

ระบบลงทะเบียนสำหรับงานประชุมทางวิชาการ  
CONFERENCE REGISTRATION SYSTEM

ธัญชา แสงเพชร  
สิริวิชญ์ วนรฐีกาล

ปฏิญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2560

ระบบลงทะเบียนสำหรับงานประชุมทางวิชาการ  
CONFERENCE REGISTRATION SYSTEM

ธัญชา แสงเพชร  
สิริวิษณุ วนรัฐีกาล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2560

ปริญญาโทปีการศึกษา 2560

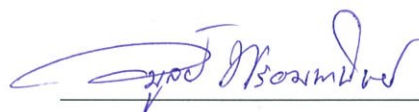
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบลงทะเบียนสำหรับงานประชุมทางวิชาการ

CONFERENCE REGISTRATION SYSTEM

ผู้จัดทำ

- |                          |                       |
|--------------------------|-----------------------|
| 1. นายรัชชา แสงเพชร      | รหัสนักศึกษา 57010574 |
| 2. นายสิริวิชญ์ วนรัฐกาล | รหัสนักศึกษา 57011363 |



อาจารย์ที่ปรึกษา

(อาจารย์วิบูลย์ พร้อมพานิชย์)

# ระบบลงทะเบียนสำหรับงานประชุมทางวิชาการ

นายธนัชชา	แสงเพชร	57010574
นายสิริวิษญ์	วนรัฐกาล	57011363
อาจารย์วิบูลย์	พร้อมพานิชย์	อาจารย์ที่ปรึกษา ปีการศึกษา 2560

## บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้ เป็นการจัดทำระบบกลางขึ้นมาเพื่อสนับสนุนระบบลงทะเบียนสำหรับการประชุมทางวิชาการ ซึ่งมีรูปแบบวิธีการชำระเงินที่หลากหลายเช่น ค่าลงทะเบียนที่มีส่วนลดพิเศษ, การชำระเงินแบบราคาพิเศษ เช่นเข้าร่วมงาน 3 ท่าน ชำระเงิน 2 ท่าน นอกจากนี้ระบบยังมีระบบการจัดการทางการเงินของผู้จัดงานเช่น การออกใบเสร็จรับเงิน, มีระบบคำนวณรายได้สุทธิสำหรับแต่ละงานประชุมเพื่อใช้สำหรับคำนวณทางภาษี, การตรวจสอบประวัติรายได้ของงานประชุมย้อนหลังเพื่อแสดงการจ่ายภาษี เป็นต้น

ระบบที่พัฒนาขึ้นมาพัฒนาโดยใช้ Framework ที่ยืดหยุ่นต่อการเปลี่ยนแปลง ได้แก่ ES6 Script และ React & Redux โดยเชื่อมต่อกับ SQL Database และส่วนเบื้องหลังของเว็บไซต์ผ่าน RESTful API โดยใช้ Node.js

ระบบสามารถรับชำระเงินได้หลากหลายช่องทาง, มีระบบส่งเอกสารงานประชุม, มีการเก็บประวัติการเข้าร่วมงานและใบเสร็จการชำระเงินค่าลงทะเบียนย้อนหลัง, มีระบบแก้ไขใบเสร็จรับเงิน และระบบยื่นคำขอคืนเงินค่าลงทะเบียน ในส่วนของผู้จัดงานจะมีระบบติดตามสถานะการชำระเงินของผู้เข้าร่วมงาน, ระบบรวบรวมข้อมูล รายการเดินบัญชี (Statement), ระบบการบริหารจัดการผู้เข้าร่วมงาน เป็นต้น

# CONFERENCE REGISTRATION SYSTEM

Thanatcha Sangphet 57010574

Sirawit Wanarattikal 57011363

Wiboon Promphanich Advisor

Academic Year 2017

## Abstract

This thesis develops the centralized platform to support attendees and conference organizers to solve conference registration problems that have many types of registration and registration rates such as special registration price rates, special registration price rates for many paper submissions, problems about receipt release. For organizers, this will support financial management. This system will help calculate net conference income, store past conferences' financial information for tax calculation.

This system has been developed using up-to-date and flexible frameworks by using React and Redux based on ES6 (JavaScript Standard) for the user interface and connected to SQL database and website's back-end with RESTful API by Node.js.

After this system release, it will support payment by many channels, have conference documents upload channel for review and approval, store joined past conference and receipts, have information editor for changing receipt information, personal information and more.

For organizers, we have attendee payment status tracking system, statement system, net income of conference calculator, attendee management system.

## กิตติกรรมประกาศ

โครงการเรื่องระบบลงทะเบียนสำหรับงานประชุมทางวิชาการนี้จะสำเร็จไม่ได้ ถ้าหากกลุ่มของข้าพเจ้าไม่ได้รับความช่วยเหลือจากบุคคลเหล่านี้

บุคคลแรกที่ต้องกล่าวขอบคุณเป็นอย่างสูงคืออาจารย์ที่ปรึกษาโครงการของข้าพเจ้า อาจารย์วิบูลย์พร้อมพานิชย์ ที่คอยให้คำปรึกษา แนะนำแนวทางของโครงการของโครงการ ทิศทางการดำเนินงานต่างๆ รวมถึงเปิดโอกาสสำคัญให้กลุ่มข้าพเจ้าได้มีโอกาสไปศึกษาดูงานประชุมทางวิชาการที่จัดขึ้นจริง ทำให้ทราบถึงปัญหาที่เกิดขึ้นจริง

ขอขอบพระคุณคณะผู้จัดงานประชุมทางวิชาการ ICSEC 2017 ที่ให้การต้อนรับและดูแลกลุ่มของข้าพเจ้าเป็นอย่างดี และได้เปิดโอกาสให้เข้าไปศึกษาและสัมผัสประสบการณ์จริงในการจัดงาน ซึ่งมีประโยชน์ต่อการนำมาปรับใช้ในการทำโครงการในครั้งนี้เป็นอย่างมาก

ธัชชา แสงเพชร

สิริวิษณุ วนรัฎีกาล

# สารบัญ

	หน้า
กิตติกรรมประกาศ.....	IV
สารบัญ.....	V
สารบัญรูปภาพ.....	VII
สารบัญตาราง.....	X
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาของปัญหา.....	1
1.1.1 กรณีศึกษา: งานประชุมทางวิชาการ ICSEC 2017.....	1
1.2 วัตถุประสงค์ของโครงการ.....	3
1.3 ขอบเขตของโครงการ.....	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	4
1.6 ส่วนประกอบของปริิณญาานิพนธ์.....	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	5
2.1 Java Script Framework: React & Redux.....	5
2.1.1 การสร้างไฟล์โปรเจคเพื่อใช้ React.....	7
2.1.2 โครงสร้างของไฟล์โปรเจคที่สร้าง.....	9
2.1.3 ไฟล์ index.html.....	11
2.1.4 ไฟล์ app.js.....	11
2.2 ECMA Script: ES6.....	14
2.2.1 Syntax ในการเขียน โปรแกรมแบบใหม่.....	14
2.2.2 เพิ่ม Classes และ Modules.....	15
2.2.3 ความสามารถเพิ่มเติม.....	15
2.3 การเชื่อมต่อ RESTful API.....	16
บทที่ 3 การออกแบบโครงการ.....	21
3.1 ภาพรวมของระบบ Client-Server.....	22
3.2 Software Requirement Specification.....	23

3.3 Use Cases Diagram .....	25
3.4 การจัดการบัญชีผู้ใช้.....	30
3.5 การออกแบบฐานข้อมูล .....	31
3.5.1 ฐานข้อมูลส่วนที่ใช้ในการลงชื่อเข้าใช้งานระบบ.....	32
3.5.2 ฐานข้อมูลส่วนที่เก็บข้อมูลของงานประชุม.....	32
3.5.3 ฐานข้อมูลส่วนการกำหนดอัตราค่าลงทะเบียน .....	33
3.5.4 ฐานข้อมูลส่วนที่เก็บข้อมูลการลงทะเบียน.....	34
3.5.5 ฐานข้อมูลส่วนที่เก็บข้อมูลส่วนบุคคล .....	35
3.5.6 ฐานข้อมูลส่วนการออกใบเสร็จ .....	36
บทที่ 4 การทดสอบและความสามารถของระบบ .....	37
4.1 การทดสอบการจ่ายเงินผ่านระบบ K-Payment Gateway.....	37
4.2 การทดสอบการจ่ายเงินผ่านระบบ PayPal.....	40
4.3 การทดสอบการทำงานในส่วนการออกใบเสร็จรับเงิน.....	43
4.4 การทดสอบทำงานเชื่อมต่อ ฟังก์ชัน client และฟังก์ชัน server ผ่าน API.....	45
4.5 ความสามารถของระบบ .....	47
4.5.1 รองรับการชำระเงินผ่านบัตรเครดิตและเครดิต .....	47
4.5.2 การลงชื่อเข้าใช้งานระบบด้วยรหัส OTP.....	48
4.5.3 ระบบสำหรับสร้างงานประชุม .....	49
4.5.4 ระบบสำหรับอัปโหลดไฟล์.....	49
4.5.5 ระบบสำหรับแสดงรายการเดินบัญชี (Statement) .....	50
4.5.6 ระบบสำหรับดาวน์โหลดเอกสาร.....	51
บทที่ 5 สรุปผลและข้อเสนอแนะ .....	53
5.1 สรุปผลการดำเนินงาน .....	53
5.2 แนวทางการพัฒนาต่อ.....	55
บรรณานุกรม.....	56
ภาคผนวก ก การติดตั้งเครื่องมือที่ใช้ในการพัฒนาระบบ. ....	57
ภาคผนวก ข Source Code. ....	66
ภาคผนวก ค การ Deploy เว็บไซต์.....	79
ภาคผนวก ง. คู่มือการใช้งานระบบ .....	86

# สารบัญรูปภาพ

	หน้า
รูป 1.1 หน้าเว็บไซต์หลักของงาน ICSEC 2017 .....	2
รูป 1.2 หน้าเว็บไซต์สำหรับชำระเงินของงาน ICSEC 2017.....	2
รูป 2.1 เครื่องหมายการค้าของ React และ Redux .....	5
รูป 2.2 ภาพแสดงโครงสร้างการทำงานของ React.....	6
รูป 2.3 ตัวอย่างการเขียน Component เปรียบเทียบกับภาษา HTML.....	6
รูป 2.4 ภาพเปรียบเทียบการทำงานของ Store และไม่ใช่ Store .....	7
รูป 2.5 หน้าจอ Terminal เมื่อสร้างโปรเจกสำเร็จ.....	8
รูป 2.6 โครงสร้างข้อมูลโปรเจกที่ถูกสร้างขึ้น .....	9
รูป 2.7 โครงสร้างข้อมูลของโปรเจก .....	10
รูป 2.8 ภาพแสดงเครื่องหมายของ ECMA Script 6 (ES6) .....	14
รูป 2.9 ภาพแสดงเครื่องหมายของ Swagger.....	16
รูป 2.10 หน้าจอการจัดการ API ของ Swagger.....	17
รูป 2.11 การทดสอบ API โดยใช้ Swagger ในการส่งรหัส OTP ไปยัง email address .....	17
รูป 3.1 แผนภาพแสดงขั้นตอนการลงทะเบียนสำหรับงานประชุมทางวิชาการ .....	21
รูป 3.1 ภาพรวมของระบบแบบ Client-Server .....	22
รูป 3.2 แผนภาพ UML ของระบบ.....	25
รูป 3.3 การทำงานของ Database Config .....	26
รูป 3.4 การทำงานของ Database Editor .....	27
รูป 3.5 การทำงานของ Document Review .....	28
รูป 3.6 การทำงานของ Payment และ Receipt Release .....	29
รูป 3.3 แผนภาพแสดงลำดับขั้นของบัญชีผู้ใช้.....	30
รูป 3.4 แผนภาพ ER Diagram การออกแบบฐานข้อมูล .....	31
รูป 4.1 ส่วนที่ใช้ทดสอบสำหรับระบบทดลองจ่ายเงิน .....	37
รูป 4.2 ส่วนสำหรับกรอกข้อมูลเพื่อส่งให้ธนาคาร .....	38
รูป 4.3 ส่วนสำหรับกรอกรหัส OTP ของระบบ Payment Gateway .....	39
รูป 4.4 ส่วนแสดงผลการชำระเงิน.....	40
รูป 4.5 หน้าเว็บสำหรับเลือกช่องทางการชำระเงินผ่าน PayPal.....	41
รูป 4.6 ภาพแสดงส่วนกรอกข้อมูลบัตรเครดิตของระบบ PayPal.....	42

รูป 4.7 ภาพแสดงผลลัพธ์การชำระเงิน.....	43
รูป 4.10 ผลลัพธ์การออก ใบเสร็จเป็นไฟล์ pdf.....	44
รูป 4.11 ใบเสร็จที่สร้างขึ้นจากการทดสอบระบบ.....	44
รูป 4.13 ผลการรัน React เชื่อมกับ server เพื่อดึงข้อมูลจาก database .....	45
รูป 4.14 ตัวอย่างการ post แก่ใจชื่อจาก pid ก่อนกดปุ่ม submit .....	46
รูป 4.15 หลังจากกดปุ่ม submit แล้ว .....	46
รูป 4.14 ฐานข้อมูลที่เปลี่ยนไปเมื่อทำการแก้ไขข้อมูล .....	47
รูป 4.15 ข้อมูลที่ถูกส่งกลับมาเมื่อชำระเงินสำเร็จ .....	48
รูป 4.17 หน้าจอสำหรับการสร้างงานประชุม.....	49
รูป 4.18 หน้าจอส่วนการอัปโหลดเอกสารงานประชุม .....	50
รูป 4.19 หน้าจอส่วนแสดงรายการเดินบัญชี .....	51
รูป 4.20 หน้าจอส่วนการดาวน์โหลดเอกสาร .....	52
รูป 1ก. สัญลักษณ์เครื่องหมายของ Semantic Ui React .....	58
รูป 2ก. สัญลักษณ์เครื่องหมายของ Styled-Components .....	58
รูป 3ก. สัญลักษณ์เครื่องหมายของ Material-Ui .....	59
รูป 4ก. สัญลักษณ์เครื่องหมายของ npm .....	59
รูป 5ก. สัญลักษณ์เครื่องหมายของ yarn .....	60
รูป 6ก. สัญลักษณ์เครื่องหมายของ swagger.....	60
รูป 7ก. หน้าจอโปรแกรม MySQL Workbench .....	61
รูป 8ก. สัญลักษณ์เครื่องหมายของ GitHub .....	61
รูป 9ก. สัญลักษณ์เครื่องหมายของ Visual Studio Code.....	62
รูป 10ก. สัญลักษณ์เครื่องหมายของ ESLint.....	62
รูป 11ก. สัญลักษณ์เครื่องหมายของ DigitalOcean.....	62
รูป 12ก. หน้าจอของระบบปฏิบัติการ Linux Ubuntu.....	63
รูป 13ก. หน้าจอของโปรแกรม VS Code ขณะทำการติดตั้งส่วนขยาย ESLint .....	64
รูป 1ข. โครงสร้างข้อมูลของโฟลเดอร์ ps-api.....	67
รูป 2ข. ไฟล์บางส่วนภายในโฟลเดอร์ routes .....	67
รูป 3ข. โครงสร้างข้อมูลของโฟลเดอร์ ps-app .....	71
รูป 4ข. โครงสร้างข้อมูลของโฟลเดอร์ ps-app/src .....	72
รูป 5ข. โครงสร้างข้อมูลของโฟลเดอร์ Component ที่ชื่อ PaymentInfo .....	73
รูป 1ค. หน้าหลักของ digitalOcean เมื่อทำการลงชื่อเข้าใช้.....	80

รูป 2ค. หน้าจอเมื่อทำการกดปุ่ม Droplets.....	80
รูป 3ค. หน้าจอเมื่อทำการกรอก SSH.....	81
รูป 4ค. การ Export ฐานข้อมูลจากโปรแกรม MySQL Workbench .....	82
รูป 5ค. Dump File ที่ Export ได้จากโปรแกรม MySQL Workbench.....	82
รูป 6ค. การ Import ฐานข้อมูล.....	83
รูป 7ค. การใช้งานฐานข้อมูลใน Server.....	84
รูป 8ค. การเข้าถึงฐานข้อมูลใน Server โดยใช้ภาษา SQL.....	84
รูป 1ง. หน้าจอการลงชื่อเข้าใช้งาน .....	87
รูป 3ง. หน้าจอสำหรับกรอกข้อมูลส่วนตัวของผู้ใช้งาน .....	88
รูป 4ง. หน้าจอสรุปผลข้อมูลการลงทะเบียน .....	89
รูป 5ง. หน้าจอเลือกวิธีการชำระเงิน .....	89
รูป 6ง. หน้าจอ Dashboard.....	90
รูป 7ง. หน้าจอของหน้า Documents.....	91
รูป 8ง. หน้าจอของหน้า Payments ส่วนที่ 1 .....	92
รูป 9ง. หน้าจอของหน้า Payments ส่วนที่ 2 .....	92
รูป 10ง. หน้าจอของหน้า History.....	93
รูป 12ง. หน้าจอของหน้า Summary .....	95
รูป 13ง. หน้าจอของหน้า Conferences.....	96
รูป 14ง. หน้าจอของหน้าปรับอัตราค่าลงทะเบียน.....	97
รูป 15ง. หน้าจอของหน้าสร้างงานประชุม .....	97
รูป 16ง. หน้าจอของหน้า Financial.....	98
รูป 17ง. หน้าจอของหน้า Statement.....	99
รูป 18ง. หน้าจอของหน้า Conference Bank Account .....	99
รูป 19ง. หน้าจอของหน้า Documents Review .....	100
รูป 20ง. หน้าจอของหน้า Data Export .....	101

# สารบัญตาราง

	หน้า
ตารางที่ 3.1 แสดงรายการความสามารถของระบบ .....	23
ตารางที่ 3.2 โครงสร้างข้อมูลสำหรับฐานข้อมูลส่วนลงชื่อเข้าใช้ .....	32
ตารางที่ 3.3 คำอธิบายสำหรับฐานข้อมูลส่วนการลงชื่อใช้งานระบบ .....	32
ตารางที่ 3.4 โครงสร้างข้อมูลสำหรับฐานข้อมูลส่วนข้อมูลงานประชุม .....	33
ตารางที่ 3.5 คำอธิบายสำหรับฐานข้อมูลส่วนข้อมูลงานประชุม .....	33
ตารางที่ 3.6 โครงสร้างข้อมูลสำหรับฐานข้อมูลอัตราค่าลงทะเบียน .....	33
ตารางที่ 3.7 คำอธิบายสำหรับฐานข้อมูลส่วนอัตราค่าลงทะเบียน .....	33
ตารางที่ 3.8 โครงสร้างข้อมูลสำหรับฐานข้อมูลส่วนข้อมูลการลงทะเบียน .....	34
ตารางที่ 3.9 คำอธิบายสำหรับฐานข้อมูลส่วนการลงทะเบียน .....	34
ตารางที่ 3.10 โครงสร้างข้อมูลสำหรับฐานข้อมูลส่วนบุคคล .....	35
ตารางที่ 3.11 คำอธิบายสำหรับฐานข้อมูลส่วนข้อมูลส่วนบุคคล .....	35
ตารางที่ 3.12 โครงสร้างข้อมูลสำหรับฐานข้อมูลการออกไปเสร็จ .....	36
ตารางที่ 3.13 คำอธิบายสำหรับฐานข้อมูลส่วนข้อมูลส่วนบุคคล .....	36
ตารางที่ 5.1 สรุปรายการความสามารถของระบบที่พัฒนา .....	53

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาของปัญหา

ในทุกๆปี จะมีการจัดงานประชุมทางวิชาการระดับชาติขึ้นมาอยู่หลายงาน ทั้งงานที่จัดขึ้นในประเทศและต่างประเทศ ซึ่งผู้เข้าร่วมงานก็มีทั้งจากในและต่างประเทศเช่นกัน แต่ละท่านเป็นบุคลากรที่มีบทบาทสำคัญทางด้านวิชาการ บุคลากรทางการศึกษา นักวิชาการ ผู้นำเสนอผลงาน และผู้ที่สนใจ จากหลากหลายสถาบัน

ในปัจจุบัน ได้มีระบบต่าง ๆ ที่จะช่วยให้ผู้เข้าร่วมงานสามารถลงทะเบียนเข้าร่วมงานได้ง่ายขึ้น เช่นระบบคัดเลือกผลงานในการเข้าร่วมงานที่มีชื่อว่า *EasyShare* แต่ทว่าระบบยังไม่ครอบคลุมทุกกระบวนการ โดยยังไม่มีระบบสำหรับลงทะเบียนเข้าร่วมงานและชำระเงินค่าลงทะเบียน

ทางผู้จัดทำได้สังเกตเห็นถึงปัญหาดังกล่าวที่ผู้เข้าร่วมงานต้องเผชิญอยู่บ่อยครั้งอาทิ ปัญหาการลงทะเบียนและชำระเงินค่าลงทะเบียนในการเข้าร่วมงานที่มีความยุ่งยากเช่น ปัญหาการลงทะเบียนและชำระเงินให้ผู้ติดตาม, การชำระเงินกรณีที่ต้องการเข้าร่วมงานประชุมมากกว่า 1 งาน, การชำระเงินกรณีส่งผลงานมากกว่า 1 ผลงาน, การให้ส่วนลดค่าลงทะเบียน, ปัญหาในการออกไปเสร็จรับเงิน เป็นต้น

นอกจากนี้ยังมีปัญหาสำคัญที่ยังคงเป็นปัญหาคือ การลงทะเบียนสำหรับบุคลากรพิเศษ เช่นแขกรับเชิญที่ได้รับการยกเว้นค่าเข้าร่วมงาน, บุคคลที่ผู้จัดงานเชิญให้เข้าร่วมงาน เป็นต้น ซึ่งเป็นปัญหาสำคัญของทางผู้จัดงานซึ่งยังไม่มีระบบกลางรองรับปัญหาในส่วนนี้

จากปัญหาที่กล่าวมาทำให้ข้าพเจ้าได้สังเกตเห็นว่า หากมีระบบกลางที่คอยจัดการประสานงานระหว่างผู้เข้าร่วมงานกับผู้จัดงานก็จะสามารถช่วยลดปัญหาและขั้นตอนที่ยุ่งยากต่างๆ ได้ จึงริเริ่ม “ระบบลงทะเบียนสำหรับงานประชุมทางวิชาการ” นี้ขึ้นมา

#### 1.1.1 กรณีศึกษา: งานประชุมทางวิชาการ ICSEC 2017

งานประชุมทางวิชาการ ICSEC 2017 ถูกจัดขึ้นระหว่างวันที่ 15-18 พฤศจิกายน พ.ศ. 2560 ซึ่งเป็นงานประชุมทางวิชาการระดับชาติที่เกี่ยวข้องกับคอมพิวเตอร์และอิเล็กทรอนิกส์

รูป 1.1 หน้าเว็บไซต์หลักของงาน ICSEC 2017

1. Credit Card  
Most events accept credit card payment. VISA CARD and MASTER CARD are welcomed by ECTI association.

2. Bank Transfer (Domestic)  
การโอนเงินผ่านธนาคาร ห้างผ่านจุดบริการ (counter service) เพื่อลงทะเบียนขึ้นมีรายละเอียด ดังนี้  
ชื่อธนาคาร : ธนาคารกรุงไทย (มหาชน) จำกัด  
สาขา : สถานีรถไฟกรุงเทพ (หัวลำโพง) สาขาแจ้งวัฒนะ  
ชื่อ บ/ช : Eng\_Conference 1  
เลขที่ บ/ช : 631-2-09992-4  
หมายเหตุ : 1) ค่าใช้จ่ายในการโอนเงิน (หากมี) ไม่สามารถที่จะหักจากค่าลงทะเบียนได้  
2) เพื่อการลงทะเบียนที่สมบูรณ์ ผู้ลงทะเบียนจะต้องอัปโหลดหลักฐานการโอนเงินขึ้นสู่ระบบ (ไม่รวมหมายเลขบัตรเครดิต)

3. Bank Transfer (International)  
For any registration made from outside the Kingdom of Thailand, bank transferring can be made based on bank's information given below:  
Bank Name: KASIKORN BANK PUBLIC COMPANY LIMITED  
Bank Address: Chalong Krung Road, KMITL, Ladkrabang, Bangkok, Thailand  
Branch: KMITL  
Account Name: Eng\_Conference 1  
Account Number: 631-2-09992-4  
SWIFT Code: KASITHBK

Remark: 1) Transferring fee (if any) cannot deduct from the registration fee  
2) Bank transfer evident must be uploaded during the process (please note down the paper ID)

4. Scholarship  
If the author had been sponsored by an organization, the payment is quite simple as just entering the payment code. The payment code should be made available (or sent to the author) once the paper is accepted or the final manuscript is uploaded.

5. Sponsorship  
An organization who has sponsored to the event can usually allow her delegate to participate the event. Sponsored person or delegate who has no paper and want to attend the event can make a registration via "non-paper registration" channel. However, a payment code may be required.

6. On Site  
To ease the publication process, it is compulsory that at least one registration must be made at the early state. Additional registration can be made at authors' convenience. In general, on site registration is provided for any walk-in registrant. For payment, CASH in local currency is usually preferred.

NOTE: Refunding is not the policy of ECTI association. If it is a case, all the expence concerns the refunding process must be charged to the registrant.  
-->

รูป 1.2 หน้าเว็บไซต์สำหรับชำระเงินของงาน ICSEC 2017

ในงานนี้กลุ่มของข้าพเจ้าได้ไปศึกษาเกี่ยวกับระบบการลงทะเบียนและชำระเงินพบว่า การออกใบเสร็จรับเงินนั้นมีปัญหาที่ผู้เข้าร่วมงานมาขอเปลี่ยนแปลงข้อมูลหน้างาน แต่ระบบของทางเจ้าหน้าที่ไม่รองรับการแก้ไขข้อมูลเฉพาะจุด ต้องทำการเขียนข้อมูลลงฐานข้อมูลใหม่แล้วจึงคัดลอกข้อมูลมารอกในโปรแกรมสำหรับออกใบเสร็จอีกครั้ง ซึ่งทำให้เสียเวลารอนาน

อีกประการหนึ่งที่สำคัญคือผู้ที่ไม่ได้ลงทะเบียนมาก่อนมาชำระเงินหน้างาน, ผู้ที่เข้าร่วมงานต้องมาติดต่อเจ้าหน้าที่เพื่อรับใบเสร็จ ซึ่งถ้าหากมีคนจำนวนมากจะทำให้เจ้าหน้าที่ทำงานไม่ทันและเกิดข้อผิดพลาดได้ ดังนั้นหากมีระบบที่ช่วยเหลือในส่วนนี้ได้จะทำให้แก้ปัญหาเหล่านี้ไปได้

## 1.2 วัตถุประสงค์ของโครงการ

- 1) การสร้างระบบขึ้นมาเพื่อสนับสนุนการจัดงานประชุมทางวิชาการ โดยจะทำหน้าที่เป็นระบบกลางประสานงานระหว่างผู้จัดงานและผู้เข้าร่วมงานเกี่ยวกับการลงทะเบียนและการรับชำระเงิน
- 2) เพื่อให้ผู้เข้าร่วมงานสามารถลงทะเบียนงานประชุมทางวิชาการหลายๆงานประชุมได้จากระบบเดียว
- 3) เพื่อช่วยเหลือผู้จัดงานในการแก้ปัญหาการออกใบเสร็จรับเงินให้มีประสิทธิภาพมากขึ้น ลดปัญหาความผิดพลาดในการออกใบเสร็จรับเงินผิดพลาด, การเสียเวลาในการแก้ไขใบเสร็จรับเงินหน้างาน, ลดระยะเวลาในการออกใบเสร็จรับเงินให้ผู้เข้าร่วมงาน
- 4) เพื่อแก้ปัญหาการจ่ายเงินซ้ำซ้อนของผู้เข้าร่วมงาน
- 5) เพื่ออำนวยความสะดวกในการจัดการงานประชุมให้แก่ผู้จัดงาน ให้สามารถติดตามจำนวนผู้เข้าร่วมงาน, สถานภาพของผู้เข้าร่วมงาน อาทิ แยกปราศรัยรับเชิญ, ผู้นำเสนอผลงาน, ผู้เข้าชมงาน เป็นต้น
- 6) เป็นระบบที่สามารถรองรับการชำระเงินได้หลากหลายช่องทาง เพื่ออำนวยความสะดวกให้ผู้เข้าร่วมงานจากต่างประเทศ

## 1.3 ขอบเขตของโครงการ

- 1) โครงการนี้ทางกลุ่มของข้าพเจ้าได้จัดทำระบบที่ครอบคลุมเฉพาะการรับลงทะเบียนและชำระเงินค่าลงทะเบียนเท่านั้น ไม่รองรับการส่งบทความเพื่อประเมินคุณสมบัติในการเข้าร่วมงาน
- 2) โครงการนี้ ทางกลุ่มของข้าพเจ้าไม่ได้พัฒนาระบบสำหรับชำระเงินขึ้นมาเอง แต่ใช้ระบบ K-Payment Gateway ของทางธนาคารกสิกรในการรับชำระเงิน, ใช้ระบบชำระเงินของ PayPal และใช้วิธีการโอนเงินผ่านบัญชีธนาคาร
- 3) ระบบที่พัฒนาขึ้นมาจะไม่มีกรเชื่อมโยงข้อมูลกับฐานข้อมูลอื่น ผู้จัดงานที่ใช้งานระบบจำเป็นต้องลงทะเบียนกับทางระบบก่อน และต้องทำการให้ข้อมูลผู้ที่ผ่านการคัดเลือกให้เข้าร่วมงานกับทางระบบก่อน จากนั้นจึงสามารถให้ผู้ชำระเงินในภายหลังเช่น ผู้ติดตาม ผู้ที่ชำระเงินล่าช้า สามารถชำระเงินเพิ่มในภายหลังได้

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้เรียนรู้การพัฒนาเว็บไซต์โดยใช้เทคโนโลยีใหม่ๆ อย่างเช่นภาษา JSX มาตรฐาน ES6, การใช้ React และ Redux Framework เป็นต้น
- 2) เป็นแนวทางให้สามารถพัฒนาระบบต่อจนเกิดเป็นระบบที่สามารถใช้งานได้จริงและสามารถนำไปประยุกต์ใช้งานระบบกับงานประชุมทางวิชาการต่างๆได้
- 3) ได้เรียนรู้จากการนำความรู้ที่เรียน มาใช้พัฒนาระบบได้จริง เช่น การออกแบบ Relational Database, การทำ User Experience

## 1.6 ส่วนประกอบของปฏิญานិพนธ์

เนื้อหาของปฏิญานิพนธ์ฉบับนี้ประกอบด้วย 5 บท ได้แก่ บทนำ, ทฤษฎีที่เกี่ยวข้อง, การออกแบบและพัฒนา, การทดลองและผลการทดลอง โดยมีรายละเอียดดังนี้

บทที่ 1 บทนำ กล่าวถึง ความสำคัญและที่มาของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ วิธีดำเนินการ และประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึงหลักการทำงานของเทคโนโลยีใหม่ที่ใช้ในการพัฒนา

บทที่ 3 การออกแบบและพัฒนา กล่าวถึงการออกแบบทั้งหมดของโครงการ

บทที่ 4 ผลการทดลอง กล่าวถึงการทดลองเชื่อมต่อ API และการทดสอบการใช้งาน

บทที่ 5 บทสรุปและข้อเสนอแนะ ประกอบไปด้วย บทสรุปที่ได้จากการทำปฏิญานิพนธ์

ภาคผนวก จะแบ่งออกเป็นแต่ละส่วนซึ่งจะมี 4 ส่วน โดยเริ่มจากการติดตั้งระบบในการพัฒนา, อธิบายโปรแกรมที่สำคัญและการเขียนโปรแกรมอย่างคร่าวๆ, การ Deploy ไฟล์โปรเจกขึ้น Server และคู่มือการใช้งานระบบ

## บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

### 2.1 Java Script Framework: React & Redux

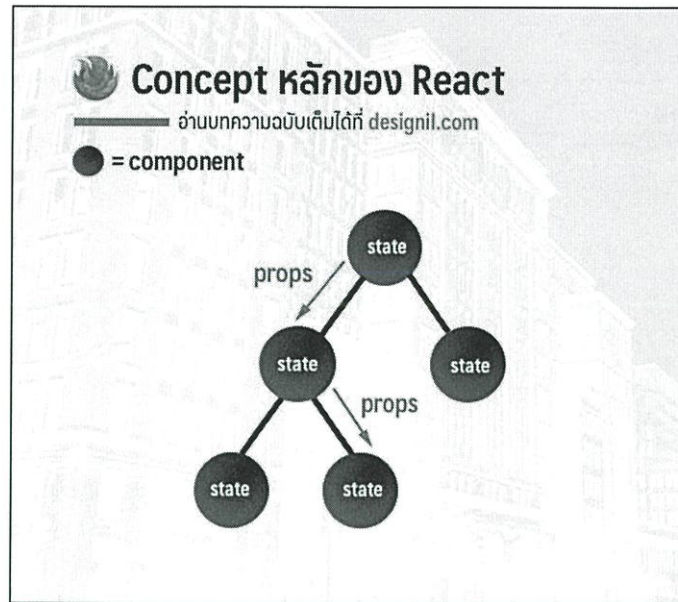


รูป 2.1 เครื่องหมายการค้าของ React และ Redux

React คือ JavaScript Library ที่ Facebook เป็นผู้พัฒนาขึ้นมา และเปิดให้คนทั่วไปนำมาใช้ฟรี ซึ่งเว็บไซต์ในปัจจุบันของ Facebook ยังคงนำ react มาใช้เองด้วย สรุปคือ คอนเซปต์ที่เราต้องรู้เพื่อเขียน React หลัก ๆ มีแค่ 3 Concept เท่านั้น

- 1) **Component** ส่วนต่าง ๆ ในเว็บเราจะมองเป็น Component
- 2) **State** ข้อมูลที่อยู่ใน Component แต่ละชิ้น เราเรียกว่า State
- 3) **Props** ข้อมูลที่ถูกส่งต่อจาก Component ชั้นบนลงไปยังชั้นล่าง เราเรียกว่า Props (Properties)

สาเหตุที่ทางผู้จัดทำเลือกใช้ React ในการพัฒนาระบบเพราะ React เป็น Framework ที่ช่วยในการลดภาระของ Server ได้เช่น ปกติในการจำกัดสิทธิ์เข้าถึงส่วนต่างๆของเว็บไซต์ หากเขียนด้วย HTML ปกติก็จำเป็นต้องตรวจสอบกับ Server ทุกครั้งโดยทุกครั้งที่มีการร้องขอ คำสั่งก็จะถูกส่งไปยัง Server เพื่อตรวจสอบ แต่หากใช้ React จะสามารถกระทำได้จาก Front-End เลย โดย React จะทำการตรวจสอบ URL ว่ามีสิทธิ์เข้าถึงได้หรือไม่ เป็นต้น และที่สำคัญการใช้งาน React สามารถใช้หลักการเขียนโปรแกรมของภาษา ES6 ได้เลย ซึ่งต่างกับ Angular.js หรือ Vue.js ซึ่งเป็นภาษา Java Script ที่ค่อนข้างจะมี Syntax เฉพาะซึ่งจำเป็นต้องใช้เวลาในการทำความเข้าใจค่อนข้างมากกว่า React



รูป 2.2 ภาพแสดงโครงสร้างการทำงานของ React

ในการเขียน โปรแกรมนั้น จะมีความแตกต่างจากการเขียนด้วยภาษา HTML ปกติ ซึ่งในการเขียน โปรแกรมโดยใช้ React นั้น จะใช้ภาษา JSX ด้วยมาตรฐาน ECMA Script ทั้งหมด ซึ่งจะสามารถเขียน โดยใช้ HTML ปกติร่วมกับ JavaScript ได้ในไฟล์เดียว ซึ่งข้อแตกต่างในการเขียน โปรแกรมจะแสดงในภาพที่ 2.3 เพื่อเปรียบเทียบดังนี้

HTML	React (JSX)
<pre>&lt;div class="header"&gt;   &lt;img class="logo"&gt; &lt;/div&gt; &lt;div class="content"&gt;...&lt;/div&gt; &lt;div class="footer"&gt;   &lt;p&gt;Copyright&lt;/p&gt; &lt;/div&gt;</pre>	<pre>&lt;Header&gt;   &lt;Logo /&gt; &lt;/Header&gt; &lt;Content&gt;...&lt;/Content&gt; &lt;Footer&gt;   &lt;Copyright /&gt; &lt;/Footer&gt;</pre>

รูป 2.3 ตัวอย่างการเขียน Component เปรียบเทียบกับภาษา HTML

### โปรแกรม 2.1 การเขียน Event ใน React

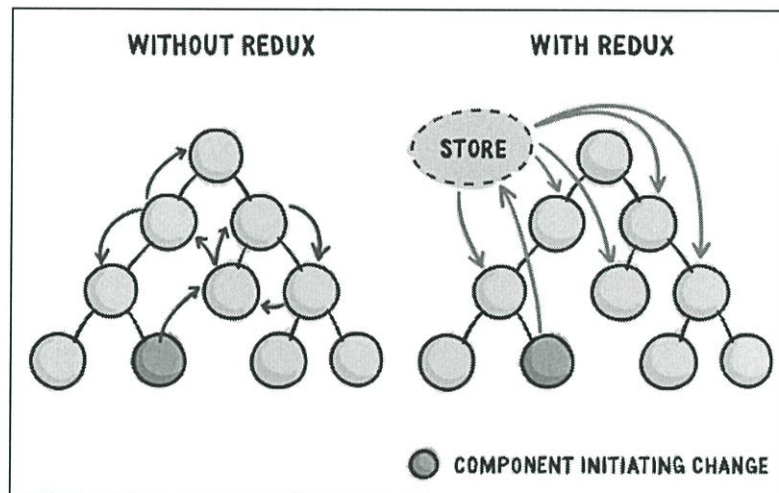
```
<button onClick={this.btnClick} > Click Me </button>
```

จากตัวอย่างโปรแกรมที่ 2.1 นั้นเป็นการบอกว่าหากคลิกที่คำว่า click me แล้วให้ทำงานที่ฟังก์ชัน btnClick ซึ่งถูกประกาศไว้ก่อนล่วงหน้าแล้ว

**Redux** เป็นไลบรารีตัวหนึ่งที่ช่วยคุณให้การไหลเวียนข้อมูลภายใน React เพื่อให้สามารถจัดการข้อมูลได้ง่ายขึ้น โดยสามารถคาดการณ์ได้ว่า เมื่อเกิดเหตุการณ์ใดเหตุการณ์หนึ่งในคอมโพเนนต์ เช่น มีการส่งค่าข้อมูลตัวหนึ่งมาซึ่งอาจเป็นสิทธิ์ในการใช้งานระบบแล้วจะมีผลในแอปพลิเคชันอย่างไร โดยสามารถส่งข้อมูลไปยัง Component นั้นโดยตรงได้

Redux สามารถแบ่งได้ 3 ส่วนดังนี้

- 1) **Store** ทำหน้าที่ในการรวม State ข้อมูล



รูป 2.4 ภาพเปรียบเทียบการทำงานของ Store และไม่มี Store

เนื่องจากการรวมข้อมูลทั้งเว็บไซต์ มาไว้ที่ Store จะทำให้จัดการกับ State ได้ง่ายขึ้นเพราะไม่ต้องส่งข้อมูลข้าม Component และสามารถนำไปใช้กับส่วนไหนของโปรแกรมก็ได้ทำให้ข้อมูลที่ได้มีความถูกต้อง

- 2) **Action** ทำหน้าที่ในการแก้ไขข้อมูลใน State ที่อยู่ใน Store
- 3) **Reducers** เป็นฟังก์ชันมีหน้าที่ในการดูว่าตอนนี้ State เป็นอะไร และถ้ามี Action เกิดขึ้น State ใหม่จะเป็นอะไร

### 2.1.1 การสร้างไฟล์โปรเจกต์เพื่อใช้ React

การสร้างไฟล์จะสร้างผ่าน Terminal โดยใช้คำสั่งผ่าน Command Line ซึ่งเป็นการสร้างไฟล์ Starter Project ของ React โดยการสร้างจำเป็นต้องมีการเรียกใช้ Node Package Manager ซึ่งจะทำการใช้ Node.js ในการสร้างไฟล์อัตโนมัติ ซึ่งในโครงการนี้จะใช้ Yarn และ npm เมื่อติดตั้งแล้วให้ใช้คำสั่งดังนี้

## โปรแกรม 2.2 คำสั่งการสร้างโปรเจก

```
npm install -g create-react-app
create-react-app <app_name>
```

เมื่อใช้คำสั่งแล้วสามารถสร้างโปรเจกสำเร็จ จะได้น้ำจอตงภาพที่ 2.5

```
= window-size@0.1.0
= wordwrap@0.0.2
= wrap-ansi@2.1.0
= write-file-atomic@2.3.0
= write@0.2.1
= xml-name-validator@2.0.1
= xtend@4.0.1
= yallist@2.1.2
= yargs-parser@5.0.0
Done in 379.37s.

Success! Created rs-test at D:\PycharmProjects\rs-test
Inside that directory, you can run several commands:

  yarn start
    Starts the development server.

  yarn build
    Bundles the app into static files for production.

  yarn test
    Starts the test runner.

  yarn eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd rs-test
  yarn start

Happy hacking!
PS D:\PycharmProjects> _
```

รูป 2.5 หน้าจอ Terminal เมื่อสร้างโปรเจกสำเร็จ

เมื่อเสร็จสิ้นจะได้ไฟล์เคอร์ของโปรเจกขึ้นมา จากนั้นทำการทดสอบว่าโปรเจกที่สร้างมาสามารถใช้งานได้จริง โดยเข้าไปที่ไฟล์โปรเจกนั้นแล้วใช้คำสั่งดังโปรแกรมที่ 2.3 หากสามารถเริ่มต้นได้แสดงว่าการสร้างโปรเจกสำเร็จและถูกต้อง

## โปรแกรม 2.3 คำสั่งเริ่มต้นโปรเจก

```
yarn start หรือ npm start
```

## 2.1.2 โครงสร้างของไฟล์โปรเจกต์ที่สร้าง

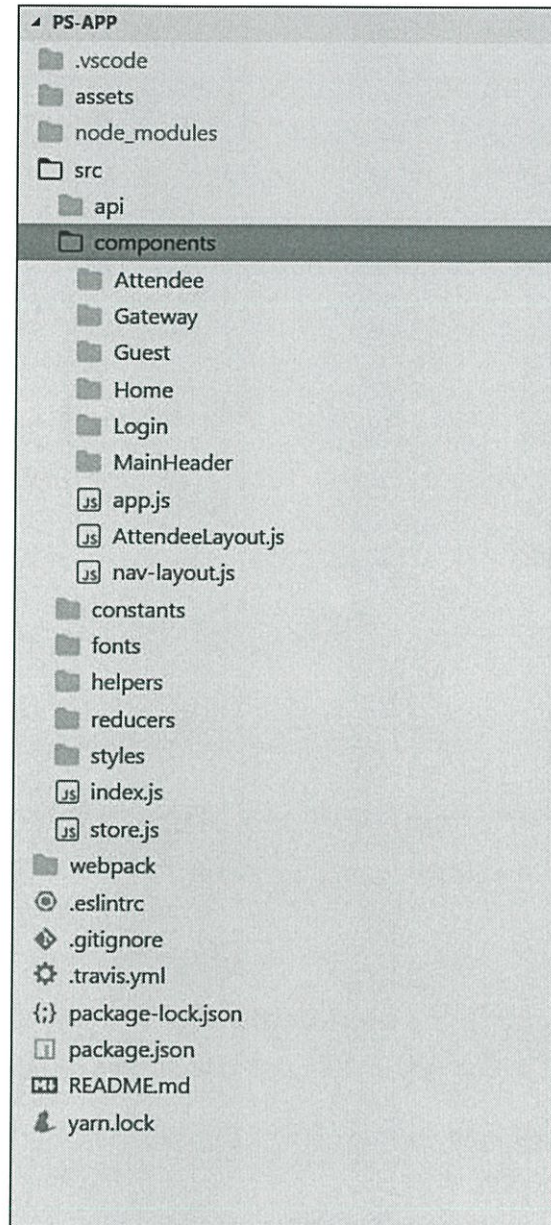
เมื่อสร้างไฟล์โปรเจกต์สำเร็จ เมื่อเปิดเข้ามาเราจะพบ โครงสร้างไฟล์ดังนี้

Name	Date modified	Type	Size
node_modules	19/04/2018 16:27	File folder	
public	19/04/2018 16:25	File folder	
src	19/04/2018 16:25	File folder	
package	19/04/2018 16:25	Text Document	1 KB
README	19/04/2018 16:25	Adobe After Effects...	1 KB
yarn.lock	19/04/2018 16:25	MD File	117 KB
yarn-error	19/04/2018 16:25	LOCK File	255 KB
	19/04/2018 16:31	Text Document	271 KB

### รูป 2.6 โครงสร้างข้อมูลโปรเจกต์ที่ถูกสร้างขึ้น

โฟลเดอร์ที่สำคัญคือ src ซึ่งจะใช้เก็บไฟล์ที่เราสร้างเว็บทั้งหมด และโฟลเดอร์ public ซึ่งใช้เก็บรูปภาพ, ของตกแต่งเว็บไซต์ และที่สำคัญคือไฟล์ index.html ซึ่งเป็นไฟล์สำหรับการ render หน้าเว็บไซต์ ส่วนโฟลเดอร์ node\_modules นั้นห้ามทำการก๊อช โดยเด็ดขาด เนื่องจากเมื่อทำงานร่วมกับผู้อื่นจะไม่ทำการส่งไฟล์ทั้งหมดในโฟลเดอร์นี้ไปให้ ทำให้ข้อมูลในโฟลเดอร์นี้ไม่เหมือนกัน และจะทำให้เกิดปัญหาในอนาคตได้

และเมื่อทำการเข้าสู่โฟลเดอร์ต่างๆจะพบกับไฟล์จำนวนมาก แต่ในการพัฒนาระบบจะมีส่วนที่สำคัญดังต่อไปนี้



รูป 2.7 โครงสร้างข้อมูลของโปรเจก

- **src/components** สำหรับเก็บไฟล์งานออกแบบส่วน Front-End ทั้งหมด โดยจะแบ่งเป็นส่วนๆ เช่น Attendee จะเก็บเฉพาะหน้าเว็บส่วนของ Attendee เท่านั้นเช่น Dashboard, Documents, Sidebar เป็นต้น
- **src/api** เป็นส่วนที่ใช้เชื่อมต่อกับ API ที่ใช้งานผ่าน Swagger
- **assets** เป็นส่วนที่ใช้เก็บไฟล์ไอคอนและรูปภาพต่างๆ
- **styles** เป็นส่วนที่ใช้เขียน style สำหรับกำหนด layout หน้าเว็บไซต์
- **index.js** เป็นไฟล์ที่ใช้ติดต่อกับไฟล์ html ในที่นี้ใช้ชื่อ template.html (เหมือนไฟล์ index.html ที่กล่าวไว้ในตอนที่ 8.3)

- `store.js` เป็นไฟล์ที่ใช้สำหรับทำ Redux โดยทำการเก็บค่าและส่งค่า Store ข้อมูลไปยัง Component ที่ข้าม Class กันได้

### 2.1.3 ไฟล์ `index.html`

การทำงานของ React นั้นจะมีไฟล์ภาษา html แค่เพียง 1 ไฟล์ โดยไฟล์นี้จะเป็นการบอกให้โปรแกรม Browser ทราบว่าจะมีการ Render ที่ตำแหน่งใด โดยทำงานร่วมกับ Webpack ซึ่งประสานงานในการแปลงภาษา JSX ให้สามารถทำงานบน Browser ได้

#### โปรแกรม 2.4 โปรแกรมที่ถูกเขียนไว้ในไฟล์ `index.html`

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="theme-color" content="#000000">
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json">
    <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">
  </head>
  <body>
    <noscript>
      You need to enable JavaScript to run this app.
    </noscript>
    <div id="root"></div>
  </body>
</html>

```

จากไฟล์ข้างต้นเราจะพบ `<div id="root"> </div>` เพียงส่วนเดียว ซึ่งเป็นตัวบอกถึงตำแหน่งที่จะทำการ Render หน้าจอ

### 2.1.4 ไฟล์ `app.js`

ไฟล์นี้เป็นไฟล์ที่สำคัญที่สุดไฟล์หนึ่งสำหรับการพัฒนาโปรแกรมด้วย react โดยจะเป็นไฟล์ที่ทำหน้าที่ส่งข้อมูลไปเพื่อ Render และจัดการทุกอย่างเช่นการเปลี่ยนหน้า, การตั้ง URL, การทำ Link ไปยังไฟล์อื่นๆ เป็นต้น

## โปรแกรม 2.5 โปรแกรมที่ถูกเขียนไว้ในไฟล์ app.js ของ Starter Project

```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';

class App extends Component {
  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h1 className="App-title">Welcome to React</h1>
        </header>
        <p className="App-intro">
          To get started, edit <code>src/App.js</code> and save to reload.
        </p>
      </div>
    );
  }
}

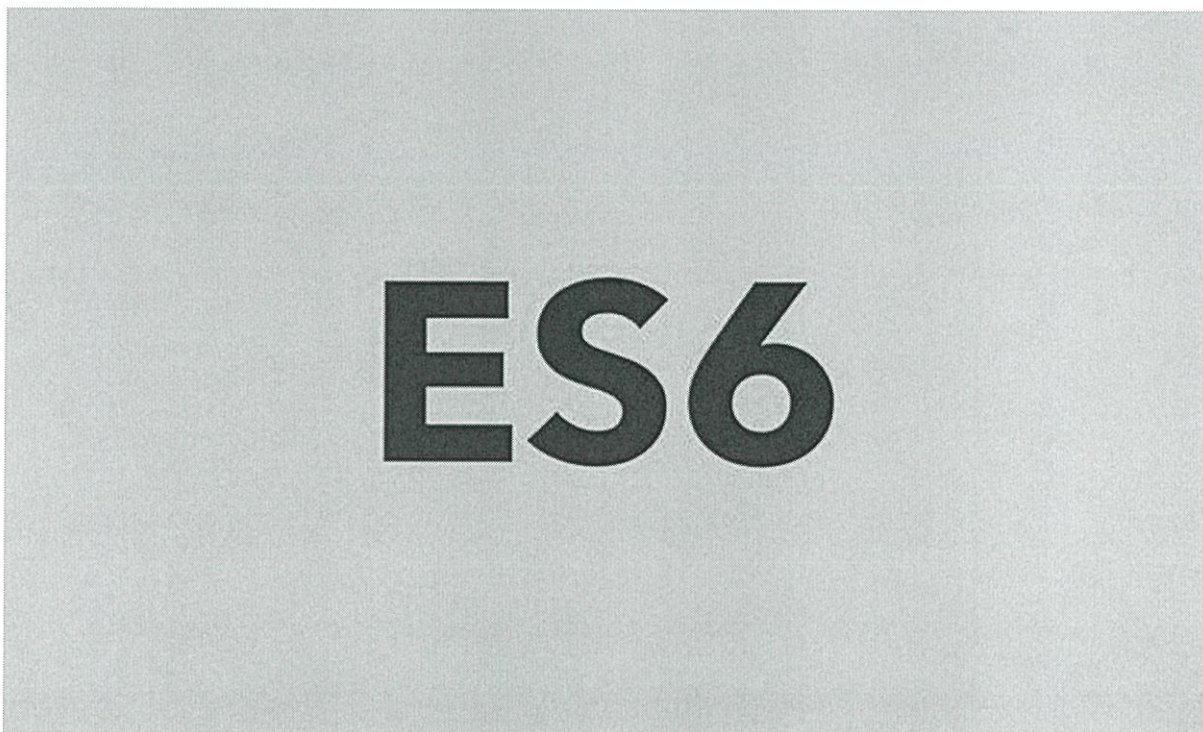
export default App;
```

และจะมีการดึง Package มาใช้ได้จากการ Import โดยข้อมูล Package จะถูกเก็บไว้ในไฟล์ package.json ซึ่งเป็น Dictionary สำหรับเก็บข้อมูล ซึ่งเมื่อเราทำการสั่งให้ระบบทำงาน ตัว Webpack จะเข้ามาดูรายชื่อของ Package ต่าง ๆ ที่ไฟล์นี้ และถ้าไม่พบ Package ที่ติดตั้งอยู่ก็จะแจ้งให้ผู้ที่นำไปใช้งานระบบติดตั้ง Package ที่ขาดไป ซึ่งไฟล์ package.json เป็นดังนี้

## โปรแกรม 2.6 รายชื่อ Packages ที่ถูกเก็บไว้ในไฟล์ package.json บางส่วน

```
"babel": {
  "presets": [
    [
      "es2015",
      {
        "modules": false
      }
    ],
    "react",
    "stage-0"
  ],
  "plugins": [
    [
      "transform-runtime",
      {
        "helpers": false,
        "polyfill": false
      }
    ]
  ],
  "env": {
    "test": {
      "plugins": [
        "transform-es2015-modules-commonjs"
      ]
    }
  }
}
```

## 2.2 ECMA Script: ES6



รูป 2.8 ภาพแสดงเครื่องหมายของ ECMA Script 6 (ES6)

ECMA Script เป็นมาตรฐานของภาษา JavaScript ซึ่งถูกกำหนดโดย ECMA International ซึ่งเป็นองค์กรที่ทำหน้าที่ในการกำหนดมาตรฐานให้กับสิ่งต่างๆ ในวงการ ไอที โดยองค์กรนี้ได้ออกมาตรฐานภาษา Script ทางฝั่ง Client ออกมาโดยใช้ชื่อว่า ECMA-262 แต่นิยมเรียกกันทั่วไปว่า ECMA Script

โดยในช่วงแรกมาตรฐานที่ออกมาจะเป็นการกำหนด Syntax ในการเขียน โปรแกรมต่างๆ ต่อมาก็ออกมาตรฐานเพื่อพัฒนาความสามารถของภาษาให้มากขึ้น โดยมีคุณสมบัติและความสามารถดังนี้

### 2.2.1 Syntax ในการเขียนโปรแกรมแบบใหม่

- **Let + Const** โดย let จะเอาไว้สร้างตัวแปรเหมือน กับ var ในภาษา JavaScript ทั่วไป เพียงแต่จะมี Scope การทำงานภายใน Block (Block คือบริเวณที่อยู่ภายใต้เครื่องหมาย { }) นั้นๆ เท่านั้น ส่วน Const จะเหมือนกับ let แต่ไม่สามารถแก้ไขค่าได้
- **Destructing** เป็นการกำหนดค่าหลายๆตัวแปรพร้อมกันได้ดังนี้ `let [x, y] = ['a', 'b'];` โดยหมายความว่า x มีค่าเท่ากับ a และ y มีค่าเท่ากับ b
- **Default Parameters** เป็นการกำหนดค่าเริ่มต้นให้กับพารามิเตอร์ของ function ได้ เช่น `function addWebsite (name = 'SiamHTML') { }`

- **For...Of Loop** เป็น loop รูปแบบใหม่ ตัวอย่างเช่นการวนดูค่าใน array สามารถเขียนได้ดังนี้ `for (let v of [1, 2, 3]) { console.log(v); }`

### 2.2.2 เพิ่ม Classes และ Modules

- **Classes** เป็นความสามารถใหม่ที่ทำให้สามารถเขียน Class แบบภาษา PHP และ Java มี Constructor และสามารถ Extend ในส่วนของ Class อื่นๆได้
- **Modules** ทำให้เราเขียน โปรแกรมออกเป็น Module ย่อยๆแล้วสามารถ import เข้ามาใช้งานได้

### 2.2.3 ความสามารถเพิ่มเติม

- **Set** เป็นการเก็บข้อมูลแบบ Set จุดเด่นของ set ก็คือถึงแม้ว่าใน set จะมี member ที่ซ้ำกัน บางตัว แต่พอแสดงค่าของ set ออกมาก็จะมีเฉพาะ member ที่ไม่ซ้ำกัน
- **Map** จะเป็น set ที่มี member เป็น key/value

ความสามารถของตัวภาษา ES6 นี้ยังมีอีกมากนอกเหนือจากที่ยกตัวอย่างมา ซึ่งสามารถเข้าไปศึกษาเพิ่มเติมได้ที่ <http://es6-features.org/>

## 2.3 การเชื่อมต่อ RESTful API

API ย่อมาจาก Application Programming Interface เป็นวิธีการเชื่อมต่อ โปรแกรมต่าง ๆ เข้ากับ Server ของเว็บไซต์

REST หรือ Representational state Transfer เป็น API ที่ใช้ส่งข้อมูลประเภทหนึ่งที่เป็นที่นิยมซึ่งจะมีการสร้าง Web Service ชนิดหนึ่งที่ใช้สื่อสารกันบนอินเทอร์เน็ต โดยใช้หลักการแบบ Stateless คือไม่มี Session ซึ่งต่างจาก Web Service แบบอื่นเช่น WSDL และ SOAP

การทำงานจะอาศัย URI/URL ของ Request เพื่อค้นหาและประมวลผลแล้วตอบกลับไปในรูปแบบของ XML, HTML, JSON โดยสามารถสั่งให้ Servers ทำงาน ได้ดังนี้

GET ทำการดึงข้อมูลภายใน URI ที่กำหนด                      POST สำหรับสร้างข้อมูล

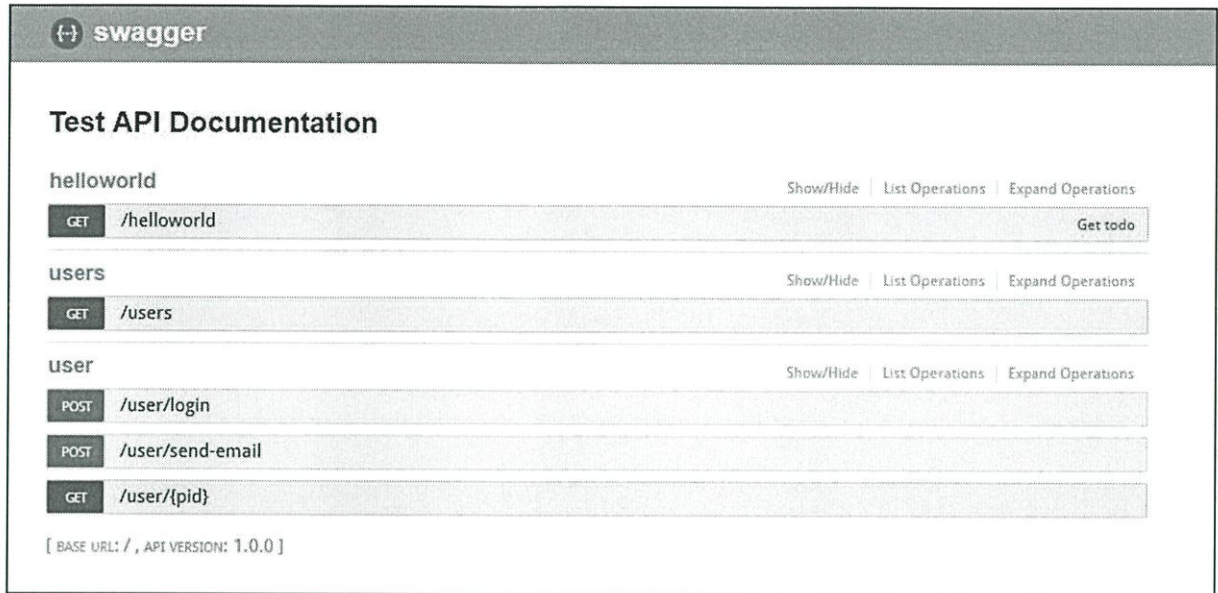
PUT ใช้แก้ไขข้อมูล    DELETE สำหรับลบข้อมูล

สำหรับการเชื่อมต่อ RESTful API นั้นในโครงการนี้ได้ใช้ **Swagger** เข้ามาใช้ในการจัดการเพื่อให้สามารถใช้ทดสอบ RESTful API ที่เขียนขึ้นมาได้ว่าสามารถใช้งานได้หรือไม่โดยที่ยังไม่ Implement เข้ากับระบบจริง อีกทั้งเป็นตัวช่วยในการเชื่อมต่อกับระบบให้สามารถทำการเชื่อมต่อได้ง่ายขึ้น



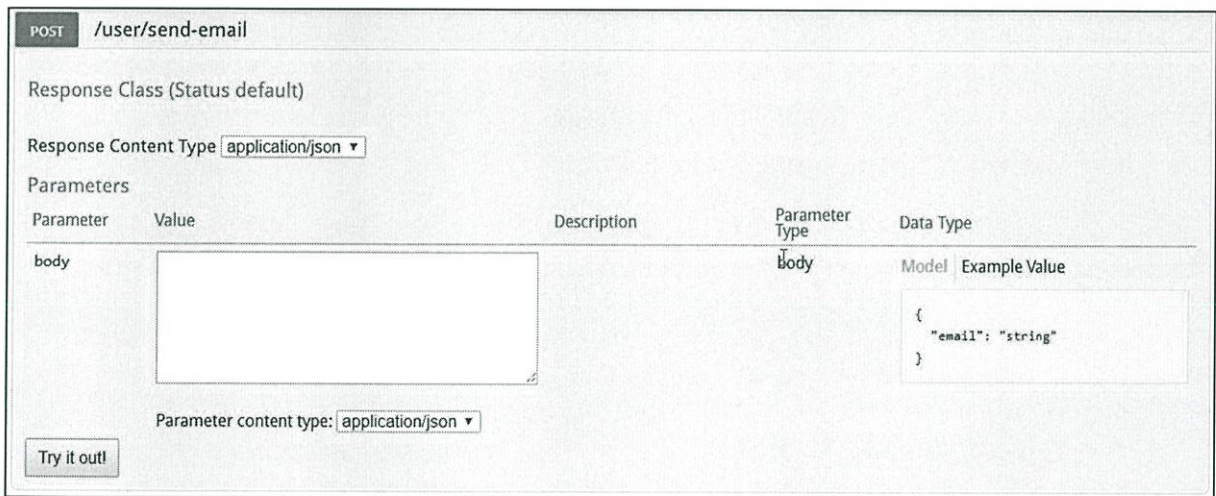
รูป 2.9 ภาพแสดงเครื่องหมายของ Swagger

การเรียกใช้งานสามารถเรียกได้จาก localhost ได้โดยเข้าไปยัง Port ที่ได้สร้างไว้ ซึ่งในที่นี้จะใช้ port 8000 และเข้าสู่ส่วนของ documentation เรียกใช้ผ่าน โปรแกรม Browser ในที่นี้เรียกใช้ผ่าน <http://localhost:8000/documentation> เมื่อสามารถเข้ามาได้ก็จะพบหน้าจอดังนี้



รูป 2.10 หน้าจอการจัดการ API ของ Swagger

ซึ่งสามารถตรวจสอบการทำงานของ API ได้ โดยยกตัวอย่างเช่น จากภาพที่ 2.10 นั้นได้ทดสอบการส่งรหัส OTP เข้าไปยัง email ที่อยู่ในฐานข้อมูล



รูป 2.11 การทดสอบ API โดยใช้ Swagger ในการส่งรหัส OTP ไปยัง email address

ในการเชื่อมต่อ API กับไฟล์โปรเจกต์นั้น ไฟล์ที่ใช้จะอยู่ในส่วน src/api โดย react นั้นจะอ่านไฟล์จากไฟล์ src/api/index.js เป็นไฟล์แรกซึ่งจะเป็นการเขียนเพื่อกำหนดค่าการเชื่อมต่อ และเป็นไฟล์ที่ทำหน้าที่ import ไฟล์ที่เขียนเชื่อมต่อ API อื่น ๆ เข้ามา

## โปรแกรม 2.7 โปรแกรมที่ถูกเขียนในไฟล์ src/api/index.js

```
import axios from 'axios';
import { GetHello } from './GetHello';
import { LoginSendOTP } from './login'

export function APIRequest(options) {
  const defaultOption = {
    baseURL: 'http://localhost:8000',
    headers: Object.assign(
      {},
      { 'Content-Type': 'application/json' },
    ),
  };
  const mergedOption = Object.assign({}, options, defaultOption);

  return axios(mergedOption);
}

export default {
  APIRequest,
  GetHello,
  LoginSendOTP,
};
```

ดังนี้

ในการเชื่อมต่อ API จะขอยกตัวอย่างการเขียนเชื่อมต่อกับ Swagger เพื่อส่งรหัส OTP ไปยัง email

## โปรแกรม 2.8 โปรแกรมที่ถูกเขียนในไฟล์ src/api/login.js

```
import { APIRequest } from '.';
import {
  LoginSendOTPSuccess,
  LoginSendOTPFailure,
} from '../reducers/loginReducer';

export function LoginSendOTP(email, otp) {
  return dispatch => {
    const options = {
      method: 'post',
      url: '/user/login',
      data: {
        otp,
        email,
      },
    };
    APIRequest(options)
      .then(res => {
        // console.log('sent OTP Success', res.data);
        localStorage.setItem('name', res.data.name);
        localStorage.setItem('role', res.data.role);
        localStorage.setItem('auth', res.data.auth);
        dispatch(LoginSendOTPSuccess(res.data));
      })
      .catch(err => {
        // console.log('sent OTP fail', err);
        dispatch(LoginSendOTPFailure(err.response));
      });
  };
}

export function LoginSendEmail(email) {
  const options = {
    method: 'post',
    url: '/user/send-email',
    data: {
      email,
    },
  };
};
```

```

APIRequest(options)

  .then(res => {
    console.log('can post api', res);
  })
  .catch(err => {
    console.log('cant post api', err);
  });
}

export function Logout() {
  return () => {
    console.log('Logout');
    localStorage.clear();
    window.location = '/home';
  };
}

export default {
  LoginSendOTP,
};

```

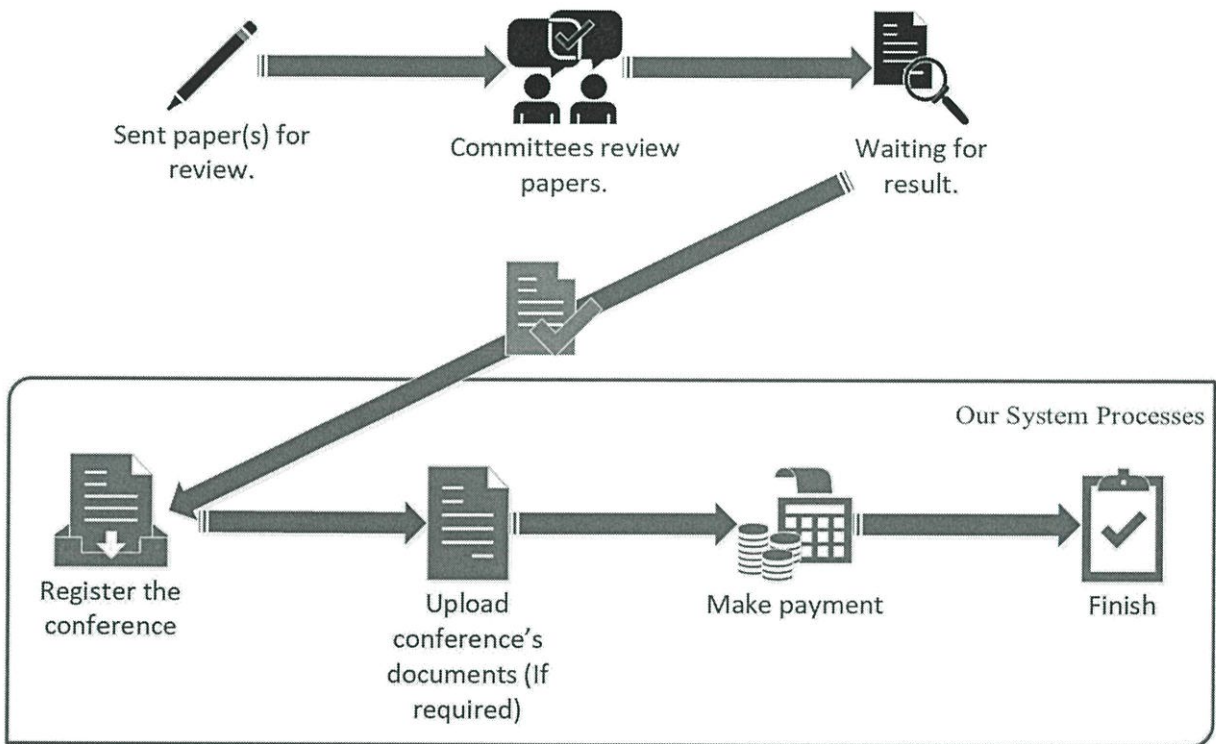
ในไฟล์นี้จะเป็นการเขียนเชื่อมต่อกับทั้ง Database และ Swagger โดยจะทำการตรวจสอบค่าในฐานข้อมูลเพื่อดู email, ชื่อ, ประเภทบัญชี และสถานะการลงชื่อเข้าใช้ ข้อมูลที่ได้จะถูกส่งมาและจัดการโดยใช้ redux

ขั้นตอนการทำงานคือตรวจสอบข้อมูลในฐานข้อมูล จากนั้นนำค่าส่งไปยังระบบเบื้องหลัง (Back-End) โดยผ่าน API ถ้าข้อมูลถูกต้องตามที่ผู้ใช้งานเข้ามา ก็จะทำการส่งรหัส OTP ไปยัง email ที่กรอกไว้ จากนั้นระบบจะเขียน OTP ลงฐานข้อมูล และเมื่อผู้ใช้งานลงชื่อเข้าใช้สำเร็จก็จะเขียนข้อมูล เปลี่ยนสถานะเป็นลงชื่อเข้าใช้งาน และเปลี่ยนหน้าเว็บไปยัง URL ที่กำหนดในไฟล์ app.js

### บทที่ 3

## การออกแบบโครงการ

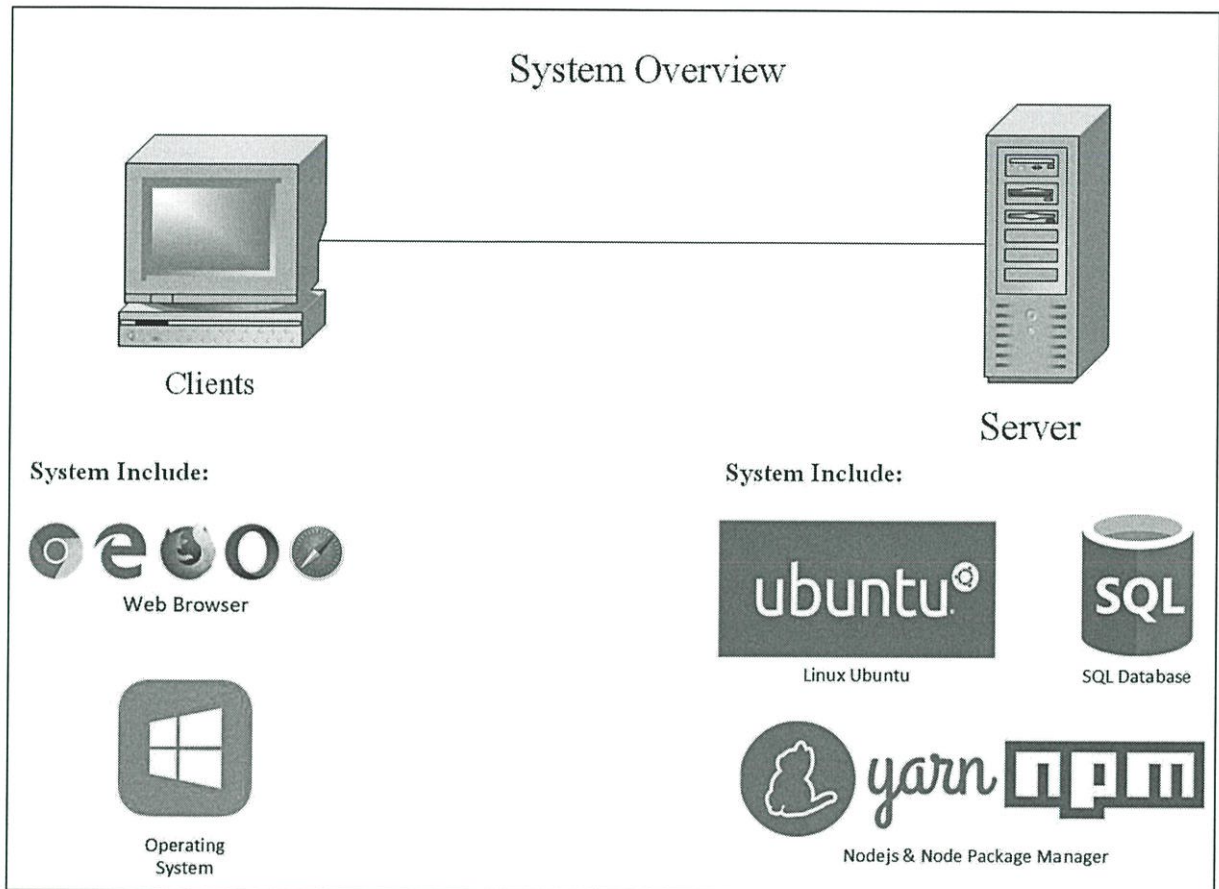
การออกแบบระบบจะเป็นการสร้างระบบกลางเพื่อรองรับการใช้งานในกระบวนการขั้นตอนการลงทะเบียนของงานประชุมทางวิชาการ โดยจะแสดงในแผนภาพดังต่อไปนี้



รูป 3.1 แผนภาพแสดงขั้นตอนการลงทะเบียนสำหรับงานประชุมทางวิชาการ

ระบบที่สร้างจะมีหน้าที่หลังจากที่ทางผู้จัดงานส่งข้อมูลรายชื่อของผู้ที่ลงทะเบียนมาให้กับทางระบบเป็นที่เรียบร้อยแล้ว จากนั้นจะนำข้อมูลเข้ามายังฐานข้อมูลของระบบแล้วให้ผู้เข้าร่วมงานชำระเงินผ่านระบบของเราที่พัฒนาขึ้น จากนั้นระบบจะทำหน้าที่จัดการและติดตามสถานะต่างๆ ให้กับทางผู้จัดงาน รวมถึงปัญหาต่างๆ ด้วยเช่นกัน

### 3.1 ภาพรวมของระบบ Client-Server



รูป 3.1 ภาพรวมของระบบแบบ Client-Server

ระบบที่ออกแบบขึ้นมาทำงานในระบบ Client-Server โดยฝั่ง Client นั้นจะเป็นส่วนของผู้ใช้งานทั้งหมดโดยส่วนนี้ตัวระบบจะประกอบไปด้วยระบบปฏิบัติการ ซึ่งผู้ใช้งานสามารถใช้ระบบปฏิบัติการใดก็ได้ทั้ง Windows, Mac OS X หรือ Linux โดยใช้งานผ่าน Web Browser ซึ่งรองรับการใช้งานสำหรับ Web Browser สำหรับเครื่องคอมพิวเตอร์เท่านั้น (เว็บไซต์ไม่ได้ออกแบบให้ใช้ในโทรศัพท์)

สำหรับฝั่ง Server นั้นจะใช้ระบบปฏิบัติการ Linux Ubuntu ซึ่งภายใน Server นั้นได้มีการติดตั้งไฟล์โปรเจกของระบบไว้ และมีการติดตั้ง SQL Database ของระบบโดยจำเป็นต้องติดตั้ง Node Package Manager และ Node.js ไว้ด้วยเพื่อให้สามารถเชื่อมต่อหน้าเว็บไซต์เข้ากับส่วนเบื้องหลังของระบบได้

### 3.2 Software Requirement Specification

ความต้องการของระบบที่ออกแบบไว้ จะแสดงในตารางดังต่อไปนี้

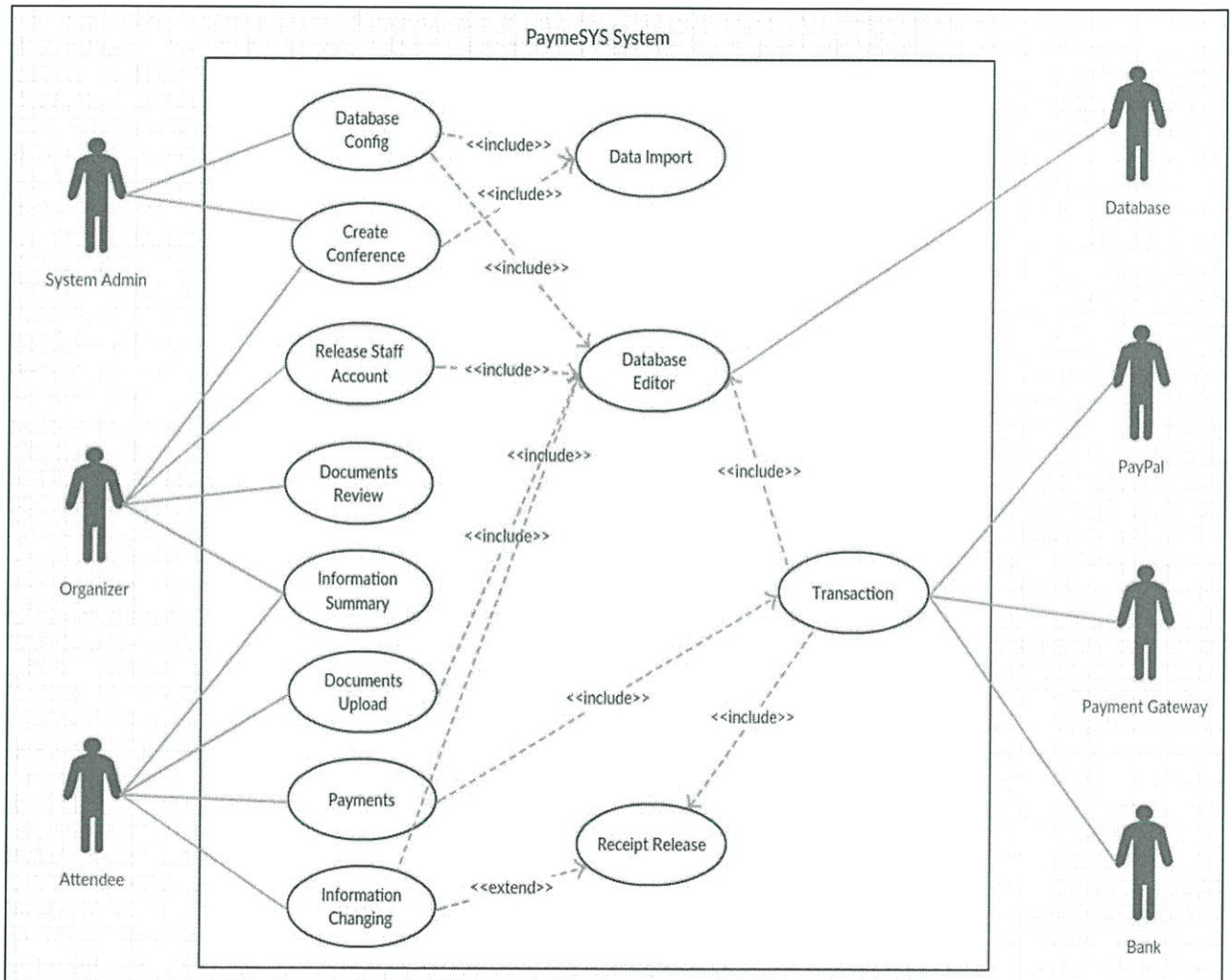
ตารางที่ 3.1 แสดงรายการความสามารถของระบบ

ID	Requirement	Priority
P-01	The system should display information about system on first page of website.	Must Have
P-02A	The system should display 'Partner Conference'.	Must Have
P-02B	The system should link to 'Partner Conference' website.	Must Have
P-03	The system should have login and sign-up link on first page.	Must Have
P-04A	The system should have support system for helping.	Must Have
P-04B	The system should have chat channel for contact supporter.	Should Have
P-05	The system should have registration system	Must Have
P-06	The system should have sign in system.	Must Have
P-07	The system should have password recovery system.	Must Have
P-08A	The system should support multi-user accounts type.	Must Have
P-08B	The system should support for change account type from permit account.	Must Have
P-09	The system should have paper information system for organizer.	Must Have
P-10	The system should release organizer's ID by manual input or random.	Must Have
P-11	The system should have files upload system.	Must Have
P-12	The system should have status tracking system.	Must Have
P-13	The system should display notification for important information after login by pop-up or another.	Must Have
P-14A	The system should have payment system.	Must Have
P-14B	The system should accept PayPal, bank transfer, credit card and debit card.	Must Have
P-14C	The system should have payment notification system to organizer when pay via bank transfer.	Must Have

P-14D	The system should accept pay for 3 <sup>rd</sup> person.	Must Have
P-14E	The system should make payment by scan Quick Response Code (QR Code).	Should Have
P-15	The system should have system visual wallet system for account uniquely.	Should Have
P-16	The system should have electronic signature system.	Must Have
P-17	The system should release invoice to customer.	Must Have
P-18	The system should release purchase order (PO).	Must Have
P-19	The system should release receipt.	Must Have
P-20	The system should cancel option for PO, receipt and relate documents.	Must Have
P-21	The system should have specification conference's information site.	Should Have
P-22	The system should release report to csv, pdf when user request.	Should Have

### 3.3 Use Cases Diagram

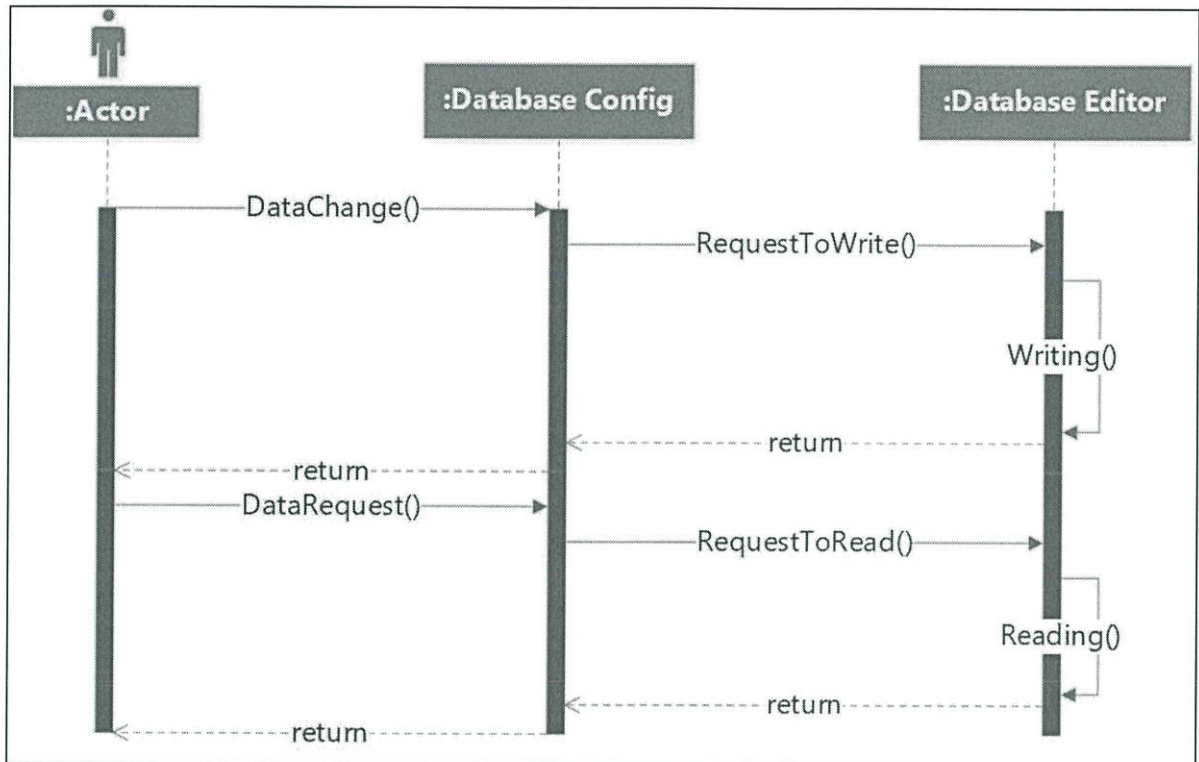
จากหัวข้อที่ผ่านมาเป็นการแสดงหัวข้อหลักสำหรับฟังก์ชันการทำงานของระบบ แต่ทว่ายังไม่ทราบว่ามีผู้ใช้ประเภทใดบ้างและความสัมพันธ์กับฟังก์ชันการทำงานอย่างไร ในส่วนนี้จะแสดงความสัมพันธ์ระหว่างผู้ใช้งาน, ระบบภายนอกที่ติดต่อกับระบบนี้ และฟังก์ชันการทำงาน ดังแผนภาพต่อไปนี้



รูป 3.2 แผนภาพ UML ของระบบ

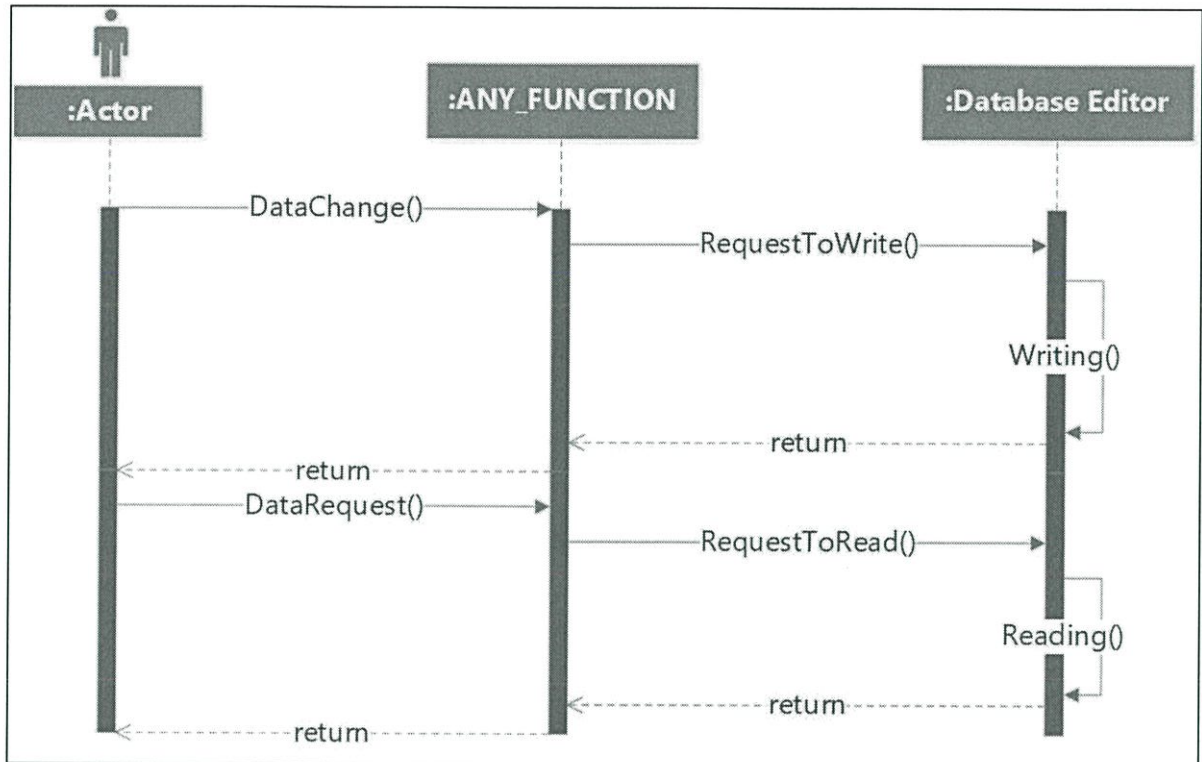
จากแผนภาพจะเห็นได้ว่าระบบนี้จะแบ่งฟังก์ชันการทำงานหลักๆเป็นดังนี้

- ระบบจัดการฐานข้อมูล (Database Config) เป็นระบบที่ผู้ดูแลระบบใช้ในการเข้าไปจัดการกับฐานข้อมูลซึ่งสามารถแก้ไขและอ่านข้อมูลได้ โดยระบบส่วนนี้จะมีการไปเรียกใช้งานส่วนที่ใช้แก้ไขฐานข้อมูลอีกครั้งเมื่อมีการส่งคำขอการอ่านหรือเขียนข้อมูล ดังนี้



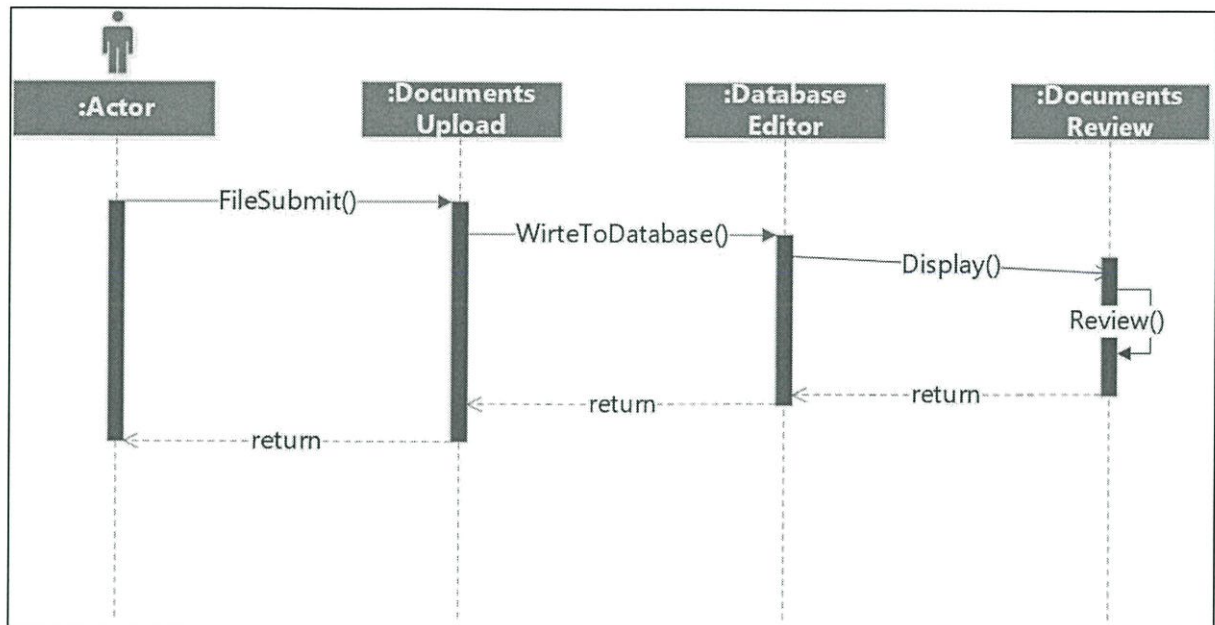
รูป 3.3 การทำงานของ Database Config

- ระบบแก้ไขฐานข้อมูล (Database Editor) เป็นระบบที่ใช้ในการอ่านหรือเขียนฐานข้อมูลของระบบ โดยส่วนนี้จะถูกเรียกใช้งานเมื่อข้อมูลหรือสถานะต่างๆในฐานข้อมูลถูกเปลี่ยนแปลง ซึ่งฟังก์ชันการทำงานนี้จะเป็ศูนย์กลางในการแก้ไขข้อมูลต่างๆที่ทุกฟังก์ชันต้องทำการเรียกใช้โดยการทำงานเป็นดังนี้



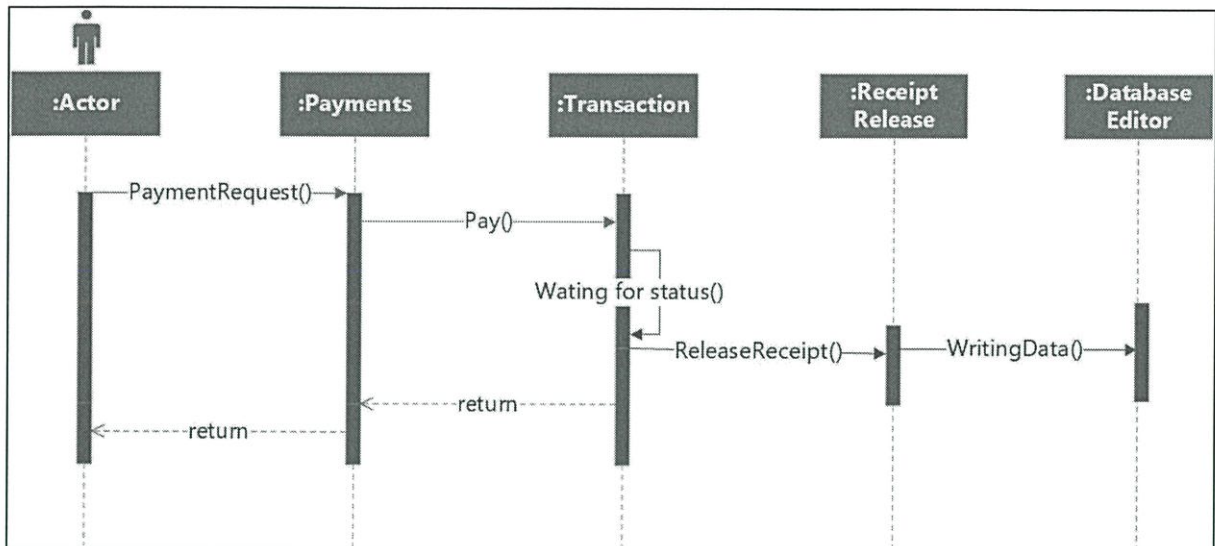
รูป 3.4 การทำงานของ Database Editor

- ระบบการสร้างงานประชุม (Create Conference) เป็นส่วนที่สามารถใช้ได้ด้วยบัญชีประเภทผู้จัดงาน ซึ่งใช้สร้างงานประชุมซึ่งผู้จัดงานสามารถที่จะกำหนดรายละเอียดต่างๆของงานประชุมได้เช่น วันที่ของการจัดงาน, อัตราค่าลงทะเบียน เป็นต้น โดยจะมีการไปเรียกใช้งานฟังก์ชันที่เกี่ยวข้องกับการเขียนข้อมูลลงฐานข้อมูล
- ระบบตรวจสอบเอกสารงานประชุม (Document Review) เป็นฟังก์ชันที่ให้ผู้เข้าร่วมงานประชุมทำการส่งเอกสารที่เกี่ยวข้องกับงานประชุมเข้ามา จากนั้นทางผู้จัดงานจะตรวจสอบเอกสารว่าผ่านการอนุมัติหรือไม่ ถ้าไม่ผ่านผู้เข้าร่วมงานก็จะต้องทำการส่งเอกสารมาเพื่อให้อนุมัติใหม่อีกครั้ง โดยฟังก์ชันนี้จะมีการเรียกใช้งานในส่วนของการทำ File Upload



รูป 3.5 การทำงานของ Document Review

- ระบบสรุปข้อมูลงานประชุม (Information Summary) จะรวบรวมข้อมูลที่เกี่ยวข้องกับงานประชุมทั้งหมดไว้ที่หน้าแรก (Dashboard) โดยจะทำหน้าที่สรุปข้อมูลสำคัญของงานประชุม เช่น จำนวนผลงานทั้งหมดที่ถูกส่ง, จำนวนผู้เข้าร่วมงาน, จำนวนผู้ที่ชำระเงินค่าลงทะเบียนเสร็จสิ้น เป็นต้น ซึ่งฟังก์ชันนี้จะไปเรียกใช้ส่วนสำหรับอ่านฐานข้อมูล
- ระบบส่งเอกสาร (Documents Upload) เป็นฟังก์ชันสำหรับใช้เพื่อ Upload ไฟล์เอกสารต่างๆ โดยเป็นฟังก์ชันที่ให้ฟังก์ชันการทำงานอื่นๆเข้ามาเรียกใช้หากมีการอัปโหลดไฟล์เอกสารต่างๆ ซึ่งจะถูกเรียกใช้เมื่อมีคำสั่ง File Submit ดังภาพที่ 3.5
- ระบบชำระเงิน (Payment) เป็นฟังก์ชันสำหรับการชำระเงินค่าลงทะเบียนต่างๆ โดยจะมีการเรียกใช้งานระบบ Transaction หรือระบบรายการเดินบัญชี สำหรับการเก็บประวัติการชำระเงินและรวบรวมข้อมูลให้กับระบบภายนอก และจะมีการเรียกใช้งานในส่วนของระบบสำหรับออกใบเสร็จรับเงิน (Receipt Release)
- ระบบสำหรับออกใบเสร็จรับเงิน (Receipt Release) ซึ่งเป็นส่วนที่สำคัญ โดยทำหน้าที่ออกใบเสร็จรับเงิน แต่ลูกค้าสามารถปรับเปลี่ยนข้อมูลได้ตามเงื่อนไข สำหรับการทำงานของฟังก์ชันนี้จะมีการ

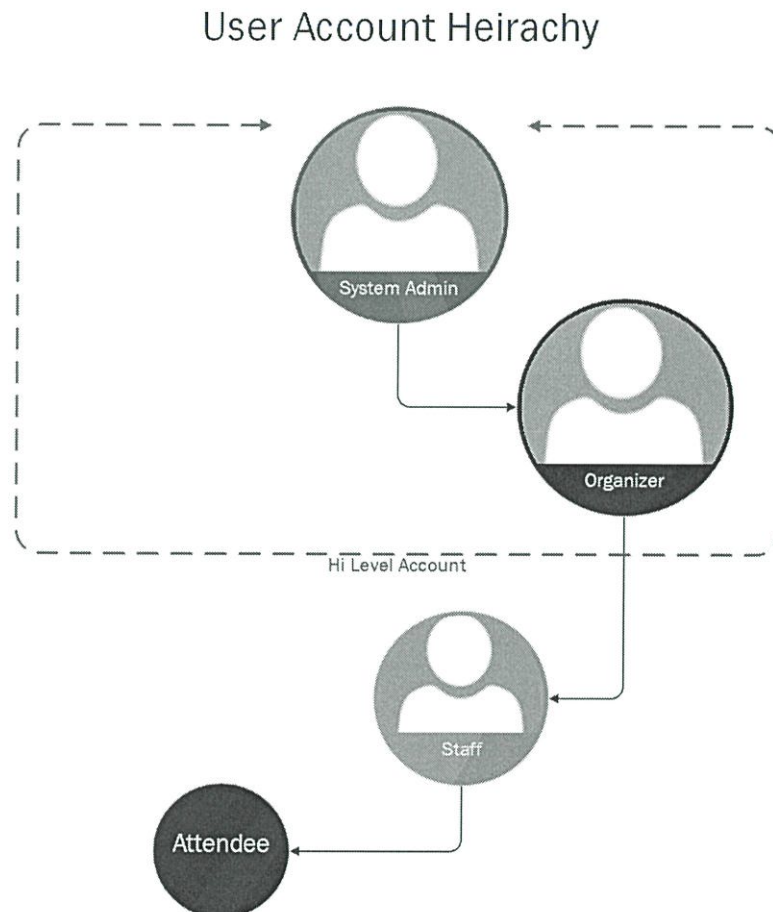


รูป 3.6 การทำงานของ Payment และ Receipt Release

- ระบบแก้ไขข้อมูล (information Changing) ใช้สำหรับให้ผู้เข้าร่วมงานแก้ไขข้อมูลส่วนตัว ข้อมูลใบเสร็จรับเงิน การขอยกเลิกเข้าร่วมงาน ข้อมูลการติดต่อ และข้อมูลสมาชิก โดยการทำงานของฟังก์ชันนี้จะคล้ายกับฟังก์ชัน Database Config คือจะมีการเรียกใช้งานฟังก์ชันที่แก้ไขฐานข้อมูลเพื่อบันทึกค่าต่างๆลงฐานข้อมูล

### 3.4 การจัดการบัญชีผู้ใช้

ตามที่ได้กล่าวไปก่อนหน้านี้ว่า ระบบนี้จะถูกจัดการโดยใช้บัญชีผู้ใช้ในระดับที่สูงขึ้น กล่าวคือ บัญชีผู้ใช้งานจะมีลำดับชั้นซึ่งทำให้สามารถจัดการระบบได้ง่าย โดยจะกำหนดลำดับชั้นของบัญชีผู้ใช้งานเป็นดังนี้



รูป 3.3 แผนภาพแสดงลำดับชั้นของบัญชีผู้ใช้งาน

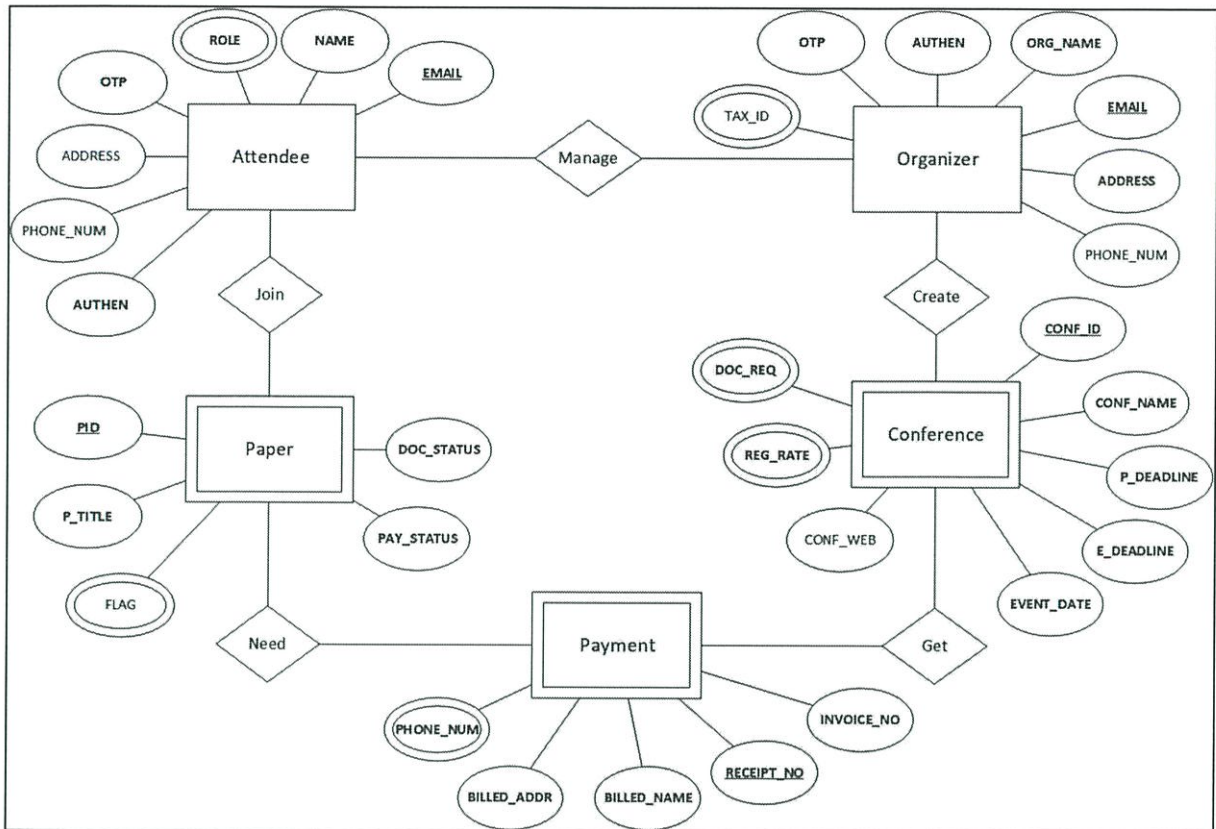
บัญชีผู้ใช้งานระดับสูงสุดของระบบนี้คือ **Administrator** ซึ่งมีบทบาทเป็น System Admin ซึ่งมีสิทธิมากที่สุด สามารถปรับเปลี่ยนและจัดการข้อมูลต่างๆทั้งแก้ไข ลบ เปลี่ยนข้อมูลได้ทั้งหมดในระบบ

บัญชีระดับที่รองลงมาคือ **Organizer** หรือผู้จัดงาน บัญชีนี้จะเปิดได้เพียง 1 บัญชีซึ่งเป็นของหน่วยงานที่จัดงานเช่น มหาวิทยาลัย, ภาครัฐหรือเอกชน โดยการเปิดบัญชีประเภทนี้จำเป็นต้องส่งเอกสารยืนยันตนมาให้ทางระบบพิจารณา ก่อน ซึ่งบัญชีระดับนี้จะสามารถปรับเปลี่ยนข้อมูลได้ทุกประการที่อยู่ภายใต้งานประชุมทางวิชาการที่ตนเองจัดขึ้น

บัญชีผู้ใช้งานระดับ **Staff** คือบัญชีที่อาจถูกสร้างขึ้นจากบัญชีของผู้จัดงาน (อาจถูกสร้างขึ้นหรือไม่ก็ได้ขึ้นอยู่กับผู้จัดงาน) ซึ่งเปรียบเสมือนบัญชีตัวแทนตามหน้าที่ที่ได้รับมอบหมาย

บัญชีผู้ใช้งานระดับ **Attendee** คือบัญชีสำหรับผู้เข้าร่วมงาน รวมถึงแขกรับเชิญที่มาร่วมงานทุกท่าน

### 3.5 การออกแบบฐานข้อมูล



รูป 3.4 แผนภาพ ER Diagram การออกแบบฐานข้อมูล

เนื่องจากระบบใช้ฐานข้อมูลที่เป็นเอกเทศ ไม่ได้เชื่อมโยงกับฐานข้อมูลเดิมของงานประชุม ดังนั้นทางระบบจึงต้องร้องขอข้อมูลตามรูปแบบที่ได้ถูกกำหนดไว้เพื่อให้ผู้จัดงานสามารถนำข้อมูลเดิมมาใช้งานได้ โดยใช้วิธีนำเข้าข้อมูล (Import) ในรูปแบบไฟล์ CSV ได้

จากแผนภาพในรูปที่ 3.5 ซึ่งเป็น ER Diagram แสดงความสัมพันธ์ระหว่างแต่ละ Entities โดยจะแบ่งเป็น 5 Entities ได้แก่ Attendee, Organizer, Conference, Payment และ Paper โดยจะทำหน้าที่เก็บข้อมูลที่เกี่ยวข้อง แต่การนำมาใช้งานจะมีการนำมาเข้าร่วมกัน (Join) แต่ละ Entity เพื่อนำ Attributes มาใช้ประกอบกัน

ฐานข้อมูลที่ระบบของเราจะใช้จะเป็นฐานข้อมูล SQL โดยใช้โปรแกรม MySQL Workbench ในการสร้างและจัดเก็บ โดยทางระบบจะแบ่งฐานข้อมูลออกเป็นส่วนดังต่อไปนี้

การตั้งค่าฐานข้อมูล ให้ทำการตั้งค่าเป็น UNICODE ก่อน เพื่อให้สามารถรองรับการจัดเก็บข้อมูลได้หลายภาษา

### 3.5.1 ฐานข้อมูลส่วนที่ใช้ในการลงชื่อเข้าใช้งานระบบ

ฐานข้อมูลส่วนนี้จะใช้ที่อยู่อีเมลที่ผู้เข้าร่วมงานลงทะเบียนไว้กับทางงานประชุมเดิม โดยลงชื่อเข้าใช้ด้วยรหัสผ่านแบบใช้ครั้งเดียว (One-Time Password) โดยมีโครงสร้างฐานข้อมูลเป็นดังนี้

ตารางที่ 3.2 โครงสร้างข้อมูลสำหรับฐานข้อมูลส่วนลงชื่อเข้าใช้

EMAIL	OTP	DIS_NAME	ROLE	AUTH
-------	-----	----------	------	------

จากตารางข้างต้นจะเห็นข้อมูลดังต่อไปนี้

ตารางที่ 3.3 คำอธิบายสำหรับฐานข้อมูลส่วนการลงชื่อเข้าใช้งานระบบ

Attributes	Data Type	Descriptions
EMAIL	varchar(45)	เก็บอีเมลที่ผู้เข้าร่วมงานที่ลงทะเบียนไว้กับงานประชุม ซึ่งจะเป็นบัญชีผู้ใช้สำหรับระบบนี้ด้วย ถึงแม้งานประชุมจะจบลงก็สามารถใช้งานด้วยอีเมลดังกล่าวได้ (ระบบจะทำการลบข้อมูลทิ้งหากไม่มีการลงทะเบียนงานประชุมในรอบ 1 ปี)
OTP	varchar(6)	เก็บรหัสผ่านสำหรับลงชื่อเข้าใช้งานระบบสำหรับแต่ละบัญชี
DIS_NAME	varchar(45)	สำหรับเก็บชื่อที่จะใช้แสดงในระบบของผู้เข้าร่วมประชุม ไม่ใช่ข้อมูลสำหรับการออกใบเสร็จรับเงินหรือข้อมูลที่เกี่ยวข้องกับงานประชุม
ROLE	varchar(12)	เก็บประเภทบัญชีผู้ใช้งาน
AUTH	boolean	เก็บค่าสถานะการลงชื่อเข้าใช้

### 3.5.2 ฐานข้อมูลส่วนที่เก็บข้อมูลของงานประชุม

สำหรับฐานข้อมูลส่วนนี้เป็นส่วนที่เก็บชื่อ และข้อมูลทั่วไปสำหรับงานประชุมทางวิชาการ โดยใช้อ้างอิงสำหรับรหัสประจำงานประชุม

ตารางที่ 3.4 โครงสร้างข้อมูลสำหรับฐานข้อมูลส่วนข้อมูลงานประชุม

CONF_ID	CONF_NAME	E_DEADLINE	P_DEADLINE	CONF_WEB
---------	-----------	------------	------------	----------

จากตารางข้างต้นจะเห็นข้อมูลดังต่อไปนี้

ตารางที่ 3.5 คำอธิบายสำหรับฐานข้อมูลส่วนข้อมูลงานประชุม

Attributes	Data Type	Descriptions
CONF_ID	varchar(45)	เก็บรหัสของงานประชุม
CONF_NAME	varchar(64)	เก็บชื่อเต็มของงานประชุม (ถ้ามี)
E_DEADLINE	timestamp	เก็บค่าวันที่วันสุดท้ายสำหรับการลงทะเบียนแบบล่วงหน้า (Early Birth)
P_DEADLINE	timestamp	เก็บค่าวันที่วันสุดท้ายสำหรับการลงทะเบียนชำระเงิน
CONF_WEB	varchar(64)	เก็บที่อยู่ของเว็บไซต์ทางการของงานประชุม

### 3.5.3 ฐานข้อมูลส่วนการกำหนดอัตราค่าลงทะเบียน

สำหรับฐานข้อมูลส่วนนี้เป็นส่วนที่ใช้ในการเก็บค่าอัตราการลงทะเบียน อัตราเข้าร่วมการลงทะเบียนในอัตราพิเศษ รวมถึงรายการเพิ่มเติมในการลงทะเบียน

ตารางที่ 3.6 โครงสร้างข้อมูลสำหรับฐานข้อมูลอัตราค่าลงทะเบียน

CONF_ID	REG_TYPE	REG_RATE	ADDITION	RATE	FLAG
---------	----------	----------	----------	------	------

จากตารางข้างต้นจะเห็นข้อมูลดังต่อไปนี้

ตารางที่ 3.7 คำอธิบายสำหรับฐานข้อมูลส่วนอัตราค่าลงทะเบียน

Attributes	Data Type	Descriptions
CONF_ID	varchar(45)	เก็บรหัสของงานประชุม
REG_TYPE	varchar(45)	เก็บประเภทการลงทะเบียนของงานประชุม
REG_RATE	varchar(45)	เก็บอัตราค่าลงทะเบียน

<b>ADDITION</b>	varchar(64)	เก็บรายการเพิ่มเติมสำหรับการลงทะเบียน เช่น บัตรรับประทานอาหาร, บัตรจอดรถเที่ยว
<b>RATE</b>	integer	เก็บอัตรารายการเพิ่มเติม
<b>FLAG</b>	varchar(45)	เก็บเงื่อนไขอัตราลงทะเบียนพิเศษ เช่น ลงทะเบียน 3 คน จ่าย 2 คน, อัตราค่าลงทะเบียนพิเศษเฉพาะ เป็นต้น

### 3.5.4 ฐานข้อมูลส่วนที่เก็บข้อมูลการลงทะเบียน

สำหรับฐานข้อมูลส่วนนี้เป็นส่วนที่ทำการแยก (Split) ออกมาจากฐานข้อมูลทำงานประชุมส่งมา โดยเป็นข้อมูลสำหรับแต่ละบัญชีผู้ใช้ซึ่งจะเก็บข้อมูลเฉพาะข้อมูลสำหรับการลงทะเบียน โดยฐานข้อมูลส่วนนี้จะเป็นส่วนหลักที่ใช้ในการประมวลผลการทำงานสำหรับแต่ละบัญชีผู้ใช้

ตารางที่ 3.8 โครงสร้างข้อมูลสำหรับฐานข้อมูลส่วนข้อมูลการลงทะเบียน

EMAIL	CONF_ID	PID	P_TITLE	ROLE	PAYMENT_STATUS	DOC_STATUS	FLAG
-------	---------	-----	---------	------	----------------	------------	------

จากตารางข้างต้นจะเห็นข้อมูลดังต่อไปนี้

ตารางที่ 3.9 คำอธิบายสำหรับฐานข้อมูลส่วนการลงทะเบียน

Attributes	Data Type	Descriptions
<b>EMAIL</b>	varchar(45)	เก็บค่าที่อยู่อีเมลที่ผู้เข้าร่วมงานลงทะเบียนไว้กับงานประชุม
<b>CONF_ID</b>	varchar(45)	เก็บรหัสของงานประชุม
<b>PID</b>	integer	เก็บรหัสผลงาน
<b>P_TITLE</b>	text	เก็บชื่อผลงาน
<b>ROLE</b>	varchar(45)	ใช้เก็บบทบาทของผู้เข้าร่วมงานเช่น Speaker (ผู้นำเสนอ), Author (ผู้เขียนผลงาน), Keynote (แขกปราชัยรับเชิญ), VIP (ผู้ทรงคุณวุฒิ, บุคคลสำคัญพิเศษ) เป็นต้น
<b>PAYMENT_STATUS</b>	boolean	เก็บสถานะการชำระเงิน

<b>DOC_STATUS</b>	boolean	สถานะการส่งเอกสารงานประชุม
<b>FLAG</b>	varchar(45)	เงื่อนไขอัตราลงทะเบียนพิเศษ เช่น ลงทะเบียน 3 คน จ่าย 2 คน, อัตรา ค่าลงทะเบียนพิเศษเฉพาะ เป็นต้น

### 3.5.5 ฐานข้อมูลส่วนที่เก็บข้อมูลส่วนบุคคล

สำหรับฐานข้อมูลส่วนนี้เป็นส่วนที่ใช้ในการเก็บข้อมูลส่วนบุคคล ซึ่งข้อมูลส่วนนี้จะใช้ในการลงทะเบียนงานประชุม และเป็นข้อมูลระบุตัวตนของผู้เข้าร่วมงาน โดยมาจกผู้จัดงานประชุมสามารถนำข้อมูลนี้ไปใช้ในการออกประกาศนียบัตร, ออกบัตรผ่านประตู เป็นต้น ที่สำคัญคือสามารถใช้ข้อมูลในส่วนนี้ออกใบเสร็จรับเงินได้

ตารางที่ 3.10 โครงสร้างข้อมูลสำหรับฐานข้อมูลส่วนบุคคล

EMAIL	TITLE	F_NAME	M_NAME	L_NAME	ADDR
M_PHONE					

จากตารางข้างต้นจะเห็นข้อมูลดังต่อไปนี้

ตารางที่ 3.11 คำอธิบายสำหรับฐานข้อมูลส่วนบุคคล

Attributes	Data Type	Descriptions
EMAIL	varchar(45)	เก็บค่าที่อยู่อีเมลที่ผู้เข้าร่วมงานลงทะเบียนไว้กับงานประชุม
TITLE	text	เก็บคำนำหน้าชื่อ
F_NAME	interger	เก็บชื่อจริง
M_NAME	varchar(64)	เก็บชื่อกลาง
L_NAME	varchar(64)	เก็บนามสกุล
ADDR	text	เก็บที่อยู่
M_PHONE	interger	เก็บหมายเลขโทรศัพท์

### 3.5.6 ฐานข้อมูลส่วนการออกใบเสร็จ

สำหรับฐานข้อมูลส่วนนี้เป็นส่วนที่ใช้ในการเก็บข้อมูลใบเสร็จรับเงิน เอกสารการเรียกเก็บเงินต่างๆ

ตารางที่ 3.12 โครงสร้างข้อมูลสำหรับฐานข้อมูลการออกใบเสร็จ

EMAIL	INVOICE_NO	RECEIPT_NO	BILLED_NAME	ADDR	PHONE	TAX_ID
-------	------------	------------	-------------	------	-------	--------

จากตารางข้างต้นจะเห็นข้อมูลดังต่อไปนี้

ตารางที่ 3.13 คำอธิบายสำหรับฐานข้อมูลส่วนข้อมูลส่วนบุคคล

Attributes	Data Type	Descriptions
EMAIL	varchar(45)	เก็บค่าที่อยู่อีเมลที่ผู้เข้าร่วมงานลงทะเบียนไว้กับงานประชุม
INVOICE_NO	interger	เก็บเลขที่ใบเรียกเก็บเงิน
RECEIPT_NO	interger	เก็บเลขที่ใบเสร็จรับเงิน
BILLED_NAME	text	เก็บชื่อที่ใช้สำหรับออกใบเสร็จรับเงิน
ADDR	text	เก็บที่อยู่สำหรับออกใบเสร็จรับเงิน
PHONE	interger	เก็บหมายเลขโทรศัพท์
TAX_ID	interger	เก็บหมายเลขประจำตัวผู้เสียภาษี

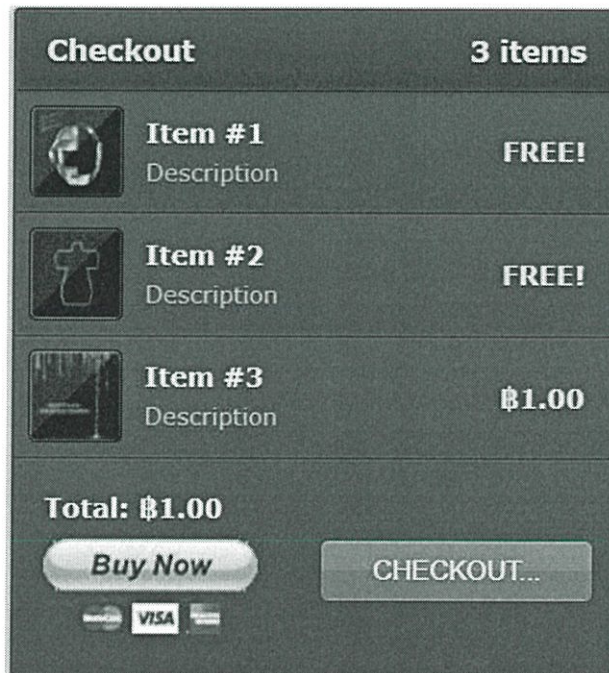
## บทที่ 4

### การทดสอบและความสามารถของระบบ

ในส่วนนี้จะแสดงให้เห็นถึงการทดสอบบางฟังก์ชันการทำงานของระบบที่ได้ทำการทดสอบไปแล้วและความสามารถของระบบที่ได้พัฒนาให้สามารถใช้งานได้ ซึ่งได้ผลลัพธ์ดังนี้

#### 4.1 การทดสอบการจ่ายเงินผ่านระบบ K-Payment Gateway

ในการทดสอบนี้จะทำการสร้างหน้าเว็บไซต์สำหรับทดสอบขึ้นมาแล้วทำการเชื่อมต่อหน้าเว็บไซต์ที่ทดสอบเข้ากับระบบ Payment Gateway ของธนาคารกสิกรไทย โดยมี API ของทางธนาคารดังนี้



รูป 4.1 ส่วนที่ใช้ทดสอบสำหรับระบบทดลองจ่ายเงิน

ส่วนที่ใช้เชื่อมต่อเข้ากับระบบ Payment Gateway ของธนาคารนั้นจะใช้ปุ่ม Checkout ที่ีเขียวทางด้านขวามือ เมื่อกดปุ่มนี้ไปแล้วจะทำการเปลี่ยนหน้าไปยังส่วนของฟอร์มกรอกข้อมูลดังภาพที่ 4.2





---

**Payment Detail**







Verified by  

Your payment is now being processed on K-Payment Gateway, an online payment system by **KASIKORNBANK**. K-Payment Gateway is the highly secure payment processor for your seller, **WWW.ISTS2012.KMITL.AC.TH**. Please verify the purchase information and enter your Credit/Debit card details below.

**1** Payment Detail

**2** Payment Confirmation

**3** Payment Result

<b>Pay To :</b>	WWW.ISTS2012.KMITL.AC.TH
<b>Invoice No :</b>	000009999999
<b>Payment For :</b>	TEST PAYMENT SYSTEM!!
<b>Amount :</b>	1.00 Baht

---

<b>Card Type :</b>	<input type="text" value="Please Select One"/>
<b>Credit/Debit Card Number :</b>	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>
<b>Security Code (CVV2) :</b>	<input type="text"/> Last 3 digits on the back of the card
<b>Expiry Date (mm/yyyy) :</b>	<input type="text" value="MM"/> <input type="text" value="YYYY"/>
<b>Card Holder Name :</b>	<input type="text"/>
<b>Country of Issuing Bank :</b>	<input type="text" value="THAILAND"/>
<b>Issuing Bank :</b>	<input checked="" type="radio"/> <input type="text" value="Please Select One"/>
<b>Not in List (Please Specify) :</b>	<input type="radio"/> <input type="text"/>






Remarks : Your Credit/Debit card information is strictly secured under KBank's security system. The online merchants will see only the first and last Four digits of your Credit/Debit card number with no expiration date nor the CVV2 code.

บริการทุกระดับประทับใจ

รูป 4.2 ส่วนสำหรับกรอกข้อมูลเพื่อส่งให้ธนาคาร

จากนั้นก็ทำการกรอกข้อมูลบัตรเครดิตและเมื่อกดปุ่ม Pay Now ก็จะทำการส่งข้อมูลไปให้ธนาคารตัดเงินจากบัตร จากนั้นระบบจะทำการให้ผู้ใช้กรอกรหัส OTP






**เพิ่มความปลอดภัยในการทำรายการชำระเงินออนไลน์ด้วย Verified by VISA**  
**Added Protection with Verified by Visa**

โปรดตรวจสอบความถูกต้องของข้อความยืนยันส่วนตัวของท่าน จากนั้นระบุรหัสผ่าน Verified by Visa เพื่อเพิ่มความปลอดภัยในการทำรายการชำระเงินนี้

Please verify your Personal Message and submit your Verified by Visa password.  
 You need to enter your Verified by Visa password in order to complete the transaction.

Merchant: WWW.ISTS2012.KMITL.AC.TH  
 Amount: 1.00 THB  
 Date: 23/11/2017  
 Card Number: \*\*\*\* \* 1204  
 Personal Message: APRCRD

กรุณากดปุ่ม "Request OTP" เพื่อรับรหัสรักษาความปลอดภัย SMS-OTP  
 Please click "Request OTP" to receive SMS-OTP password

กรณารับรหัสรักษาความปลอดภัย SMS-OTP ที่ได้รับ  :  Request OTP

Please input SMS-OTP password

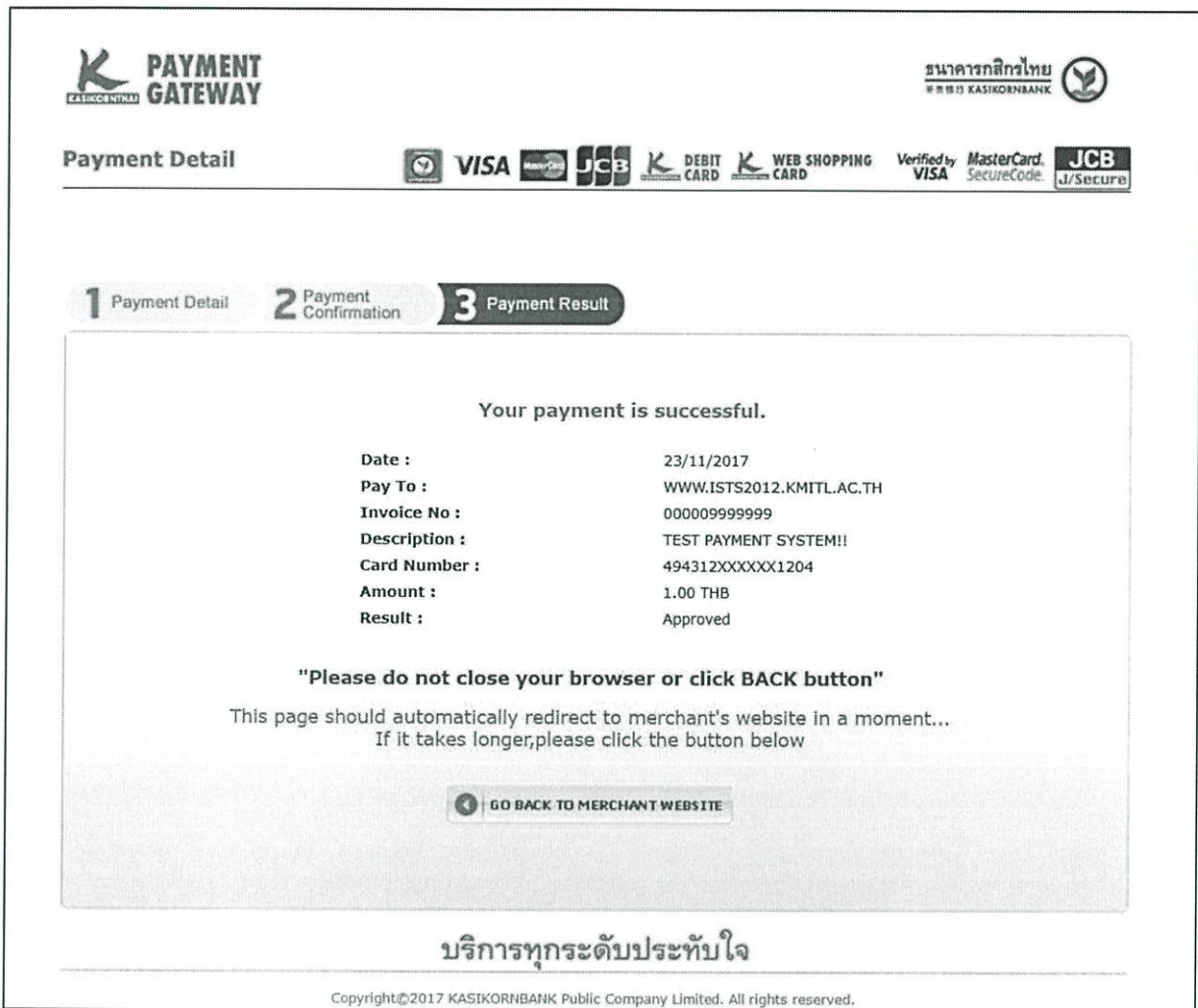
ตกลง / Submit
ยกเลิก / Exit

[เปลี่ยนรหัสรักษาความปลอดภัยเป็นแบบธรรมดา](#)  
 Change to static password

© Kasikornbank PCL 2010. All rights reserved.  
[Terms & Conditions](#) [Privacy Policy](#)

รูป 4.3 ส่วนสำหรับกรอกรหัส OTP ของระบบ Payment Gateway

เมื่อผู้ใช้กรอกรหัส OTP ถูกต้อง ระบบก็จะทำการสรุปข้อมูลและเมื่อผู้ใช้กดปุ่มยืนยัน ระบบจ่ายเงินให้กับทางบัญชีที่ทางระบบสมัครไว้กับทางธนาคาร



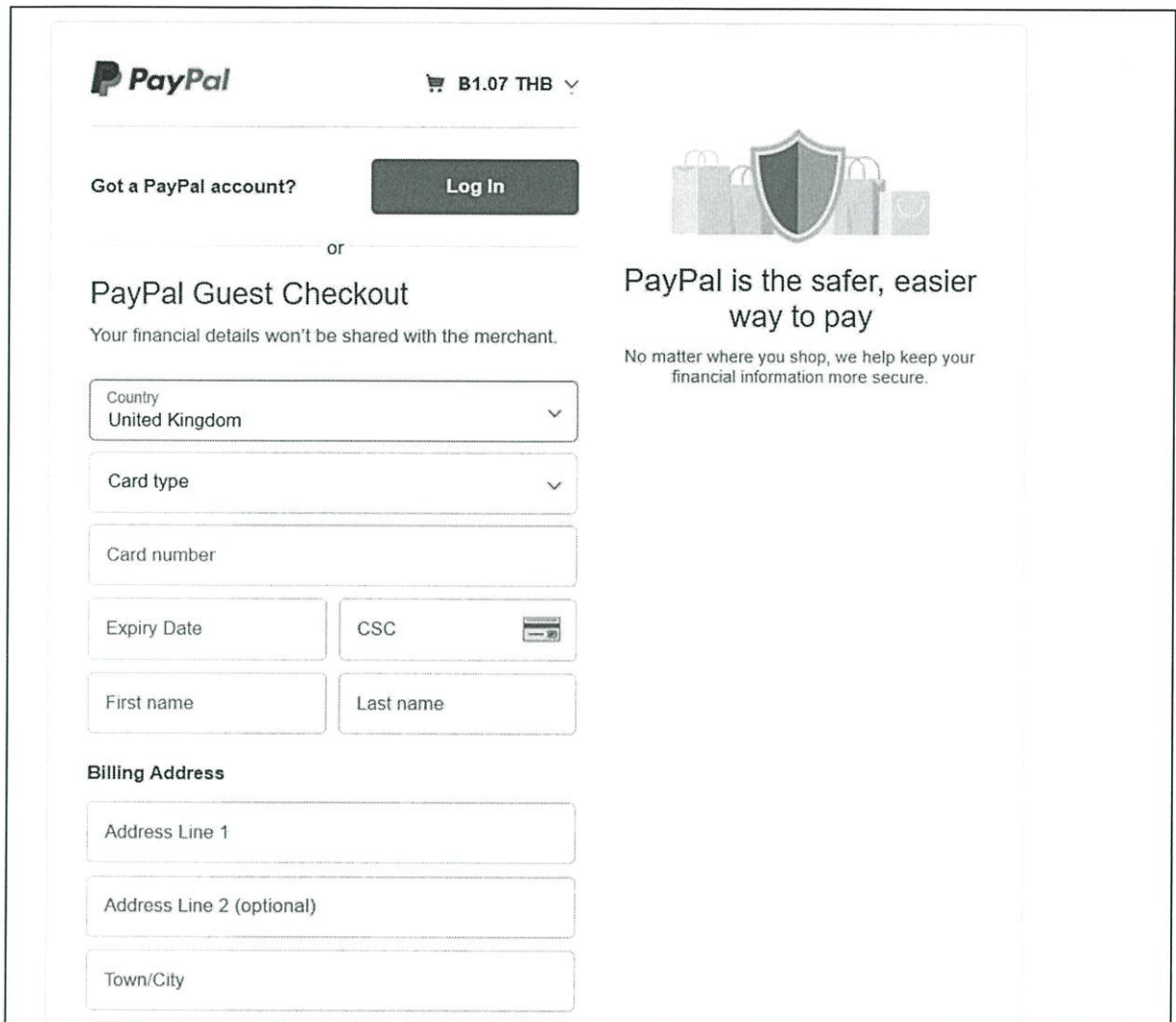
รูป 4.4 ส่วนแสดงผลการชำระเงิน

เมื่อถึงขั้นตอนนี้ พบปัญหาที่สำคัญประการหนึ่งคือระบบที่เราออกแบบจะทราบผลการจ่ายเงินได้ก็ต่อเมื่อผู้ใช้งานต้องกดปุ่ม Go Back to Merchant Website เท่านั้น หากผู้ใช้ไม่ทำการกดปุ่มจะเกิดปัญหาที่ระบบไม่ทราบผลการชำระเงินจากผู้ใช้งาน

#### 4.2 การทดสอบการจ่ายเงินผ่านระบบ PayPal

การชำระเงินผ่านช่องทางนี้ ผู้ใช้งานต้องเปิดบัญชีกับทาง PayPal ก่อน หรือสามารถที่จะกรอกข้อมูลบัตรเครดิตก็สามารถทำได้ดังนี้

เมื่อผู้ใช้งานกดปุ่มไปที่ Buy Now จากภาพที่ 4.1 ระบบจะทำการพาผู้ใช้งานไปยังหน้าของระบบกรอกข้อมูลช่องทาง PayPal ดังภาพ



**PayPal** 🛒 B1.07 THB

Got a PayPal account? Log In

or

### PayPal Guest Checkout

Your financial details won't be shared with the merchant.

Country  
United Kingdom

Card type

Card number

Expiry Date

CSC

First name


Last name

**Billing Address**

Address Line 1

Address Line 2 (optional)

Town/City



**PayPal is the safer, easier way to pay**

No matter where you shop, we help keep your financial information more secure.

รูป 4.5 หน้าเว็บสำหรับเลือกช่องทางการชำระเงินผ่าน PayPal

หากผู้ใช้มีบัญชีกับทาง PayPal ก็สามารถลงชื่อเข้าใช้ได้ทันที หากไม่มีก็สามารถทำการสมัครสมาชิกก่อนได้ หากไม่มีบัญชีก็สามารถชำระเงินโดยผ่านบัตรเครดิตได้โดยทำการกรอกข้อมูลใหม่

### Pay with debit or credit card

Your financial details won't be shared with the merchant.

Country  
United Kingdom

Card type

Card number

Expiry Date      CSC

First name      Last name

Phone number type  
Mobile      Mobile number  
+44

**Billing Address**


Address Line 1

Address Line 2 (optional)

Town/City

County (optional)      Postcode

Send to my billing address



## PayPal is the safer, easier way to pay

No matter where you shop, we help keep your financial information more secure.

รูป 4.6 ภาพแสดงส่วนกรอกข้อมูลบัตรเครดิตของระบบ PayPal

เมื่อชำระเงินสำเร็จ ในบัญชี PayPal ของร้านค้าที่จะแสดงข้อมูลการชำระเงินขึ้นมาดังภาพ

Summary Money Activity Reports Tools More

Profile Log out

---

#### Transaction details

**Payment received from Sirawit Wanarattikal**  
 13 October 2017 at 00:13:27 GMT+7      Transaction ID: 0F050711FU4074400  
 Payment status: COMPLETED

**OK to ship to**  
 Sirawit Wanarattikal  
 94/6 Ramintra 8 alley  
 Ramintra Rd  
 Bang Khen, BANGKOK 10220  
 Thailand  
 Confirmed

Ship | Mark as shipped | Add tracking info |  
 Print packing slip

**Order details**

Order details	Quantity	Price	Subtotal
Test Payment	1	฿1.00 THB	฿1.00 THB
		Purchase total	฿1.00 THB

**Payment details**

Purchase total      ฿1.00 THB

**Gross amount**  
**฿1.00 THB**

**Seller protection**  
 Eligible  
 Eligible when you ...  
 Ship to the address on this page.  
 Save your tracking or shipping info.  
 Follow the requirements for seller protection.

Can't find something? For a limited time you can still use the Classic view of transaction details

## รูป 4.7 ภาพแสดงผลลัพธ์การชำระเงิน

### 4.3 การทดสอบการทำงานในส่วนการออกใบเสร็จรับเงิน

เป็นการทำงานฝั่ง sever โดยใช้ node.js ร่วมกันฐานข้อมูล SQL โดยใช้ PDF Kit ในการออกใบเสร็จจากตัวอย่าง database ด้านล่างนำข้อมูลมาใส่ในใบเสร็จ

#	receipt	invoice	pid	pid_title	bill_name	registration_for_author	country	registration
1	0	100	135	Extension of Quadratic Means for ...	Kantaphon Torat	Kantaphon Torat	Thailand	Kantaphon
2	0	16	68	Personality Traits Analysis From Fa...	Aman Shakya	Aman Shakya	Nepa	Aman Sh
3	1	3	113	Hybrid Node Development for Distr...	Yi Yang	Yi Yang	China	Yi Yang
4	2	4	18	The new Equation for RSA?s Decr...	Kritsanapong Somsuk	Kritsanapong Somsuk	Thailand	Kritsanap
5	4	11	105	Corn Disease Identification from Le...	Mahidol Wittayanusorn School	Pichayoot Cuppaphan	Thailand	Pichayoot
6	6	15	17	Stock Selection by using an improv...	Mr. Dit Suthiwong	Dit Suthiwong	Thailand	Dit Suthiv
7	7	13	59	Faster Line of Sight Computation a...	Sanjeeb Prasad Panday	Sanjeeb Prasad Panday	Nepa	Sanjeeb F
8	9	19	94	Smart Security Guard Scheduling ...	Cheng Jiang	Cheng Jiang	China	Cheng Ji
9	11	21	114	Alignment Free Power Transfer Sy...	Franklin Bien	Franklin Bien	Korea, South	Nak-Youn
10	12	28	11	Complexity Reduction on API Call ...	Thotsaphon Tungjitviboonkun	Thotsaphon Tungjitviboonkun	Thailand	Thotsaph
11	13	32	52	Clearance Estimation Through Mob...	Puttipong Leakkaw	Puttipong Leakkaw	Thailand	Puttipong
12	14	35	10	The Fuzzy Scheduling Algorithm fo...	Pimsiri Tubthong	Pimsiri Tubthong	Thailand	Pimsiri Tu
13	15	37	29	3D Pickup Customization System	Chonlachat Srisompetch	Chonlachat Srisompetch	Thailand	Pongsago
14	17	8	33	POSITION CONTROL FOR OBJE...	Mr.Napassadol Singhata	Napassadol Singhata	Thailand	Napassad

## รูป 4.8 ตัวอย่างฐานข้อมูลจากงาน ICSEC 2017

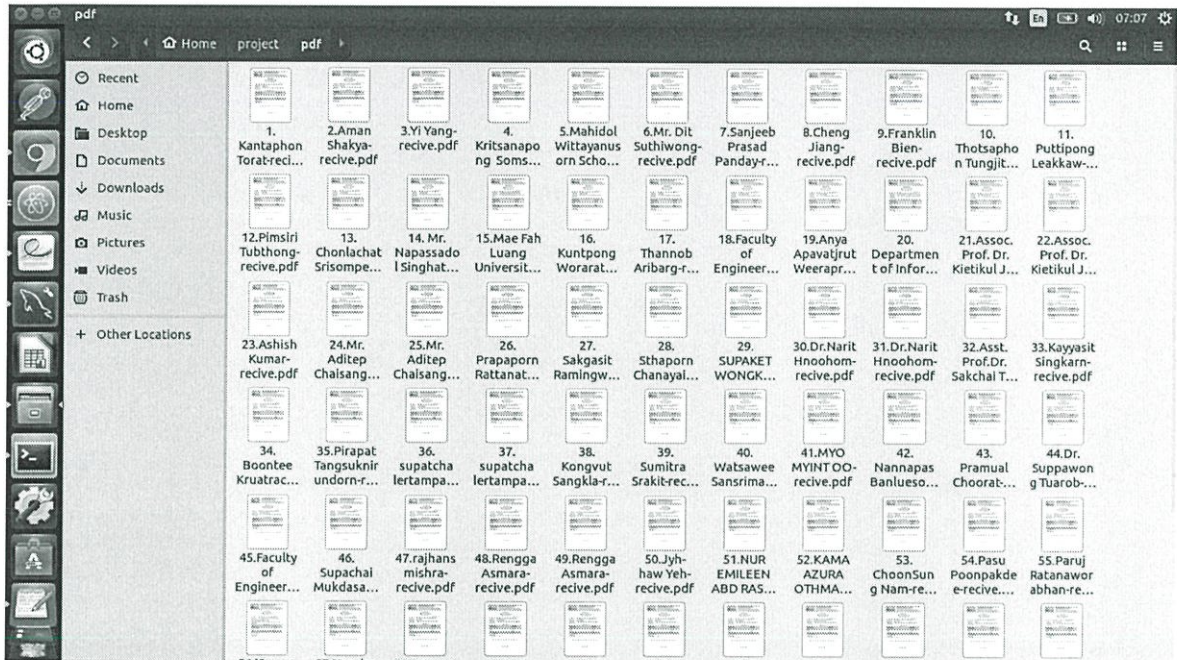
```

thnatcha@thnatcha-X550JX:~/project/nodejsFristproject$ node app1.js
[ RowDataPacket {
  receipt: 0,
  invoice: 100,
  pid: 135,
  pid_title: 'Extension of Quadratic Means for Weighted Centroid Localization with ZigBee Technology',
  bill_name: 'Kantaphon Torat',
  registration_for_author: 'Kantaphon Torat',
  country: 'Thailand',
  registration_by: 'Kantaphon Torat',
  registration_type: 'Non Member',
  payment_method: 'TRF',
  additional_item1: 0,
  total_payment: 14000,
  announce_date: '0000-00-00 00:00:00',
  finance_status: 'Approved' },
  RowDataPacket {
  receipt: 0,
  invoice: 16,
  pid: 68,
  pid_title: 'Personality Traits Analysis From Facebook Data',
  bill_name: 'Aman Shakya',
  registration_for_author: 'Aman Shakya',
  country: 'Nepa',
  registration_by: 'Aman Shakya',
  registration_type: 'Non Member',
  payment_method: 'TRF',
  additional_item1: 0,
  total_payment: 12000,
  announce_date: '0000-00-00 00:00:00',
  finance_status: 'Approved' },
  RowDataPacket {
  receipt: 1,
  invoice: 3,
  pid: 113,
  pid_title: 'Hybrid Node Development for Distributed Agriculture Service System in Beijing-Tianjin-Hebei Urbanization Region',
  bill_name: 'Yi Yang',
  registration_for_author: 'Yi Yang',
  country: 'China',
  registration_by: 'Yi Yang',
  registration_type: 'Non Member',
  payment_method: 'CRD',
  additional_item1: 0,

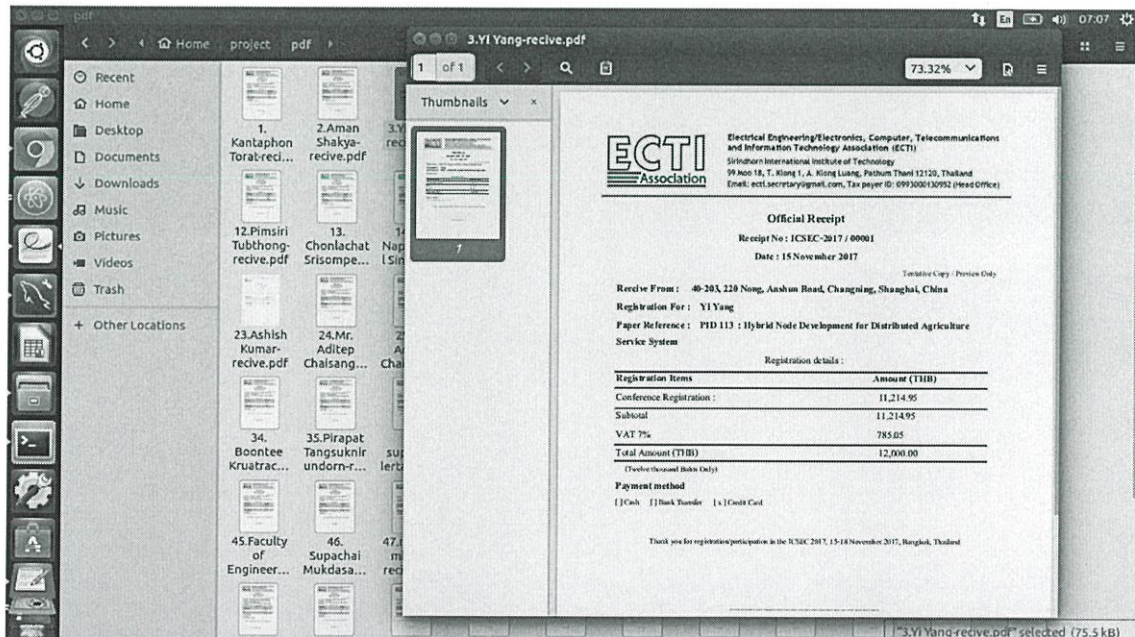
```

## รูป 4.9 ตัวอย่างการรันโปรแกรม ด้วยคำสั่ง node app1.js

จากคำสั่งดังกล่าวผู้จัดทำให้แสดงเนื้อหาข้อมูล database ทั้งหมดที่ดึงมาให้มาแสดงใน terminal พร้อมกับนำข้อมูลที่ได้ออกไปพิมพ์ใบเสร็จทั้งหมด



รูป 4.10 ผลลัพธ์การออก ใบเสร็จเป็นไฟล์ pdf



รูป 4.11 ใบเสร็จที่สร้างขึ้นจากการทดสอบระบบ

#### 4.4 การทดสอบทำงานเชื่อมต่อฝั่ง client และฝั่ง server ผ่าน API

ขั้นตอนที่ 1 รันโปรแกรมฝั่ง sever ผ่าน terminal ด้วย node.js ด้วยคำสั่งดังนี้

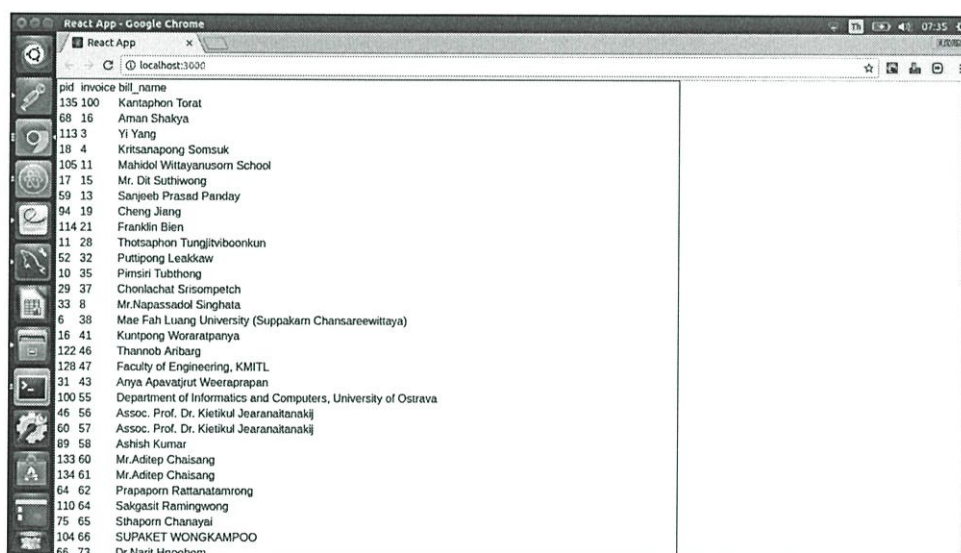
```

thanatcha@thanatcha-X550JX:~/project/NodeRestful-master$ npm start
> noderestful@1.0.0 start /home/thanatcha/project/NodeRestful-master
> node server.js

Server running at: http://localhost:8000
  
```

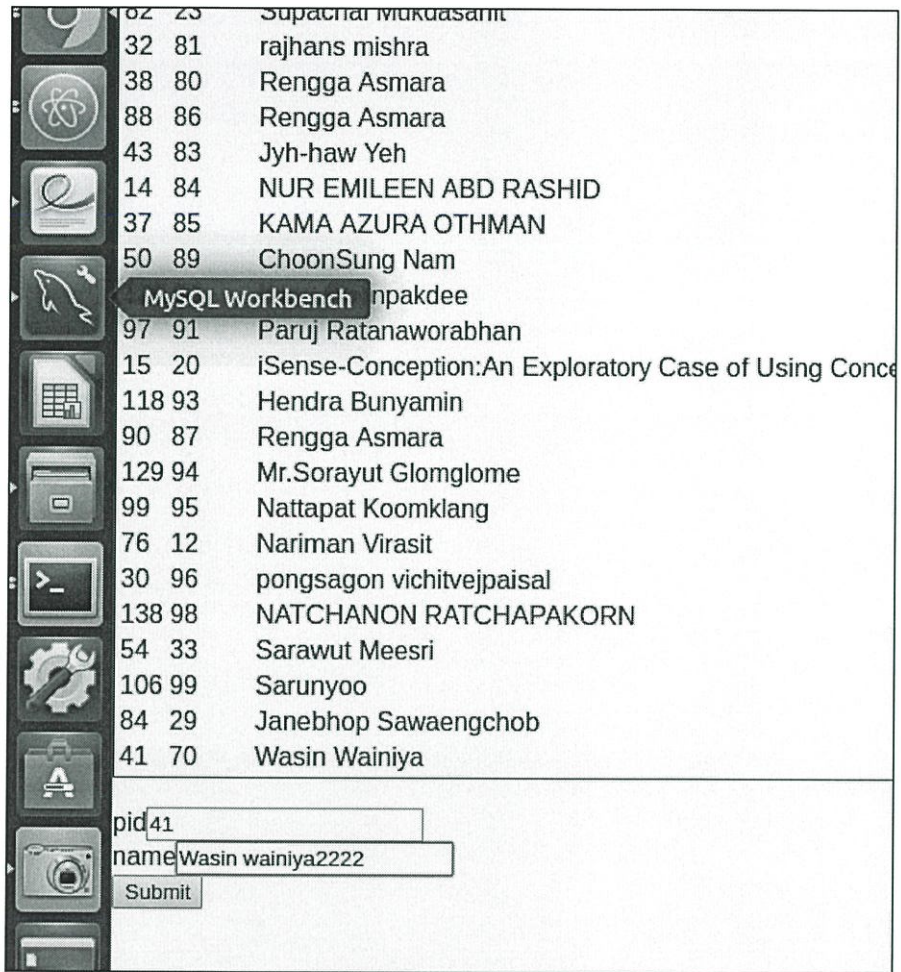
รูป 4.12 คำสั่ง localhost server เพื่อให้ Server ทำงาน

ขั้นตอนที่ 2 รันโปรแกรมฝั่ง client ที่พัฒนามาจาก react ผ่าน terminal ด้วยคำสั่ง npm start ที่ โฟล์เดอร์ my-app จะได้เว็บไซต์หน้าตาเป็นดังภาพ



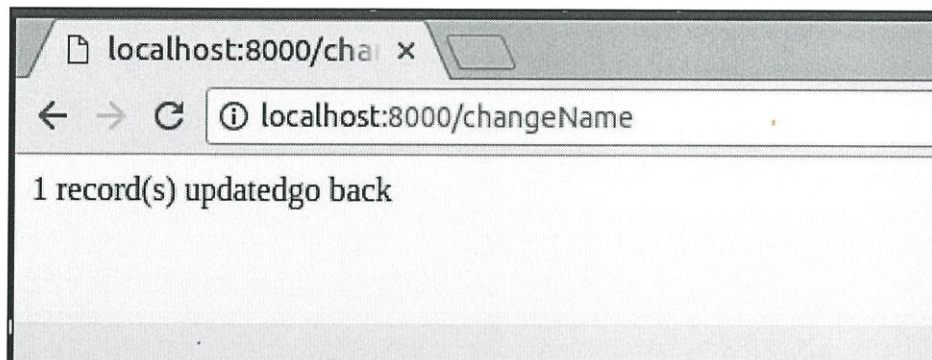
รูป 4.13 ผลการรัน React เชื่อมกับ server เพื่อดึงข้อมูลจาก database

โดยขั้นตอนดังกล่าวจะเป็นการดึงเอา pid invoice และ bill\_name จากฐานข้อมูลมาแสดงเป็นตารางให้เราเห็นซึ่งเป็นวิธีการ get ข้อมูลมาจาก database ผ่าน Restful API ที่ถูกสร้างมาจาก node.js

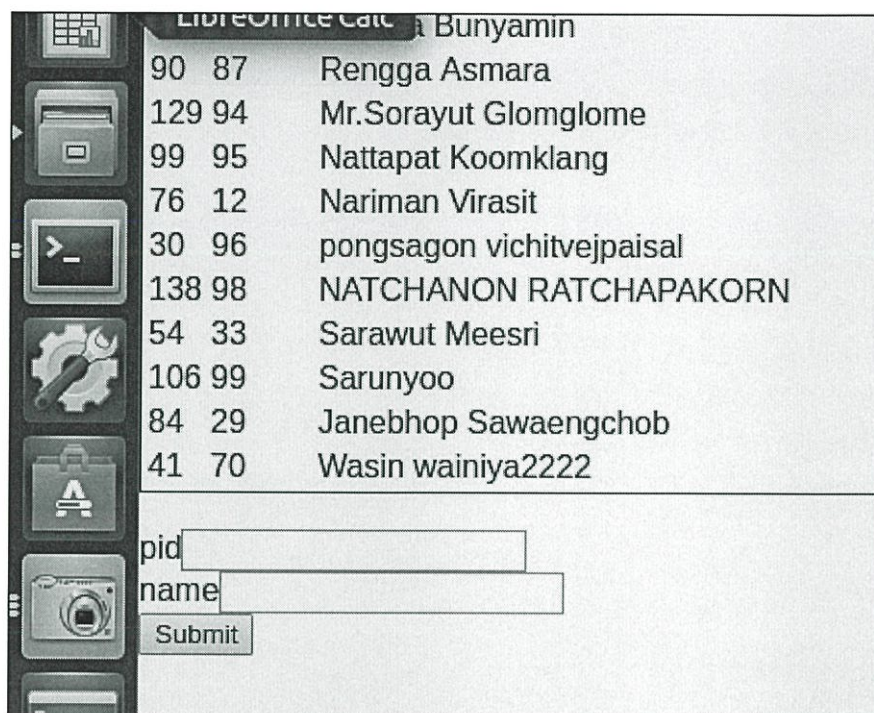


รูป 4.14 ตัวอย่างการ post แก้ไขชื่อจาก pid ก่อนกดปุ่ม submit

ต่อมาจะเป็นตัวอย่างการ post ข้อมูลเข้าไปแก้ไข database ด้วยการกรอกฟอร์ม pid และ bill\_name แล้วระบบจะทำการแก้ไขชื่อ จากเลข pid ที่กรอกไปให้ โดยในรูปจะเป็นการเข้าไปแก้ไขชื่อใน pid หมายเลข 41 ให้ wasin wainiya เป็น wasin wainiya2222



รูป 4.15 หลังจากกดปุ่ม submit แล้ว



รูป 4.14 ฐานข้อมูลที่เปลี่ยนไปเมื่อทำการแก้ไขข้อมูล

จากการทดสอบการเชื่อมต่อระบบที่เขียนขึ้น โดยใช้ React เข้ากับฐานข้อมูลที่ให้บริการของ myPHP ซึ่งทำการเชื่อมต่อผ่าน API พบว่าสามารถใช้งานแทนการเขียนด้วยภาษา php แบบเดิมได้ และสามารถที่จะทำการแก้ไขเฉพาะส่วนของฐานข้อมูลได้โดยแก้ไขผ่าน API

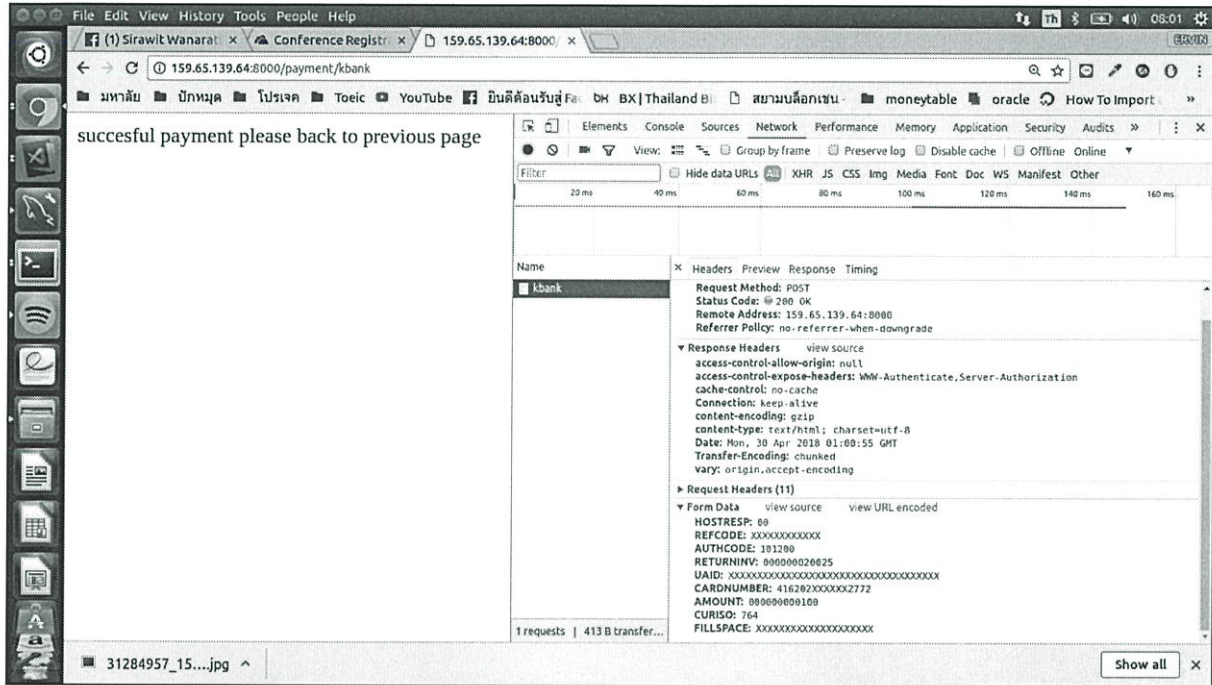
โดยระบบนี้จะนำไปใช้งานในส่วนของการเปลี่ยนแปลงข้อมูลต่างๆในฐานข้อมูล เช่นเปลี่ยนชื่อผู้เข้าร่วมงาน, เปลี่ยนชื่อหรือที่อยู่ใบเสร็จรับเงิน, ลบข้อมูลออกจากฐานข้อมูลหรือเปลี่ยนแปลงสถานะต่างๆ เป็นต้น

#### 4.5 ความสามารถของระบบ

จากการดำเนินงานทั้งหมดทางผู้จัดทำสามารถพัฒนาระบบให้สามารถใช้งานได้จริงซึ่งปัจจุบันทางผู้จัดทำสามารถทำให้ใช้งานได้ 1 งานประชุมก่อน โดยระบบที่พัฒนาขึ้นมาสามารถใช้งานได้ตามความสามารถดังต่อไปนี้

##### 4.5.1 รองรับการชำระเงินผ่านบัตรเครดิตและเครดิต

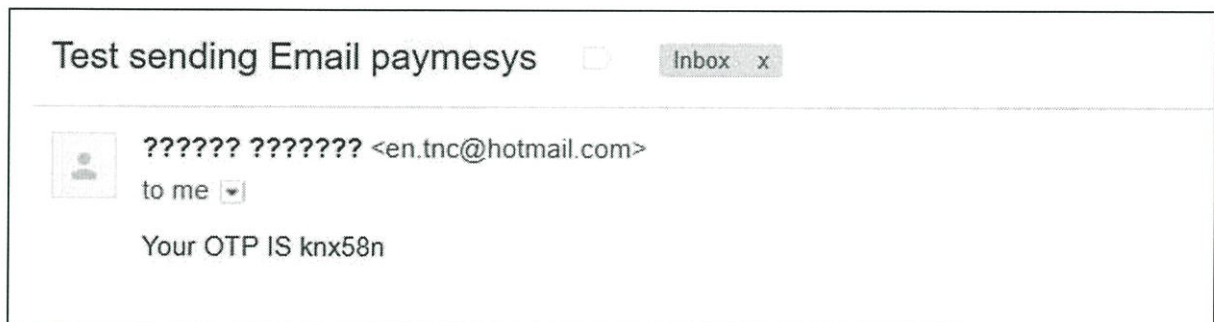
ระบบรองรับการชำระเงินผ่านบัตรเครดิตและบัตรเดบิตได้โดยใช้งานผ่านช่องทาง K-Payment Gateway และ PayPal ได้ และระบบสามารถเก็บค่าสถานะการชำระเงินที่ถูกส่งกลับมาได้ และจัดเก็บลงฐานข้อมูลเพื่อให้ผู้ใช้งานระบบสามารถทราบสถานะได้ทันที



รูป 4.15 ข้อมูลที่ถูกส่งกลับมาเมื่อชำระเงินสำเร็จ

#### 4.5.2 การลงชื่อเข้าใช้งานระบบด้วยรหัส OTP

ในการลงชื่อเข้าใช้งานระบบ ไม่จำเป็นต้องสมัครบัญชีผู้ใช้ใหม่ แต่สามารถลงชื่อเข้าใช้งานโดยใช้อีเมลที่สมัครไว้กับทางงานประชุมแล้วระบบจะจัดส่งรหัส OTP ให้สามารถลงชื่อเข้าใช้งานได้



รูป 4.16 รหัส OTP ที่ถูกส่งเข้าอีเมลเมื่อลงชื่อเข้าใช้งานระบบ

### 4.5.3 ระบบสำหรับสร้างงานประชุม

รูป 4.17 หน้าจอสำหรับการสร้างงานประชุม

เป็นฟังก์ชันการทำงานซึ่งใช้สำหรับสร้างงานประชุมขึ้นมาแล้วสามารถนำเข้าฐานข้อมูลเดิมได้โดยใช้ไฟล์ csv ผู้ใช้งานสามารถกำหนดค่าวันที่ต่างๆ, อัตราค่าลงทะเบียนเข้าร่วมงาน, ภาพ Logo, รวมถึงรหัสและชื่องานประชุมได้

### 4.5.4 ระบบสำหรับอัปโหลดไฟล์

เป็นระบบที่ใช้สำหรับอัปโหลดไฟล์เอกสารต่างๆหรือรูปภาพเข้าสู่ Server โดยฐานข้อมูลจะทำหน้าที่เก็บชื่อไฟล์ไว้เพื่อเชื่อมโยงไปยังที่เก็บไฟล์ใน Server ซึ่งเปรียบเสมือน Pointer ที่ไปยังตำแหน่งของไฟล์ ซึ่งส่วนนี้จะใช้ในส่วนของการ Upload เอกสารงานประชุม, หลักฐานการชำระเงินต่างๆ

### Documents Upload

Upload documents required by conference.

**Warning!!!**  
Don't forget to upload conference's documents.

**Conference:**

ICSEC2017

**Paper ID:**

CE6033

Conference Logo HERE

Response to Committee Comment	<input type="button" value="Choose File"/> No file chosen
IEEE Copyright Form	<input type="button" value="Choose File"/> No file chosen
Speaker Biology Form	<input type="button" value="Choose File"/> No file chosen
Camera Ready (Paper Complete Edition)	<input type="button" value="Choose File"/> No file chosen


DOCUMENTS STATUS

- Response to Committee Comment
- IEEE Copyright Form
- Speaker Biology Form
- Camera Ready (Complete Edition of Paper)

รูป 4.18 หน้าจอส่วนการอัปโหลดเอกสารงานประชุม

#### 4.5.5 ระบบสำหรับแสดงรายการเดินบัญชี (Statement)

เป็นระบบที่ใช้แสดงรายการชำระเงินของผู้เข้าร่วมงาน ซึ่งผู้จัดงานสามารถเข้ามาตรวจสอบได้ ซึ่งจะแสดงชื่อผู้จ่าย, อีเมลล์ที่ชำระเงิน, รหัสผลงาน, หมายเลขคำสั่งซื้อ, เลขที่ใบเสร็จรับเงิน, ช่องทางที่ชำระเงิน และจำนวนเงิน

 **Statement**

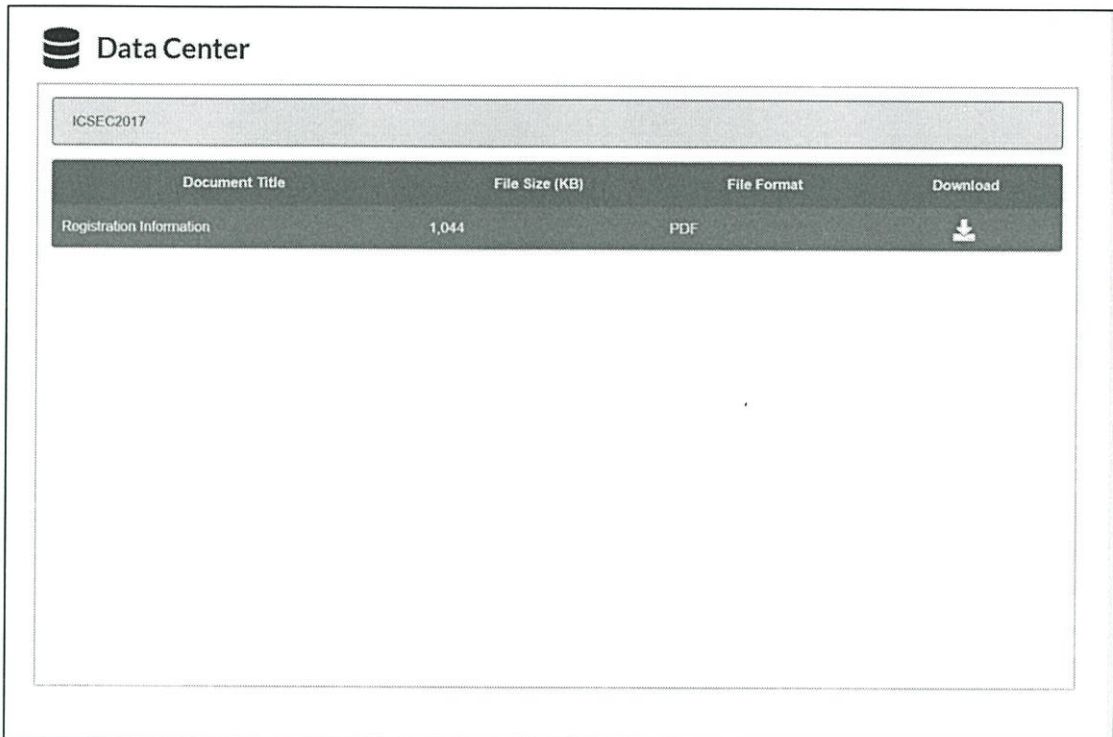
ICSEC2017

PID	Email	Name	Invoice No.	Receipt No.	PayMethod	conference	Amount
25	en.tnc@hotmail.com	sirawit Wanarattikal	214	204	credit card	ICSEC2017	1
23	kwee@hotmail.com	savitree boonmee	205	189	credit card	ICSEC2017	1
18	walinee@hotmail.com	wavitree meekhun	202	185	unpaid	ICANN2017	0
23	kwee@hotmail.com	savitree boonmee	205	189	unpaid	ICANN2017	0
24	en.tnc@hotmail.com	Thianatcha sangphet	203	189	unpaid	ICSEC2017	0


รูป 4.19 หน้าจอส่วนแสดงรายการเดินบัญชี

#### 4.5.6 ระบบสำหรับดาวน์โหลดเอกสาร

เป็นส่วนที่รวบรวมข้อมูลทั้งหมดของงานประชุม ซึ่งผู้จัดงานสามารถดาวน์โหลดมาใช้งานได้โดยจะแสดงชื่อไฟล์, ประเภทของไฟล์ และขนาดของไฟล์



The screenshot shows a web interface for a Data Center. At the top left, there is a logo consisting of three stacked cylinders and the text "Data Center". Below this, there is a search bar containing the text "ICSEC2017". Underneath the search bar is a table with the following structure:

Document Title	File Size (KB)	File Format	Download
Registration Information	1,044	PDF	

#### รูป 4.20 หน้าจอส่วนการดาวน์โหลดเอกสาร

ความสามารถของระบบที่กล่าวมาเป็นเพียงความสามารถบางส่วนที่ระบบสามารถทำได้ ซึ่งสามารถดูวิธีใช้งานระบบและความสามารถของระบบอื่นๆ ได้ที่ภาคผนวกท้ายเล่ม

## บทที่ 5

# สรุปผลและข้อเสนอแนะ

### 5.1 สรุปผลการดำเนินงาน

ระบบที่ได้พัฒนาขึ้นมาเป็นระบบที่สามารถอำนวยความสะดวกให้ทั้งผู้เข้าร่วมงานและทางผู้จัดงานได้ ซึ่งจากการพัฒนาระบบในปัจจุบันนี้สามารถรองรับงานประชุมได้เพียง 1 งานประชุมก่อนในเบื้องต้น ระบบสามารถรับชำระเงินได้และเก็บสถานะการชำระเงินเข้าสู่ฐานข้อมูลได้, สามารถรองรับการ Upload ไฟล์เอกสารที่จำเป็นต่างๆได้, สามารถลงชื่อเข้าใช้ด้วยรหัส OTP ได้โดยไม่ต้องสมัครบัญชีผู้ใช้ใหม่กับทางระบบ ซึ่งสามารถสรุปเป็นตารางได้ดังนี้

ตารางที่ 5.1 สรุปรายการความสามารถของระบบที่พัฒนา

ID	Requirement	Available
P-01	The system should display information about system on first page of website.	Yes
P-02A	The system should display 'Partner Conference'.	No
P-02B	The system should link to 'Partner Conference' website.	Yes
P-03	The system should have login page.	Yes
P-04A	The system should have support system for helping.	No
P-04B	The system should have chat channel for contact supporter.	No
P-05	The system should have registration system	Yes
P-06	The system should have sign in with OTP.	Yes
P-07	The system should have password recovery system.	No (Use OTP)
P-08A	The system should support multi-user accounts type.	Yes
P-08B	The system should support for change account type from permit account.	No
P-09	The system should have paper information system for organizer.	Yes
P-10	The system should release organizer's ID by manual input or random.	Yes
P-11	The system should have files upload system.	Yes
P-12	The system should have status tracking system.	Yes

P-13	The system should display notification for important information after login by pop-up or another.	No
P-14A	The system should have payment system.	Yes
P-14B	The system should accept PayPal, bank transfer, credit card and debit card.	Yes
P-14C	The system should have payment notification system to organizer when pay via bank transfer.	No
P-14D	The system should accept pay for 3 <sup>rd</sup> person.	Yes
P-14E	The system should make payment by scan Quick Response Code (QR Code).	No
P-15	The system should have system visual wallet system for account uniquely.	No
P-16	The system should have electronic signature system.	No
P-17	The system should release invoice to customer.	Yes
P-18	The system should release purchase order (PO).	No
P-19	The system should have receipt information changing.	Yes
P-20	The system should have refund system.	Yes
P-21	The system should have specification conference's information site.	No
P-22	The system should release report to csv, pdf when user request.	Yes
P-23	The system should import conference's database with csv.	Yes
P-24	The system should have conference information editor.	Yes
P-25	The system should have document review system.	Yes
P-26	The system should show all financial statement.	No

ฟังก์ชันการทำงานบางส่วนสามารถใช้งานได้จริง แต่บางฟังก์ชันการทำงานไม่สามารถทำงานได้จริงหากไม่ได้เชื่อมโยงเข้ากับฐานข้อมูลและต้องกำหนด Row ของฐานข้อมูลให้ตรงกันก่อนจึงจะสามารถทำงานได้

## 5.2 แนวทางในการพัฒนาต่อ

- 1) พัฒนาให้ระบบสามารถรองรับงานประชุมทั่วไปได้และสามารถพัฒนาเพื่อใช้เป็นระบบกลางในการรับชำระเงินสำหรับงานประชุมโดยเฉพาะ
- 2) พัฒนาให้ระบบสามารถรองรับการชำระเงินและแสดงผลหลายสกุลเงินได้
- 3) สามารถรองรับการใช้งานลายเซ็นอิเล็กทรอนิกส์เพื่อใช้สำหรับลงนามในเอกสารต่างๆเช่น การออกใบเสร็จรับเงิน, การลงชื่อในเอกสารคำร้องต่างๆ เป็นต้น
- 4) พัฒนาฟังก์ชันการทำงานแต่ละส่วนเพิ่มเติมให้สามารถทำงานได้ดีและมีประสิทธิภาพมากยิ่งขึ้น
- 5) พัฒนาระบบความปลอดภัยของเว็บไซต์

## บรรณานุกรม

Jon Duckett. 2011. HTML and CSS Design and Build Websites. Canada: John Wiley & Sons, Inc.

Jon Duckett. 2014. JavaScript and JQuery Interactive Front-End Web Development. Canada: John Wiley & Sons, Inc.

Nicholas C. Zakas. 2016. Understanding ECMAScript 6. San Francisco: No Starch Press, Inc.

“ทำความเข้าใจกับ Redux แบบฉบับย่อยแล้วย่อยอีก”

[ระบบออนไลน์]. แหล่งที่มา <https://microbenz.in.th/ทำความเข้าใจกับ-redux-แบบฉบับย่อยแล้วย่อยอีก-b464808aca12>

(20 พฤศจิกายน 2560)

“Node.js คืออะไร?”

[ระบบออนไลน์]. แหล่งที่มา <https://devahoy.com/posts/getting-started-with-nodejs/>

(10 พฤศจิกายน 2560).

“SQL คืออะไร?”

[ระบบออนไลน์]. แหล่งที่มา <http://www.mindphp.com/คู่มือ/73-คืออะไร/2088-sql-คืออะไร.html>

(15 พฤศจิกายน 2560).

“ECMA Script 6 คืออะไร? + วิธีใช้”

[ระบบออนไลน์]. แหล่งที่มา <http://www.siamhtml.com/ecmascript-6-tutorial/>

(17 พฤศจิกายน 2560).

ภาคผนวก ก.

การติดตั้งเครื่องมือที่ใช้ในการพัฒนาระบบ

## 1. เครื่องมือที่ใช้ในการพัฒนาระบบ

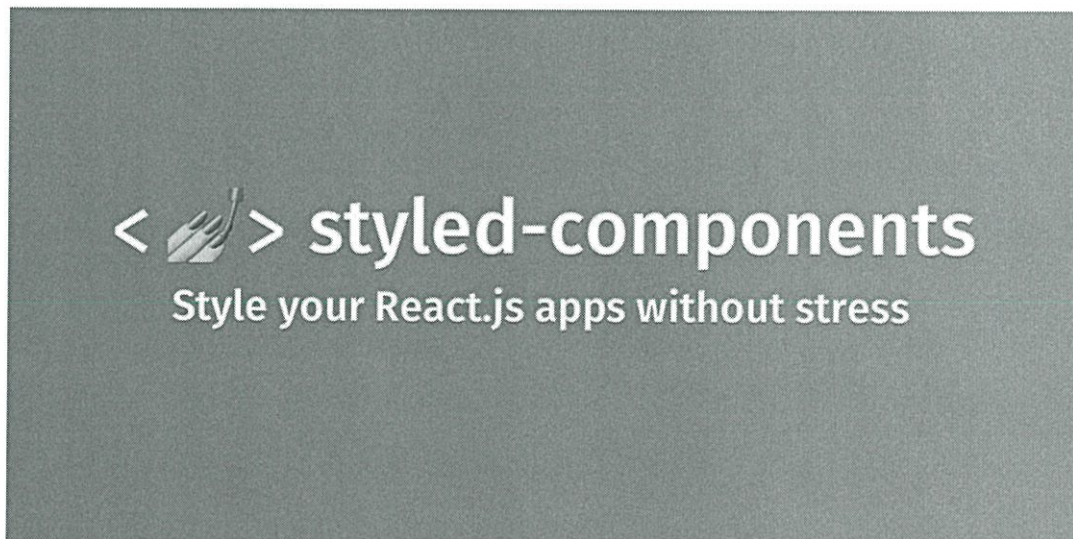
### 1.1 Frameworks ในส่วนของ Front-End

- **Semantic Ui React** ใช้สำหรับการสร้าง Component ต่างๆที่เป็นส่วนของ User Interface ต่างๆเช่น ปุ่มกด, ตาราง, หัวข้อต่างๆ เป็นต้น



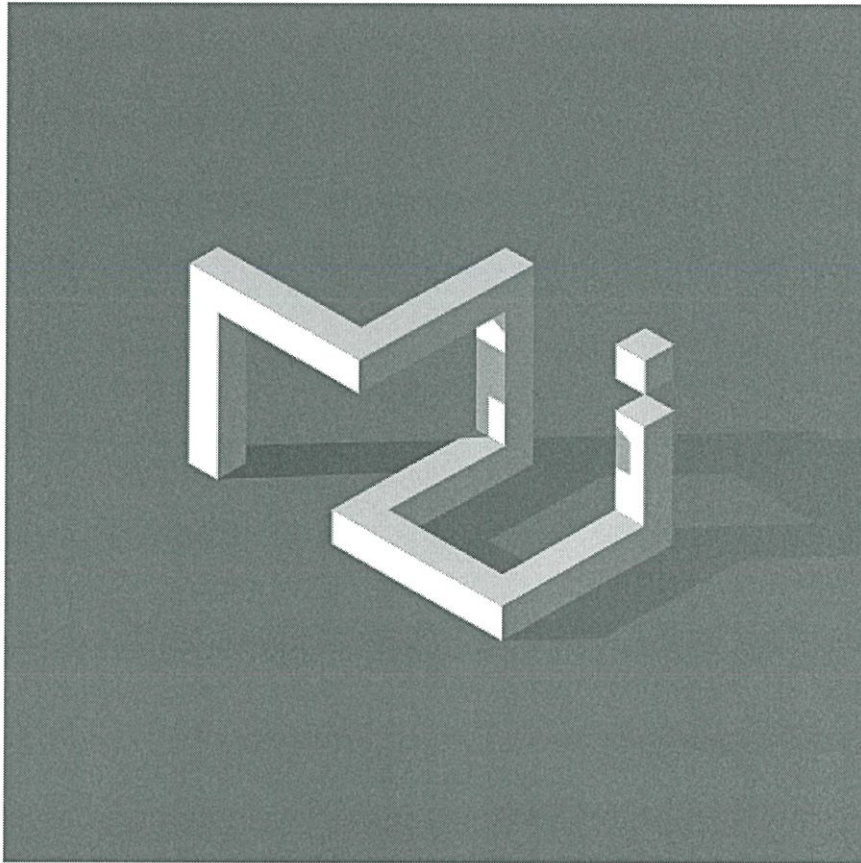
รูป 1ก. สัญลักษณ์เครื่องหมายของ Semantic Ui React

- **Styled-Components** ใช้ในการกำหนด Custom Component เองได้ ทำให้เราสามารถใช้ Tag ที่กำหนดขึ้นมาเองสำหรับการออกแบบโดยการเขียนจะใช้ CSS ในการเขียนปกติ



รูป 2ก. สัญลักษณ์เครื่องหมายของ Styled-Components

- **Material-Ui** เป็น Framework ที่ช่วยในการออกแบบ GUI ต่างๆ คล้ายกับ Semantic-UI แต่เป็นของบริษัท Google

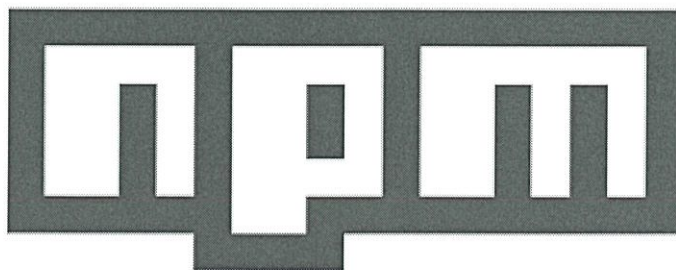


รูป 3ก. สัญลักษณ์เครื่องหมายของ Material-Ui

## 1.2 Frameworks ในส่วนของ Back-End

ส่วนระบบหลังบ้าน (Back-End) จะใช้ Node.js ในการเชื่อมต่อ โดยเรียกใช้ผ่าน Node Package Manager ดังนี้

- npm



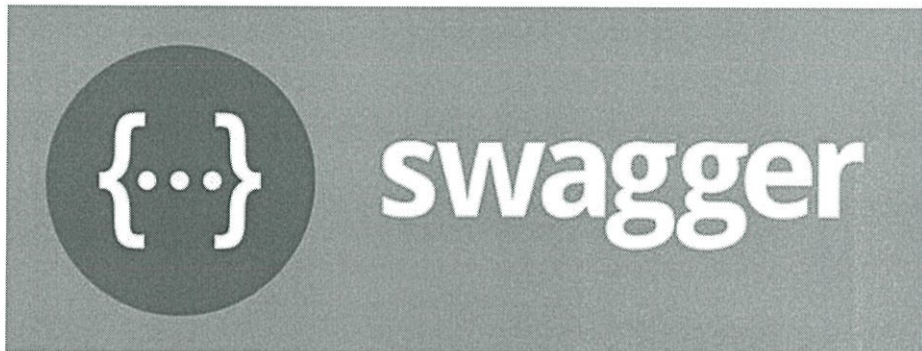
รูป 4ก. สัญลักษณ์เครื่องหมายของ npm

- Yarn (Node Package Manager)



รูป 5ก. สัญลักษณ์เครื่องหมายของ yarn

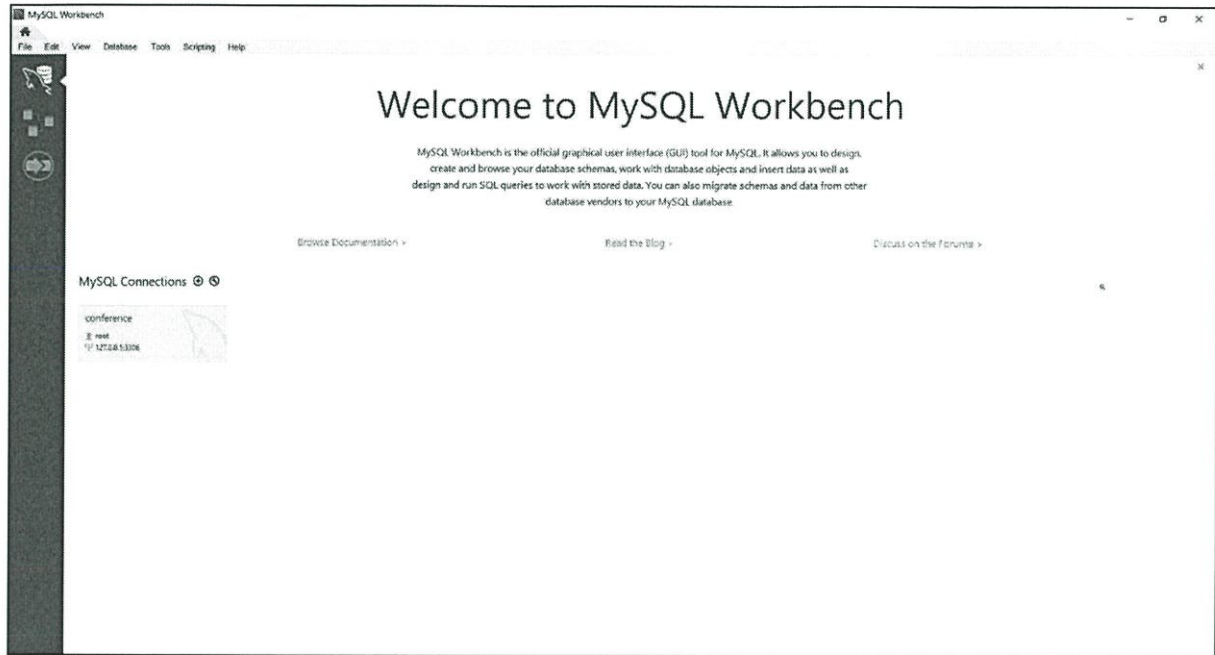
- Swagger ใช้ในการจัดการเพื่อให้สามารถให้ทดสอบ API ที่เขียนขึ้นมาได้ว่าสามารถใช้งานได้หรือไม่โดยที่ยังไม่ Implement เข้ากับระบบจริง



รูป 6ก. สัญลักษณ์เครื่องหมายของ swagger

### 1.3 ฐานข้อมูล

โครงการนี้ได้ใช้ฐานข้อมูลแบบ SQL โดยใช้โปรแกรม MySQL Workbench ซึ่งเป็นฐานข้อมูลที่สามารถใช้งานได้โดยไม่เสียค่าใช้จ่าย และสามารถจำลองเป็นเครื่อง Server แบบ Localhost ซึ่งสามารถกำหนดเป็น Connection ในการเชื่อมต่อได้



รูป 7ก. หน้าจอโปรแกรม MySQL Workbench

โดยสามารถตั้งค่าการเชื่อมต่อกับระบบที่เขียนขึ้นมาได้จากการตั้งค่า port ซึ่งใช้ port 3306 และ ให้ใช้ IP Address เป็น 127.0.0.1 จากนั้นก็สั่งให้ทำงานโดย start server หากทำงานเป็นปกติจะได้อิงภาพที่ 7.9 ซึ่งจะทำให้เราสามารถอ่านและเขียนข้อมูลผ่าน API

#### 1.4 การจัดการโปรเจก

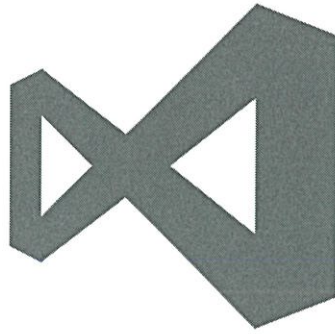
ในการจัดการไฟล์โปรเจกจะใช้ GitHub เข้ามาช่วยในการทำงานร่วมกัน ซึ่งจะใช้งานผ่าน Terminal



รูป 8ก. สัญลักษณ์เครื่องหมายของ GitHub

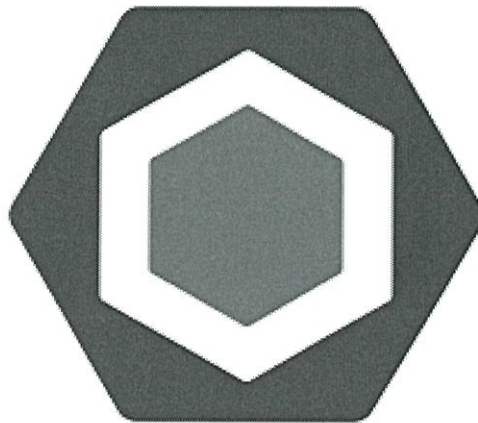
#### 1.5 IDE

ทางผู้จัดทำได้ตกลงที่จะใช้ IDE ของบริษัท Microsoft ที่ชื่อว่า Visual Studio Code เนื่องจากสามารถใช้งานได้โดยไม่เสียค่าใช้จ่ายใดๆ และสามารถลงโปรแกรมเสริม (Extensions) ได้



รูป 9ก. สัญลักษณ์เครื่องหมายของ Visual Studio Code

และได้ลงโปรแกรมจัดการในการเขียนโค้ดที่ชื่อว่า ESLint ซึ่งจะทำหน้าที่ตรวจสอบการเขียนโค้ดให้ถูกต้อง และการจัดระเบียบการเขียนด้วยสำหรับภาษา JavaScript



รูป 10ก. สัญลักษณ์เครื่องหมายของ ESLint

## 1.6 Web Hosting & Server

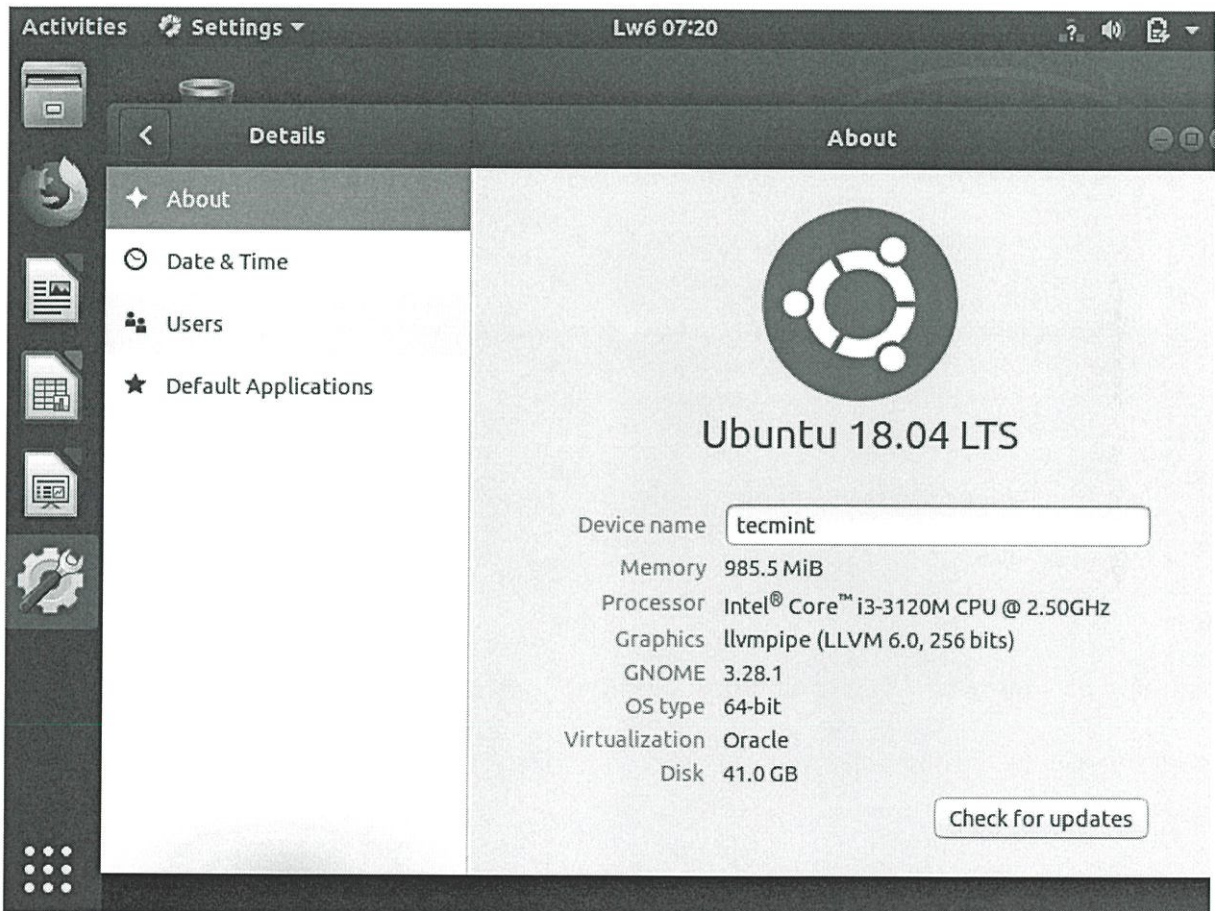


รูป 11ก. สัญลักษณ์เครื่องหมายของ DigitalOcean

## 2. ขั้นตอนการติดตั้ง

ก่อนการเริ่มพัฒนาระบบ จำเป็นต้องติดตั้งเครื่องมือต่างๆที่ใช้ในการพัฒนาระบบ โดยมีขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 ระบบปฏิบัติการที่ใช้แนะนำให้ใช้ระบบปฏิบัติการ Linux Ubuntu เพื่อให้สามารถพัฒนาได้ทั้งส่วน Front-End และ Back-End รวมถึงการ Deploy ขึ้น Server ด้วย แต่หากใช้ระบบปฏิบัติการ Windows นั้น การทำส่วนของ Back-End และการ Deploy ขึ้น Server นั้นค่อนข้างจะยุ่งยาก เพราะเครื่อง Server ใช้ระบบปฏิบัติการ Linux ซึ่งสามารถเข้าไป Download ได้ที่ <https://www.ubuntu.com/>

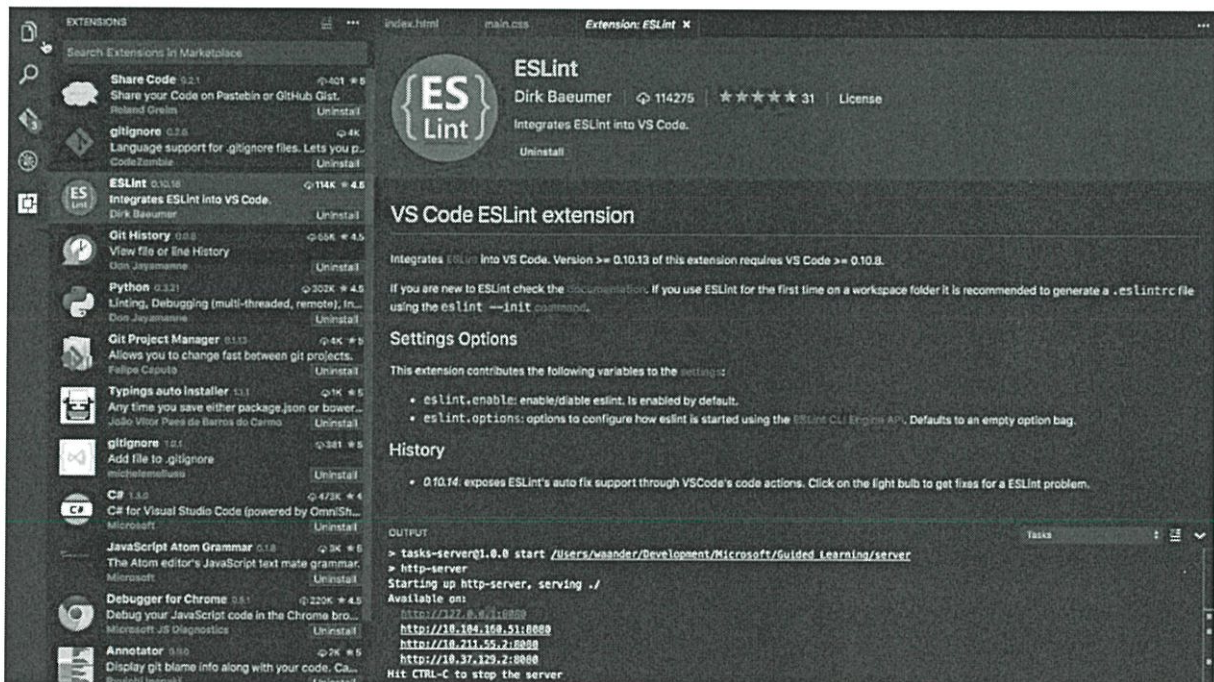


รูป 12ก. หน้าจอของระบบปฏิบัติการ Linux Ubuntu

ขั้นตอนที่ 2 ติดตั้ง Node.js และ Node Package Manager (NPM) ซึ่งตัว Node Package Manager สามารถใช้ npm หรือ yarn ก็ได้ ซึ่งสามารถใช้คำสั่งในการติดตั้งได้ดังนี้

```
sudo apt-get update
sudo apt-get install nodejs
sudo apt-get install npm
sudo apt-get install yarn
sudo apt-get update
```

ขั้นตอนที่ 3 ติดตั้ง IDE ในการพัฒนา ซึ่งแนะนำให้ใช้โปรแกรม Visual Studio Code ซึ่งสามารถติดตั้งส่วนขยาย (Extension) ที่ชื่อว่า ESLint เพื่อให้โค้ดที่เขียนมีระเบียบและถูกต้องตามหลักการเขียนโปรแกรม ซึ่งสามารถ Download โปรแกรมได้จาก <https://code.visualstudio.com/download> แล้ววิธีติดตั้งได้จาก <https://code.visualstudio.com/docs/setup/linux>



รูป 13ก. หน้าจอของโปรแกรม VS Code ขณะทำการติดตั้งส่วนขยาย ESLint

ขั้นตอนที่ 4 ติดตั้ง โปรแกรม GitHub พร้อมสมัครบัญชีผู้ใช้ก่อนที่ <https://www.github.com> จากนั้นทำการ Clone ไฟล์โปรเจกจาก GitHub เพื่อนำมาพัฒนาต่อ โดยใช้คำสั่งดังนี้

```
sudo apt-get install git
git clone https://github.com/eaotnc/ps-app.git
git clone https://github.com/eaotnc/PS-api.git
```

ขั้นตอนที่ 5 ทำการติดตั้งไฟล์ Module ของ React Framework ที่ไฟล์โปรเจกต์ทำการ Clone มาจาก Github โดยใช้คำสั่งดังนี้

```
yarn init
yarn add react react-dom
```

ขั้นตอนที่ 6 ทำการติดตั้ง Package ต่าง ๆที่ใช้ในการพัฒนาระบบ ได้แก่ semantics-ui, react-router, material-ui, styled component และ grid foundation

ขั้นตอนที่ 7 ทำการติดตั้งฐานข้อมูลโดยใช้โปรแกรม MySQL Workbench และทำการสร้างฐานข้อมูล

```
sudo apt-get update
sudo apt-get install mysql-server
mysql_secure_installation
```

ขั้นตอนที่ 8 ให้ทำการติดตั้ง nvm ซึ่งเป็นตัวจัดการ Version ของ node.js

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.11/install.sh | bash
```

ขั้นตอนที่ 9 ให้เลือกใช้ nodejs version 6.14.1 โดยใช้คำสั่ง

```
nvm install 6.14.1
```

ขั้นตอนที่ 10 ให้ทำการทดสอบระบบโดยทดลองสั่งเริ่มต้น โปรเจกต์ ซึ่งสามารถใช้คำสั่งในการรันโปรเจกต์ได้ดังนี้

```
yarn start หรือ
npm start
```

**ภาคผนวก ข.**

**Source Code**

ไฟล์ของระบบจะแบ่งออกเป็น 2 ส่วนดังต่อไปนี้

- **ps-api** เป็นไฟล์ส่วน Back-End ซึ่งใช้เขียนเชื่อมต่อกับ api
- **ps-app** เป็นไฟล์ส่วน Front-End ซึ่งเกี่ยวข้องกับการออกแบบ GUI, การเปลี่ยนหน้าเว็บ, การจัดการ State, การส่งผ่านข้อมูลโดยใช้ Redux และไฟล์อื่น ๆ ที่เกี่ยวข้อง

## 1. ps-api

Name	Date modified	Type	Size
.git	13/05/2018 13:58	File folder	
node_modules	30/04/2018 09:49	File folder	
routes	13/05/2018 13:57	File folder	
	19/03/2018 10:23	Text Document	1 KB
.server.js.swp	30/04/2018 09:48	SWP File	1 KB
dump2	30/04/2018 09:48	SQL Text File	24 KB
package	13/05/2018 13:57	Adobe After Effects...	1 KB
README	19/03/2018 10:23	MD File	1 KB
server	30/04/2018 09:48	JavaScript File	1 KB
yarn.lock	13/05/2018 13:57	LOCK File	50 KB
yarn-error	30/04/2018 11:11	Text Document	55 KB

รูป 1ข. โครงสร้างข้อมูลของโฟลเดอร์ ps-api

โดยโฟลเดอร์ต่างๆสามารถอธิบายได้ดังนี้

- **node\_modules** เป็นที่เก็บไฟล์สำคัญของ React & Redux Modules ซึ่งไฟล์ในโฟลเดอร์นี้ ห้ามทำการแก้ไขโดยเด็ดขาด
- **routes** เป็นที่เก็บไฟล์ที่เราเขียนขึ้นเพื่อใช้ติดต่อกับ api ต่าง ๆ และไฟล์ในโฟลเดอร์นี้จะถูกเรียกใช้เมื่อมีการร้องขอ

Name	Date modified	Type	Size
index	19/03/2018 10:23	JavaScript File	1 KB
login	30/04/2018 09:48	JavaScript File	5 KB
payment	30/04/2018 09:48	JavaScript File	2 KB
uploadimage	13/05/2018 13:57	JavaScript File	1 KB

รูป 2ข. ไฟล์บางส่วนภายในโฟลเดอร์ routes

โดยการเรียกใช้งาน React จะเข้ามาเรียกใช้ที่ไฟล์ index.jsx ก่อนทุกครั้งเพื่อดูสิ่งที่กำหนดไว้ว่าให้ทำอะไรและไปเรียกใช้งานที่ไฟล์ไหน เปรียบเสมือนเป็นป้ายบอกทางเท่านั้น ส่วนไฟล์อื่น ๆ จะทำหน้าที่เป็นตัวกำหนดการกระทำและการเชื่อมต่อกับ Back-End ซึ่งแต่ละไฟล์ก็จะทำหน้าที่เฉพาะ เช่น การส่งรหัส OTP, การอัปโหลดไฟล์ เป็นต้น

สำหรับการเขียนเพื่อเชื่อมต่อกับ API นั้น ให้เริ่มที่ไฟล์ index.jsx ภายในโฟลเดอร์ routes นี้ โดยในไฟล์นี้มีรายละเอียดดังนี้

```
import axios from 'axios';
import { GetHello } from './GetHello';
import { LoginSendOTP } from './login'

export function APIRequest(options) {
  const defaultOption = {
    baseURL: 'http://localhost:8000',
    headers: Object.assign(
      {},
      { 'Content-Type': 'application/json' },
    ),
  };
  const mergedOption = Object.assign({}, options, defaultOption);

  return axios(mergedOption);
}

export default {
  APIRequest,
  GetHello,
  LoginSendOTP,
};
```

จากโค้ดข้างต้นเป็นโปรแกรมที่ถูกเขียนอยู่ในไฟล์ index.jsx ภายในโฟลเดอร์ routes ซึ่งจะเป็นออกเป็นแต่ละส่วนดังนี้

ส่วนที่ 1 เป็นส่วนของการ Import ไฟล์ต่างๆเข้ามา โดยจะทำการ Import ไฟล์ API ที่เขียนขึ้นมาในส่วนนี้

ส่วนที่ 2 เป็น Function ที่ได้ทำการ export ออกไปเพื่อให้ Component อื่นๆเข้ามาเรียกใช้งานผ่านชื่อที่ถูกประกาศไว้ ในส่วนนี้จะมีการเขียน URL เพื่อเชื่อมต่อ โดยในที่นี้จะเชื่อมต่อกับ Swagger ที่ Port 8000 แบบ localhost

ส่วนที่ 3 เป็นส่วนที่ใช้ Export สำหรับ API ที่ได้เขียนขึ้นมา เช่น LoginSendOTP เป็น API ที่เขียนขึ้นเพื่อให้ส่งรหัส OTP ไปยัง email ของผู้ใช้ ซึ่งถูกเขียนขึ้นในไฟล์ login.jsx ในไฟล์เตอร์ routes เช่นเดียวกัน ซึ่งถูกเขียนไว้ดังนี้

```
import { APIRequest } from '.';
import {
  LoginSendOTPSuccess,
  LoginSendOTPFailure,
} from './reducers/loginReducer';

export function LoginSendOTP(email, otp) {
  return dispatch => {
    const options = {
      method: 'post',
      url: '/user/login',
      data: {
        otp,
        email,
      },
    };
    APIRequest(options)
      .then(res => {
        // console.log('sent OTP Success', res.data);
        localStorage.setItem('name', res.data.name);
        localStorage.setItem('role', res.data.role);
        localStorage.setItem('auth', res.data.auth);
        dispatch(LoginSendOTPSuccess(res.data));
      })
      .catch(err => {
        // console.log('sent OTP fail', err);
        dispatch(LoginSendOTPFailure(err.response));
      });
  };
}
```

```

export function LoginSendEmail(email) {
  const options = {
    method: 'post',
    url: '/user/send-email',
    data: {
      email,
    },
  };

  APIRequest(options)
    .then(res => {
      console.log('can post api', res);
    })
    .catch(err => {
      console.log('cant post api', err);
    });
}

export function Logout() {
  return () => {
    console.log('Logout');
    localStorage.clear();
    window.location = '/home';
  };
}

export default {
  LoginSendOTP,
};

```

ในไฟล์นี้จะเป็นการเขียนเชื่อมต่อกับทั้ง Database และ Swagger โดยจะทำการตรวจสอบค่าในฐานข้อมูลเพื่อดู email, ชื่อ, ประเภทบัญชี และสถานะการลงชื่อเข้าใช้ ข้อมูลที่ได้จะถูกส่งมาและจัดการโดยใช้ redux

ขั้นตอนการทำงานคือตรวจสอบข้อมูลในฐานข้อมูล จากนั้นนำค่าส่งไปยังระบบเบื้องหลัง (Back-End) โดยผ่าน API ถ้าข้อมูลถูกต้องตามที่ผู้ใช้ป้อนเข้ามาก็จะทำการส่งรหัส OTP ไปยัง email ที่กรอกไว้ จากนั้นระบบจะเขียน OTP ลงฐานข้อมูล และเมื่อผู้ใช้ลงชื่อเข้าใช้สำเร็จก็จะเขียนข้อมูล เปลี่ยนสถานะเป็นลงชื่อเข้าใช้งาน และเปลี่ยนหน้าเว็บไปยัง URL ที่กำหนดในไฟล์ app.js

- **package.json** เป็นไฟล์ที่เก็บรวบรวมรายชื่อ Packages ที่ได้ทำการติดตั้งไว้ โดยจะเก็บในรูปแบบของ Dictionary ซึ่งได้กล่าวไว้ในบทที่ 2 แล้ว
- **server.jsx** เป็นไฟล์ที่ใช้สำหรับเขียนคำสั่งเกี่ยวกับ server เช่นการเชื่อมต่อกับ server

## 2. ps-app

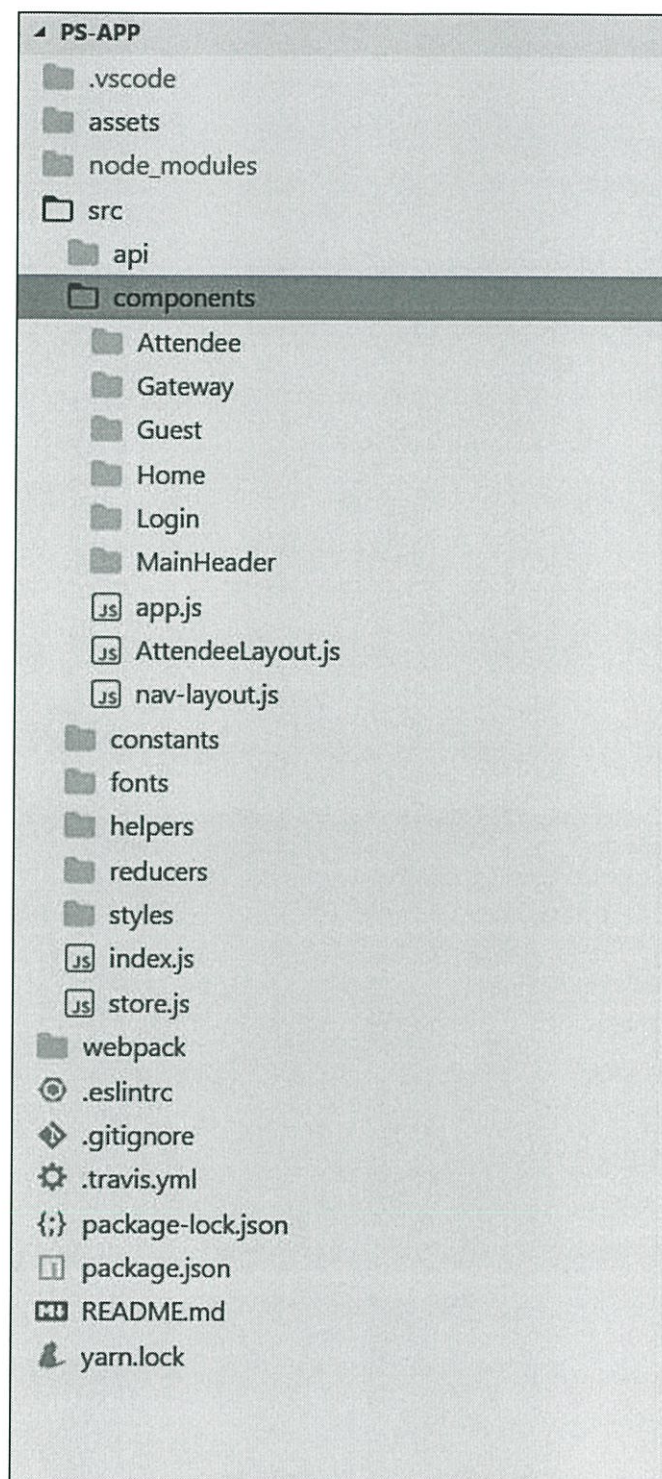
Name	Date modified	Type	Size
.git	13/05/2018 15:31	File folder	
.vscode	19/04/2018 16:52	File folder	
assets	21/04/2018 16:16	File folder	
node_modules	21/04/2018 13:13	File folder	
src	24/02/2018 14:04	File folder	
webpack	19/02/2018 15:49	File folder	
.eslintrc	21/04/2018 14:13	ESLINTRC File	1 KB
	19/02/2018 15:49	Text Document	1 KB
.travis.yml	19/02/2018 15:49	YML File	1 KB
package	21/04/2018 13:13	Adobe After Effects...	3 KB
package-lock	21/04/2018 13:13	Adobe After Effects...	441 KB
README	19/02/2018 15:49	MD File	1 KB
yarn.lock	23/04/2018 09:29	LOCK File	281 KB
yarn-error	21/04/2018 13:08	Text Document	297 KB

รูป 3ข. โครงสร้างข้อมูลของโฟลเดอร์ ps-app

โดยโฟลเดอร์ต่างๆสามารถอธิบายได้ดังนี้

- **node\_modules** เป็นที่เก็บไฟล์สำคัญของ React & Redux Modules ซึ่งไฟล์ในโฟลเดอร์นี้ ห้ามทำการแก้ไขโดยเด็ดขาด
- **assets** เป็นที่เก็บไฟล์รูปภาพ, ไอคอนต่างๆ ที่ใช้ในเว็บไซต์
- **src** เป็นที่เก็บไฟล์ Source Code ของหน้าเว็บและส่วนประกอบต่าง ๆ ของเว็บไซต์ทั้งหมด ซึ่งจะอธิบายอย่างละเอียดในตอนสุดท้าย
- **webpack** เป็นที่เก็บไฟล์ webpack ซึ่งติดต่อสื่อสารกับโปรแกรม Web Browser และแปลงเป็นภาษาที่ Web Browser เข้าใจ ซึ่งได้กล่าวไว้ในบทที่ 2
- **.eslintrc** เป็นไฟล์ซึ่งสามารถแก้ไขการตั้งค่าของโปรแกรมส่วนเสริม ESLint

ไฟล์ที่อยู่ในโฟลเดอร์ src เป็นสิ่งที่สำคัญเพราะเป็นส่วนของหน้าเว็บไซต์ ภายในประกอบไปด้วยโฟลเดอร์ย่อยๆและไฟล์สำคัญที่ React เรียกใช้ ซึ่งมีไฟล์ดังนี้



รูป 4x. โครงสร้างข้อมูลของโฟลเดอร์ ps-app/src

เห็นไฟล์สำคัญดังนี้

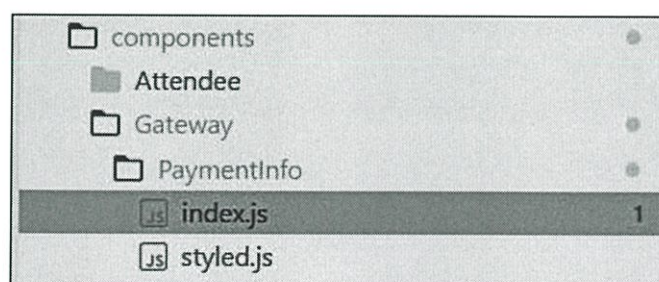
- **src/components** สำหรับเก็บไฟล์งานออกแบบส่วน Front-End ทั้งหมด โดยจะแบ่งเป็นส่วนๆ โดยจากภาพจะเห็นได้ว่าโฟลเดอร์ย่อยนั้นคือแต่ละหน้าของเว็บไซต์ หรือเป็นส่วน

ของหน้าเว็บ เช่น Attendee เป็นส่วนของหน้าเว็บที่ให้สิทธิเฉพาะบัญชีประเภท Attendee เข้าถึง และภายในก็จะประกอบไปด้วยโฟลเดอร์ย่อย ๆ ซึ่งเป็นหน้าเว็บแต่ละหน้าสำหรับ Attendee เช่น Dashboard, Documents, Payment, History เป็นต้น

- **styles** เป็นส่วนที่ใช้เขียน style สำหรับกำหนด layout หน้าเว็บไซค์
- **index.js** เป็นไฟล์ที่ใช้ติดต่อกับไฟล์ html ในที่นี้ใช้ชื่อ template.html (เหมือนไฟล์ index.html ที่กล่าวไว้ในบทที่ 2)
- **store.js** เป็นไฟล์ที่ใช้สำหรับทำ Redux โดยทำการเก็บค่าและส่งค่า Store ข้อมูลไปยัง Component ที่ข้าม Class กัน ได้
- **app.js** เป็นไฟล์ที่ใช้จัดการกับหน้าเว็บ โดยจะมีการ Import ไฟล์ต่างๆรวมถึงการตรวจสอบ การเปลี่ยนหน้าเว็บไซค์, การกำหนด URL ต่างๆ เป็นต้น (รายละเอียดมีกล่าวไว้ในบทที่ 2)
- **AttendeeLayout.js** เป็นไฟล์ที่ใช้จัดหน้า Layout เว็บไซค์ในส่วนของบัญชีผู้ใช้ประเภท Attendee โดยภายในจะมีการกำหนดตำแหน่งในการแสดงผลต่างเช่น Sidebar, Header เป็นต้น

### 3. การเขียนโปรแกรม

ในการเขียนโปรแกรม จะมองหน้าเว็บไซค์แต่ละส่วนเป็น Components ซึ่งถูกเก็บไว้ในโฟลเดอร์ components โดยการเขียนโปรแกรม จะสามารถ export ได้จำเป็นต้องมี tag ใหญ่เพียง tag เดียวครอบเนื้อหา ของเว็บไซค์ทั้งหมดอยู่ ไม่สามารถมีมากกว่าได้ ดังนั้นจำเป็นต้องเขียนทุกอย่างไว้ภายใต้ tag นั้นๆ เช่นเขียน `<div ....></div>` ครอบ



รูป 5ข. โครงสร้างข้อมูลของโฟลเดอร์ Component ที่ชื่อ PaymentInfo

โดยในหนึ่ง Component นั้นจะประกอบไปด้วยไฟล์ 2 ไฟล์คือ index.js ซึ่งจะเป็นไฟล์ที่ถูกอ่าน ขึ้นมาเป็นไฟล์แรกเมื่อมีการเรียกใช้งาน Component และไฟล์ styled.js ซึ่งเป็นไฟล์ที่กำหนดการตกแต่ง Component หรือสร้าง Custom Component ได้ซึ่งใช้ Styled Component เขียนด้วย CSS ซึ่งมีการเขียน โปรแกรมจากไฟล์ทั้ง 2 ไฟล์ดังนี้

```

import React, { Component } from 'react';
import { Header, Icon, Divider, Grid, Image, Button } from 'semantic-ui-react';
import { ContainBox, OptionsBox, Alert, VerifyBox, Complete, Incomplete } from './styled';
import paypal from '../././././assets/Payments/paypal.png';
import credit from '../././././assets/Payments/credit.png';
import bank from '../././././assets/Payments/bank.png';
import handon from '../././././assets/Payments/handon.png';
import BrowseBox from '.././Attendee/Documents/BrowseBox';

class PaymentsInfo extends Component {
  static propTypes = {
  }
  constructor() {
    super();
    this.state = {
      status: 0, // Status of next button
      PaymentStatus: 'Incomplete', // Have 2 status: Complete, Incomplete
    };
  }
  render() {
    return (
      <div>
        <ContainBox>
          <Header size="huge">
            <Icon name="payment" />
            <Header.Content>
              Payments Options
            <Header.Subheader>
              Make payment for your conference(s).
            </Header.Subheader>
            </Header.Content>
          </Header>
          <h3>
            {
              (this.state.PaymentStatus === 'Complete') ?
                <Complete> {this.state.PaymentStatus}
                  <Icon name="check circle" color="green" size="large" />
                </Complete>
              :
            }
          </h3>
        </ContainBox>
      </div>
    );
  }
}

```

```

<Incomplete> {this.state.PaymentStatus}
    <Icon name="remove circle outline" color="red" size="large" />
  </Incomplete>
}

</h3>
<Divider horizontal>Payment Options</Divider>
<br />
<OptionsBox>
  <Grid columns={4} divided>
    <Grid.Row>
      <Grid.Column>
        <Image src={paypal} size="tiny" verticalAlign="middle" /> <h3>PayPal</h3>
      </Grid.Column>
      <Grid.Column>
        <Image src={credit} size="tiny" verticalAlign="middle" /> <h3>Credit Card</h3>
      </Grid.Column>
      <Grid.Column>
        <Image src={bank} size="tiny" verticalAlign="middle" /> <h3>Bank Transfer</h3>
      </Grid.Column>
      <Grid.Column>
        <Image src={handon} size="tiny" verticalAlign="middle" /> <h3>Cash (On Desk)</h3>
      </Grid.Column>
    </Grid.Row>
    <Grid.Row>
      <Alert>*If you pay by cash, you must verify with organizer by yourself.</Alert>
    </Grid.Row>
  </Grid>
</OptionsBox>
<br />
<br />
<Divider horizontal>Payment Verify</Divider>
<br />
<VerifyBox>
  <h3>For Bank Transfer, Please upload payment slip.</h3>
  <BrowseBox />
  <br />
  <Button color="orange" size="large" onClick={this.upload}>
    Upload
  </Button>
</VerifyBox>

```

```

</ContainBox>
  </div>
)
}
}
}

export default PaymentsInfo;

```

ข้างต้นเป็นไฟล์หน้าเว็บไซท์ที่ถูกเขียนอยู่ในไฟล์ index.js ของ PaymentInfo ซึ่งเป็นหน้าเว็บไซท์สำหรับการชำระเงิน โดยโค้ดจะแบ่งเป็นส่วนๆ ได้ดังนี้

ส่วนแรก เป็นส่วนที่ทำหน้าที่ในการ Import ไฟล์ต่างๆ เข้ามา

ส่วนที่ 2 เป็นส่วนที่ใช้ประกาศ class name ของ Component และจะมีการประกาศ Props, Constructor และ State ต่างๆ ของ class

ส่วนที่ 3 คือ Render โดยส่วนนี้จะเป็นส่วนที่ใช้แสดงผลหน้าเว็บไซท์ โดยเป็นส่วนของการออกแบบ GUI ต่างๆ

ส่วนที่ 4 เป็นส่วนสำหรับการ export ชื่อ Component ซึ่งสำคัญมากๆ โดยชื่อนี้จะถูกไฟล์ app.js ใช้ import อีกครั้งเพื่อระบุหน้าเว็บไซท์ โดยจะเขียนดังนี้ export default <ClassName>

ส่วนการกำหนด style หรือการเขียน CSS นั้นจะเขียนอยู่ที่อีกไฟล์ โดยการเขียนจะมีการเรียกใช้ Styled Component เพื่อให้สามารถเขียน Custom Tag เพื่อใช้ในการเขียนหน้าเว็บไซท์ได้ ซึ่งสามารถเขียนได้ดังนี้

```

import styled from 'styled-components';

export const ContainBox = styled.div`
  height: 700px;
  width: auto;
  padding-top: 3rem;
  padding-left: 2rem;
  padding-right: 2rem;
  text-align: left;
  background-color:white;
;

```

```
export const SelectiveBox = styled.div`
```

```
  height: 430px;  
  width: 650px;  
  padding-top: 1rem;  
  padding-left: 1rem;  
  padding-right: 1rem;  
  text-align: left;  
  display: block;  
  margin: auto;  
  border: 1px solid black;  
  border-style: rounded;  
  background-color: white;
```

```
`;
```

```
export const OptionsBox = styled.div`
```

```
  height: 150px;  
  width: auto;  
  padding-top: 1rem;  
  padding-left: 1rem;  
  padding-right: 1rem;  
  text-align: center;  
  display: block;  
  margin: auto;  
  background-color: white;
```

```
`;
```

```
export const VerifyBox = styled.div`
```

```
  height: 180px;  
  width: auto;  
  padding-top: 1.5rem;  
  padding-left: 1rem;  
  padding-right: 1rem;  
  padding-bottom: 1.5rem;  
  text-align: center;  
  display: block;  
  margin: auto;  
  border: 1px solid black;  
  border-style: rounded;  
  background-color: white;
```

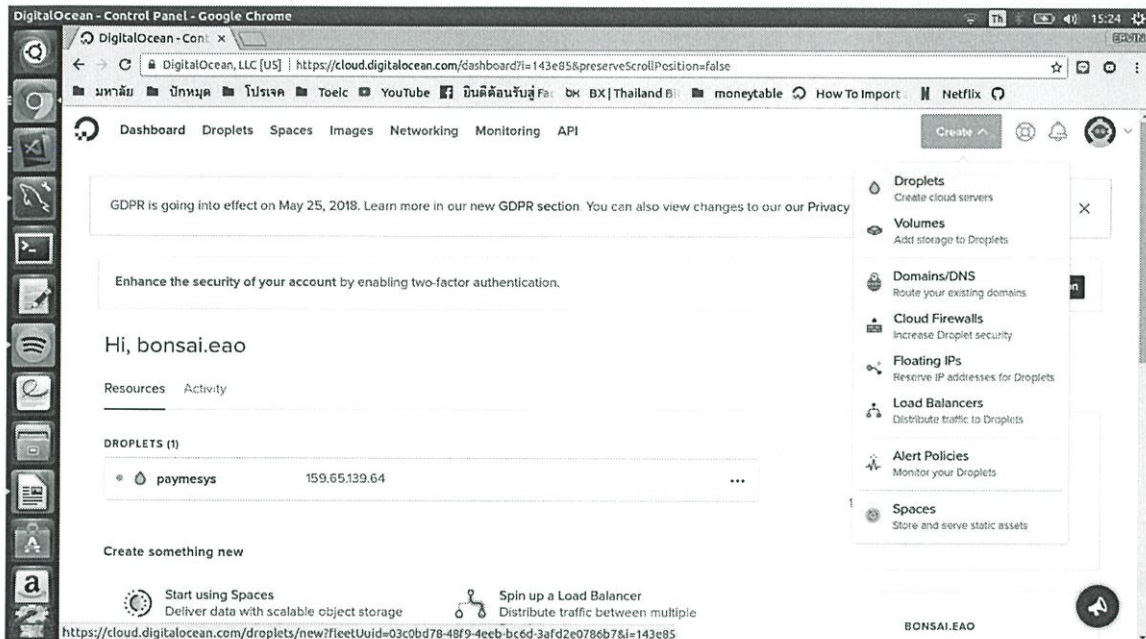
```
`;
```

โดยการเขียน Custom Tag นั้นจะถูกเขียนโดยใช้ CSS ปกติ แต่มี Syntax พิเศษซึ่งถูกกำหนดไว้โดย Style Component และจะต้องทำการ export ชื่อของ Component นั้นๆ แล้วนำไป import ที่ไฟล์ index.js ส่วนการใช้งานก็สามารถใช้งานได้ตามปกติเหมือนภาษา HTML ทั่วไป

ในบาง Component นั้นอาจมี Component ย่อยภายในอีก ซึ่งจะเป็น โพลเดอร์ย่อยภายใน แต่เมื่อทำการเปิดดูก็จะพบว่ารูปแบบการเขียน Component นั้นเหมือนกัน มีไฟล์และรูปแบบการเขียน โปรแกรมที่เหมือนกัน

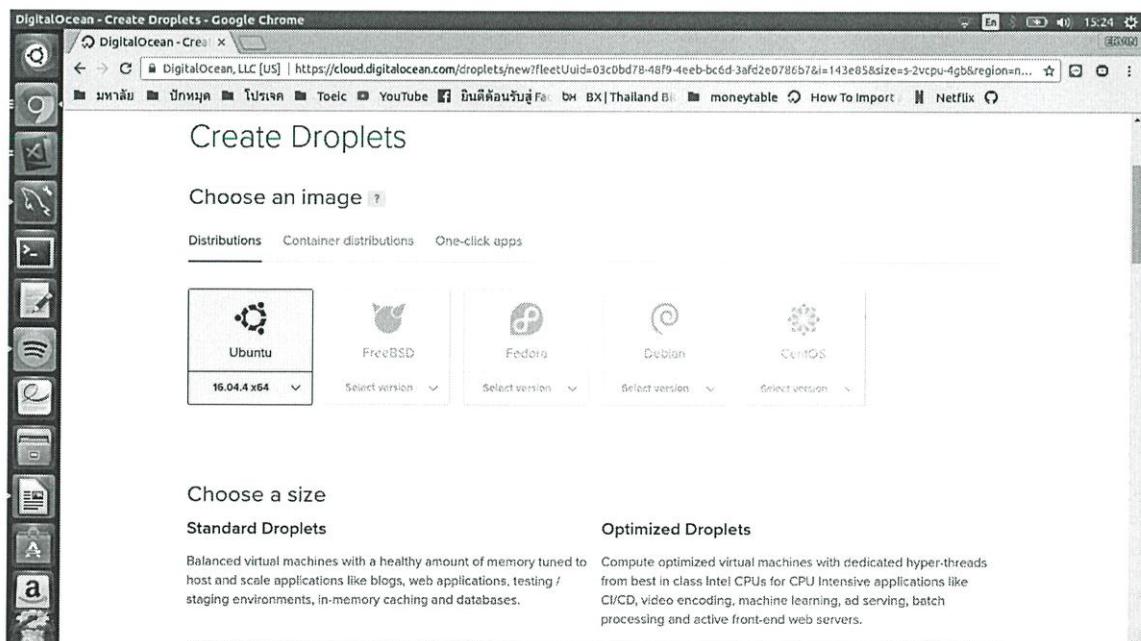
ภาคผนวก ค.  
การ Deploy เว็บไซต์

## 1. การสร้าง server บน digitalOcean



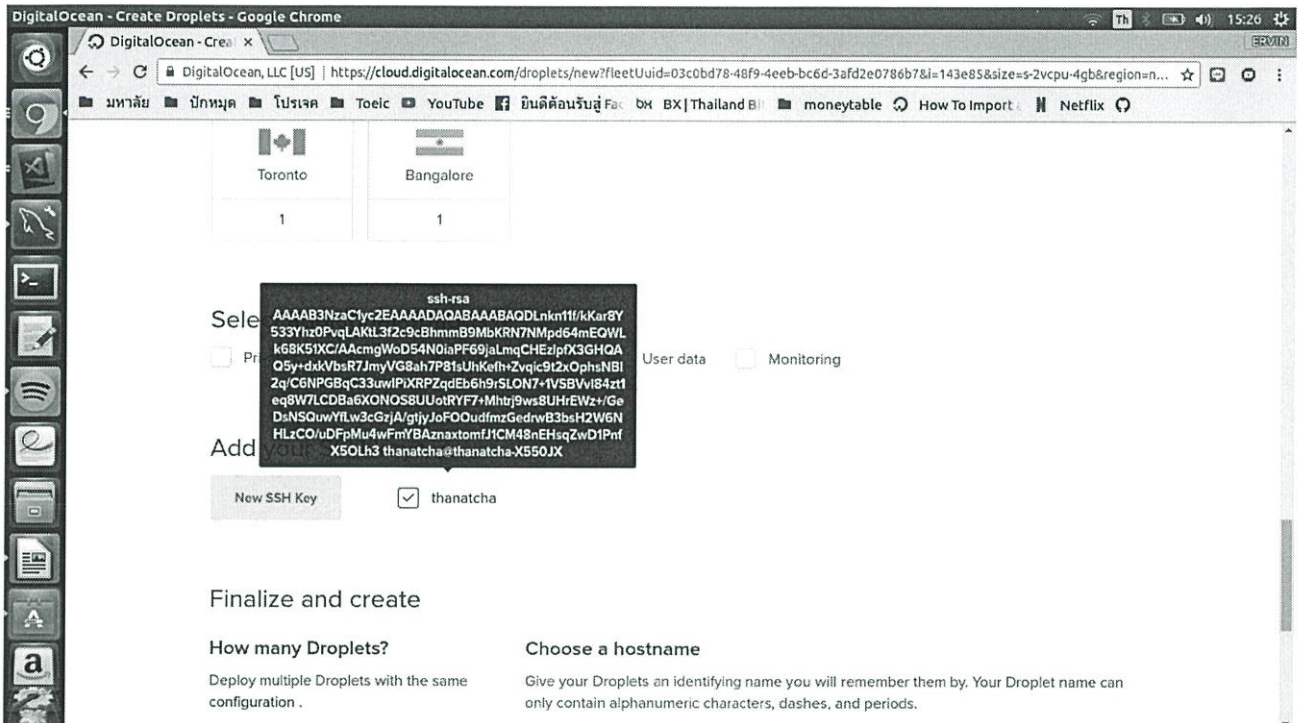
รูป 1ก. หน้าหลักของ digitalOcean เมื่อทำการลงชื่อเข้าใช้

สมัครบัญชีผู้ใช้งานของ digitalOcean หลังจากนั้น กดที่ Create > Droplets เพื่อสร้าง server ใหม่ขึ้นมา



รูป 2ก. หน้าจอเมื่อทำการกดปุ่ม Droplets

หลังจากนั้นให้กำหนดระบบปฏิบัติการของเครื่อง Server เป็น Ubuntu 16.04 x64 ซึ่งเป็น Linux เพื่อให้สามารถติดตั้งระบบได้ง่าย



### รูป 3ค. หน้าจอเมื่อทำการกรอก SSH

เมื่อเลือกเสร็จแล้วให้ทำการเลือกใส่ SSH เพื่อให้สามารถ Remote เข้า Server จากเครื่องเราได้ โดยหากไม่ทราบวิธีสร้าง SSH ก็สามารถ กด New SSH Key แล้วกด ดู How to Create SSH Key ได้ แล้วทำการคัดลอก SSH Key ของเครื่องเราใส่ในช่อง New SSH Key

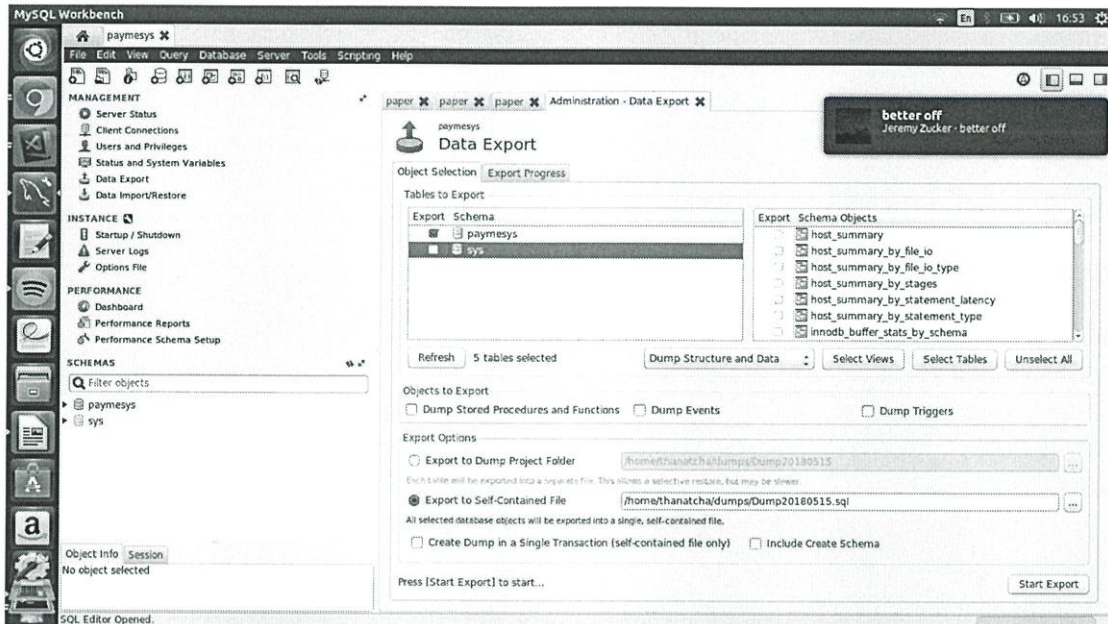
หลังจากสร้าง Server เสร็จก็จะได้ Droplets ที่เป็น ชื่อ Server ของเรา และ IP Address โดยสามารถใช้คำสั่ง SSH root@159.65.139.64 เข้าไปเพื่อทำการ Remote Server ได้

## 2. การ Remote เข้า Server

ด้วยคำสั่ง ssh root@159.65.139.64 เพื่อ Remote เข้าไปที่ Server ซึ่งอาจจะมีการถาม Private Key ที่ติดตั้งไว้ในเครื่อง เพื่อเป็นรหัสผ่าน 2 ขั้นตอน หากไม่ได้ตั้ง Private Key ไว้ สามารถใช้คำสั่งดังนี้

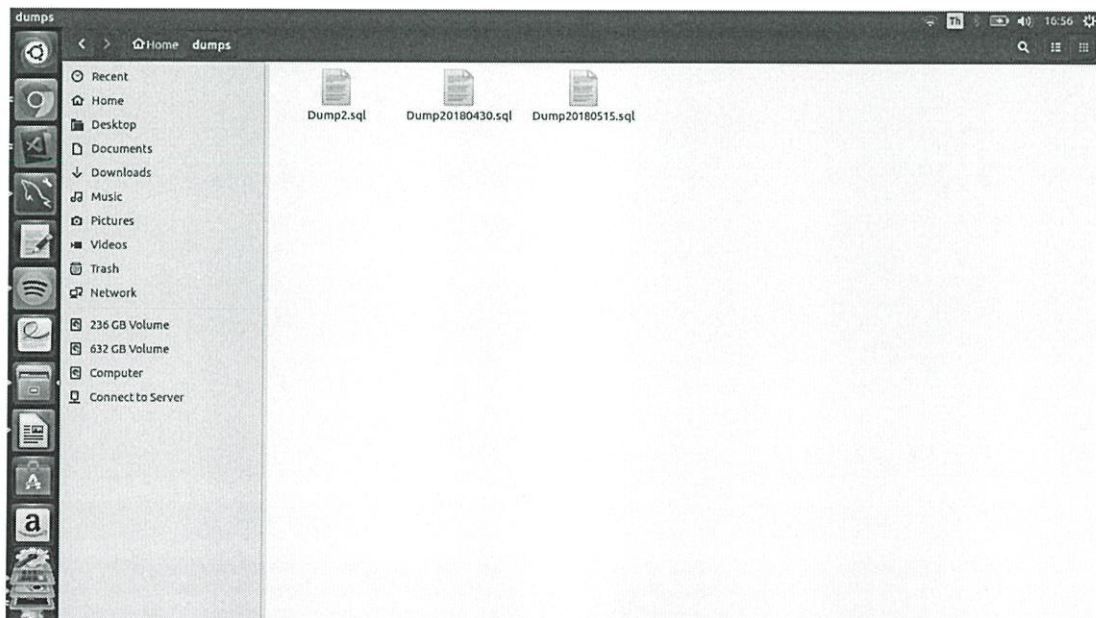
```
ssh-add -K ~/.ssh/<Your_Own_Private_Key>
```

ทำการ Import ฐานข้อมูล จาก localhost โดยให้ทำการ Export ฐานข้อมูลจากโปรแกรม MySQL Workbench ก่อนดังภาพ



รูป 4ก. การ Export ฐานข้อมูลจากโปรแกรม MySQL Workbench

เมื่อ Export ออกมาแล้วจะได้ Dump File ดังภาพ



รูป 5ก. Dump File ที่ Export ได้จากโปรแกรม MySQL Workbench

หลังจากที่ได้ Dump File ออกมาแล้วให้ทำการ Upload ไฟล์ดังกล่าวขึ้นไป Server โดยในที่นี้จะขอ Upload ไปพร้อมกับไฟล์โปรเจกต์ชื่อว่า ps-api โดยจะเพิ่มเข้าไปในส่วนของ MySQL ซึ่งใช้คำสั่ง

```
mysql -u username -p database_name < FILE_NAME.sql>
```

ซึ่งจะได้ผลลัพธ์ดังภาพ

```
root@paymesys:~# ls
PS-api ps-app
root@paymesys:~# cd PS-api/
root@paymesys:~/PS-api# ls
README.md      package.json      server.js
dump2.sql      routes            yarn.lock
node_modules  routesScreenshot from 2018-04-28 22-13-06.png
root@paymesys:~/PS-api# mysql -u root -p paymesys < dump.sql
-bash: dump.sql: No such file or directory
root@paymesys:~/PS-api# mysql -u root -p paymesys < dump2.sql
Enter password:
root@paymesys:~/PS-api# █
```

### รูป เค. การ Import ฐานข้อมูล

หลังจากเพิ่มฐานข้อมูลเสร็จสิ้นแล้ว ให้ทำการตรวจสอบฐานข้อมูลโดยใช้คำสั่งดังนี้

```
mysql -u root -p
SHOW DATABASES;
USE paymesys;
SHOW TABLES;
DESCRIBE person;
```

ซึ่งคำสั่งดังกล่าวจะเป็นการเข้าสู่ฐานข้อมูลแล้วทำการเปิดดูฐานข้อมูลในเครื่อง Server ซึ่งจะได้ผลลัพธ์ดังนี้

```
mysql> SHOW DATABASES;
+-----+
| Database           |
+-----+
| information_schema |
| mysql              |
| paymesys           |
| performance_schema |
| sys                |
+-----+
5 rows in set (0.00 sec)

mysql> SHOW paymesys;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the right syntax to use
near 'paymesys' at line 1
mysql> USE paymesys;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW paymesys;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the right syntax to use
near 'paymesys' at line 1
mysql> SHOW TABLES;
+-----+
| Tables_in_paymesys |
+-----+
| conference          |
| paper               |
| person              |
| receipt_data        |
| statement            |
+-----+
5 rows in set (0.00 sec)
```

รูป 7ค. การใช้งานฐานข้อมูลใน Server

```
mysql> select * from statement;
+-----+-----+-----+-----+-----+-----+-----+
| pid | email                | name                | invoice | receipt | paymethod | conf       | amount |
+-----+-----+-----+-----+-----+-----+-----+
| 12  | kw@kmitl.ac.th       | wiboon prompanit   | 24      | 219     | paypal    | icsec2017 | 21000 |
| 33  | sirawit.sw16@gmail.com | Sirawit Wanarattikal | 26      | 221     | creditcard | icsec2017 | 21000 |
| 52  | en.tnc@hotmail.com   | Thanatcha sangphet | 25      | 220     |           |           | NULL  |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

รูป 8ค. การเข้าถึงฐานข้อมูลใน Server โดยใช้ภาษา SQL

เมื่อทำการตรวจสอบแล้วว่าฐานข้อมูลถูกต้องแล้ว จากนั้นก็ทำการ Clone ไฟล์โปรเจกของเราทั้งส่วน Front-End และ Back-End ด้วยคำสั่งดังนี้

```
git clone https://github.com/eaotnc/ps-app.git
```

```
git clone https://github.com/eaotnc/PS-api.git
```

หลังจากนั้นให้ทำการติดตั้ง Package ในโฟลเดอร์ที่ Clone มาอีกครั้งทั้ง ps-app และ ps-api ทั้ง 2 โฟลเดอร์โดยใช้คำสั่ง

```
yarn install
```

```
yarn start
```

เมื่อทำการติดตั้ง Package ทั้ง 2 โฟลเดอร์สำเร็จแล้ว ก็จะสามารถเข้าใช้งานได้โดยใช้ IP Address ผ่านโปรแกรม Web Browder ดังต่อไปนี้

<http://159.65.139.64:8080> เป็น IP Address สำหรับหน้าเว็บไซต์

<http://159.65.139.64:8000/documentation> เป็น IP Address สำหรับการใช้งานการจัดการ API ด้วย Swagger

หากสามารถเข้าถึงได้ก็เป็นอันสิ้นสุดขั้นตอนการ Deploy เว็บไซต์ขึ้น Server

ภาคผนวก ง.

คู่มือการใช้งานระบบ

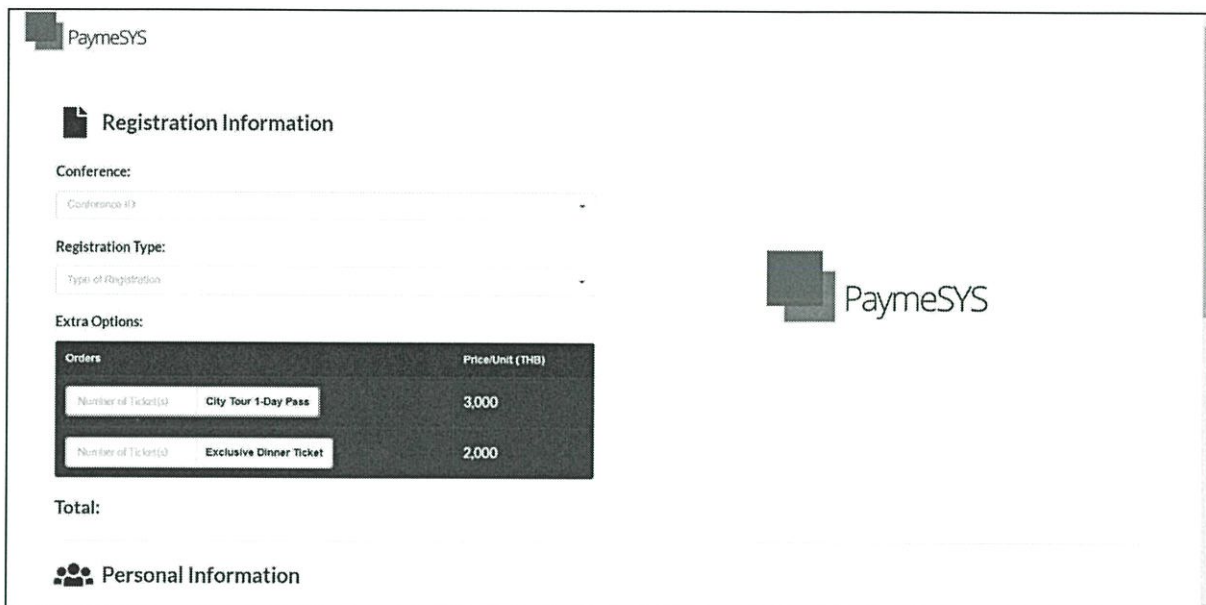
การใช้งานระบบนั้น สามารถแบ่งประเภทของการใช้งานได้เป็น 3 ประเภท ได้แก่ผู้เข้าร่วมงาน (Attendee), ผู้จัดงาน (Organizer) และผู้เข้าร่วมงานที่ไม่ได้ส่งผลงาน (Guest) ซึ่งผู้ใช้งานแต่ละประเภทจะมีฟังก์ชันและขั้นตอนการใช้งานที่แตกต่างกัน ซึ่งมีรายละเอียดดังต่อไปนี้

### 1. ผู้เข้าร่วมงานที่ไม่ได้ส่งผลงาน (Guest)

สำหรับผู้ใช้งานประเภทนี้ อาจเป็นบุคคลที่ไม่ได้ลงทะเบียนส่งผลงานเพื่อนำเสนอ อาจเป็นผู้ที่มาเยี่ยมชมหรือลงทะเบียนใกล้วันงาน ซึ่งระบบสามารถรองรับการชำระเงินเข้าร่วมงานได้ ซึ่งมีขั้นตอนดังนี้

รูป 1ง. หน้าจอการลงชื่อเข้าใช้งาน

เมื่อเข้าสู่หน้าจอสำหรับลงชื่อเข้าใช้งานระบบ ให้กดไปที่ Signup as Visitor เพื่อทำการลงทะเบียนและกรอกข้อมูลผู้ใช้งานต่างๆ ดังนี้



PaymeSYS

### Registration Information

Conference:  
Conference #3

Registration Type:  
Type of Registration

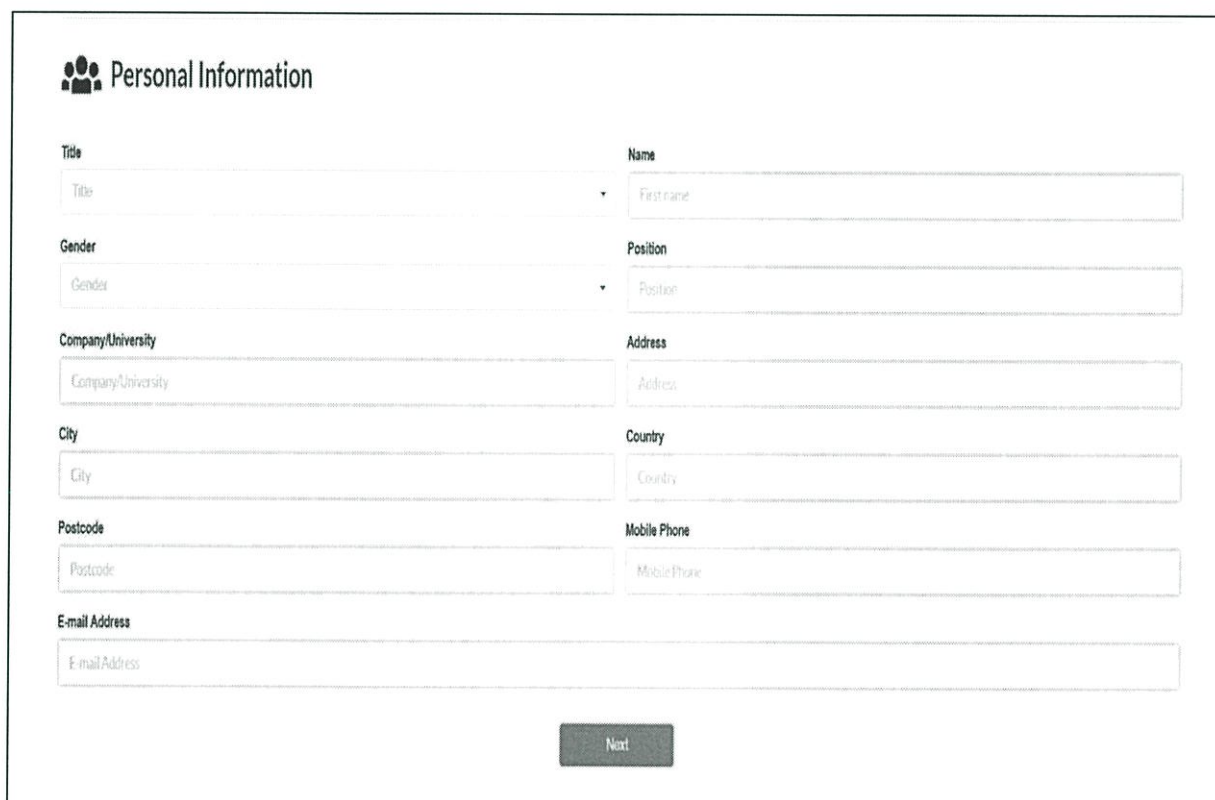
Extra Options:

Orders	Price/Unit (THB)
Number of Ticket(s) City Tour 1-Day Pass	3,000
Number of Ticket(s) Exclusive Dinner Ticket	2,000

Total:

Personal Information

รูป 2ง. หน้าจอสำหรับเลือกงานประชุมเพื่อลงทะเบียน



### Personal Information

Title: Title

Name: Firstname

Gender: Gender

Position: Position

Company/University: Company/University

Address: Address

City: City

Country: Country

Postcode: Postcode

Mobile Phone: Mobile Phone

E-mail Address: E-mail Address

Next

รูป 3ง. หน้าจอสำหรับกรอกข้อมูลส่วนตัวของผู้ใช้งาน

เมื่อผู้ใช้งานกรอกข้อมูลต่างๆเสร็จสิ้นให้กดที่ปุ่ม Next ซึ่งระบบก็จะสรุปข้อมูลการลงทะเบียน พร้อมแจ้งจำนวนเงินที่ต้องชำระให้ทราบ

### Summary

**Conference:** ICSEC2017  
**Paper ID:** N/A  
**Billed Name:** Mr.Test Admin  
**Billed Address:** KMITL, 1 Chalongkrung Road, Ladkrabang, Bangkok  
**Payfor:** Mr.Test Admin


**Your Registration and Orders:**

Late Registration with IEEE Member: 11,000 THB  
 Exclusive Dinner : 3,000 THB

Total 14,000 THB

รูป 4ง. หน้าจอสรุปผลข้อมูลการลงทะเบียน


เมื่อข้อมูลถูกต้องก็จะเข้าสู่หน้าจอสำหรับเลือกวิธีการชำระเงินเมื่อผู้ใช้คลิกปุ่ม Pay โดยจะสามารถเลือกรูปแบบการชำระเงินได้ดังนี้


 PaymeSYS


**Payments Options**  
 Make payment for your conference(s).


**Incomplete** ✖

PAYMENT OPTIONS

  
 PayPal

  
 Credit Card

  
 Bank Transfer

  
 Cash (On-Desk)

\*If you pay by cash, you must verify with organizer by yourself.

รูป 5ง. หน้าจอเลือกวิธีการชำระเงิน

ระบบรองรับการชำระเงิน 4 วิธี ได้แก่ ชำระเงินผ่านระบบ PayPal, Credit Card, การโอนเงินผ่านบัญชีธนาคาร และการชำระเงินหน้างาน

เมื่อผู้ใช้งานชำระเงินเสร็จสิ้น ระบบก็จะเก็บข้อมูลเข้าฐานข้อมูลของงานประชุมที่ลงทะเบียน แล้ว email ก็จะถูกบันทึกไว้ เมื่อกลับมาใช้งานระบบอีกครั้งก็จะทำให้สามารถลงชื่อเข้าใช้งานแบบผู้เข้าร่วมงานปกติได้

## 2. การใช้งานแบบผู้เข้าร่วมงาน (Attendee)

การลงทะเบียนใช้งานแบบผู้เข้าร่วมงาน คือผู้ที่ส่งผลงานและทางคณะกรรมการอนุมัติผลงานให้เข้าร่วมในงานประชุม ซึ่งผู้ใช้งานประเภทนี้จะมี email ที่ถูกบันทึกไว้ในฐานข้อมูลของงานประชุมอยู่แล้วโดยผู้ใช้งานสามารถใช้ email ที่ได้ให้ไว้กับทางงานประชุมลงทะเบียนใช้งานได้ทันที โดยจะใช้รหัสผ่าน OTP ในการลงทะเบียนใช้งาน ซึ่งบัญชีผู้ใช้งานประเภทนี้ได้มีฟังก์ชันการใช้งานดังนี้

### - Dashboard

**Dashboard**  
Manage all of conferences at one place.

Conference:  
ICSEC2017

Paper ID:  
CE6033

Payment Status:  
Incomplete ❌

Role(s): Speaker

Early Bird Deadline:

Payment Deadline:

MORE

**Active Conference(s)**

#	Conferences	PID	Receipt	Info
1	ICSEC 2017	CE6033		Conference Website
2	ICAN 2018	IP120		Conference Website

รูป 6ง. หน้าจอ Dashboard

สำหรับหน้า Dashboard นี้เป็นหน้าแรกที่ทำหน้าที่รวบรวมข้อมูลที่สำคัญของแต่ละงานประชุมเช่น สถานะการชำระเงิน, วันสุดท้ายของการชำระเงิน, วันสุดท้ายของการลงทะเบียนแบบล่วงหน้า (Early Birth) เป็นต้น ซึ่งผู้ใช้งานสามารถเลือกงานประชุมและรหัสผลงานเพื่อดูสถานะได้

## - Documents

### Documents Upload

Upload documents required by conference.

**Warning!!!**  
Don't forget to upload conference's documents

**Conference:**

ICSEC2017

▼

PaymeSYS

**Paper ID:**

CE6033

▼

Conference Logo HERE

Response to Commitee Comment	<input type="button" value="Choose File"/> No file chosen
IEEE Copyright Form	<input type="button" value="Choose File"/> No file chosen
Speaker Biology Form	<input type="button" value="Choose File"/> No file chosen
Camera Ready (Paper Complete Edition)	<input type="button" value="Choose File"/> No file chosen

DOCUMENTS STATUS

---

- Response to Commitee Comment
- IEEE Copyright Form
- Speaker Biology Form
- Camera Ready (Complete Edition of Paper)

รูป 7ง. หน้าจอของหน้า Documents

สำหรับหน้า Documents นี้จะเป็นส่วนที่ให้ผู้ใช้งานได้ทำหารอัปโหลดไฟล์เอกสารที่ทางงานประชุมร้องขอเช่น ข้อมูลผู้นำเสนอผลงาน, เอกสารลิขสิทธิ์ต่างๆ และเอกสารผลงานฉบับสมบูรณ์ เป็นต้น ซึ่งในหน้านี้จะแสดงสถานะของแต่ละเอกสารว่าผ่านหรือต้องปรับปรุงแก้ไขแล้วทำการส่งมาใหม่

## - Payments

### Payments

Make payment for your conference(s).

**Incomplete** ⊕

**Conference:**

**Paper ID:**

**Pay for:**

**Billed Name:**

**Registration Type:**

**Extra Options:**

Orders	Price/Unit (THB)
<input type="text" value="Number of Ticket(s)"/> <input type="text" value="City Tour 1-Day Pass"/>	3,000
<input type="text" value="Number of Ticket(s)"/> <input type="text" value="Exclusive Dinner Ticket"/>	2,000

รูป 8ง. หน้าจอของหน้า Payments ส่วนที่ 1

**Billed Address:**

**Address 1**

**Address 2**

**City**  **Country**  **Postcode**

**PAYMENT OPTIONS**

PayPal

Credit Card

Bank Transfer

Cash (On Desk)

*\*If you pay by cash, you must verify with organizer by yourself.*

**PAYMENT VERIFY**

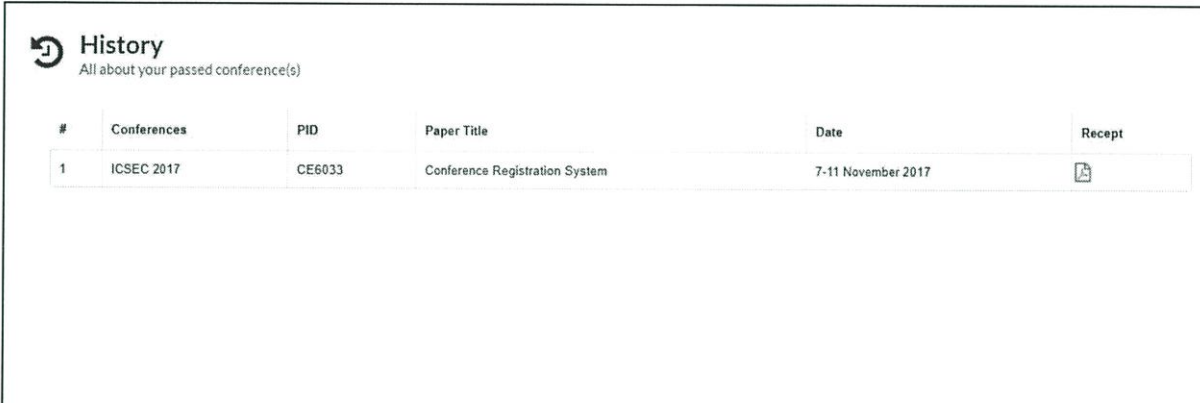
For Bank Transfer, Please upload payment slip.


No file chosen

รูป 9ง. หน้าจอของหน้า Payments ส่วนที่ 2

สำหรับหน้า Payments นี้เป็นส่วนสำหรับการลงทะเบียนและชำระเงินในงานประชุม โดยผู้ใช้สามารถเลือกงานประชุมและรหัสผลงานเพื่อทำการลงทะเบียน จากนั้นทำการเลือกประเภทการลงทะเบียน แล้วเลือกว่าจะชำระเงินให้กับใคร ชื่อที่จะใช้สำหรับออกใบเสร็จรับเงิน และสุดท้ายคือที่อยู่สำหรับออกใบเสร็จรับเงิน รวมถึงการลงทะเบียนรายการเสริมต่างๆ จากนั้นก็เลือกวิธีการชำระเงิน และหากต้องการ Upload หลักฐานการชำระเงินก็สามารถ Upload ได้ที่นี่

#### - History

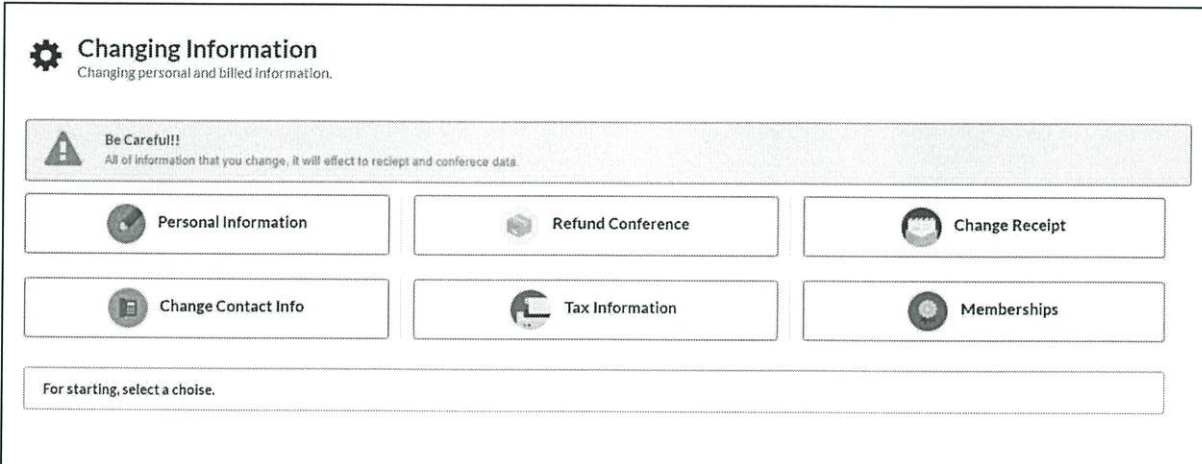


#	Conferences	PID	Paper Title	Date	Receipt
1	ICSEC 2017	CE6033	Conference Registration System	7-11 November 2017	

รูป 10ง. หน้าจอของหน้า History

สำหรับส่วนนี้จะเป็นส่วนที่เก็บประวัติการเข้าร่วมงานของผู้ใช้ โดยจะเก็บใบเสร็จรับเงินไว้ด้วย เพื่อให้ผู้ใช้สามารถ Download ไปใช้งานได้ แต่จะไม่สามารถแก้ไขได้แล้ว รวมถึงไม่สามารถปรับเปลี่ยนข้อมูลการลงทะเบียนงานประชุมด้วย โดยจะสามารถเก็บประวัติย้อนหลังได้สูงสุด 5 ปี (เนื่องมาจากเกี่ยวข้องกับการคำนวณภาษีต่างๆ)

#### - Changing Information



**Changing Information**  
Changing personal and billed information.

**Be Careful!!**  
All of information that you change, it will effect to receipt and conference data.

For starting, select a choice.

รูป 11ง. หน้าจอของหน้า Changing Information

สำหรับหน้านี้ถือว่าเป็นหน้าที่สำคัญที่สุดอีกส่วนหนึ่ง โดยส่วนนี้จะทำหน้าที่ในการเปลี่ยนแปลงข้อมูลต่างๆของผู้ใช้งานซึ่งสามารถแบ่งได้ดังนี้

**Personal Information** เป็นส่วนสำหรับแก้ไขข้อมูลส่วนบุคคลเช่นที่อยู่, ชื่อและนามสกุล, ตำแหน่งปัจจุบัน ซึ่งข้อมูลส่วนนี้ปกติจะทำการอัปเดตจากฐานข้อมูลงานประชุมที่ผู้ใช้งานลงทะเบียนไว้ล่าสุดอยู่แล้ว แต่หากมีข้อผิดพลาดก็สามารถแก้ไขได้ ข้อควรระวังคือการแก้ไขทุกครั้งจะส่งผลกระทบต่อข้อมูลของงานประชุม

**Changing Contact Info** เป็นส่วนสำหรับแก้ไขข้อมูลติดต่อเช่น หมายเลขโทรศัพท์ที่ติดต่อได้, หมายเลขโทรศัพท์ของหน่วยงานที่สังกัดอยู่, หมายเลขแฟกซ์ เป็นต้น

**Refund Conference** เป็นส่วนสำหรับส่งคำขอเพื่อขอคืนเงินค่าลงทะเบียนงานประชุม แต่เป็นเพียงแค่ระบบส่งข้อความไปยังผู้จัดงานเท่านั้น ซึ่งจะสามารถคืนเงินได้หรือไม่ขึ้นอยู่กับทางผู้จัดงาน

**Tax Information** เป็นส่วนที่ใช้สำหรับแก้ไขข้อมูลทางภาษีต่างๆ

**Change Receipt** เป็นส่วนสำหรับแก้ไขข้อมูลใบใบเสร็จรับเงิน ข้อควรระวังคือสามารถแก้ไขข้อมูลได้เพียงแค่ครั้งเดียวเท่านั้น เพื่อป้องกันการออกใบเสร็จซ้ำซ้อน

**Memberships** เป็นส่วนที่ใช้สำหรับแก้ไขข้อมูลสมาชิกต่างๆที่ผู้ใช้เข้าร่วมอยู่ ซึ่งระบบจะให้ผู้ใช้เพิ่มหรือลบเองได้ แต่จำเป็นต้องส่งหลักฐานการเป็นสมาชิกให้อนุมัติก่อน

### 3. การใช้งานแบบผู้จัดงาน (Organizer)

สำหรับการใช้งานประเภทนี้เป็นเฉพาะหน่วยงานที่ทางระบบได้ออกบัญชีผู้ใช้งานให้ ซึ่งจะเป็นบัญชีที่ใช้ในการสร้าง, ติดตามสถานะ, จัดการงานประชุม รวมถึงตรวจสอบข้อมูลทางการเงินต่างๆ ซึ่งมีฟังก์ชันการใช้งานดังต่อไปนี้

## - Summary

**Overview**

ICSEC2017

**Conference Status**

- Event Date : 15-19 November 2017
- Early Birth Acceptance: No
- Payment Acceptance: Yes

**Conference Data**

- All of Attendee : 250
- All of Paper : 89
- Completed Documents (Papers) : 50
- Completed Payment (Papers) : 82

รูป 12ง. หน้าจอของหน้า Summary

ในส่วนนี้จะเป็นหน้าสรุปข้อมูลงานประชุม ซึ่งจะแสดงวันที่ที่จัดงานประชุม, สถานะการลงทะเบียนล่วงหน้าและชำระเงิน รวมถึงจำนวนผู้เข้าร่วมงานทั้งหมด, สรุปจำนวนผลงานที่ส่งเข้าร่วมงานประชุมทั้งหมด, จำนวนผู้ที่ชำระเงินแล้ว และจำนวนผู้ที่ Upload เอกสารงานประชุมเสร็จสิ้นแล้ว

ผู้ใช้จำเป็นต้องเลือกงานประชุมที่จะให้ระบบแสดงในหน้านี้ ซึ่งระบบจะทำการเก็บค่าสถานะไว้เพื่อใช้สำหรับการจัดการในส่วนอื่นๆ ซึ่งหากผู้ใช้งานจะเปลี่ยนงานประชุมต้องดำเนินการเปลี่ยนในหน้านี้เท่านั้น

## - Conferences

## Conferences

**Beware to config conference data.**  
It will effect for payment rate and attendee accounts.

Configuration

ICSEC2017

**Changing Start Date:**

Starting Date

**Changing Finish Date:**

Finishing Date

**Changing Early Birth Deadline:**

Early Birth Deadline

**Changing Payment Deadline:**

Payment Deadline

Cancel
Confirm

Attendees Management

Registration Rate Config

Conference Options:

Create Conference

รูป 13ง. หน้าจอของหน้า Conferences

ในส่วนนี้จะเป็นส่วนที่เกี่ยวข้องกับงานประชุม โดยผู้ใช้สามารถปรับเปลี่ยนวันที่ของงาน, วันสุดท้ายของการชำระเงินได้ สามารถกำหนดอัตราค่าลงทะเบียนให้สำหรับผู้เข้าร่วมงานแต่ละคนได้ รวมถึงปรับเปลี่ยนอัตราค่าลงทะเบียนต่างๆ ได้เช่นกันจากหน้านี้ ดังภาพ

### 1 Registration Rate Configuration

<b>Early Birth Registration</b> THB Amount .00	<b>Member Early Birth Registration</b> THB Amount .00	<b>Regular Late Registration</b> THB Amount .00
<b>Member Late Registration</b> THB Amount .00	<b>Student Late Registration</b> THB Amount .00	<b>Visitor (Non-Paper)</b> THB Amount .00
<b>Extra Registration:</b>		
<b>Exclusive Dinner:</b> THB Amount .00	<b>1-Day Pass City Tour</b> THB Amount .00	

รูป 14. หน้าจอของหน้าปรับอัตราค่าลงทะเบียน

### Create a Conference

<b>Conference Logo</b> <input type="button" value="Choose File"/> No file chosen	<b>Conference Name</b> <input type="text" value="Conference Name/ID"/>	
<b>Starting Date:</b> <input type="text" value="Starting Date"/>		
<b>Finishing Date:</b> <input type="text" value="Finishing Date"/>		
<b>Early Birth Deadline:</b> <input type="text" value="Early Birth Deadline"/>		
<b>Payment Deadline:</b> <input type="text" value="Payment Deadline"/>		
<b>Registration Rate:</b>	<b>Member Early Birth Registration</b> THB Amount .00	<b>Regular Late Registration</b> THB Amount .00
<b>Early Birth Registration</b> THB Amount .00	<b>Student Late Registration</b> THB Amount .00	<b>Visitor (Non-Paper)</b> THB Amount .00
<b>Member Late Registration</b> THB Amount .00	<b>Extra Registration:</b>	
<b>Exclusive Dinner:</b> THB Amount .00	<b>1-Day Pass City Tour</b> THB Amount .00	
<b>Database Import:</b> <input type="button" value="Choose File"/> No file chosen We accept only csv file		


รูป 15. หน้าจอของหน้าสร้างงานประชุม

นอกจากนี้ผู้ใช้สามารถสร้างงานประชุมขึ้นมาใหม่ได้พร้อมทั้งนำเข้าฐานข้อมูลในรูปแบบไฟล์ .csv จากฐานข้อมูลงานประชุมเดิมได้ ทำให้ไม่ต้องสร้างฐานข้อมูลขึ้นมาใหม่

## - Financial

**Financial**

ICSEC2017


Net Income:  3


**Refund Request:**

**PI503**  
Request By: example@test.com ignore  
Reason: incidence events. Please refund it.

Approve Decline


**Options:**

 Statement

 Conference's Bank Account

รูป 16ง. หน้าจอของหน้า Financial


สำหรับส่วนต่อมาก็คือส่วนของการจัดการทางการเงิน โดยในหน้านี้จะมีการแสดงยอดเงินที่ได้รับทั้งหมดของงานประชุม, ค่าของค่าน้ำตาลงทะเบียนงานประชุม ซึ่งจะเป็นการรออนุมัติว่าจะให้ค่าน้ำตาลงทะเบียนหรือไม่ และมีประวัติการชำระเงินของผู้เข้าร่วมงานอยู่ในส่วนของ Statement

 **Statement**

ICSEC2017

PID	Email	Name	Invoice No.	Receipt No.	PayMethod	conference	Amount
25	en.tnc@hotmail.com	sirawit Wanarattikal	214	204	credit card	ICSEC2017	1
23	kwee@hotmail.com	savitree boonmee	205	189	credit card	ICSEC2017	1
18	walineee@hotmail.com	wavitree meekhun	202	185	unpaid	ICANN2017	0
23	kwee@hotmail.com	savitree boonmee	205	189	unpaid	ICANN2017	0
24	en.tnc@hotmail.com	Thanatcha sangphet	203	189	unpaid	ICSEC2017	0

รูป 17ง. หน้าจอของหน้า Statement

 **Conference's Bank Account**

**Account Name**  
Account name

**Account Address**  
Account address

**Bank**  
Bank

OR

**K-Payment Gateway Merchant ID:**

**Merchant ID**  
Merchant ID

รูป 18ง. หน้าจอของหน้า Conference Bank Account

ส่วนที่สำคัญอีกอย่างหนึ่งคือบัญชีธนาคารของงานประชุมที่จะให้ทางระบบ โอนเงินให้กับผู้ใช้งาน ซึ่งทางผู้ใช้งานสามารถกรอกรายละเอียดได้ที่เมนู Conference Bank Account

## - Documents Review

Documents Review

ICSEC2017

Auto Accept

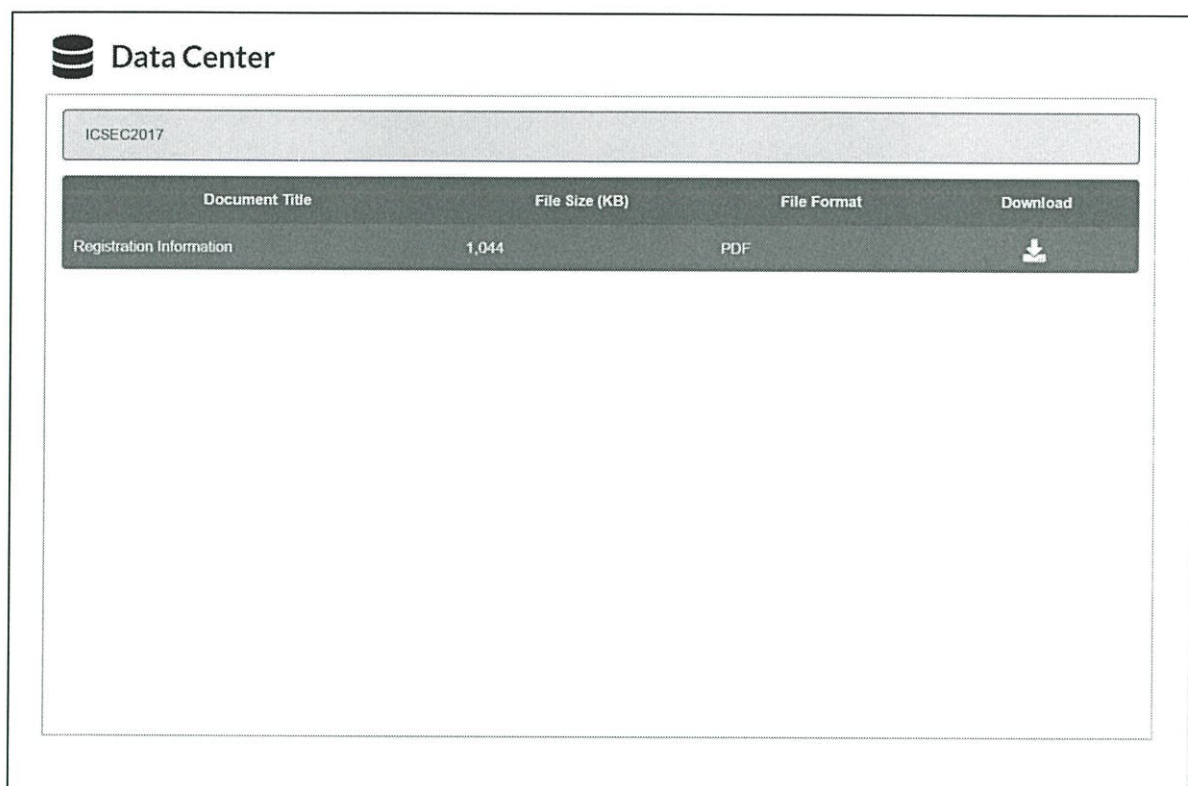
PID	Commenteter Review	IEEE Copyright	Speaker Biology	Camera Ready	Confirm All
CE6033	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Confirm


รูป 19ง. หน้าจอของหน้า Documents Review

การใช้งานในส่วนนี้เป็นการใช้งานเพื่อทำการอนุมัติหรือปฏิเสธเอกสารที่ทางผู้เข้าร่วมงานส่งมา ซึ่งผู้ใช้สามารถเลือกที่จะใช้การอนุมัติแบบอัตโนมัติหรือกำหนดเองก็ได้

## - Data Center



The screenshot shows a web interface titled "Data Center" with a sub-header "ICSEC2017". Below this is a table with the following structure:

Document Title	File Size (KB)	File Format	Download
Registration Information	1,044	PDF	

รูป 20ง. หน้าจอของหน้า Data Export

เป็นส่วนที่ใช้สำหรับดาวน์โหลดเอกสารข้อมูลของต่างๆของงานประชุม โดยจะรวบรวมเอกสารสำคัญไว้เช่น รายชื่อผู้ลงทะเบียนเข้าร่วมงาน, รายชื่อและรหัสผลงานที่ถูกส่งเข้ามาร่วมงานประชุม, รายการสถานะการชำระเงินของผู้เข้าร่วมงาน เป็นต้น