

บทสรุป	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญตาราง	ช
สารบัญภาพประกอบ	ซ

บทที่ 1

1. บทนำ	1
1.1 ชื่อและที่ตั้งของสถานประกอบการ	1
1.2 ลักษณะธุรกิจของสถานประกอบการ	1
1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร	6
1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย	7
1.5 พนักงานที่ปรึกษาและตำแหน่งของพนักงานที่ปรึกษา	7
1.6 ระยะเวลาปฏิบัติงาน	7
1.7 ที่มาและความสำคัญของปัญหา	7
1.8 วัตถุประสงค์ของโครงการ	7

บทที่ 2

2. ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน	9
2.1 เทคโนโลยีที่ใช้ในการปฏิบัติงาน	9
2.1.1. Git	9
2.1.1.1. ทำไมถึงต้องใช้ Git	10

สารบัญ (ต่อ)

หน้า

2.1.2. GitHub	11
2.1.3. AWS CodeCommit	12
2.1.4. Firebase	13
2.1.4.1. Service ต่างๆบน Firebase	13
2.1.5. Visual Code Studio	15
2.2 ภาษาที่ใช้ในการปฏิบัติงาน	16
2.2.1. Ionic Framework	16
2.2.1.1. HTML5	17
2.2.1.2. CSS	18
2.2.1.3. AngularJS	19
บทที่ 3	
3. แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน	20
3.1 แผนงานการฝึกงาน	20
3.2 รายละเอียดที่นักศึกษาปฏิบัติในการฝึกงาน	21
3.2.1. การศึกษาเทคโนโลยีใหม่ๆเพื่อนำมาสนับสนุนและประยุกต์ใช้ในการพัฒนาระบบได้	21
3.2.2. งานอื่นๆ	21
3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงาน	21
3.3.1. ศึกษาเรื่อง Git	21
3.3.2. ฝึกใช้งานคำสั่งต่างๆ Git	21
3.3.3. ศึกษา Web Service ที่ใช้รองรับไฟล์ Git (GitHub)	22
3.3.4. ศึกษา Web Service ที่ใช้รองรับไฟล์ Git (AWSCodeCommit)	22

สารบัญ (ต่อ)

หน้า

3.3.5. นำโปรเจกต์ต่างๆของทีมขึ้นไปยัง AWSCodeCommit	22
3.3.6. ศึกษาเรื่อง Firebase	22
3.3.7. จัดทำแอปพลิเคชันใช้เทคโนโลยีของ Firebase และ Git	23
3.3.8. ปรับปรุงและแก้ไขปัญหาที่เกิดขึ้นของแอปพลิเคชัน	23
3.3.9. จัดทำเอกสารที่เกี่ยวข้อง	23

บทที่ 4

4. ผลการดำเนินงาน การวิเคราะห์และสรุปผล 24

4.1 ขั้นตอนและผลการดำเนินงาน 24

4.1.1. การศึกษาเกี่ยวกับ Git 24

4.1.1.1. ทำไมถึงต้องใช้ Git 25

4.1.1.2. Git Status 25

4.1.1.3. ติดตั้ง Git 26

4.1.1.4. Git Command Line 26

4.1.2. การศึกษา Web Service ที่ให้บริการฝากไฟล์ Git 32

4.1.2.1. GitHub 32

4.1.2.2. AWSCodeCommit 35

4.1.3. การอัปโหลดไฟล์โปรเจกต์ขึ้นไปยัง Web Service ที่ให้บริการฝากไฟล์ Git 56

4.1.4. ศึกษาข้อมูลเกี่ยวกับ Firebase 57

4.1.4.1. Firebase Authentication 58

4.1.4.2. Firebase Real-time Database 59

4.1.4.3. Firebase Storage 60

4.1.5. การทำแอปพลิเคชันที่ใช้บริการของ Firebase โดยใช้ Git เป็นตัวควบคุมเวอร์ชัน 61

สารบัญ (ต่อ)

	หน้า
บทที่ 5	
5. บทสรุปและข้อเสนอแนะ	68
5.1 สรุปผลการดำเนินงาน	68
5.2 แนวทางการแก้ไขปัญหา	68
5.3 ข้อเสนอแนะจากการดำเนินงาน	69
เอกสารอ้างอิง	70
ภาคผนวก	
ก. การติดตั้ง Ionic 2	71
ประวัติผู้จัดทำโครงการ	74

TNI

สารบัญตาราง

ตารางที่

หน้า

1.1 งานที่ได้รับมอบหมาย

7

3.1 ตารางปฏิบัติงานในโครงการ

20



สารบัญภาพประกอบ

ภาพที่	หน้า
1.1 แผนที่ บริษัท เอ-โฮสต์ จำกัด	1
1.2 A-HOST Proud Awards	3
1.3 คณะผู้บริหารบริษัท เอ-โฮสต์ จำกัด แต่ละแผนก	6
2.1 สัญลักษณ์ Git	9
2.2 รูปแบบการทำงานของ Git	11
2.3 สัญลักษณ์ GitHub	11
2.4 สัญลักษณ์ AWS CodeCommit	12
2.5 สัญลักษณ์ Firebase	13
2.6 หน้า Dashboard บน Firebase	13
2.7 สัญลักษณ์ Visual Studio Code	15
2.8 Visual Code Studio	15
2.9 สัญลักษณ์ Ionic Framework	16
2.10 สัญลักษณ์ HTML5	17
2.11 สัญลักษณ์ CSS	18
2.12 สัญลักษณ์ AngularJS	19
3.1 โปรเจกบน Web service AWSCodeCommit	22
4.1 Workflow of Git	26
4.2 คำสั่งที่ใช้ในการ setup ชื่อและอีเมลล์ของ Git	26
4.3 คำสั่ง git init	26
4.4 คำสั่ง git status	27
4.5 คำสั่ง git add	27
4.6 คำสั่ง git commit	28
4.7 คำสั่ง git log	28

สารบัญภาพประกอบ (ต่อ)

ภาพที่	หน้า
4.8 คำสั่งที่ใช้ในการสร้าง branch และเช็คว่ามีกี่ branch	28
4.9 Branching Workflow	29
4.10 คำสั่ง git checkout กรณีที่ใช้ในการเข้า branch	29
4.11 คำสั่ง git checkout กรณีไฟล์หายหรือเผลอลบ	29
4.12 คำสั่ง git merge	30
4.13 คำสั่ง git remote	30
4.14 คำสั่ง git push	30
4.15 คำสั่ง git fetch	31
4.16 คำสั่ง git clone	31
4.17 ภาพการสร้าง Repository บน GitHub	32
4.18 url ของ remote ใช้เชื่อมต่อฝั่ง Local และ Server	33
4.19 การ Forking	33
4.20 ตัวอย่างการทำ Pull Request	34
4.21 หน้า Pull Request	34
4.22 ปุ่ม Sign In เข้าไปใน Console	35
4.23 หน้า Sign In	35
4.24 หน้า Service บน AWSCodeCommit	36
4.25 ปุ่ม Add User บนหน้า IAM Dashboard	36
4.26 ชื่อ IAM User	36
4.27 หน้าประเภทการเข้าถึงของ IAM User	37
4.28 หน้า Permission ของ IAM User	38
4.29 หน้า Review	39
4.30 ทำการ Add IAM User สำเร็จ	39

สารบัญภาพประกอบ (ต่อ)

ภาพที่	หน้า
4.31 link login หน้า Dashboard	40
4.32 หน้า login IAM User	40
4.33 หน้า User บน IAM Dashboard	40
4.34 Generate เพื่อดูค่า Credentials	41
4.35 Git credentials generated	41
4.36 การ push ไฟล์ขึ้นไปยัง AWSCodeCommit	42
4.37 Login with Root User	42
4.38 AWSCodeCommit Dashboard	43
4.39 Create Repository หน้า Dashboard	43
4.40 Create Repository	44
4.41 URL เพื่อใช้เชื่อมต่อระหว่าง Local และ Server	44
4.42 IAM Dashboard	45
4.43 แสดงถึงการ Login ด้วย Root Account	45
4.44 Create Policies	46
4.45 วิธีสร้าง Policy	46
4.46 วิธีสร้าง Policy	46
4.47 วิธีสร้าง Policy	47
4.48 วิธีการหา ARN	47
4.49 วิธีการหา ARN	48
4.50 วิธีการหา ARN	48
4.51 วิธีสร้าง Group	48
4.52 Set Group Name	49
4.53 Attach Policy for Group	49

สารบัญภาพประกอบ (ต่อ)

ภาพที่	หน้า
4.54 Review Group	50
4.55 Add User to Group	50
4.56 AWSCodeCommit Dashboard	51
4.57 Setting Repository	51
4.58 Change Name of Repository	52
4.59 Repository Description	52
4.60 Default branch	53
4.61 Delete Repository	53
4.62 Repository Dashboard	53
4.63 การเข้าถึง Commits	54
4.64 หน้า Commits	54
4.65 Commit ID	55
4.66 การเปรียบเทียบระหว่าง 2 commit	55
4.67 หน้า Compare	56
4.68 การเปรียบเทียบระหว่าง 2 commit	56
4.69 โปรเจกต์อยู่บน AWSCodeCommit	57
4.70 Dashboard บน Firebase	57
4.71 Demo Application หน้า Register	58
4.72 Firebase Authentication	58
4.73 Demo Application หน้าโน้ตเตือนความจำ	59
4.74 การอัปเดตแบบ Real-time	60
4.75 Demo Application หน้า Upload ไฟล์	60
4.76 ภาพบน Firebase Storage	61

สารบัญภาพประกอบ (ต่อ)

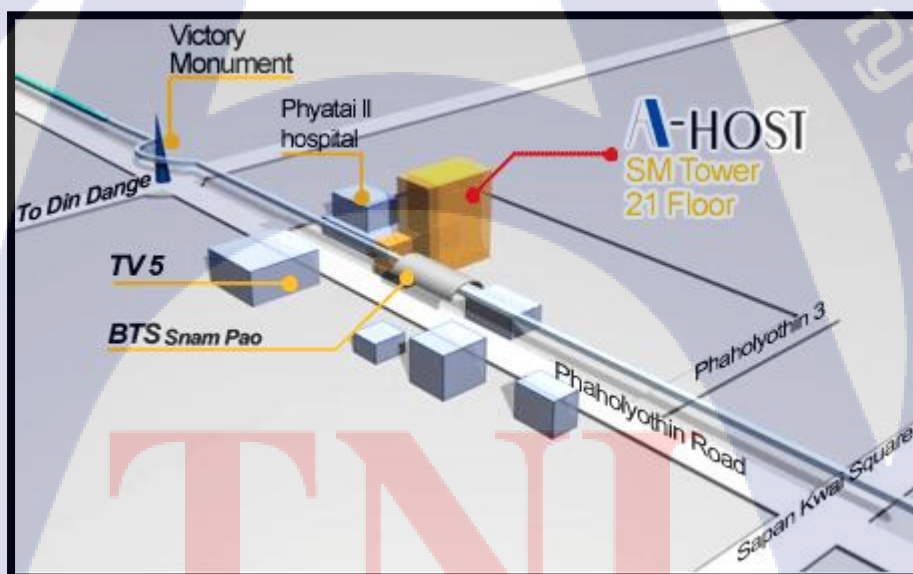
ภาพที่	หน้า
4.77 หน้า Login	62
4.78 หน้า Register	62
4.79 หน้า My Project	63
4.80 หน้า New Project	63
4.81 หน้า Backlog	64
4.82 หน้า To-do	64
4.83 หน้า Doing	65
4.84 หน้า Done	65
4.85 หน้า Add Task	66
4.86 หน้า Detail	66
4.87 หน้า Edit	67
ก.1 หน้า Setup Node JS	72
ก.2 หน้า Setup Node JS	73
ก.3 ติดตั้ง ionic เสร็จสมบูรณ์	73

บทที่ 1

บทนำ

1.1 ชื่อและสถานที่ตั้งของสถานประกอบการ

ชื่อสถานประกอบการ : บริษัท เอ-โฮสต์ จำกัด A-Host Company Limited
ที่ตั้งสถานประกอบการ : 979/52-55 ชั้น 21 อาคาร SM Tower ถนนพหลโยธิน
แขวงสามเสนใน เขตพญาไท กรุงเทพฯ 10400
โทรศัพท์ : (66) 2298-0625-32
โทรสาร : (66) 2298-0053



ภาพที่ 1.1 แผนที่บริษัท เอ-โฮสต์ จำกัด

1.2 ลักษณะธุรกิจของสถานประกอบการหรือการให้บริการขององค์กร

บริษัทเอ-โฮสต์จำกัดได้ก่อตั้งขึ้นเมื่อปีพ.ศ.2542 ในฐานะหนึ่งบริษัทในเครือของบริษัทซิสเต็มส์คอร์ปอเรชั่น (มหาชน) จำกัดและเป็นผู้เชี่ยวชาญด้านการบริการจัดวางระบบสารสนเทศ (Information Technology : IT) และบริการเสริมต่างๆสำหรับลูกค้าตั้งแต่ธุรกิจขนาดย่อมไปจนถึงขนาดกลางธุรกิจหลักของบริษัทเอ-โฮสต์คือการให้บริการโฮสติ้ง (Hosting) ซึ่งเป็นการเปิดเครื่องคอมพิวเตอร์แม่ข่ายในการให้เช่าพื้นที่เพื่อวางระบบและบริการระบบสารสนเทศด้วยผลิตภัณฑ์ของออราเคิล (Oracle) เป็นซอฟต์แวร์สำหรับการวางแผนบริหารทรัพยากรของระดับแนวหน้าของโลก

เอ-โฮสต์ถือกำเนิดขึ้นจากกลุ่มผู้เชี่ยวชาญด้านสารสนเทศท่ามกลางภาวะเศรษฐกิจตกต่ำทั่วภูมิภาคเอเชียแต่เอ-โฮสต์ก็สามารถเติบโตอย่างรวดเร็วและมั่นคงตั้งแต่แรกก่อตั้งด้วยจุดแข็งในฐานะผู้บุกเบิกธุรกิจโฮสติ้งเซิร์ฟเวอร์พร้อมทั้งนำธุรกิจแนวใหม่อย่างการให้บริการระบบโปรแกรมประยุกต์ หรือ ASP (Application Services Provider) เข้ามาให้บริการเป็นรายแรกในเมืองไทย อีกทั้งยังถือเป็นผู้ให้บริการรายแรกนอกประเทศสหรัฐอเมริกาด้วย

ในฐานะผู้นำในอุตสาหกรรมนี้เป็นเวลามากกว่า 10 ปีเอ-โฮสต์ได้เสริมสร้างความแข็งแกร่งทางธุรกิจด้วยบริการที่มีความโดดเด่นและรวบรวมเอาทรัพยากรบุคคลซึ่งได้สั่งสมประสบการณ์และความชำนาญไว้อย่างพร้อมเพรียงส่งผลให้ศูนย์ข้อมูลของเอ-โฮสต์ในปัจจุบันมีความสมบูรณ์ด้วยกลุ่มเซิร์ฟเวอร์ (Server) ที่เชื่อมต่อกันในลักษณะการจัดกลุ่ม (Clustering) ซึ่งเปี่ยมสมรรถนะสามารถให้บริการแก่ผู้ใช้งานจำนวนมากได้ในเวลาเดียวกัน

นอกจากนี้เอ-โฮสต์ยังคิดตั้งระบบรักษาความปลอดภัยระบบสำรองข้อมูลและระบบบริหารจัดการรวมถึงอุปกรณ์ต่างๆอย่างครบครันเพื่อให้เอ-โฮสต์สามารถตอบสนองต่อระดับความต้องการในระดับสูงสุดที่ลูกค้าคาดหวังได้ตลอดจนเป็นการสร้างความมั่นใจให้แก่ลูกค้าที่ใช้บริการโฮสติ้งและแอปพลิเคชันต่างๆของเอ-โฮสต์ว่าจะได้รับทั้งประสิทธิภาพและความปลอดภัยอย่างครบครัน ธุรกิจการให้บริการระบบโปรแกรมประยุกต์ในรูปแบบ ASP เอ-โฮสต์ไม่เพียงแต่ให้บริการด้านโปรแกรมประยุกต์ด้านการดำเนินธุรกิจทางอิเล็กทรอนิกส์ระดับโลกของออราเคิลพร้อมโครงสร้างพื้นฐานทางเทคโนโลยีสารสนเทศเท่านั้นแต่ยังมีบริการที่ครอบคลุมตั้งแต่การให้คำปรึกษาการสนับสนุนและการให้บริการทั่วไปอย่างพร้อมพร้อมครบครัน

นอกจากธุรกิจโฮสติ้งและธุรกิจการให้บริการโปรแกรมประยุกต์ในรูปแบบ ASP ซึ่งถือเป็นธุรกิจหลักเอ-โฮสต์ยังดำเนินธุรกิจอย่างต่อเนื่องโดยการขยายหน่วยงานใหม่เพิ่มขึ้นนั่นก็คือ Core Technology Division หน่วยงานเทคโนโลยีหลักที่ให้คำตอบเบ็ดเสร็จแก่ลูกค้าด้วยระบบฐานข้อมูลและเครื่องมือต่างๆของออราเคิลซึ่งช่วยเพิ่มความสามารถของลูกค้าในการออกแบบพัฒนา

ปรับเปลี่ยนระบบแอปพลิเคชันให้เหมาะสมกับธุรกิจนั้นๆภายใต้คำปรึกษาแนะนำและการวางระบบของเอ-โฮสต์ลูกค้าสามารถบริหารระบบฐานข้อมูลของตนเองและดูแลระบบดังกล่าวได้อย่างมีประสิทธิภาพ

ตลอดระยะเวลามากกว่า 10 ปีในการดำเนินธุรกิจของเอ-โฮสต์ไม่เพียงแต่ในฐานะผู้บุกเบิกธุรกิจโฮสติ้งและธุรกิจการให้บริการโปรแกรมประยุกต์ในรูปแบบ ASP เท่านั้นแต่เอ-โฮสต์ยังได้ทำการติดตั้งระบบสารสนเทศรวมทั้งผลิตภัณฑ์ของออราเคิลให้กับลูกค้าจนประสบความสำเร็จเป็นจำนวนมากซึ่งหลายรายเป็นหนึ่งในร้อยบริษัทชั้นนำของประเทศไทยแต่สิ่งที่สำคัญกว่านั้นก็คือการที่เอ-โฮสต์ได้สานสัมพันธ์กับลูกค้าและพันธมิตรทางธุรกิจอย่างแนบแน่นจนกลายเป็นหุ้นส่วนทางกลยุทธ์และได้รับแต่งตั้งให้เป็น OCAP (Oracle Certified Advantage Partner) รายแรกในประเทศไทย



รูปที่ 1.2 รางวัลที่บริษัท A-HOST ได้รับ

ปัจจุบันเอ-โฮสต์มีประเภทของสินค้าและการบริการซึ่งสามารถแบ่งออกเป็น 3 กลุ่มใหญ่ได้แก่

1.2.1 Hosting & Outsource Service

เอ-โฮสต์ได้ปรับปรุงและขยายการบริการ Hosting และการให้บริการภายนอก (Outsource) จนสามารถครอบคลุมความต้องการของลูกค้าได้อย่างหลากหลายโดยยึดหลักในการให้บริการที่เรียกว่า “Peace of Mind for the customer” ซึ่งหมายถึงการที่จะทำงานให้กับลูกค้าแบบครบวงจรเพื่อที่ลูกค้าจะสามารถใช้งานระบบสารสนเทศได้อย่างมีประสิทธิภาพได้อย่างสบายใจไร้ความกังวลต่อความเสี่ยงต่างๆ ไม่ว่าจะเป็นเรื่องของปัญหาทางเทคนิคการจัดทำระบบและข้อมูลสำรองการปรับแต่งระบบให้ได้ประสิทธิภาพสูงสุด (Performance Tuning) และที่สำคัญที่สุดคือการที่เข้ามารับภาระในด้านการบริหารจัดการบุคลากรทางด้านสารสนเทศทั้งหมดแทนลูกค้า

การใช้บริการ Hosting และ Outsource จะทำให้ลูกค้าสามารถทุ่มเทกำลังสมองเวลาและทรัพยากรขององค์กรให้กับธุรกิจที่เป็นแกนหลัก (Core Business) ซึ่งเป็นสิ่งที่ลูกค้าเน้นกว่าโดยทั่วไปแล้วบริการ Hosting และ Outsource จะประกอบด้วยส่วนประกอบและบริการย่อยๆ ดังต่อไปนี้

1. High Availability and High Performance IT Infrastructure
2. Dedicated or Co-Location Service
3. Disaster Site
4. Oracle E-Business Application (ERP, CRM, SCM)
5. ERP Implementation Service
6. System and Database Administration
7. Help Desk
8. On-Request Service i.e. On-Site Support, Software Customization

ทั้งนี้การบริการ Hosting และ Outsource สามารถครอบคลุมได้ทั้งระบบที่ใช้เทคโนโลยีของออราเคิลและระบบที่ใช้เทคโนโลยีอื่นๆ

1.2.2 Oracle Core Technology Products and Advanced Services

เอ-โฮสต์เป็นผู้ดำเนินการดำเนินธุรกิจในฐานะผู้แทนจำหน่ายเพิ่มมูลค่าให้กับออราเคิลโดยไม่เพียงแต่ทำหน้าที่ในการจัดจำหน่ายสินค้าในกลุ่มแกนหลักของเทคโนโลยี (Core Technology) ของออราเคิลทุกประเภทแต่ยังมีทีมผู้เชี่ยวชาญที่จะให้การสนับสนุนและบริการเสริมอย่างครบวงจรแก่บริษัทลูกค้าและลูกค้าไม่ว่าจะเป็นการร่วมจัดกิจกรรมทางการตลาดการฝึกอบรมการติดตั้งระบบ

และการให้คำปรึกษาเพื่อแก้ไขปัญหาต่างๆสินค้าและบริการที่อยู่ในกลุ่ม Oracle Core Technology Products and Advanced Services ได้แก่

1. Oracle Database and database options
2. Oracle Business Intelligence Suite
3. Business Partner Development
4. System Installation, Integration and Optimization
5. Oracle Fusion Middleware (รวมถึง BEA)
6. สินค้าอื่นๆทุกประเภทของออราเคิล
7. Marketing and Lead Generation Activities
8. SOA-Based Development and Implementation

ผลสำเร็จในการดำเนินธุรกิจประเภทนี้ทั้งในด้านการตลาดและบริการทำให้เอ-โฮสต์ได้รับรางวัล ASEAN Partner of the Year ในปี 2005

1.2.3 Oracle Enterprise Performance Management (EPM) และ Hyperion Business

Intelligence Products and Services

ความต้องการสูงสุดประการหนึ่งของผู้บริหารในการนำเอาระบบสารสนเทศมาใช้ในการองค์กรไม่ว่าจะเป็นภาครัฐหรือเอกชนคือการทำให้ผู้บริหารสามารถได้ข้อมูลที่แสดงให้เห็นถึงสถานะในการดำเนินธุรกิจได้อย่างแม่นยำรวดเร็วและการนำเอาข้อมูลมาวิเคราะห์และวางแผนทั้งในระดับปฏิบัติการและในระดับกลยุทธ์เพื่อให้ธุรกิจสามารถได้เปรียบในการแข่งขันปรับตัวตามเศรษฐกิจได้ในทุกสถานการณ์ Oracle Enterprise Performance Management (EPM) และ Hyperion Business Intelligence จัดเป็นซอฟต์แวร์ชั้นนำของโลกที่สามารถสนองต่อความต้องการในลักษณะดังกล่าวได้เป็นอย่างดี

เอ-โฮสต์มีทีมงานที่ปรึกษาที่มีประสบการณ์ทั้งทางด้านธุรกิจและทางด้านเทคนิค รวมถึงความเข้าใจในระบบ ERP ของออราเคิลอย่างลึกซึ้งจึงทำให้สามารถให้บริการที่ปรึกษาเพื่อออกแบบติดตั้งเชื่อมโยงและปรับใช้ระบบให้กับลูกค้าจนเกิดประสิทธิภาพสูงสุดอีกทั้งยังมีความยืดหยุ่นและให้การตอบสนองที่เร็วกว่าเมื่อเทียบกับการว่าจ้างที่ปรึกษาจากต่างประเทศ

1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร



รูปที่ 1.3 คณะผู้บริหาร A-HOST Company Limited

1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย

ตำแหน่ง : Programmer

หน่วยงาน : BS

ตารางที่ 1.1 แสดงงานที่ได้รับมอบหมาย

KPI Description	Expected Results
1. ช่วยสนับสนุนในการทำ Mobile Application โดยนำเสนอเทคโนโลยีใหม่เข้ามาประยุกต์ใช้ในการพัฒนา	- นำเสนอเทคโนโลยีใหม่ที่สามารถช่วยทีมพัฒนาระบบได้ - สามารถพัฒนาระบบที่สามารถพร้อมใช้งานได้ - เมื่อระบบมีปัญหา สามารถช่วยแก้ไขปัญหาได้ - สามารถเขียนโปรแกรมได้อย่างมีประสิทธิภาพ
2. ทำหน้าที่ผู้ทดสอบระบบ (Tester) ให้กับระบบต่างๆที่เกิดขึ้นของแผนก	- สามารถช่วยกำหนด Test Case ได้ - สามารถทดสอบระบบได้อย่างมีประสิทธิภาพ
3. พัฒนา Mobile Application - Firebase พัฒนา Mobile Application ที่ทำงานร่วมกับ Firebase โดยมีการใช้งาน Tools ต่างๆภายใน Firebase - Github ศึกษาการทำงานและเครื่องมือต่างๆเพื่อนำมาใช้งานร่วมกับการพัฒนา Mobile Application	- สามารถพัฒนาระบบได้อย่างมีประสิทธิภาพ - ทำเอกสารสำหรับการพัฒนาระบบเช่น Code Diff. หรือ programming flow - จัดทำเอกสาร User Manual - จัดทำเอกสาร Programmer Manual
4. ศึกษาเทคโนโลยีใหม่ หรือ หัวข้อที่น่าสนใจ เพื่อสนับสนุนงานโครงการต่างๆที่ได้รับมอบหมาย	- สามารถนำไปใช้งานได้จริง - สามารถถ่ายทอดให้ผู้อื่นได้ - ศึกษาเทคโนโลยี และจัดทำ Programmer Manual - 1. Github - 2. Firebase
5. จัดทำ Individual Work List	- นำเสนอ IWL เป็นประจำทุกเดือน

1.5 พนักงานที่ปรึกษา และ ตำแหน่งของพนักงานที่ปรึกษา

นาย ณัฐวุฒิ พิทักษ์นรเศรษฐ์

ตำแหน่ง : Programmer

1.6 ระยะเวลาที่ปฏิบัติงาน

ระยะเวลาปฏิบัติงานสหกิจศึกษาเป็นเวลา 4 เดือน

- เริ่มต้นปฏิบัติงานสหกิจศึกษา วันที่ 16 พฤษภาคม 2560
- สิ้นสุดงานสหกิจศึกษา วันที่ 30 กันยายน 2560

1.7 ที่มาและความสำคัญของปัญหา

1. เพื่อศึกษาการและหาประสบการณ์การทำงานจริงภายในองค์กร
2. เพื่อนำความรู้ที่ได้ไปใช้ในการพัฒนาโครงการและการทำงานในอนาคต

1.8 วัตถุประสงค์หรือจุดมุ่งหมายของโครงการ

1. ได้ประสบการณ์ในการทำงานจริง
2. ได้นำความรู้มาประยุกต์ใช้ในการทำงาน

บทที่ 2

ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

ในการปฏิบัติงานสหกิจศึกษาครั้งนี้ เป็นการนำความรู้ทางด้านทฤษฎีและเทคโนโลยีมาใช้ในการปฏิบัติงานทุกส่วนตลอดการปฏิบัติงานสหกิจศึกษา ซึ่งเป็นการนำความรู้ทั้งที่เคยเรียนมาประยุกต์ใช้และเป็นการศึกษาเรียนรู้สิ่งใหม่ ๆ ที่ได้จากการปฏิบัติงาน

2.1 เทคโนโลยีที่ใช้ในการปฏิบัติงาน

2.1.1 Git



ภาพที่ 2.1 สัญลักษณ์ Git

Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดการเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจกต์เรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชันต่างๆของโปรเจกต์ที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ Version Control ก็เหมาะสมอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียวโดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม

2.1.1.1 ทำไมถึงต้องใช้ Git

1) Track version ของ Source Code ย้อนกลับได้

เมื่อจัดเก็บไฟล์เข้าไปในระบบของ Git จะเรียกว่า Git Repository ซึ่งเก็บสำรองข้อมูลและการเปลี่ยนแปลงของ Source Code ทำให้สามารถย้อนกลับไปเวอร์ชันใดๆ ก่อนหน้า และดูรายละเอียดการเปลี่ยนแปลงของแต่ละเวอร์ชันได้ นอกจากนั้นยังสามารถดูได้ว่าใครเป็นคนแก้ไข

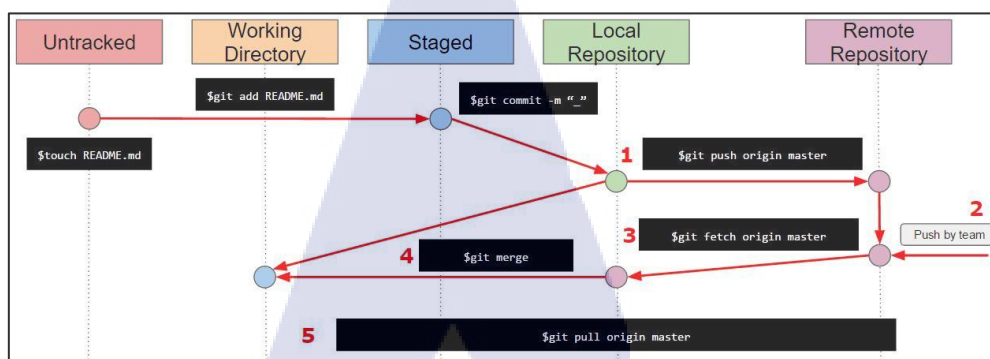
2) ช่วยในการพัฒนาซอฟต์แวร์เป็นทีม

Git สามารถเก็บบันทึกการเปลี่ยนแปลงของ Source Code เวอร์ชันล่าสุดไว้ที่ Local Repository ซึ่งสามารถทำงานได้โดยไม่ต้องต่อกับอินเทอร์เน็ต และเมื่อต้อง Update การเปลี่ยนแปลงของ Source Code เวอร์ชันล่าสุดให้กับเพื่อนร่วมทีมก็สามารถที่จะ Push ขึ้นไปเก็บที่ Remote Repository (Git Hosting) และเพื่อนร่วมทีมก็สามารถ Pull เวอร์ชันล่าสุดนั้นมารวม (Auto Merge) ที่เครื่องของเขาเอง ทำให้ Source Code ที่พัฒนาร่วมกันกับคนภายในทีมเป็นเวอร์ชันล่าสุดเสมอ

3) Git Status

สถานะของ Source Code ที่เก็บอยู่ในระบบของ Git นั้นมีดังนี้

- Untracked เป็นสถานะที่ Source Code ถูกเพิ่มเข้ามาใหม่และยังไม่ได้ถูกเก็บไว้ในระบบของ Git
- Working Directory เป็นสถานะที่กำลังมีการเปลี่ยนแปลงหรือแก้ไข Source Code หรืออาจจะเรียกสถานะนี้ว่า Modified
- Staged เป็นสถานะที่ Source Code กำลังเตรียมที่จะ Commit เพื่อยืนยันการเปลี่ยนแปลงก่อนที่จะเก็บลงในสถานะ Local Repository
- Local Repository เป็นสถานะที่มีการเก็บบันทึกข้อมูลการเปลี่ยนแปลงของ Source Code ลงไปที่ Git Repository ที่เป็น Local (ที่เครื่องตัวเอง)
- Remote Repository เป็นสถานะที่มีการเก็บบันทึกข้อมูลการเปลี่ยนแปลงของ Source Code ลงไปที่ Git Repository ที่เป็น Hosting (ที่เครื่องเซิร์ฟเวอร์)



ภาพที่ 2.2 รูปแบบการทำงานของ Git

2.1.2 GitHub



ภาพที่ 2.3 สัญลักษณ์ GitHub

GitHub คือ website Git (version control repository) ที่อยู่บน internet มีการทำงานแบบเดียวกับ Git แต่สามารถเข้าถึงข้อมูลและจัดการไปผ่าน web โดยไม่ต้องเสียเงิน หรือลงทุนตั้ง server เพื่อติดตั้ง Git เองเลย แต่ code project ทั้งหมดจะถูกแจกจ่ายให้ผู้อื่นสามารถเห็นได้ด้วย ซึ่ง GitHub ก็มีการเสนอ plan แบบส่วนตัวให้ถ้าอยากให้ code ไม่ถูกแจกจ่ายออกไปโดยจะมีค่าใช้จ่ายตรงนี้ ปัจจุบันมีมากกว่า 20 ล้าน user รวมกันกว่า 60 ล้าน repository บนระบบ

2.1.3 AWS CodeCommit



ภาพที่ 2.4 สัญลักษณ์ AWS CodeCommit

AWS CodeCommit คือ version control service hosted โดย Amazon Web Services ที่สามารถจัดเก็บและจัดการไฟล์ git repository แบบ private ได้

นอกจากนี้ยังช่วยลดความจำเป็นในการระบบควบคุมแหล่งที่มาหรือกังวลเกี่ยวกับการปรับโครงสร้างพื้นฐานอีกทั้งยังสามารถใช้ AWS CodeCommit ในการจัดเก็บตั้งแต่ code ไปยัง binaries สนับสนุนฟังก์ชันมาตรฐานของ git ดังนั้นจึงทำงานได้อย่างราบรื่นด้วยเครื่องมือ git ที่มีอยู่

ข้อดีของ AWS CodeCommit

1. ได้รับผลประโยชน์แบบเต็มรูปแบบจาก Amazon Web Service
2. มีการเก็บ code ไว้อย่างปลอดภัย
3. สามารถปรับแต่งหรือควบคุมเวอร์ชันของโปรเจกต์ได้อย่างง่ายดาย
4. เก็บทุกอย่างรายละเอียดและทุกเวลา
5. สามารถผสมผสานรวมกับ service อื่นๆบน Amazon Web Service
6. ย้ายไฟล์จาก remote ตัวอื่นๆได้อย่างง่ายดาย
7. ใช้เครื่องมือ git เบื้องต้นที่สามารถใช้งานได้ง่าย

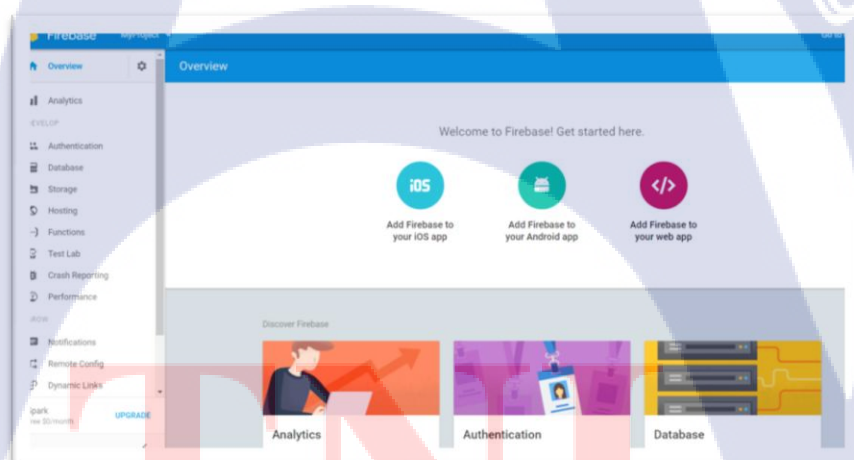
2.1.4 Firebase



ภาพที่ 2.5 สัญลักษณ์ Firebase

Firebase คือ Project ที่ถูกออกแบบมาให้เป็น API และ Cloud Storage สำหรับพัฒนา Realtime Application รองรับหลาย Platform ทั้ง IOS App, Android App, Web Application

2.1.4.1 Service ต่างๆบน Firebase



รูปที่ 2.6 หน้า Dashboard บน Firebase

1) Firebase Authentication จะเป็นบริการที่เข้ามาจัดการ backend ทั้งหมด ทั้ง การ register, การ sign-in, การ reset password, โดยจะมี SDK ให้ทั้ง Android, iOS และ Web นำไปติดตั้งและใช้งาน ซึ่งรองรับการ sign-in หลากหลายรูปแบบทั้งจาก social network ยอดนิยม, จาก Email และ Password ของผู้ใช้งาน หรือแบบไม่ระบุตัวตน(Anonymous)

2) Firebase Realtime Database เป็น NoSQL cloud database ที่เก็บข้อมูลในรูปแบบของ JSON และมีการ sync ข้อมูลแบบ realtime กับทุก devices ที่เชื่อมต่อแบบอัตโนมัติ รองรับการทำงานแบบ offline คือข้อมูลจะเก็บไว้ใน local และจะ sync ข้อมูลอัตโนมัติ

3) Firebase Storage เป็นบริการที่สามารถ Upload หรือ Download ไฟล์บน Google Cloud Storage โดยถ้าใช้งานฟรีจะมีพื้นที่ให้เก็บ 5GB และสามารถ upload และ download ได้ 50000 ครั้ง

4) Firebase Analytics บริการวิเคราะห์ข้อมูล ดึงเทคโนโลยีมาจาก Google Analytics แดมยังเปิดให้ใช้ฟรีแบบไม่จำกัดปริมาณข้อมูลใดๆ

5) Firebase Cloud Messaging (FCM) ระบบส่งข้อความแจ้งเตือน ใช้งานฟรีไม่จำกัดปริมาณข้อความ

6) Firebase Remote Config ตัวช่วยอัปเดตคอนฟิกของแอป สำหรับปรับแต่งค่าต่างๆ ในแอปจากระยะไกล (เช่น เกมที่อยากปรับสมดุลของเกมตลอดเวลา) สามารถใช้ร่วมกับ Firebase Analytics เพื่อกำหนดผู้ใช้งานแยกเป็นกลุ่มๆ ได้

7) Firebase Crash Reporting ตัวรายงานการแครชของแอป รองรับทั้ง iOS และ Android

8) Firebase Test Lab for Android บริการทดสอบแอปบนฮาร์ดแวร์จริง

9) Firebase Notifications เป็นคอนโซลสำหรับนักพัฒนา เพื่อยิงข้อความผ่าน FCM ไปยังผู้ใช้ สำหรับโปรโมทหรือกระตุ้นให้ผู้ใช้กลับมาเปิดแอปของเรา (เช่น แจกของในเกม)

10) Firebase Dynamic Links บริการ URL กลางที่สามารถชี้ทางไปยังเพจต่างๆ แปรผันตามอุปกรณ์หรือคุณสมบัติของผู้ใช้ (เช่น แต่ละประเทศคลิกเดียวกัน เข้าคนละเพจกัน)

11) Firebase Invites ระบบเชิญเพื่อนมาใช้แอป มีฟีเจอร์ referral คนชวนได้สิทธิประโยชน์

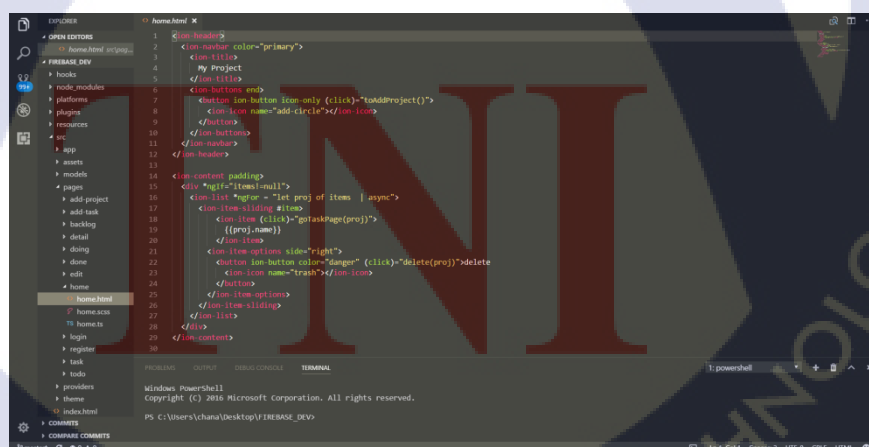
12) Firebase App Indexing เปลี่ยนชื่อมาจาก Google App Indexing ที่ช่วยให้ Google Search ค้นหาเนื้อหาภายในแอป

2.1.5 Visual Code Studio



ภาพที่ 2.7 แสดงถึงสัญลักษณ์ Visual Studio Code

Visual Studio Code คือเครื่องมือสำหรับนักพัฒนาโปรแกรม เครื่องมือชิ้นนี้จะตอบรับกับความต้องการระดับพื้นฐานอย่างเต็มรูปแบบ สามารถใช้งานได้บนวินโดวส์ แมค และลินุกซ์ ซึ่งทางไมโครซอฟท์ให้ใช้ฟรี โปรแกรมใช้งานง่ายและไม่ซับซ้อน มีความเป็นมืออาชีพรองรับมากกว่า 30 โปรแกรมภาษาอะไรบางอย่าง เช่น C++, C#, CSS, Dockerfile, HTML, JavaScript, JSON, Less, Markdown, PHP, Python, Sass, TypeScript ที่สำคัญรองรับภาษา Java ด้วย



ภาพที่ 2.8 Visual Code Studio

2.2 ภาษาที่ใช้ในการปฏิบัติงาน

2.2.1 Ionic Framework



ภาพที่ 2.9 สัญลักษณ์ Ionic Framework

ionic framework เป็นเครื่องมือที่ใช้ในการพัฒนา Mobile Application แบบ Hybrid หรือ "Hybrid App หรือ Hybrid Mobile App" คือสามารถพัฒนาแอปครั้งเดียวแล้วรันได้หลาย Platform เช่น Android , iOS ที่ได้รับความนิยมในปัจจุบัน ตัว ionic framework เองจะใช้เทคโนโลยีในการพัฒนาคือ HTML5, CSS3 และ JAVA Script ตัวการพัฒนาหลักๆเลยที่เราจะต้องใช้คือ JAVA Script เพราะว่า ionic framework ใช้ Core หลักเป็น Angular JS

ข้อดีที่เด่นๆของ ionic framework ที่เห็นได้ชัดคือมีความใกล้เคียงกับ Native App มาก ทั้ง หน้าตา UI การใช้งาน แต่ก็ไม่ถึงกับเหมือน Native App เลยซะทีเดียว ข้อดีอีกอย่างคือ ionic framework สามารถพัฒนาให้ติดต่อกับ Hardware ของอุปกรณ์ได้ด้วย เช่น กล้อง , ไมโครโฟน การตรวจสอบการเชื่อมต่ออินเทอร์เน็ต เป็นต้น โดยใช้งานร่วมกับ PhoneGap/Cordova

2.2.1.1 HTML5



ภาพที่ 2.10 สัญลักษณ์ HTML5

HTML5 คือ ภาษามาร์กอัพ ที่ใช้สำหรับเขียน website ซึ่ง HTML5 นี้เป็นภาษาที่ถูกพัฒนาต่อมาจากภาษา HTML และพัฒนาขึ้นมาโดย WHATWG (The Web Hypertext Application Technology Working Group) โดยได้มีการปรับปรุงเพิ่ม Feature หลายอย่างเข้ามาเพื่อให้ผู้พัฒนาสามารถใช้งานได้ง่ายมากยิ่งขึ้น

ข้อดีของ HTML5

1. เว็บไซต์ที่สร้างจากภาษา HTML5 สามารถแสดงผลได้กับทุก web browser
2. HTML5 จะช่วยลดการใช้พวกปลั๊กอินพิเศษอย่างพวก Adobe Flash, Microsoft Silverlight, Apache Pivot สนับสนุน วิดีโอ และ องค์ประกอบเสียง รวมทั้ง สื่อมัลติมีเดียต่าง ๆ มากขึ้น โดยไม่ต้องใช้ Flash
3. มีการจัดการข้อผิดพลาดที่ดีขึ้น
4. สคริปต์ใหม่ ที่จะมาแทนที่สคริปต์เดิม (เขียนโค้ดสั้นลง)
5. HTML5 มีความเป็นอิสระสูง (คล้ายๆ XML)
6. HTML5 ทำงานควบคู่กับ CSS3 ได้ดี ช่วยให้สามารถเพิ่มลูกเล่นต่าง ๆ บนเว็บไซต์ได้สวยงามมากยิ่งขึ้น (CSS คือ ส่วนแสดงผล ที่นักออกแบบสามารถกำหนดสี สัน ตำแหน่ง ลักษณะ เวลามาเมาส์ไปแตะแล้วมีกระด้างโผล่ออกมาจากโพรง หรือจับก้อนวัตถุในหน้าเว็บฯ ให้ชัดเจน ชัดขาว ส่วน CSS3 คือ เวอร์ชันที่ 3 ของ CSS)

2.2.1.2 CSS



ภาพที่ 2.11 สัญลักษณ์ CSS

CSS ย่อมาจาก Cascading Style Sheet มักเรียกโดยย่อว่า “สไคล์ชีท” คือภาษาที่ใช้เป็นส่วนของการจัดรูปแบบการแสดงผลเอกสาร HTML โดยที่ CSS กำหนดกฎเกณฑ์ในการระบุรูปแบบ (หรือ “Style”) ของเนื้อหาในเอกสาร อันได้แก่ สีของข้อความ สีพื้นหลัง ประเภทตัวอักษร และการจัดวางข้อความ ซึ่งการกำหนดรูปแบบ หรือ Style นี้ใช้หลักการของการแยกเนื้อหาเอกสาร HTML ออกจากคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผล กำหนดให้รูปแบบของการแสดงผลเอกสาร ไม่ขึ้นอยู่กับเนื้อหาของเอกสาร เพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลลัพท์ของเอกสาร HTML โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลงเนื้อหาเอกสารบ่อยครั้ง หรือต้องการควบคุมให้รูปแบบการแสดงผลเอกสาร HTML มีลักษณะของความสม่ำเสมอทั่วกันทุกหน้าเอกสารภายในเว็บไซต์เดียวกัน โดยกฎเกณฑ์ในการกำหนดรูปแบบ (Style) เอกสาร HTML ถูกเพิ่มเข้ามาครั้งแรกใน HTML 4.0 เมื่อปีพ.ศ. 2539 ในรูปแบบของ CSS level 1 Recommendations

2.2.1.3 AngularJS



ภาพที่ 2.12 สัญลักษณ์ AngularJS

AngularJS คือ JavaScript Framework รูปแบบหนึ่งที่พัฒนามาจาก Google หน้าที่ของมัน
คือเป็น engine ที่ใช้ควบคุมส่วน frontend ของเว็บได้เป็นอย่างดีมีการทำงานแบบ Model View
Controller (MVC)

บทที่ 3

แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

3.1 แผนงานการฝึกงาน

ตารางที่ 3.1 ตารางปฏิบัติงานในโครงการ

หัวข้องาน	เดือนที่ 1	เดือนที่ 2	เดือนที่ 3	เดือนที่ 4
ศึกษาเรื่อง Git	■			
ฝึกใช้งานคำสั่งต่างๆของ Git	■	■		
ศึกษา Web Service ที่ใช้รองรับไฟล์ Git (GitHub)		■		
ศึกษา Web Service ที่ใช้รองรับไฟล์ Git (AWSCodeCommit)		■		
นำโปรเจกต์ต่างๆของทีมขึ้นไปยัง AWSCodeCommit		■		
ศึกษาเรื่อง Firebase			■	
จัดทำแอปพลิเคชันที่ใช้เทคโนโลยีของ Firebase และ Git			■	■
ปรับปรุงและแก้ไขปัญหาที่เกิดขึ้นของ แอปพลิเคชัน			■	■
จัดทำเอกสารที่เกี่ยวข้อง				■

3.2 รายละเอียดที่นักศึกษาปฏิบัติในการฝึกงาน

3.2.1 การศึกษาเทคโนโลยีใหม่ๆ เพื่อนำมาสนับสนุนและประยุกต์ใช้ในการพัฒนาระบบได้

การปฏิบัติงานสหกิจศึกษา ข้าพเจ้าทำงานในตำแหน่ง Programmer ของแผน Business Solution จึงมีแผนการปฏิบัติงานไปทางด้านการศึกษาเทคโนโลยีใหม่ๆ เพื่อนำเทคโนโลยีเหล่านั้นมาใช้ในการช่วยเหลือหรือสนับสนุนคนในแผนกพัฒนาระบบได้อย่างมีประสิทธิภาพ

3.2.2 งานอื่นๆ

1) การพัฒนาแอปพลิเคชันรุ่นทดลองโดยใช้เทคโนโลยีของ Firebase และมี Git เป็นตัวควบคุมเวอร์ชันของแอปพลิเคชัน

3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงาน

3.3.1 ศึกษาเรื่อง Git

ศึกษาข้อมูลที่เป็นจำเป็นอย่างก่อนการใช้ Git เป็นตัวควบคุมเวอร์ชันอย่างละเอียดในด้านของการทำงานของ Git, รูปแบบการทำงานและคำสั่งที่จำเป็น โดยศึกษาจากเอกสารของผู้พัฒนาที่ได้นำเอกสารโพลลงบนฟอรัมตามเว็บไซต์ต่างๆ

3.3.2 ฝึกใช้งานคำสั่งต่างๆ Git

ฝึกใช้งานคำสั่งของ Git ตั้งแต่การเพิ่มไฟล์ Git เข้าไปในชิ้นงาน วิธีในการ commit โปรเจกต์ลงบนเครื่อง Local การเชื่อมต่อไปยัง Web service, วิธีการนำไฟล์ Git ขึ้นไปยัง Web service, การโคลนนิ่งไฟล์ Git จาก Web service ลงมาบนเครื่อง Local ต่างๆ, การสร้าง Branch เพื่อนำมาพัฒนากรณีที่ต้องการทดลองพัฒนาแอปพลิเคชันในรูปแบบต่างๆ และรวมถึงวิธีแก้ไขปัญหาต่างๆที่เกิดขึ้นกับ Git

3.3.3 ศึกษา Web Service ที่ใช้รองรับไฟล์ Git (GitHub)

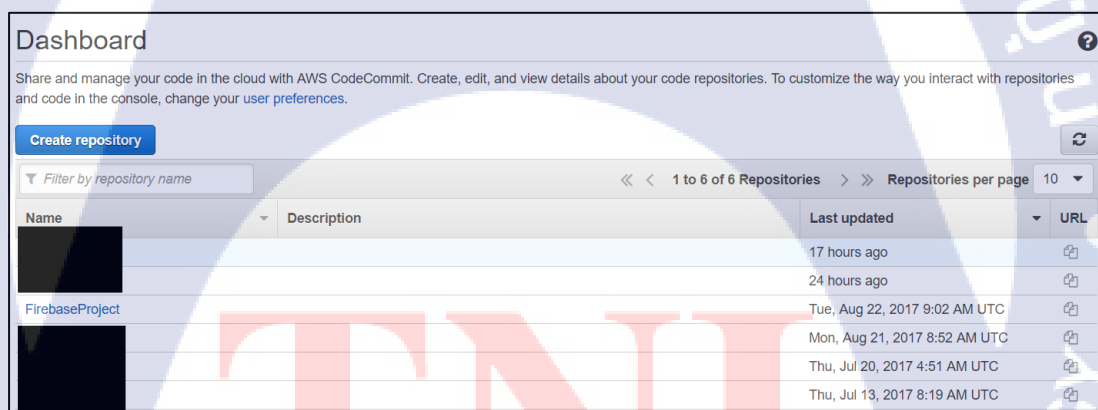
ศึกษาวิธีการทำงานต่างๆของ Web service ที่ชื่อ GitHub เพื่อนำไปใช้ในการฝากไฟล์ Git รวมไปถึง feature ต่างๆบน GitHub รวมถึงวิธีการ Forking ไฟล์ Git บน GitHub และการทำ Pull Request

3.3.4 ศึกษา Web Service ที่ใช้รองรับไฟล์ Git (AWSCodeCommit)

ศึกษาวิธีโครงสร้างและ feature ต่างๆบน AWSCodecommit เพื่อการนำมาใช้ในการนำโปรเจกต์ต่างๆของทีมพัฒนาอัพโหลดขึ้นไปยัง AWSCodeCommit รวมไปถึงการศึกษากการสร้าง IAM User เพื่อใช้ในการใช้งานคำสั่ง Git ต่างๆ, การสร้างข้อมูลของ Git สำหรับการเชื่อมต่อแบบ HTTPS ไปยัง AWSCodeCommit, การสร้างและการโคลน Repository, การสร้าง Repository และ IAM Policies ให้กับ Group ของ User, การตั้งค่า Repository และรวมไปถึงการเปรียบเทียบของแต่ละ Commit บน AWSCodecommit

3.3.5 นำโปรเจกต์ต่างๆของทีมขึ้นไปยัง AWSCodeCommit

การใช้คำสั่ง Git เพื่อนำไฟล์โปรเจกต์ของทีมผู้พัฒนาอัพโหลดขึ้นไปยัง AWSCodeCommit



ภาพที่ 3.1 แสดงถึงโปรเจกต์บน Web service AWSCodeCommit

3.3.6 ศึกษาเรื่อง Firebase

ศึกษาและเรียนรู้บริการต่างๆบน Firebase ซึ่งมี 3 อันหลักที่ได้ศึกษาได้แก่ Authentication, Real-time Database และ Storage นำมาพัฒนาร่วมกับ Mobile Application แบบ Real-time เพื่อเกิดประโยชน์สูงสุดแก่ทีมผู้พัฒนา

3.3.7 จัดทำแอปพลิเคชันใช้เทคโนโลยีของ Firebase และ Git

ได้มีการจัดการทำแอปพลิเคชันรุ่นทองเกี่ยวกับ To-do List Application หรือก็คือ แอปพลิเคชันที่ใช้จัดการหน้าที่ที่ได้รับมอบหมายในแต่ละวัน โดยใช้เทคโนโลยีและบริการของ Firebase ในการทำอีกทั้งยังใช้ Git เป็นตัวควบคุมเวอร์ชันเพื่อสนับสนุนการพัฒนาตัวแอปพลิเคชันร่วมกันเป็นทีมในภายภาคหน้า

3.3.8 ปรับปรุงและแก้ไขปัญหาที่เกิดขึ้นของแอปพลิเคชัน

ปรับปรุงและแก้ไขปัญหาต่างๆที่เกิดขึ้น เช่น task งานของแต่ละวันไม่อัปเดตไปตามชนิดของงานหรือไม่อัปเดตบน Firebase Real-time Database การเกิดบัคต่างๆบนแอปพลิเคชันเป็นต้น

3.3.9 จัดทำเอกสารที่เกี่ยวข้อง

จัดทำเอกสารการใช้งานเทคโนโลยีต่างๆที่ได้ศึกษาเพื่อใช้เป็นเอกสารการใช้งานให้แก่บริษัท

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

บทที่ 4

สรุปผลการดำเนินงาน การวิเคราะห์และสรุปผลต่าง ๆ

4.1 ขั้นตอนและผลการดำเนินงาน

ในช่วงระยะเวลาหนึ่งเดือนแรกของการปฏิบัติงานสหกิจศึกษา จะเป็นช่วงของการศึกษาข้อมูลที่เป็นสำหรับ Git ก่อนจะนำไปใช้ในการฝากไฟล์โปรเจกต์ขึ้นไปยัง Web Service และต่อมาได้มีการศึกษา Web Service 2 อย่างและได้นำมาเปรียบเทียบกัน ซึ่งได้ข้อสรุปว่าใช้ Web Service ของ AWSCodeCommit เพราะสามารถสร้างเป็น repository (ไฟล์ Git) แบบ private แต่ในส่วน of GitHub ต้องมีค่าบริการจึงสามารถสร้าง repository แบบ private ได้ และได้นำไฟล์โปรเจกต์งานต่างๆ ของทีมพัฒนาอัพโหลดขึ้นไปยัง Web Service เพื่อความสะดวกในการพัฒนางานเป็นทีม ต่อมาได้มีศึกษาเพิ่มเติมเกี่ยวกับบริการต่างๆบน Firebase จากนั้นได้นำเทคโนโลยีของ Git และ Firebase มาทำเป็นแอปพลิเคชันรุ่นทดลอง ที่มีชื่อแอปพลิเคชันว่า To-do List Application เป็นแอปพลิเคชันที่ช่วยจัดการงานของแต่ละคนในหนึ่งอาทิตย์ ซึ่งปกติแผนกจะมีการจัดการประชุมในทุกเช้าของทุกวัน ซึ่งจะมีการวางแผนงานในแต่ละอาทิตย์และบันทึกงานของแต่ละคนลงบนไวท์บอร์ดของแผนก แต่แอปพลิเคชันตัวนี้ได้มีการอำนวยความสะดวกให้ทีมผู้พัฒนาได้เป็นอย่างดี

โดยหลักๆแล้วจะแบ่งการทำงานหลักๆออกเป็น 5 ขั้นตอนคือ

4.1.1 การศึกษาข้อมูลเกี่ยวกับ Git

4.1.2 การศึกษา Web Service ที่ให้บริการฝากไฟล์ Git

4.1.3 การอัพโหลดไฟล์โปรเจกต์ขึ้นไปยัง Web Service ที่ให้บริการฝากไฟล์ Git

4.1.4 การศึกษาข้อมูลเกี่ยวกับ Firebase

4.1.5 การทำแอปพลิเคชันที่ใช้บริการของ Firebase โดยใช้ Git เป็นตัวควบคุมเวอร์ชัน

4.1.6 จัดทำเอกสารที่เกี่ยวข้อง

4.1.1 การศึกษาเกี่ยวกับ Git

เริ่มต้นด้วยการศึกษาความหมายของ Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจกต์ มีการ backup code สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชันต่างๆของโปรเจกต์ที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆถูกเพิ่มหรือ

แก้ไข ดังนั้น Version Control เหมาะอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียวโดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม

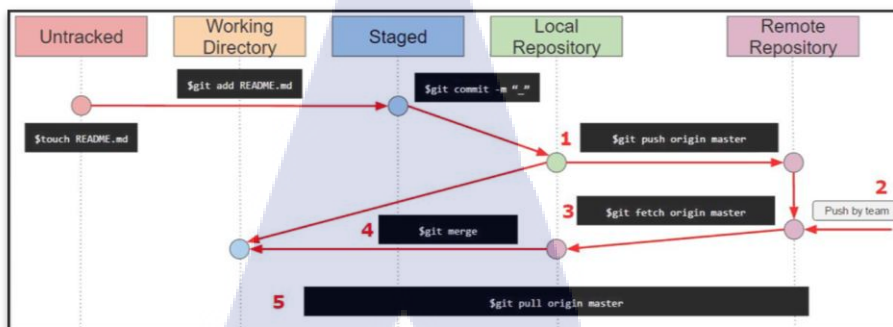
4.1.1.1 ความสำคัญของการใช้ Git

Track version ของ Source Code ย้อนกลับได้ เมื่อจัดเก็บไฟล์เข้าไปในระบบของ Git จะเรียกว่า Git Repository ซึ่งเก็บสำรองข้อมูลและการเปลี่ยนแปลงของ Source Code ทำให้สามารถย้อนกลับไปเวอร์ชันใดๆ ก่อนหน้า และดูรายละเอียดการเปลี่ยนแปลงของแต่ละเวอร์ชันได้นอกจากนั้นสามารถดูได้ว่าในทีมพัฒนานั้นใครเป็นคนแก้ไข ช่วยในการพัฒนาซอฟต์แวร์เป็นทีม Git สามารถเก็บบันทึกการเปลี่ยนแปลงของ Source Code เวอร์ชันล่าสุดไว้ที่ Local Repository ซึ่งสามารถทำงานได้โดยไม่ต้องต่อกับอินเทอร์เน็ต และเมื่อต้อง Update การเปลี่ยนแปลงของ Source Code เวอร์ชันล่าสุดให้กับเพื่อนร่วมทีมสามารถที่จะ Push ขึ้นไปเก็บที่ Remote Repository (Git Hosting) และเพื่อนร่วมทีมสามารถ Pull เวอร์ชันล่าสุดนั้นมารวม(Auto Merge) ที่เครื่องอื่นๆ ทำให้ Source Code ที่พัฒนาร่วมกันกับคนภายในทีมเป็นเวอร์ชันล่าสุดเสมอ

4.1.1.2 Git Status

สถานะของ Source Code ที่อยู่ในระบบของ Git มีดังนี้

- 1) Untracked เป็นสถานะที่ Source Code ถูกเพิ่มเข้ามาใหม่และยังไม่ได้ถูกเก็บไว้ในระบบของ Git
- 2) Working Directory เป็นสถานะที่กำลังมีการเปลี่ยนแปลงหรือแก้ไข Source Code หรืออาจจะเรียกสถานะนี้ว่า Modified
- 3) Staged เป็นสถานะที่ Source Code กำลังเตรียมที่จะ Commit เพื่อยืนยันการเปลี่ยนแปลงก่อนที่จะเก็บลงในสถานะ Local Repository
- 4) Local Repository เป็นสถานะที่มีการเก็บบันทึกข้อมูลการเปลี่ยนแปลงของ Source Code ลงไปที่ Git Repository ที่เป็น Local (ที่เครื่องตัวเอง)
- 5) Remote Repository เป็นสถานะที่มีการเก็บบันทึกข้อมูลการเปลี่ยนแปลงของ Source Code ลงไปที่ Git Repository ที่เป็น Hosting (ที่เครื่องเซิร์ฟเวอร์)



ภาพที่ 4.1 Workflow of Git

4.1.1.3 ติดตั้ง Git

Git สามารถที่จะติดตั้งได้ทั้ง Windows, Mac OS X, Linux ซึ่ง Download ได้จาก <https://git-scm.com> เมื่อติดตั้งเสร็จแล้วคือทำการ setup ชื่อและอีเมลล์สำหรับการใช้งาน git โดยใช้คำสั่งดังรูปข้างต้น

```
git config --global user.name "YOURNAME"
git config --global user.email "your@email.com"
```

ภาพที่ 4.2 คำสั่งที่ใช้ในการ setup ชื่อและอีเมลล์ของ Git

4.1.1.4 Git Command Line

1) Git Init เป็นคำสั่งที่ใช้สร้างระบบของ Git ขึ้นมาภายใต้โฟลเดอร์หรือ Path นั้น โดยจะสร้างโฟลเดอร์ .git ขึ้นมาเพื่อใช้เก็บ สำรองข้อมูล การเปลี่ยนแปลงและคุณสมบัติอื่นๆ ของ Git

```
C:\Users\chana\Desktop\gittest\TemplateApp>git init
Reinitialized existing Git repository in C:/Users/chana/Desktop/gittest/TemplateApp/.git/
```

ภาพที่ 4.3 คำสั่ง git init

2) Git Status เป็นคำสั่งที่ใช้ตรวจสอบสถานะของ source code ในระบบของ Git ซึ่งจะแสดงสถานะดังที่อธิบายไว้ในข้อ 4.1.1.2

```
C:\Users\chana\Desktop\gittest\TemplateApp>git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to track)

        .editorconfig
        .gitignore
        .idea/
        README.md
        config.xml
        hooks/
        ionic.config.json
        node_modules/
        package.json
        resources/
        src/
        tsconfig.json
        tslint.json
        www/

nothing added to commit but untracked files present (use "git add" to track)
```

ไฟล์ที่ยังไม่ได้ถูก track (สีแดง)

ภาพที่ 4.4 คำสั่ง git status

- 3) Git Add เป็นคำสั่งที่ใช้เพิ่มการเปลี่ยนแปลงของ Source Code เข้าไปสู่สถานะ Staged

```
C:\Users\chana\Desktop\gittest\TemplateApp>git add .
warning: LF will be replaced by CRLF in .editorconfig.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in config.xml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in ionic.config.json.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in package.json.
```

ภาพที่ 4.5 คำสั่ง git add

- 4) Git Commit เป็นคำสั่งที่ใช้ยืนยัน source code ที่อยู่ในสถานะ staged เข้าไปเก็บไว้ที่

Local Repository

```
C:\Users\chana\Desktop\gittest\TemplateApp>git commit -m "Initial Committed."
[master (root-commit) 55e0e50] Initial Committed.
112 files changed, 1199 insertions(+)
create mode 100644 .editorconfig
create mode 100644 .gitignore
create mode 100644 README.md
create mode 100644 config.xml
create mode 100644 ionic.config.json
create mode 100644 package.json
create mode 100644 resources/android/icon/drawable-hdpi-icon.png
create mode 100644 resources/android/icon/drawable-ldpi-icon.png
create mode 100644 resources/android/icon/drawable-mdpi-icon.png
create mode 100644 resources/android/icon/drawable-xhdpi-icon.png
create mode 100644 resources/android/icon/drawable-xxhdpi-icon.png
create mode 100644 resources/android/icon/drawable-xxxhdpi-icon.png
```

ภาพที่ 4.6 คำสั่ง git commit

5) Git log เป็นคำสั่งที่ใช้แสดงประวัติของ Commit ที่เก็บไว้ใน Repository

```
C:\Users\chana\Desktop\gittest\TemplateApp>git log
commit 4247a71629fc204ede25f1368149da5b60b5799c (HEAD -> master)
Author: Chanasit <yo.chanasit_st@tni.ac.th>
Date: Mon Jun 12 14:13:22 2017 +0700

    Already commit
```

ภาพที่ 4.7 คำสั่ง git log

6) Git Branch เป็นคำสั่งที่ใช้ในแสดงและแตกกิ่งสาขาในการพัฒนา ซึ่งทำให้การพัฒนาซอฟต์แวร์มีความยืดหยุ่นมากขึ้น

```
C:\Users\chana\Desktop\gittest\TemplateApp>git branch test3
C:\Users\chana\Desktop\gittest\TemplateApp>git branch
* master
test3
```

ภาพที่ 4.8 คำสั่งที่ใช้ในการสร้าง branch และเช็คว่ามีกี่ branch



ภาพที่ 4.9 Branching Workflow

7) Git Checkout เป็นคำสั่งที่ใช้สลับ Working Directory ไปยัง Branch หรือ commit ที่เราระบุ คำสั่งนี้ยังสามารถใช้งานได้อีกหลายได้อีกหลายๆแบบ

```
C:\Users\chana\Desktop\gittest\TemplateApp>git checkout test3
Switched to branch 'test3'
```

ภาพที่ 4.10 คำสั่ง git checkout กรณีที่ใช้ในการเข้า branch

```
C:\Users\chana\Desktop\gittest\TemplateApp>git checkout README.md
C:\Users\chana\Desktop\gittest\TemplateApp>
```

ภาพที่ 4.11 คำสั่ง git checkout กรณีไฟล์หายหรือเผลอลบ

- 8) Git Merge เป็นคำสั่งที่ใช้ในการรวม Branch หรือ commit ทั้งสองเข้าด้วยกัน

```
C:\Users\chana\Desktop\gittest\TemplateApp>git checkout master
Switched to branch 'master'

C:\Users\chana\Desktop\gittest\TemplateApp>git merge test3
Updating 4247a71..94fd704
Fast-forward
Auto packing the repository in background for optimum performance.
See "git help gc" for manual housekeeping.
Counting objects: 17721, done.
```

ภาพที่ 4.12 คำสั่ง git merge

- 9) Git Remote (เริ่มต้นการทำงานกับ Git Hosting) เพิ่ม URL ของ Remote Repository เข้าไปยังคุณสมบัติของ Git โดยชื่อว่า origin ส่วนใหญ่จะเป็นชื่อ Default ที่หลายๆคนเข้าใจตรงกัน แต่สามารถตั้งชื่ออื่นๆได้

```
C:\Users\chana\Desktop\gittest\TemplateApp>git remote add origin https://github.com/chanasityordsri/
```

ภาพที่ 4.13 คำสั่ง git remote

- 10) Git Push เป็นคำสั่งที่ใช้ส่งการเปลี่ยนแปลงของ source code ที่เก็บอยู่บน Local Repository ขึ้นไปยัง Remote Repository

```
C:\Users\chana\Desktop\gittest\TemplateApp>git push origin master
Counting objects: 136, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (122/122), done.
Writing objects: 100% (136/136), 1.60 MiB | 58.00 KiB/s, done.
Total 136 (delta 23), reused 0 (delta 0)
remote: Resolving deltas: 100% (23/23), done.
To https://github.com/chanasityordsri/Test2.git
* [new branch] master -> master
```

ภาพที่ 4.14 คำสั่ง git push

11) Git Fetch เป็นคำสั่งที่ใช้รับการเปลี่ยนแปลงบน source code ล่าสุดที่อยู่บน Remote Repository ลงมายัง Local Repository แต่ยังไม่ได้ทำการรวม source code

```
C:\Users\chana\Desktop\gittest\TemplateApp>git fetch
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 2), reused 3 (delta 2), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/chanasityordsri/Test2
55e0e50..ee4cb96 master -> origin/master
```

ภาพที่ 4.15 คำสั่ง git fetch

12) Git Pull (Fetch + Merge) เป็นคำสั่งที่ใช้รับการเปลี่ยนแปลงของ source code ล่าสุดที่อยู่บน Remote Repository ลงมายัง Local Repository และทำการ Auto Merge โดยมีคำสั่งว่า git pull origin master

13) Git Clone เป็นคำสั่งที่ใช้ดึงประวัติทั้งหมดบน Remote Repository ของทีมผู้พัฒนาของคนอื่นหรือของเราเองที่มีอยู่แล้วบน Git Hosting มาที่เครื่องของเรา คำสั่งนี้จะคล้ายๆ Git Init ที่ใช้สร้างระบบ Git ขึ้นมาตอนเริ่มต้น แต่เราจะได้ประวัติเดิมของ Repository มาด้วย ทำให้เราเริ่มพัฒนาต่อจากตรงจุดนี้ได้เลย

```
C:\Users\chana\Desktop\gittest>git clone https://github.com/chanasityordsri/Test2.git TemplateApp2
Cloning into 'TemplateApp2'...
remote: Counting objects: 136, done.
remote: Compressing objects: 100% (99/99), done.
remote: Total 136 (delta 23), reused 136 (delta 23), pack-reused 0
B/s
Receiving objects: 100% (136/136), 1.60 MiB | 206.00 KiB/s, done.
Resolving deltas: 100% (23/23), done.
```

ภาพที่ 4.16 คำสั่ง git clone

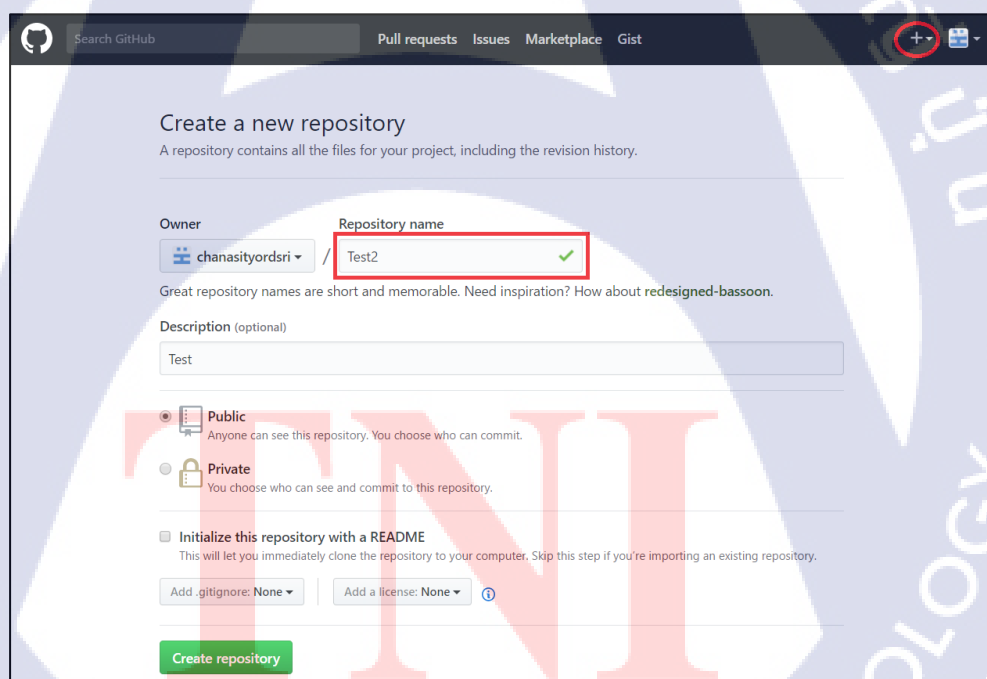
4.1.2 การศึกษา Web Service ที่ให้บริการฝากไฟล์ Git

4.1.2.1 GitHub

GitHub เป็นเว็บเซิร์ฟเวอร์ที่ให้บริการในการฝากไฟล์ Git (ทั่วโลกมักนิยมใช้ในการเก็บโปรเจกต์ Open Source ต่างๆ ที่ต่างๆ ไม่ว่าจะเป็น Bootstrap, Rails, Node.js, Angular เป็นต้น) ขั้นตอนแรกให้ทำการสมัครสมาชิกกับ Github จากนั้นเมื่อสมัครสมาชิกแล้ว เราสามารถที่จะสร้าง Repository ของเราแล้วเอาไปฝากไว้บน Github ได้ (การทำงานทุกอย่างก็เหมือนตอนทำ local) จุดเด่นของ Github คือใช้ฟรี และสร้าง repository ได้ไม่จำกัด แต่ต้องเป็น public repository เท่านั้น หากอยากใช้แบบ private ก็ต้องมีการเสียค่าบริการ

1) การสร้าง Repository บน GitHub

เมื่อสมัครสมาชิกกับ GitHub เสร็จ สามารถสร้าง Repository ได้โดยกดที่รูป + ตรงบนขวามือและเลือก “New Repository” จากนั้นใส่ชื่อของ Repository



ภาพที่ 4.17 การสร้าง Repository บน GitHub

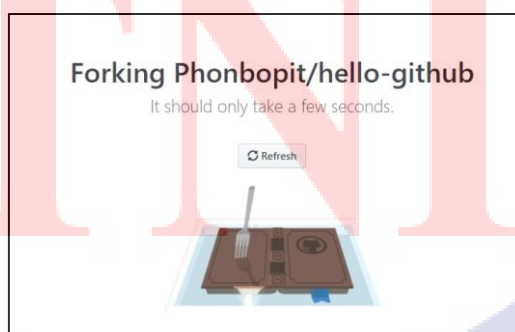


ภาพที่ 4.18 url ของ remote ใช้เชื่อมต่อฝั่ง Local และ Server

2) Feature ต่างๆบน GitHub

- Forking

Fork เป็น feature ของทาง Github เอาไว้สำหรับ copy โปรเจกหรือว่า clone โปรเจกอื่นมาเป็นของเรา เพียงแต่การ clone จะเกิดขึ้นบน remote(server) เท่านั้น ข้อดีของการ Fork ก็คือ กรณีที่เจอโปรเจกที่น่าสนใจ แล้วอยากจะช่วยพัฒนา สามารถที่จะ Fork มาเป็นของตนเอง จากนั้น clone มาที่ local เมื่อแก้ไขโค้ดเสร็จ ก็ทำการ Push โค้ดขึ้นไป ที่นี้ก็จะทำการ Pull Request เพื่อ merge โค้ด เมื่อการ Forking สำเร็จแล้ว โปรเจกก็จะมาอยู่ที่ account ของตนเอง และทำการเปลี่ยน url ให้แต่จะบอกว่าเรา fork มาจากที่ใด



ภาพที่ 4.19 การ Forking

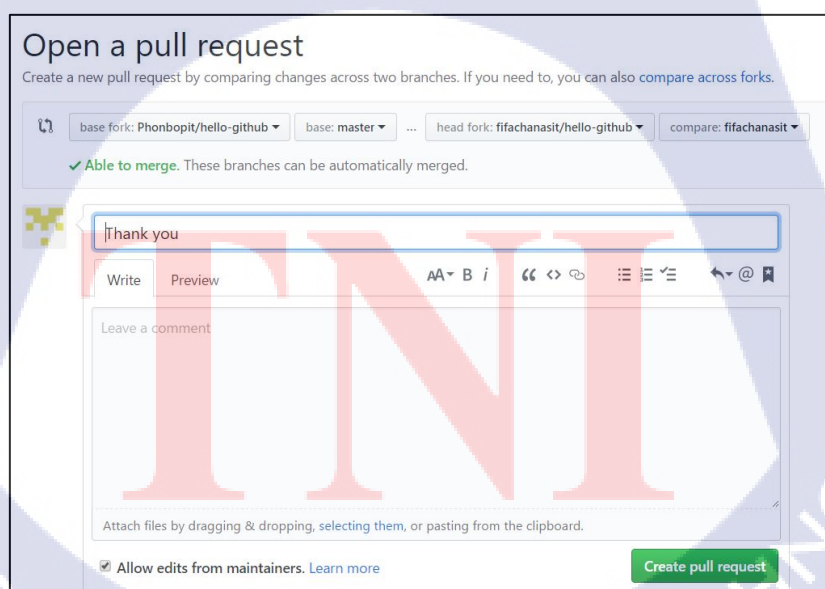
- Pull Request

Pull Request เป็น feature ที่ช่วยให้นักพัฒนาคนอื่นๆ ที่ไม่ใช่เจ้าของ repository สามารถที่จะร่วมกันส่งโค้ดเข้ามา merge รวมกับโค้ดต้นฉบับได้ โดยการใช้งาน Pull Request ต้องทำการ Fork โปรเจกต์มาก่อน ตัวอย่างการ Pull Request ในกรณีที่ทำการแก้ไขโปรเจกต์ที่ Fork แล้วทำการ commit และ push โค้ดขึ้นมา จะเห็นปุ่ม Pull Request ขึ้นมาแสดงว่าเราต้องการ Pull Request หรือไม่



ภาพที่ 4.20 ตัวอย่างการทำ Pull Request

เมื่อกดเข้าไปจะเจอหน้า Pull Request เพื่อให้ใส่รายละเอียด มีการเปรียบเทียบระหว่าง Original Repository และตัว Fork ของเรา รวมถึงเปรียบเทียบและให้เลือก branch ที่ต้องการ Pull Request ก็ได้ เพื่อ Pull Request ไปแล้ว ก็แค่ออให้ทางเจ้าของ Repository รออนุมัติว่าจะ merge โค้ดหรือไม่

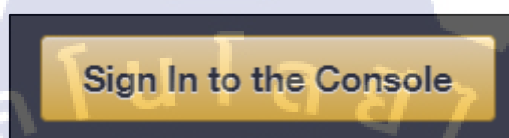


ภาพที่ 4.21 หน้า Pull Request

4.1.2.2 AWSCodeCommit

AWSCodeCommit เป็น web server อีกตัวหนึ่ง ซึ่งให้บริการฝากไฟล์ Git เช่นเดียวกับ GitHub แต่สามารถสร้าง repository แบบ private ได้โดยไม่เสียค่าบริการ โดยในภายหลังได้ย้าย web server จาก GitHub มาใช้ AWSCodeCommit แทน

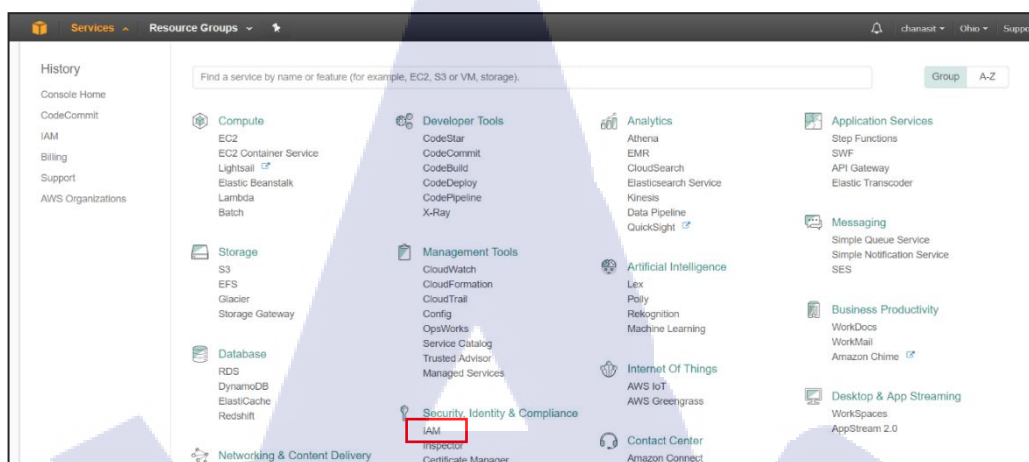
- 1) การสมัคร AWS account และ IAM User เพื่อเชื่อมต่อกับ Git Credentials
- สมัคร AWS account โดยไปที่ <https://aws.amazon.com> กรอกข้อมูลให้ครบถ้วน



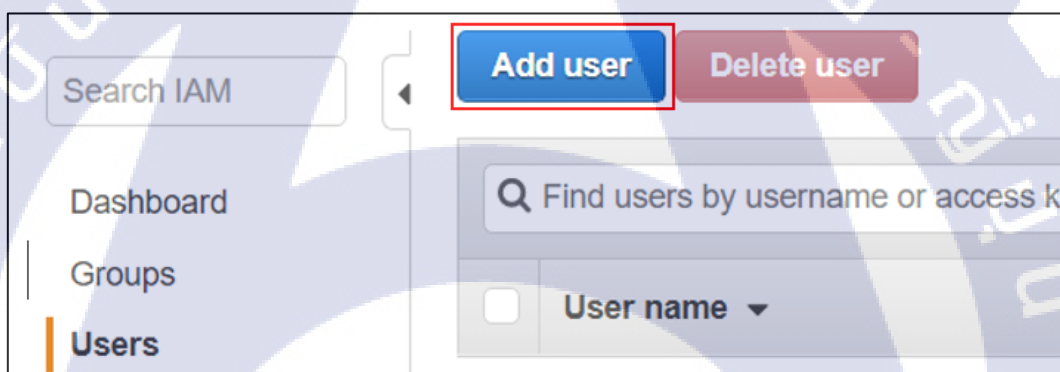
ภาพที่ 4.22 ปุ่ม Sign In เข้าไปใน Console

ภาพที่ 4.23 หน้า Sign In

- เข้าไปสร้าง IAM user โดยไปที่หน้า IAM Dashboard โดยไปที่ Services แล้วเลือกคำว่า IAM จากนั้นเลือกคำว่า Users ตรงแถบด้านข้างซ้าย และกด Add User

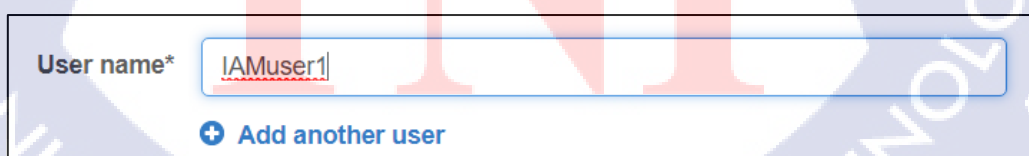


ภาพที่ 4.24 หน้า Service บน AWSCodeCommit



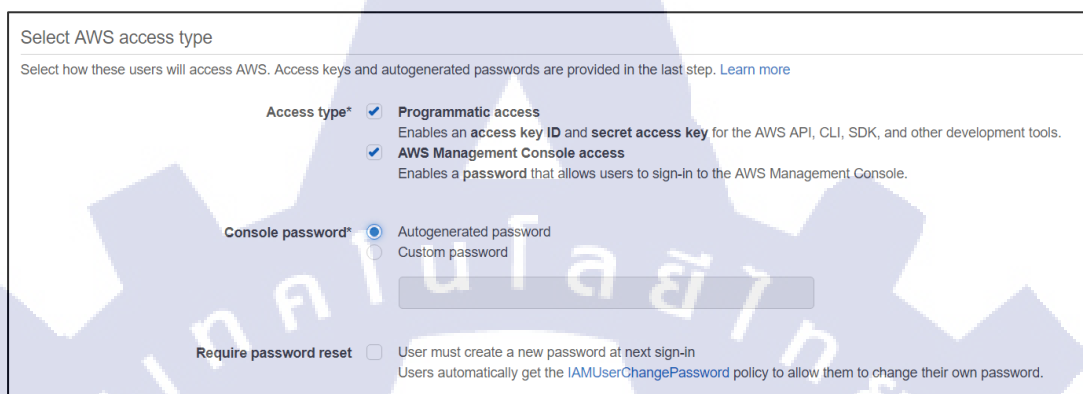
ภาพที่ 4.25 ปุ่ม Add User บนหน้า IAM Dashboard

- กรอกชื่อ User name



ภาพที่ 4.26 ชื่อ IAM User

- ในส่วนประเภทการเข้าถึง ทำการตั้งค่าตามภาพ ตรง console password หากตั้งตามภาพ ระบบจะทำการสุ่ม password มาให้ แต่หากเลือก custom password จะเป็นการกำหนด password เอง แล้วทำการกดปุ่ม Next: Permissions



Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* ☒ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

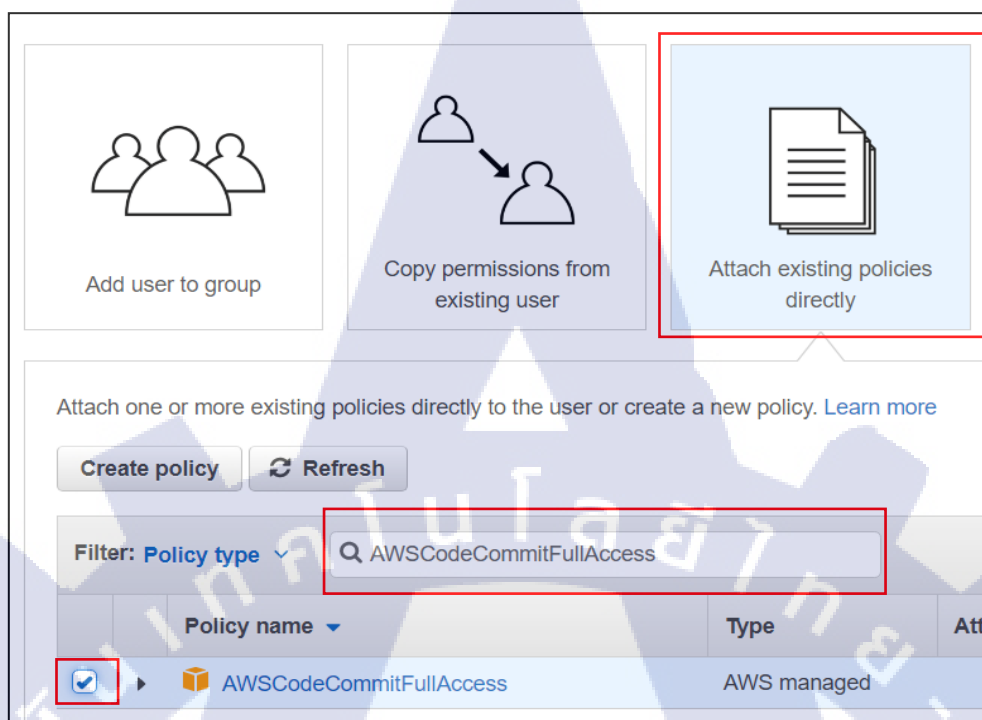
☒ **AWS Management Console access**
Enables a password that allows users to sign-in to the AWS Management Console.

Console password* ☒ Autogenerated password
☐ Custom password

Require password reset ☐ User must create a new password at next sign-in
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

ภาพที่ 4.27 หน้าประเภทการเข้าถึงของ IAM User

- มาที่หน้า Permission ให้เลือก Attach existing policies directly เพื่อกำหนดว่า IAM user นี้สามารถเข้าถึงอะไรได้บ้าง ในช่อง search ให้หาคำว่า AWSCodeCommitFullAccess และเลือกช่องเครื่องหมายถูกด้านหน้า



ภาพที่ 4.28 หน้า Permission ของ IAM User

- หากใช้ Git เพื่อเชื่อมต่อกับ AWS CodeCommit ให้เพิ่ม IAMSelfManageServiceSpecificCredentials และ IAMReadOnlyAccess
- หากใช้ SSH เพื่อเชื่อมต่อกับ AWS CodeCommit ให้เพิ่ม IAMUserSSHKeys และ IAMReadOnlyAccess ด้วย
- หลังจากทำการ Add policy เสร็จเรียบร้อยแล้วให้ทำการ Next: Review เมื่ออยู่ในหน้า Review ให้ทำการตรวจสอบว่าใส่ policy ถูกหรือไม่ หากถูกต้องกด Create user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	IAMUser1
AWS access type	Programmatic access and AWS Management Console access
Console password type	Custom
Require password reset	Yes

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AWSCodeCommitFullAccess
Managed policy	IAMUserChangePassword

Cancel Previous **Create user**

ภาพที่ 4.29 หน้า Review

- วิธีดู password หากเลือก **Autogenerated password** ให้ save ไว้ดังภาพ
(สามารถ download เป็นไฟล์ .csv ได้)

Add user

1 Details 2 Permissions 3 Review 4 **Complete**

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

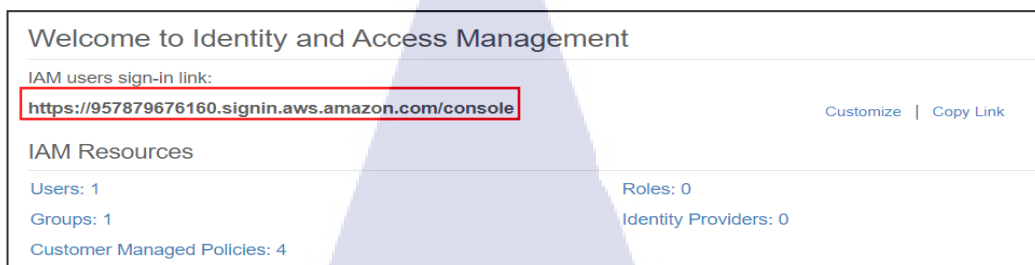
Users with AWS Management Console access can sign-in at: <https://957879676160.signin.aws.amazon.com/console>

Download .csv

User	Access key ID	Secret access key	Password	Email login instructions
mcocwpofkewofew	AKIAJU5GYVVUHAWOU45Q	***** Show	4*W4E+hiBxx Hide	Send email

ภาพที่ 4.30 ทำการ Add IAM User สำเร็จ

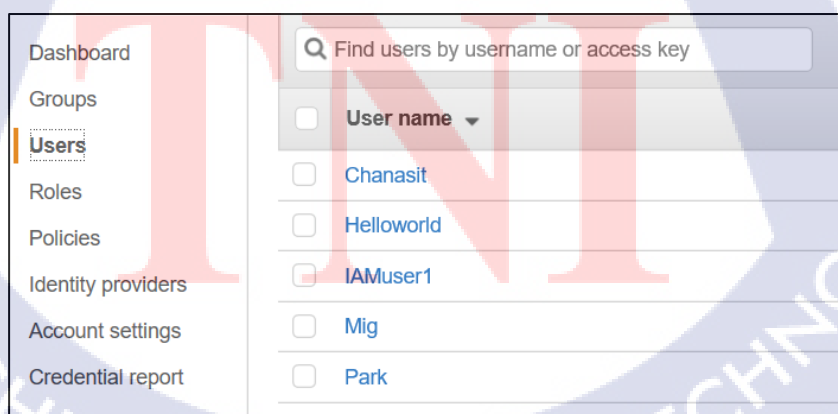
- สร้างข้อมูลของ Git สำหรับการเชื่อมต่อแบบ HTTPS ไปยัง AWSCodeCommit
- เข้าสู่ระบบด้วย IAM user โดยไปดู link login หน้า IAM Dashboard ที่
<https://console.aws.amazon.com/iam/>



ภาพที่ 4.31 link login หน้า Dashboard

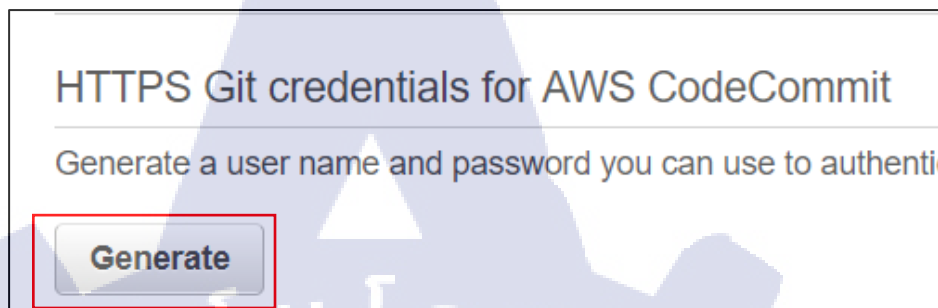
ภาพที่ 4.32 หน้า login IAM User

- บนหน้า Dashboard แถบด้านซ้ายให้เลือกหัวข้อ Users จากนั้นเลือก IAM user ที่ต้องการ



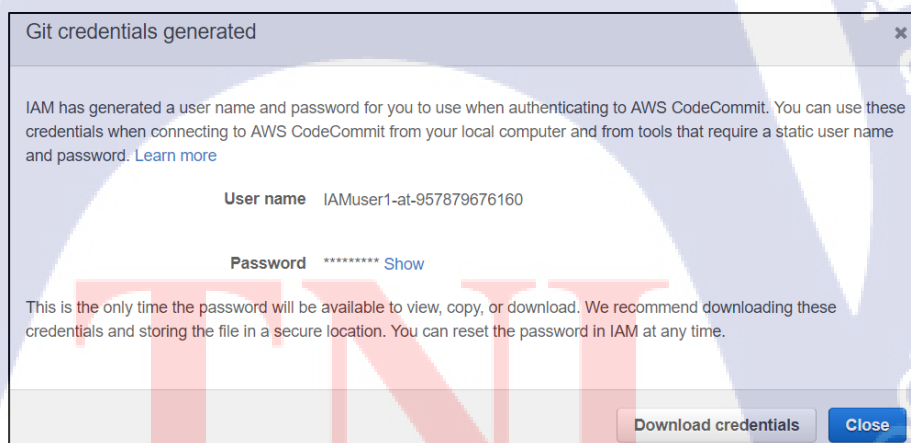
ภาพที่ 4.33 หน้า User บน IAM Dashboard

- บนหน้า Summary ให้เลือกหัวข้อ Security Credential จากนั้นเลื่อนลงมาจะเจอหัวข้อ HTTPS Git Credential for AWS CodeCommit เลือก Generate



ภาพที่ 4.34 ปุ่ม Generate เพื่อค่า Credentials

- ให้ทำการกดปุ่ม Download credentials เพื่อทำการ download username และ password เก็บไว้ (ไฟล์จะอยู่ในรูปแบบ .csv)



ภาพที่ 4.35 Git credentials generated

- Username และ password นี้จะใช้ในการเชื่อมต่อกับ awscodecommit เพื่อ push ไฟล์ขึ้นไปยัง repository บน awscodecommit และหากไม่ download เก็บไว้ จะไม่สามารถดู password ของ

git credential ได้ต้องการทำการรีเซ็ต password iam user ใหม่ แต่จะสามารถดู username บนหน้า Dashboard ได้

```
C:\Users\Admin\Desktop\push>git push origin master
Username for 'https://git-codecommit.us-east-2.amazonaws.com': IAMuser1-at-957879676160
Password for 'https://IAMuser1-at-957879676160@git-codecommit.us-east-2.amazonaws.com':
```

ภาพที่ 4.36 การ push ไฟล์ขึ้นไปยัง AWSCodeCommit

3) การสร้าง AWSCodeCommit Repository และการ Clone Repository

- ทำการ logout จาก IAM user และเข้าสู่ระบบด้วย root account ที่

<https://console.aws.amazon.com/codecommit> จากนั้นกด Services และเข้าที่ CodeCommit



amazon web services

Sign In or Create an AWS Account

What is your email (phone for mobile accounts)?

E-mail or mobile number:

fifa.ne@hotmail.com

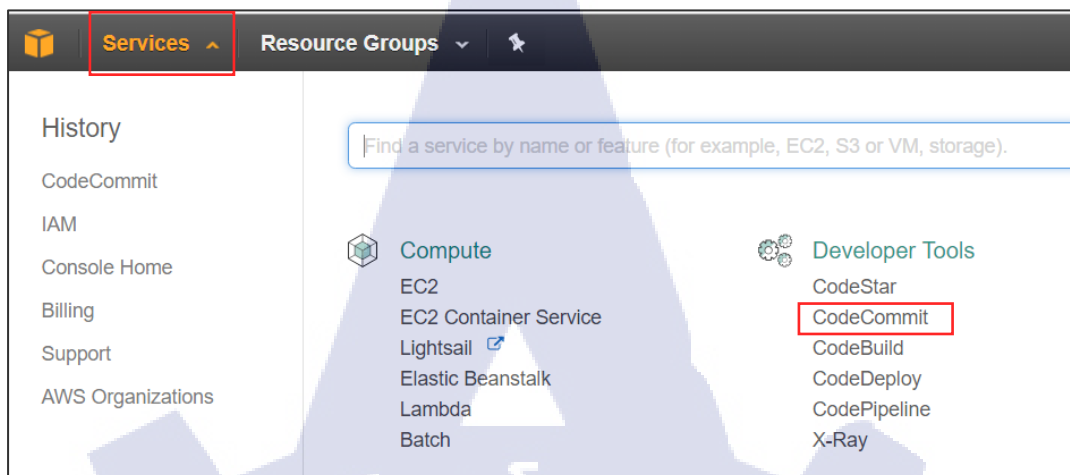
☐ I am a new user.

☒ I am a returning user and my password is:

.....

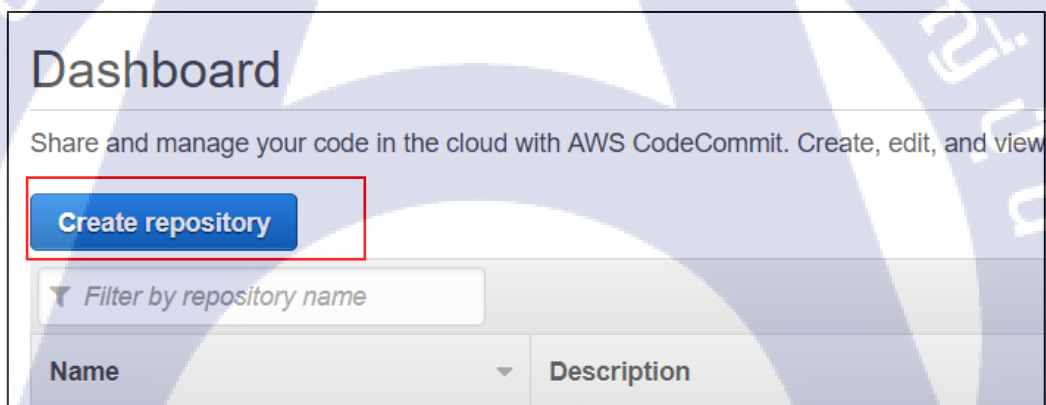
Sign in using our secure server

ภาพที่ 4.37 Login with Root User



ภาพที่ 4.38 AWSCodeCommit Dashboard

- บนหน้า Dashboard เลือก Create repository



ภาพที่ 4.39 Create Repository หน้า Dashboard

- ในหน้า Create repository ให้ใส่ชื่อ Repository ลงในช่อง Repository name และสามารถใส่คำอธิบายให้ Repository ได้ในช่อง Description หากใส่ทุกอย่างถูกต้องกด Create repository

Create repository ?

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Access to the repository
Users connecting to an AWS CodeCommit repository for the first time must complete setup steps before they can use it. [Learn more](#)

Repository name*

Description

*Required Cancel Create repository

ภาพที่ 4.40 Create Repository

- เมื่อสร้างเสร็จจะอยู่บนหน้า Code หากเรามี Project หรือไฟล์บางอย่างที่ต้องการ clone ลงมาบนเครื่อง ให้ทำการ copy url ตรงปุ่ม Clone url แล้วเลือก HTTPS

Code

for test create repo

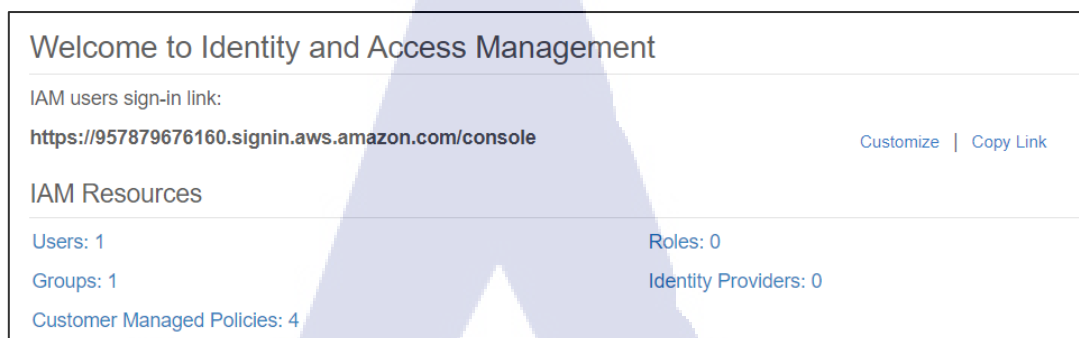
Clone URL ▲

Repository **HTTPS** [Select](#) ✕

<https://git-codecommit.us-east-2.amazonaws.com/v1/repos/AWSTestRepo>

ภาพที่ 4.41 URL เพื่อใช้เชื่อมต่อระหว่าง Local และ Server

- 4) การแชร์ AWSCodeCommit Repository!และสร้าง IAM Policies ให้กับ Group
 - เข้าสู่ระบบด้วย root account โดยไปดู link login หน้า IAM Dashboard ที่ <https://console.aws.amazon.com/iam/>



Welcome to Identity and Access Management

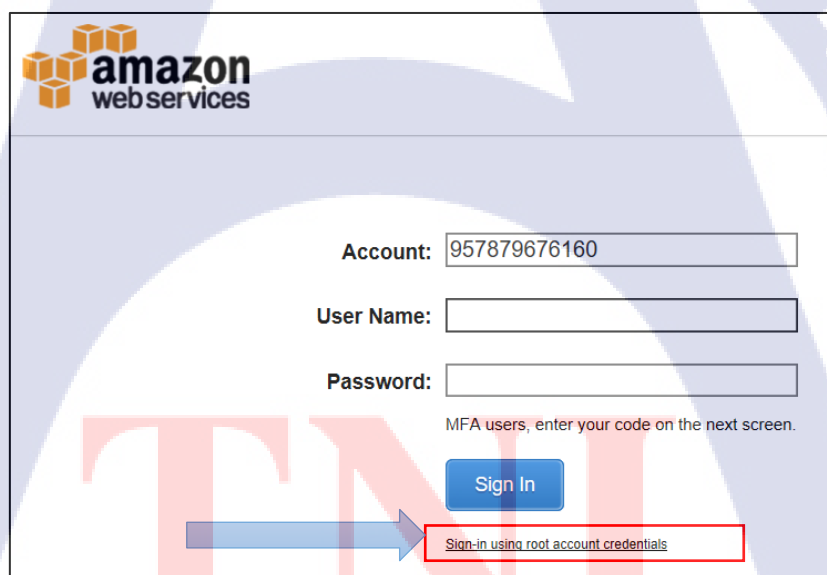
IAM users sign-in link:
<https://957879676160.signin.aws.amazon.com/console> [Customize](#) | [Copy Link](#)

IAM Resources

Users: 1	Roles: 0
Groups: 1	Identity Providers: 0
Customer Managed Policies: 4	

ภาพที่ 4.42 IAM Dashboard

- หาก login IAM user อยู่ให้ทำการ Logout ก่อน แล้วให้กด Sign-in using root account credentials



amazon web services

Account: 957879676160

User Name:

Password:

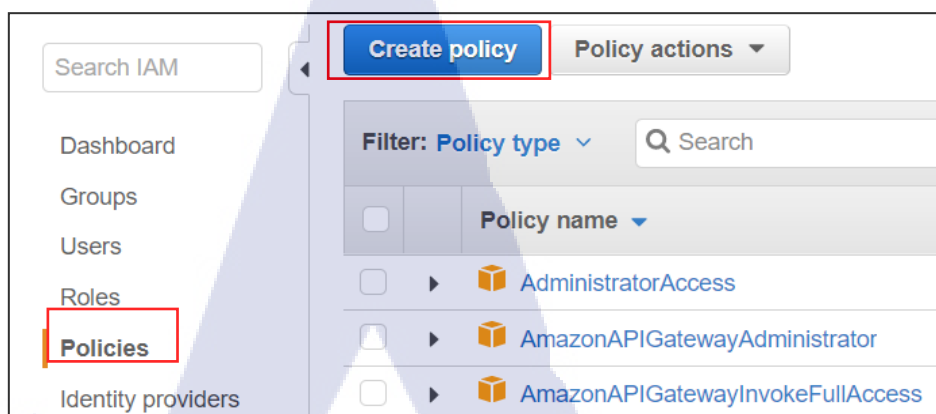
MFA users, enter your code on the next screen.

[Sign In](#)

[Sign-in using root account credentials](#)

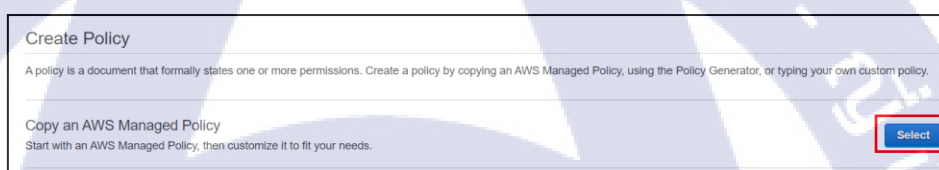
ภาพที่ 4.43 แสดงถึงการ Login ด้วย Root Account

- ในหน้า Dashboard เลือก Policies จากนั้นเลือก Create Policy



ภาพที่ 4.44 Create Policies

- บนหน้า Create Policy เลือก Copy an AWS Managed Policy เลือก Select



ภาพที่ 4.45 วิธีสร้าง Policy

- บนหน้า Copy an AWS Managed Policy พิมพ์คำว่า AWSCodeCommitPowerUser ลงไปในช่อง Search Policies จากนั้นกด Select



ภาพที่ 4.46 วิธีสร้าง Policy

- บนหน้า Review Policy ให้ใส่ชื่อของ Policy เช่น AWSCodeCommitPowerUser-AWSTestRepo ในช่องข้อความของ Policy Document บรรทัดที่มีคำว่า Resource ให้ใส่ ARN

(Amazon Resource Name) แทนเครื่องหมาย “ * ” ** วิธีหา ARN ให้ไปที่ Repository ของตัวเอง จากนั้นเลือก Setting ** จากนั้นกด Validate Policy และ Create Policy

Review Policy

Customize permissions by editing the following policy document. For more information about the access policy language, see [Overview of Policies](#) in the *Using IAM* guide. To test the effects of this policy before applying your changes, use the [IAM Policy Simulator](#).

Policy Name
AWSCodeCommitPowerUser-AWSTestRepo

Description
Provides full access to AWS CodeCommit repositories, but does not allow repository deletion.

Policy Document

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:CreateRepository",
        "codecommit:DeleteBranch",
        "codecommit:Get*",
        "codecommit:GitPull",
        "codecommit:GitPush",
        "codecommit:List*",
        "codecommit:Put*",
        "codecommit:Test*",
        "codecommit:Update*"
      ],
      "Resource": "arn:aws:codecommit:us-east-2:957879676160:AWSTestRepo"
    }
  ]
}

```

☒ Use autoformatting for policy editing

[Cancel](#) [Validate Policy](#) [Previous](#) [Create Policy](#)

ภาพที่ 4.47 วิธีสร้าง Policy

** วิธีหา ARN ให้ไปที่หน้า CodeCommit Dashboard เลือก Repository ของตัวเองจากนั้นเลือก Setting **

Dashboard

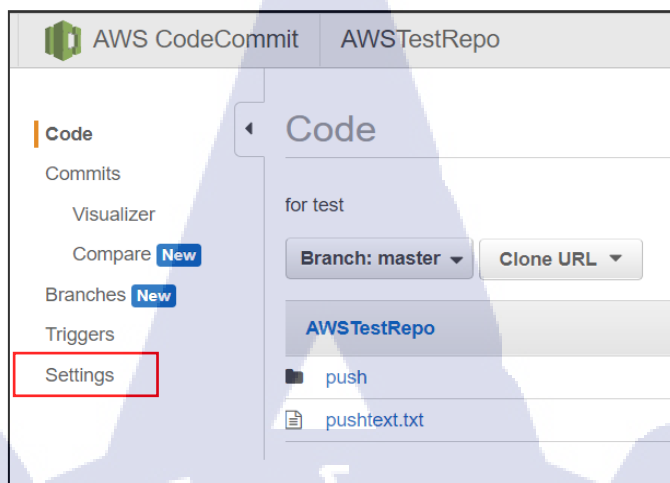
Share and manage your code in the cloud with AWS CodeCommit. Create, edit,

[Create repository](#)

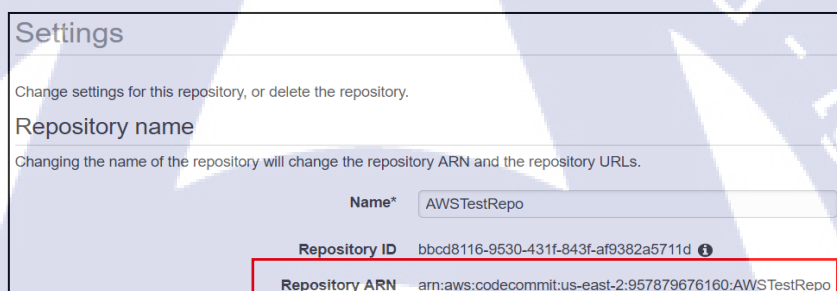
Filter by repository name

Name	Description
AWSTestRepo	for test

ภาพที่ 4.48 วิธีการหา ARN

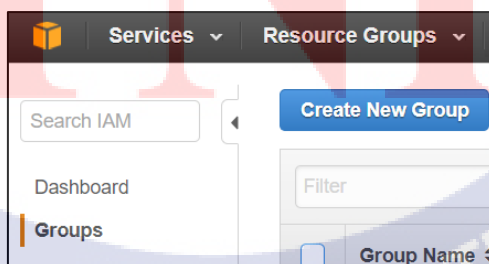


ภาพที่ 4.49 วิธีการหา ARN



ภาพที่ 4.50 วิธีการหา ARN

- ต่อไปจะเป็นการสร้าง Group ให้ IAM user โดยไปที่หน้า Dashboard จากนั้นเลือก Groups ตรงแถบด้านซ้าย และเลือก Create New Group



ภาพที่ 4.51 วิธีสร้าง Group

- บนหน้า Set Group Name ใส่ชื่อ Group ในช่อง Group Name จากนั้นเลือก Next Step (ชื่อ Group ต้องไม่ซ้ำกันในบัญชี AWS) ต่อไปเลือก Next Step



Set Group Name

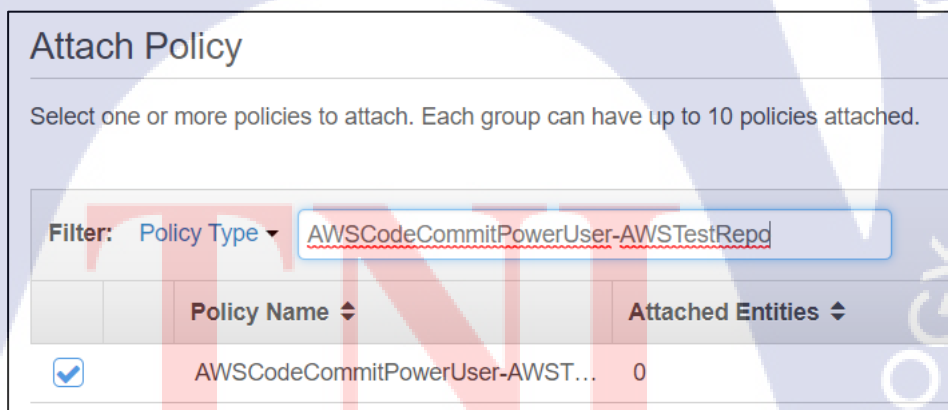
Specify a group name. Group names can be edited any time.

Group Name:

Example: Developers or ProjectAlpha
Maximum 128 characters

ภาพที่ 4.52 Set Group Name

- ให้ทำการเลือกเงื่อนไขที่เราต้องการลงใน Group นี้ (เลือกอันที่เราสร้างเมื่อขั้นตอนที่แล้วด้วย) เช่น AWSCodeCommitPowerUser-AWSTestRepo



Attach Policy

Select one or more policies to attach. Each group can have up to 10 policies attached.

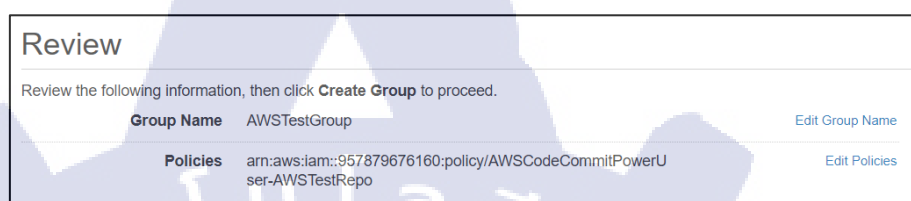
Filter: Policy Type

	Policy Name	Attached Entities
☒	AWSCodeCommitPowerUser-AWSTestRepo	0

ภาพที่ 4.53 Attach Policy for Group

- หากใช้ Git เพื่อเชื่อมต่อกับ AWS CodeCommit ให้เพิ่ม IAMSelfManageServiceSpecificCredentials และ IAMReadOnlyAccess ด้วย

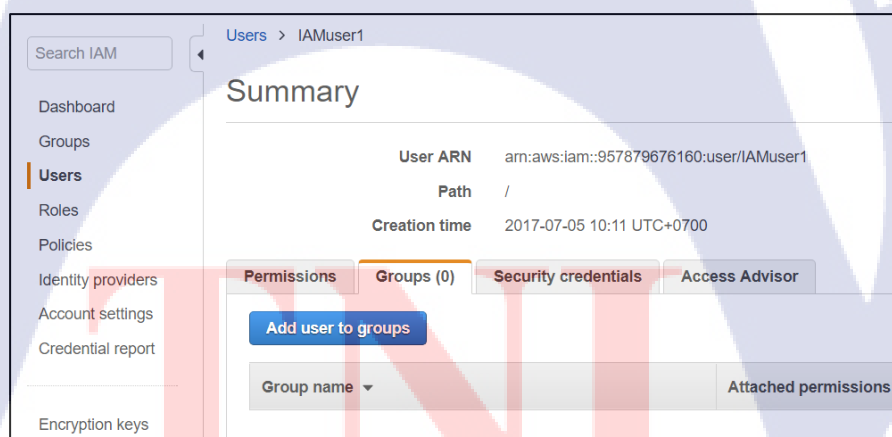
- หากใช้ SSH เพื่อเชื่อมต่อกับ AWS CodeCommit ให้เพิ่ม IAMUserSSHKeys และ IAMReadOnlyAccess ด้วย
- บนหน้า Review ให้เลือก Create Group ต่อไปจะเป็นการ Add Users เข้า Group



Review	
Review the following information, then click Create Group to proceed.	
Group Name	AWSTestGroup Edit Group Name
Policies	arn:aws:iam::957879676160:policy/AWSCodeCommitPowerUser-AWSTestRepo Edit Policies

ภาพที่ 4.54 Review Group

- ให้เลือกแถบ Users และเลือก Add User to Group เมื่อเลือกเสร็จ เลือก Add Users เป็นอันเสร็จสิ้น



Search IAM

Users > IAMUser1

Summary

User ARN: am:aws:iam::957879676160:user/IAMUser1
Path: /
Creation time: 2017-07-05 10:11 UTC+0700

Permissions Groups (0) Security credentials Access Advisor

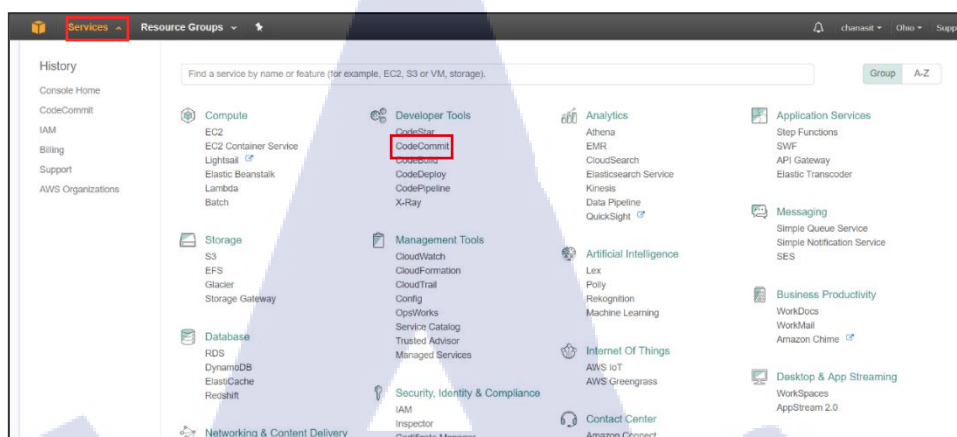
Add user to groups

Group name Attached permissions

ภาพที่ 4.55 Add User to Group

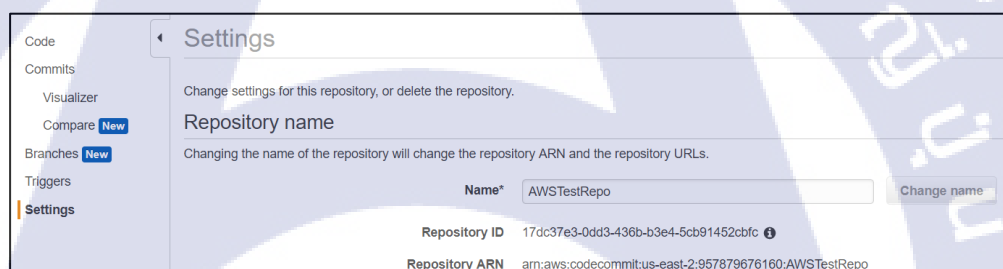
5) การเปลี่ยนการตั้งค่าของ Repository

- เปิด AWS CodeCommit Console <https://console.aws.amazon.com/codecommit>



ภาพที่ 4.56 AWSCodeCommit Dashboard

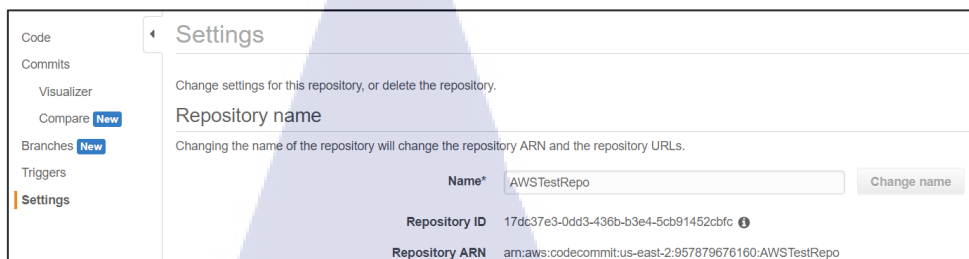
- เลือก Repository ของเรานั้นเลือก Settings



ภาพที่ 4.57 Setting Repository

- หากต้องการเปลี่ยนชื่อ Repository ให้กรอกลงในช่อง Name text box และเลือก Change

Name



Code
Commits
Visualizer
Compare **New**
Branches **New**
Triggers
Settings

Settings

Change settings for this repository, or delete the repository.

Repository name

Changing the name of the repository will change the repository ARN and the repository URLs.

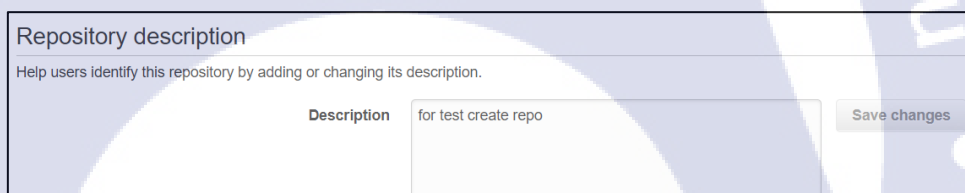
Name* Change name

Repository ID 17dc37e3-0dd3-436b-b3e4-5cb91452cbfc ⓘ

Repository ARN arn:aws:codecommit:us-east-2:957879676160:AWSTestRepo

ภาพที่ 4.58 Change Name of Repository

- หากเปลี่ยนชื่อ repository จะเปลี่ยน HTTPS และ SSH URLs ไปด้วยดังนั้นหากต้องการเชื่อมต่อจากเครื่อง Local มายัง AWS CodeCommit อีกครั้ง ควรใช้คำสั่ง `git remote set-url` เพื่อให้มันทำการจำ URLs ตัวใหม่เพื่อเชื่อมต่อยัง Server เช่นหากต้องการเปลี่ยนชื่อจาก MyDemoRepo เป็น MyRenamedDemoRepo ใช้คำสั่ง `git remote set-url origin <url>`
- สามารถเปลี่ยนคำอธิบาย Repository ได้เช่นกัน โดยพิมพ์ลงยังช่อง Description text box และเลือก Save changes



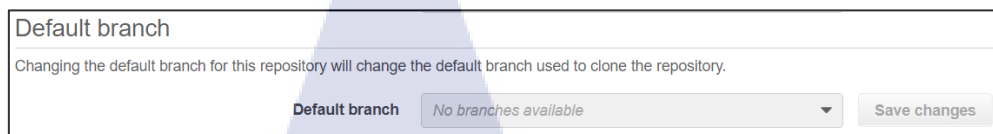
Repository description

Help users identify this repository by adding or changing its description.

Description Save changes

ภาพที่ 4.59 Repository Description

- สามารถกำหนด Default branch โดยเลือก Default branch แล้วหา Branch ที่เราต้องการให้เป็นค่าเริ่มต้นจากนั้นกด Save changes และกด Change default



Default branch

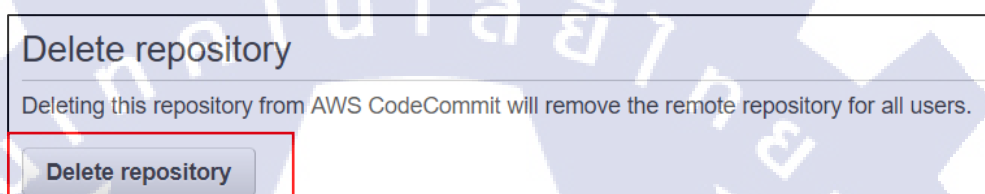
Changing the default branch for this repository will change the default branch used to clone the repository.

Default branch No branches available Save changes

ภาพที่ 4.60 Default branch

- สามารถลบ repository โดยเลือก Delete repository จากนั้นใส่ชื่อ repository แล้วกด

Delete



Delete repository

Deleting this repository from AWS CodeCommit will remove the remote repository for all users.

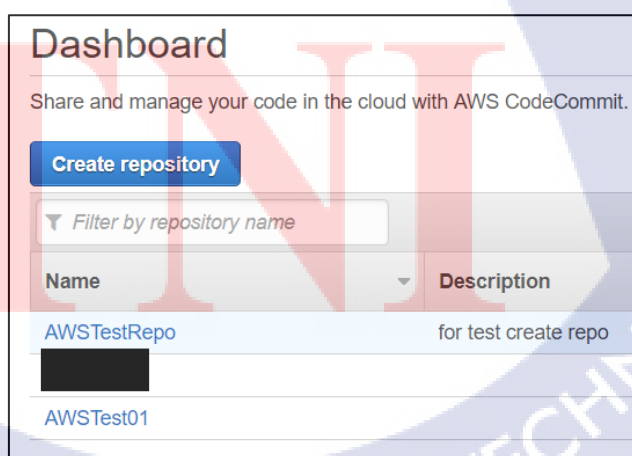
Delete repository

ภาพที่ 4.61 Delete Repository

6) การเปรียบเทียบการ commit บน AWSCodeCommit

- เปิด AWS CodeCommit Console <https://console.aws.amazon.com/codecommit> บนหน้า

Dashboard เลือก repository ที่ต้องการดูความแตกต่างของการ commit



Dashboard

Share and manage your code in the cloud with AWS CodeCommit. C

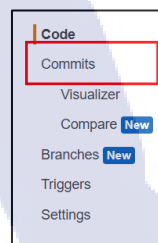
Create repository

Filter by repository name

Name	Description
AWSTestRepo	for test create repo
AWSTest01	

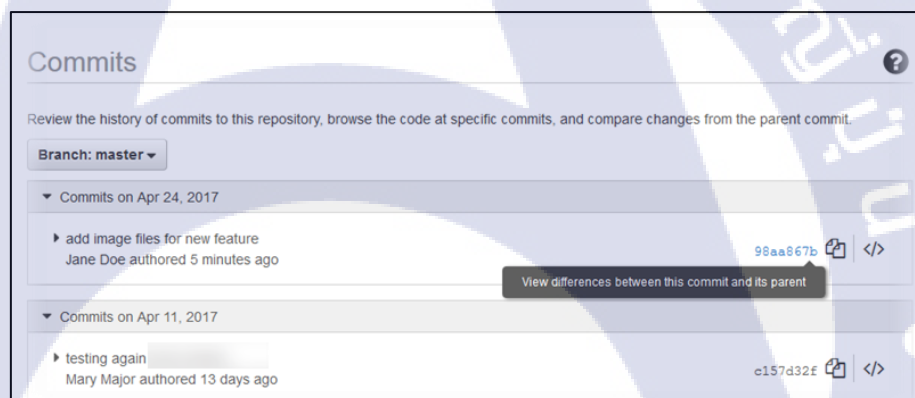
ภาพที่ 4.62 Repository Dashboard

- แถบด้านซ้าย เลือก Commits



ภาพที่ 4.63 การเข้าถึง Commits

- เลือก commit ID ในรายการ จะขึ้นข้อมูลการ commit ที่เราเลือก



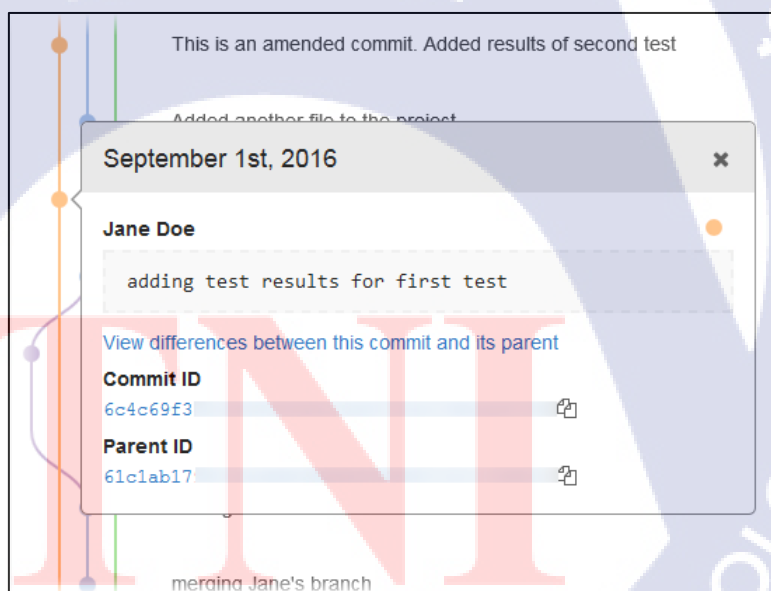
ภาพที่ 4.64 หน้า Commits

- สามารถแสดงแบบ Split view หรือ Unified view ได้



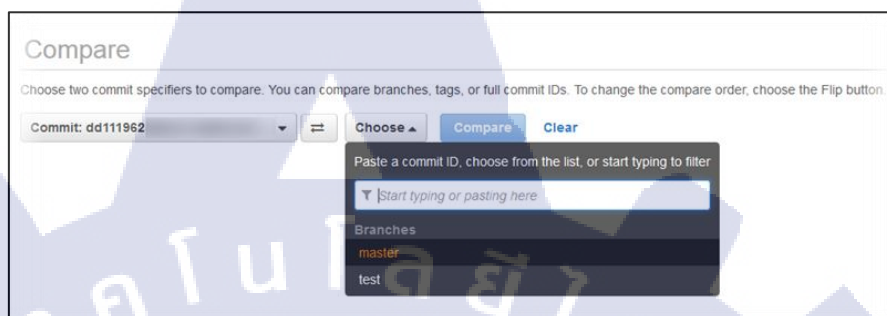
ภาพที่ 4.65 Commit ID

- จะสามารถเปรียบเทียบกับ commit หลักได้โดยการเข้าไปที่ Commit Visualizer page เลือกจุดของกราฟและกด View differences between this commit and its parent



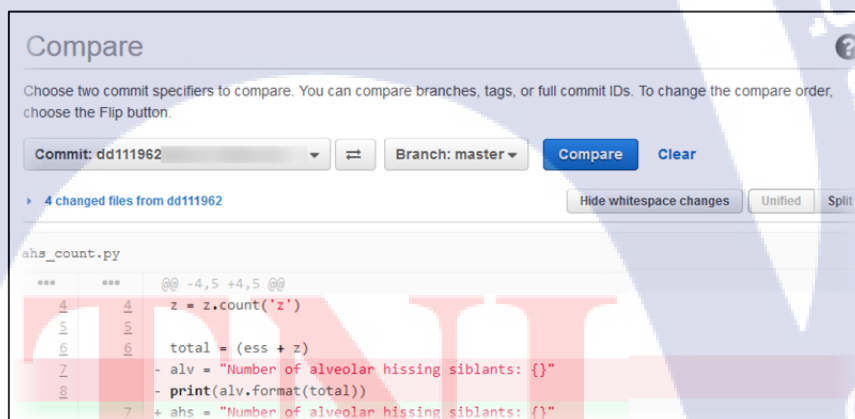
ภาพที่ 4.66 การเปรียบเทียบระหว่าง 2 commit

- คราวนี้มาลองเปรียบเทียบระหว่าง 2 Compare โดยบนแถบด้านซ้าย เลือก Compare กดปุ่ม Choose เพื่อเลือก commit มาเปรียบเทียบ (สามารถเปรียบเทียบ branch หรือหาก commit ไหนที่มี tag name สามารถใส่ tag name แทนได้และใส่ commit ID ก็ได้)



ภาพที่ 4.67 หน้า Compare

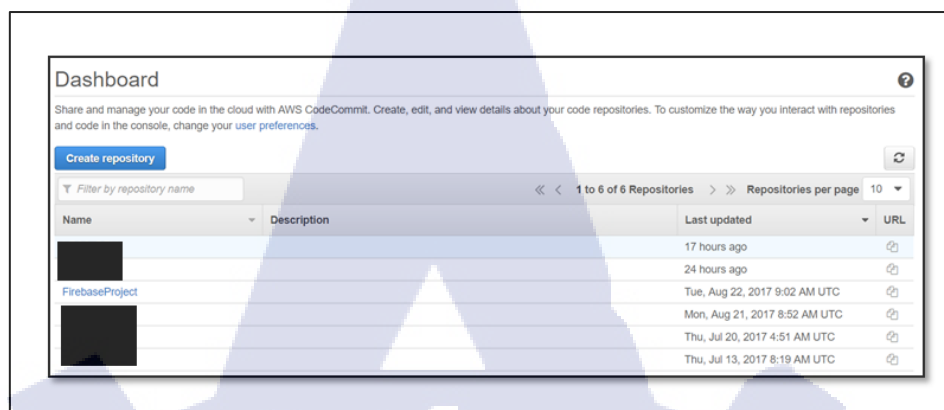
- หลังจากเลือก commit เสร็จแล้ว ให้กด Compare



ภาพที่ 4.68 แสดงถึงการเปรียบเทียบระหว่าง 2 commit

4.1.3 การอัปโหลดไฟล์โปรเจกต์ขึ้นไปยัง Web Service ที่ให้บริการฝากไฟล์ Git

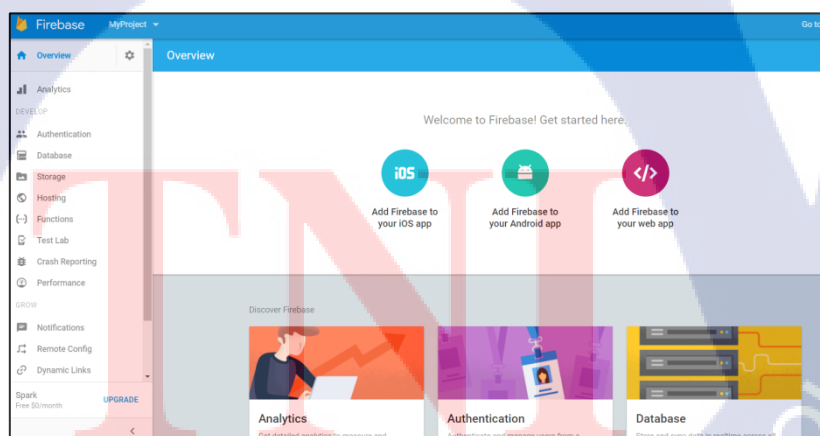
ได้ทำการอัปโหลดหรือ push ไฟล์โปรเจกต์ทุกโปรเจกต์ขึ้นบน AWSCodeCommit เพื่ออำนวยความสะดวกสบายในการพัฒนาร่วมกันเป็นทีม ไม่จำเป็นต้อง backup ข้อมูลลงบนเครื่อง Local หรือ Flash Drive อีกต่อไป



ภาพที่ 4.69 โปรเจกต์ที่อยู่บน AWSCodeCommit

4.1.4 ศึกษาข้อมูลเกี่ยวกับ Firebase

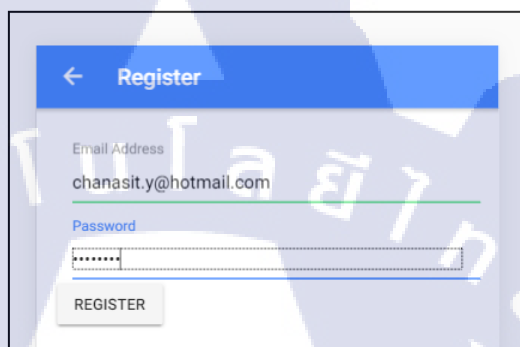
เรียนรู้ service ต่างๆ บนไฟเบส เพื่อนำ Service ไปใช้ในการพัฒนาร่วมกับ Mobile Application ซึ่ง Firebase คือ Project ที่ถูกออกแบบมาให้เป็น API และ Cloud Storage สำหรับพัฒนา Real-time Application รองรับหลาย Platform ทั้ง iOS App, Android App, Web App ซึ่งจะมี 3 อันหลักที่ได้ศึกษา ก็คือ Authentication, Real-time Database และ Storage



ภาพที่ 4.70 Dashboard บน Firebase

4.1.4.1 Firebase Authentication

Firebase Authentication จะเป็นบริการที่เข้ามาจัดการ backend ให้คุณทั้งหมด ทั้ง การ register, การ sign-in, การ reset password, โดยจะมี SDK ให้ทั้ง Android, iOS และ Web นำไปติดตั้งและใช้งาน ซึ่งรองรับการ sign-in หลากหลายรูปแบบทั้งจาก social network ยอดนิยม, จาก Email และ Password ของผู้ใช้งาน หรือแบบไม่ระบุตัวตน(Anonymous)



ภาพที่ 4.71 Demo Application หน้า Register

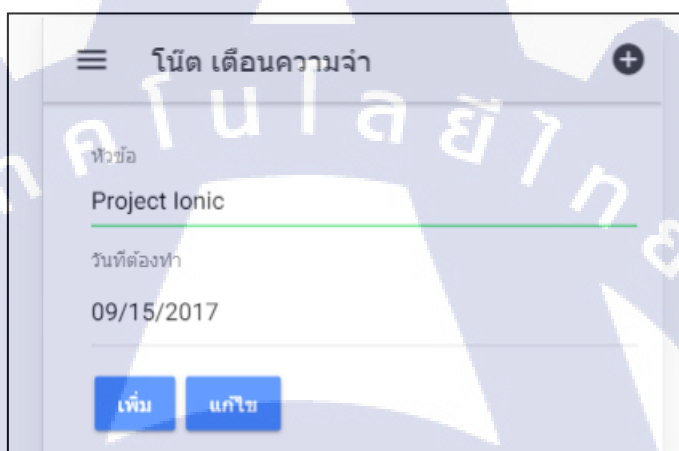
ได้มีการทำแอปพลิเคชันทดลองกับ service ตัวนี้โดยเมื่อกด Register จากนั้นไปที่ Firebase Console เพื่อที่จะตรวจสอบว่า id ที่เรามาสมัครไปได้ทำการ Authentication กับ Firebase เสร็จสิ้นแล้ว

Identifier	Providers	Created	Signed In	User UID
fifaf08@gmail.com	Google	Aug 10, 2017	Aug 10, 2017	ORunJLEllgdNd7PEs0QKtKT10s12
chanasit.y@hotmail.com	Email	Sep 15, 2017	Sep 15, 2017	FFp7vvUVXrZhZoL5peDJCF90N1s1
test@test.com	Email	Aug 7, 2017	Aug 21, 2017	HReVFUQidVgGXzxE00px16Jg62
test2@test.com	Email	Aug 23, 2017	Aug 23, 2017	lIBOFo1NBAa6g10uMcvUFCukOo83
mangkudza@hotmail.com	Email	Aug 10, 2017	Aug 10, 2017	tn72fjHgloedNCKQIGIs0la1uHC3
fifa.ne@hotmail.com	Facebook	Aug 16, 2017	Aug 16, 2017	xuQQsMnSEdM8maY04Henri3fcc...

ภาพที่ 4.72 Firebase Authentication บนหน้า Dashboard

4.1.4.2 Firebase Real-time Database

Firebase Real-time Database เป็น NoSQL cloud database ที่เก็บข้อมูลในรูปแบบของ JSON และมีการ sync ข้อมูลแบบ real-time กับทุก devices ที่เชื่อมต่อแบบอัตโนมัติในเสี้ยววินาที รองรับการทำงานเมื่อ offline(ข้อมูลจะถูกเก็บไว้ใน local จนกระทั่งกลับมา online ก็จะทำการ sync ข้อมูลให้อัตโนมัติ) รวมถึงมี Security Rules ให้เราสามารถออกแบบเงื่อนไขการเข้าถึงข้อมูลทั้งการ read และ write ได้ทั้ง 3 Platform Android, iOS และ Web



ภาพที่ 4.73 Demo Application หน้าโน้ตเดือนความจำ

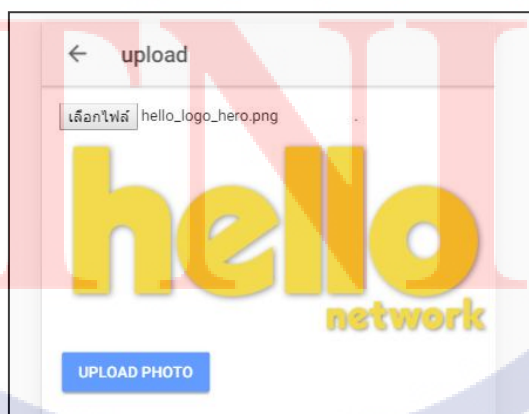
ได้มีการทำแอปพลิเคชันทดลองกับ service ตัวนี้โดยเมื่อกดเพิ่มโน้ตเดือนความจำลงไป จากนั้นเข้าไปที่ Firebase Console และไปยัง Real-time Database เพื่อตรวจสอบว่าข้อมูลได้อัพเดทแล้ว



ภาพที่ 4.74 แสดงถึงการอัปเดตแบบ Real-time

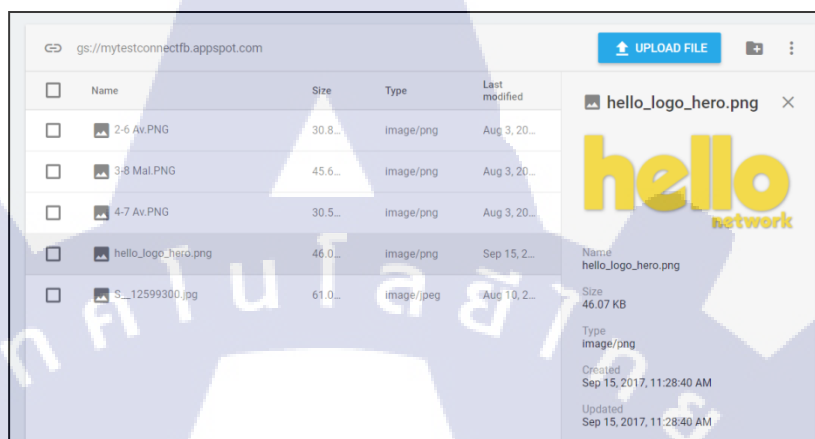
4.1.4.2 Firebase Storage

Firebase Storage เป็นบริการที่สามารถ Upload หรือ Download ไฟล์บน Google Cloud Storage โดยถ้าใช้งานฟรีจะมีพื้นที่ให้เก็บ 5GB และสามารถ upload และได้ download ได้ 50000 ครั้ง



ภาพที่ 4.75 Demo Application หน้า Upload ไฟล์

ได้มีการทำแอปพลิเคชันทดลองกับ service ตัวนี้โดยเมื่อกดอัปโหลดไฟล์ภาพชื่อ “Hello” ขึ้นไปยัง Firebase แล้วจากนั้นเข้าไปที่ Firebase Console และไปยัง Storage เพื่อตรวจสอบว่าภาพได้ถูกอัปโหลดขึ้นเสร็จสิ้น

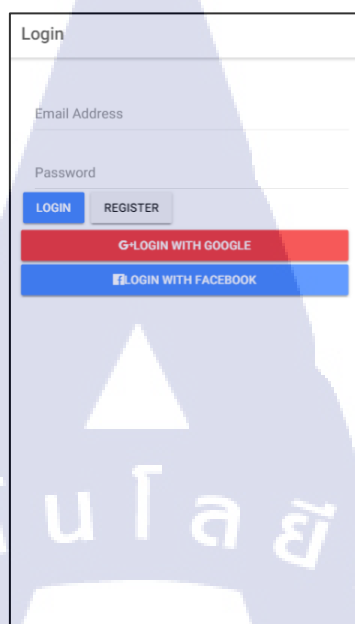


ภาพที่ 4.76 ภาพบน Firebase Storage

4.1.5 การทำแอปพลิเคชันที่ใช้บริการของ Firebase โดยใช้ Git เป็นตัวควบคุมเวอร์ชัน

พัฒนา To-do List Application โดยใช้ Service ของ Firebase และมี Git เป็นตัวควบคุมเวอร์ชัน โดยพัฒนามาเป็นตัวแอปพลิเคชันเวอร์ชันทดลอง เป็นแอปพลิเคชันที่ใช้เก็บการวางแผนการทำงานต่างๆไว้ โดยใช้ Database แบบ Real-time Database เก็บงานที่ต้องทำในแต่ละอาทิตย์ หรือแต่ละวัน โดยมี 4 ประเภทคือ Backlog, Todo, Doing, Done ใช้ 2 service ของ Firebase คือ Authentication และ Real-time Database เข้ามาช่วยพัฒนา

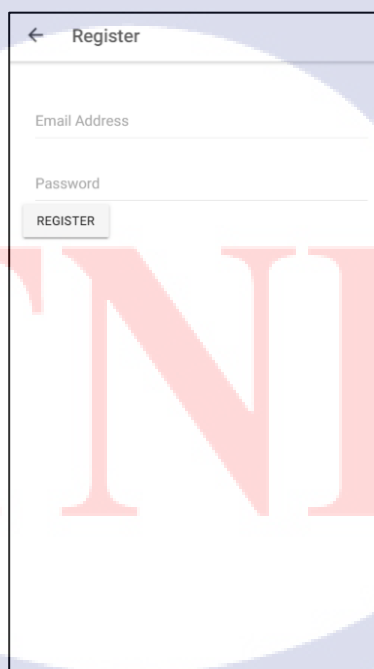
หน้า Login ประกอบไปด้วย รองรับการ login 3 อย่างคือ Google +, Facebook และอีเมลที่ไม่ระบุตัวตน และมีปุ่ม register เพื่อไปยังหน้าสมัครสมาชิก



The screenshot shows a mobile application login screen. At the top, there is a title bar with the word "Login". Below it are two input fields: "Email Address" and "Password". Under the "Password" field, there are two buttons: "LOGIN" (in blue) and "REGISTER" (in grey). Below these buttons are two social login options: "G+ LOGIN WITH GOOGLE" (in red) and "F LOGIN WITH FACEBOOK" (in blue).

ภาพที่ 4.77 หน้า Login

หน้า Register ใช้ในการสมัครสมาชิกอีเมลแบบไม่ระบุตัวตน



The screenshot shows a mobile application register screen. At the top, there is a title bar with a back arrow and the word "Register". Below it are two input fields: "Email Address" and "Password". Under the "Password" field, there is a single button: "REGISTER" (in grey).

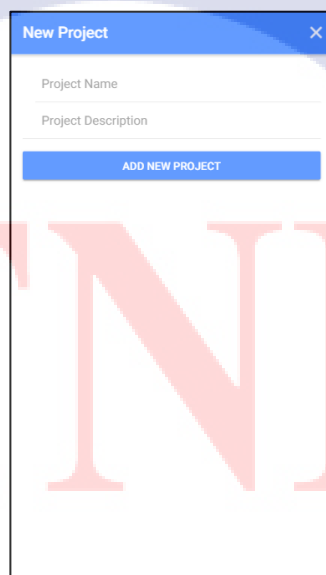
ภาพที่ 4.78 หน้า Register

หน้า My Project คือหน้าที่จะแสดงโปรเจกในแต่ละอันที่เคยเพิ่มไว้



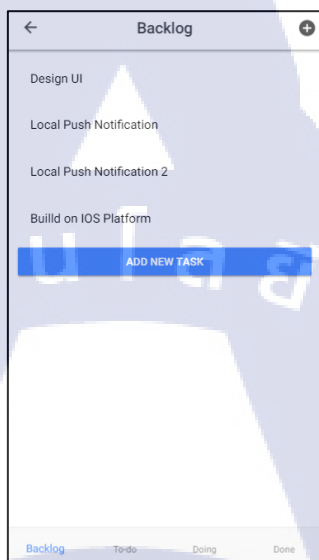
ภาพที่ 4.79 หน้า My Project

หน้า New Project ไว้สำหรับเพิ่มโปรเจกและคำอธิบายเกี่ยวกับโปรเจก



ภาพที่ 4.80 หน้า New Project

หน้า Backlog, To-do, Doing, Done ไว้แสดงงานที่ต้องทำจัดเก็บแต่ละประเภท และสามารถอัปเดตจากหน้า Backlog ไปไว้ยังแต่ละหน้าได้ หน้า Backlog สามารถกดปุ่ม“Add New Task” เพื่อเพิ่มงานได้



ภาพที่ 4.81 หน้า Backlog



ภาพที่ 4.82 หน้า To-do



ภาพที่ 4.83 หน้า Doing



ภาพที่ 4.84 หน้า Done

หน้า Add Task ไว้สำหรับเพิ่ม Task งานที่จะทำ ประกอบไปด้วย ชื่องาน, คำอธิบายของงาน, ประเภทของงานและวันที่ต้องส่ง



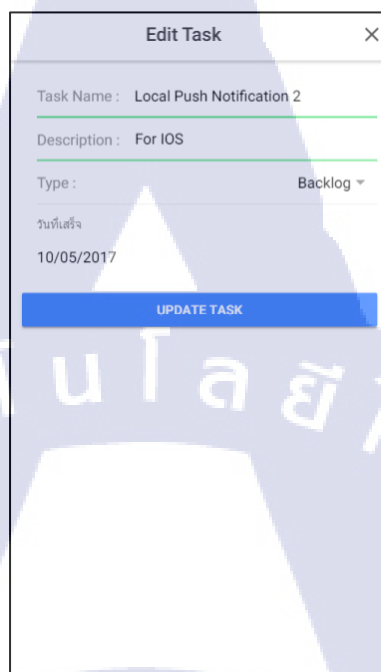
ภาพที่ 4.85 หน้า Add Task

หน้า Detail จะบอกรายละเอียดของงานที่ถูกเพิ่มเข้าไป ในแต่ละหัวข้องาน



ภาพที่ 4.86 หน้า Detail

หน้า Edit ใช้แก้ไขงานที่เพิ่มเข้าไป หรืออัปเดตงานใหม่



Edit Task	
Task Name :	Local Push Notification 2
Description :	For IOS
Type :	Backlog ▾
วันที่เสร็จ	10/05/2017
UPDATE TASK	

ภาพที่ 4.87 หน้า Edit

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการที่ได้ศึกษาเทคโนโลยีใหม่ที่สามารถช่วยสนับสนุนทีมพัฒนาระบบได้ซึ่งได้ผลลัพธ์ดังนี้

1. สามารถนำเทคโนโลยีใหม่มาพัฒนาร่วมกันเป็นทีมได้มีประสิทธิภาพ
2. เมื่อระบบมีปัญหา สามารถช่วยแก้ปัญหาได้
3. สามารถนำเทคโนโลยีที่ศึกษามาไปใช้งานได้จริง

จากผลลัพธ์ทั้งหมด สามารถสรุปได้ การช่วยสนับสนุนการทำ Mobile Application โดยนำเสนอเทคโนโลยีใหม่เข้ามาประยุกต์ใช้ในการพัฒนาระบบได้เป็นอย่างดีพร้อมทั้งสามารถนำเทคโนโลยีนี้ไปต่อยอดและพัฒนาในอนาคตได้

5.2 แนวทางการแก้ไขปัญหา

ปัญหาที่พบระหว่างการศึกษาและนำเสนอเทคโนโลยีใหม่เข้ามาประยุกต์ใช้ในการพัฒนาระบบนั้น เป็นเรื่องของเนื้อหาใหม่ทั้งหมด ซึ่งต้องใช้เวลาในการทำความเข้าใจในการศึกษาเพราะเป็นเทคโนโลยีที่ยังไม่เคยลงศึกษาและยังไม่มีการใช้งานเกิดขึ้นกับคนในทีม อีกทั้งยังมีเวลาเตรียมความพร้อมในการเขียนแอปพลิเคชันโดยใช้ ionic เวอร์ชัน 2 น้อยเกินไป การพัฒนา To-do List แอปพลิเคชัน จึงใช้เวลานานในการคิดค้นและพัฒนาแอปพลิเคชันตัวนี้ ซึ่งวิธีการแก้ปัญหาคือเรียนรู้ผ่านเว็บไซต์ที่มีผู้พัฒนาได้มาเขียนไว้ก่อนหน้านี้ หรือศึกษาจากเอกสารหลักของเรื่องนั้นๆ และในกรณีการเขียนแอปพลิเคชัน หากเกิดปัญหาหรือข้อขัดข้อง ทำการปรึกษาพี่เลี้ยง และเพื่อนที่ชำนาญด้านในด้านการเขียนแอปพลิเคชัน

5.3 ข้อเสนอแนะจากการดำเนินงาน

- 1) ควรมีความรู้และทักษะในด้านการค้นคว้าเทคโนโลยีใหม่ๆ เพื่อลดระยะเวลาในการดำเนินงานในแต่ละขั้นตอน
- 2) ในการทำงานเป็นต้องหาข้อมูลเพื่อเตรียมความพร้อมก่อนลงมือและต้องวางแผนให้การทำงานเสมอ



เอกสารอ้างอิง

devahoy, 2015, **Git คืออะไร ? + พร้อมสอนใช้งาน Git และ Github** [Online], Available: <https://devahoy.com/posts/introduction-to-git-and-github/>, [29 พฤษภาคม 2560].

Amazon Web Service, 2015, **What Is AWS CodeCommit?** [Online], Available: <http://docs.aws.amazon.com/codecommit/latest/userguide/welcome.html>, [16 กรกฎาคม 2560].

javebratt, 2016, **AngularFire2 Authentication for Ionic Framework** [Online], Available: <https://javebratt.com/ionic-firebase-authentication/>, [12 สิงหาคม 2560].

Matthieu drula, 2017, **Using Firebase and AngularFire2 in an Ionic Real-time TODO application** [Online], Available: <https://www.javascripttuts.com/using-firebase-and-angularfire2-in-an-ionic-real-time-todo-application/>, [20 สิงหาคม 2560].

GitHub, 2015, **The official library for Firebase and Angular** [Online], Available: <https://github.com/angular/angularfire2>, [3 กันยายน 2560].

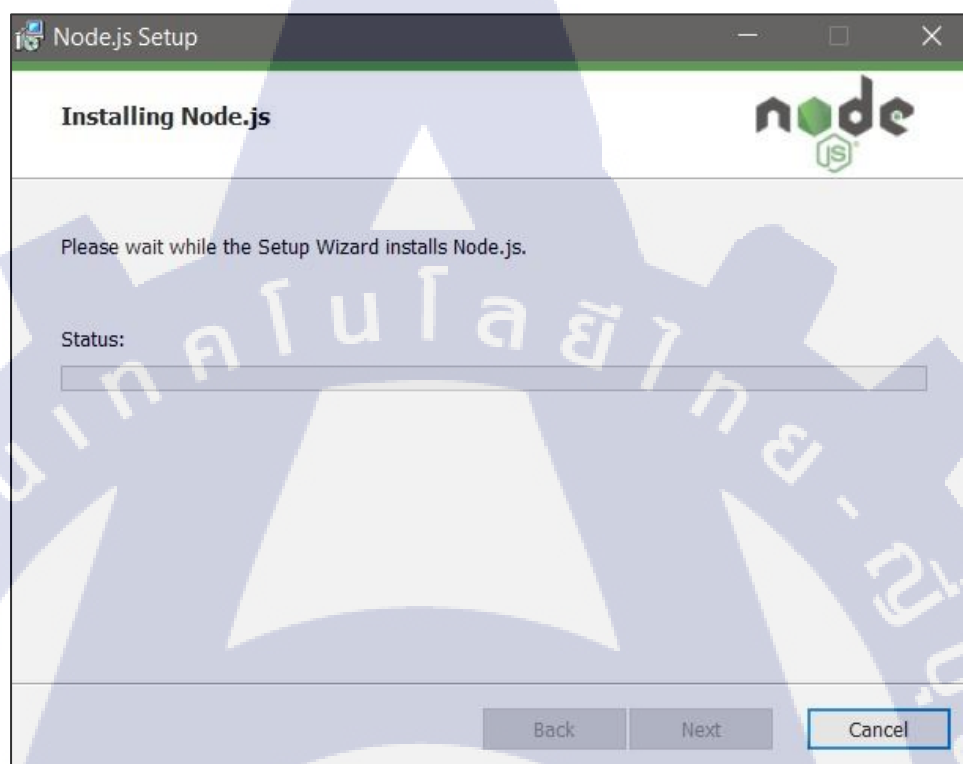
TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

ภาคผนวก ก
การติดตั้ง Ionic 2

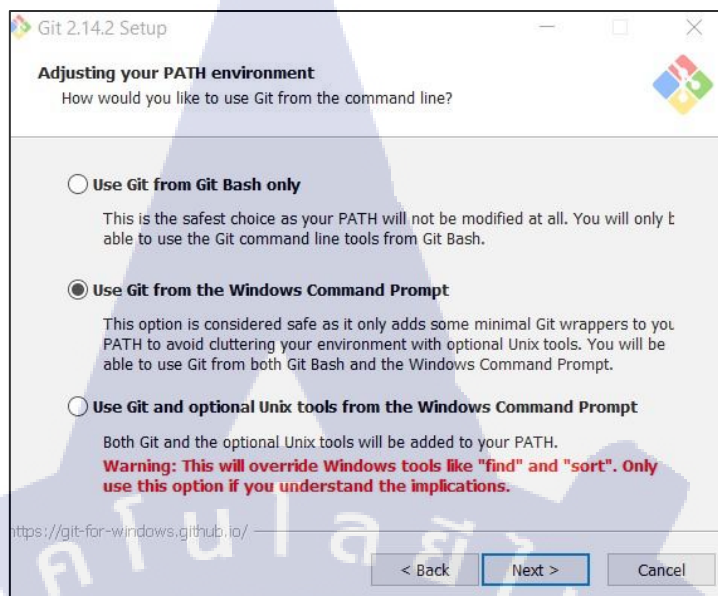
TNI

1. ติดตั้ง Node JS โดยสามารถดาวน์โหลดได้จาก <https://nodejs.org/en/> จากนั้นให้ทำการติดตั้งโดยจะใช้เวลาประมาณ 2-3 นาที



ภาพที่ ก.1 แสดงหน้า Setup Node JS

2. ติดตั้ง Git สำหรับใช้งาน PhoneGap/Cordova Plugin สามารถดาวน์โหลดได้จาก <https://git-scm.com/download/win>
3. มาถึงในส่วนของ Adjust your PATH environment ให้เลือก Run Git from the Windows Command Prompt จากนั้นกด Next และรอการติดตั้งเสร็จสมบูรณ์



ภาพที่ ก.2 แสดงหน้า Setup Git

4. ติดตั้งชุดคำสั่ง Cordova CLI และ Ionic Framework โดยใช้คำสั่ง `npm install -g cordova ionic` ใน Command Prompt แล้วกด enter
5. สามารถทดสอบว่าติดตั้งเสร็จสมบูรณ์หรือไม่โดยการเปิด Command Prompt แล้วพิมพ์คำสั่ง Ionic

```
C:\Users\chana\Desktop>ionic

 ionic
  CLI 3.10.3

Usage:
  $ ionic <command> [<args>] [--help] [--verbose] [--quiet] [--no-interactive] [--confirm] [options]

Global Commands:
  config <subcommand> ..... Manage CLI and project config values (subcommands: get, set)
  docs ..... Open the Ionic documentation website
  info ..... Print system/environment info
  login ..... Login with your Ionic ID
  signup ..... Create an Ionic account
  start ..... Create a new project
  telemetry ..... Opt in and out of telemetry

Project Commands:
  You are not in a project directory.
```

ภาพที่ ก.3 ติดตั้ง ionic เสร็จสมบูรณ์

ประวัติผู้จัดทำโครงการงาน

ชื่อ – สกุล

นายชนสิทธิ์ ยอดศรี

วัน เดือน ปีเกิด

30 กรกฎาคม 2539

ประวัติการศึกษา

ระดับประถมศึกษา

โรงเรียนอิสราณูสรณ์

ระดับมัธยมศึกษา

โรงเรียนอำมาตย์พานิชนุกูล

ระดับอุดมศึกษา

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ทุนการศึกษา

- ไม่มี -

ประวัติการฝึกอบรม

Training Pre-Cooperative Education โครงการสหกิจฯ
บริษัท A-HOST Co.,Ltd.

ผลงานที่ได้รับการตีพิมพ์ - ไม่มี -

TNI

TAI - NICHI INSTITUTE OF TECHNOLOGY