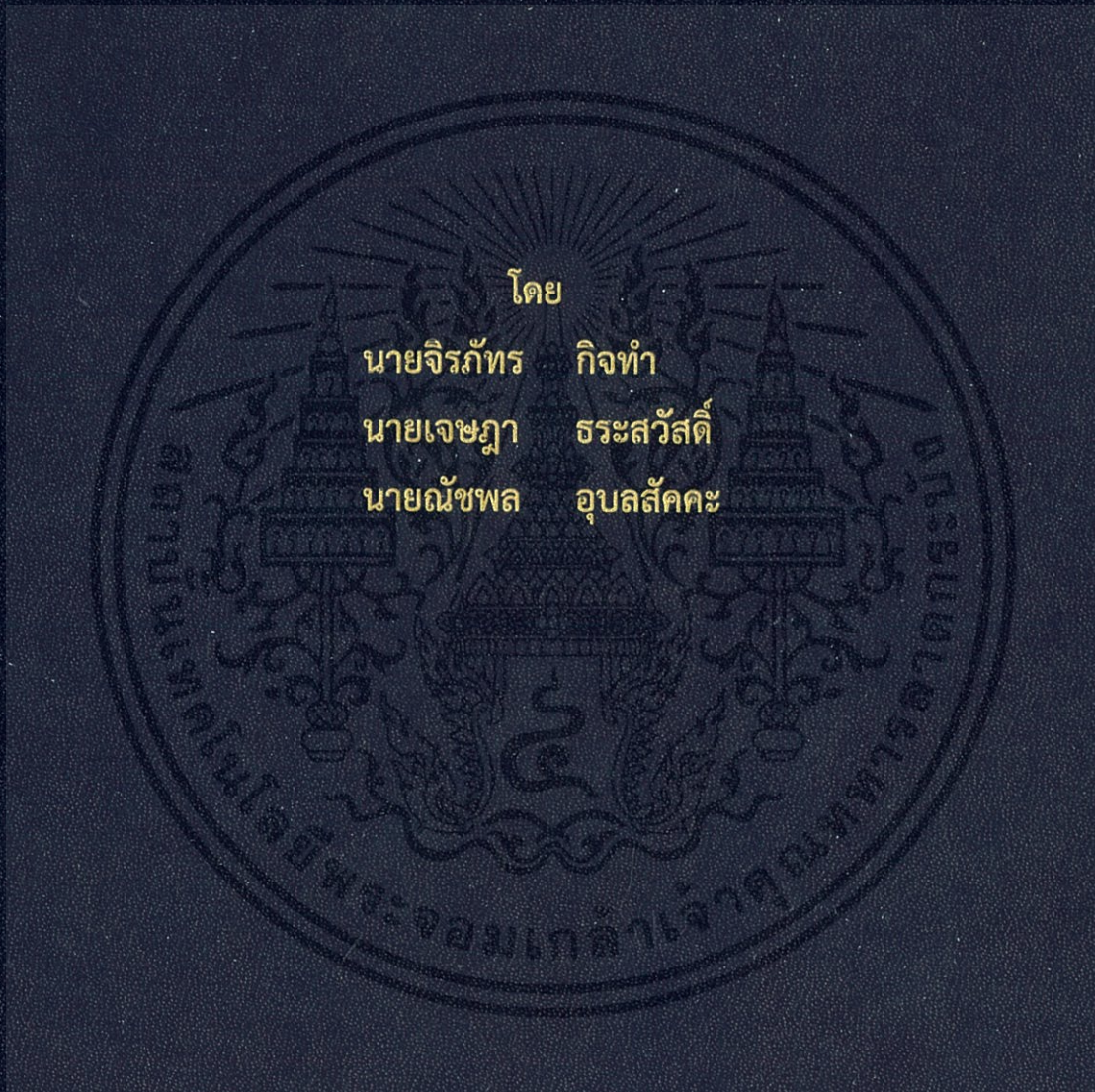


ระบบการจัดการการใช้งานห้องเรียนโดยการใช้บล็อกเชน
CLASSROOM MANAGEMENT SYSTEM BY USING BLOCKCHAIN



โดย

นายจิรภัทร กิจท่า

นายเจษฎา ธรรมสวัสดิ์

นายณัชพล อุบลศักดิ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

ระบบการจัดการการใช้งานห้องเรียนโดยการใช้บล็อกเชน
CLASSROOM MANAGEMENT SYSTEM BY USING BLOCKCHAIN

โดย

นายจิรภัทร กิจท่า	58010187
นายเจษฎา ธรรมสวัสดิ์	58010211
นายณัฏพล อุบลศักดิ์	58010353

อาจารย์ที่ปรึกษา

รศ.ดร. ปราโมทย์ วาดเขียน

รศ.ดร. จีรสุดา โกษียาภรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

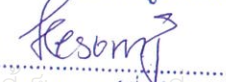
ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

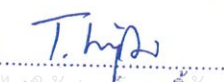
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

ผ่านการตรวจรูปเล่มแล้ว


.....
อาจารย์ที่ปรึกษา
20/5/62

ผ่านการตรวจชิ้นงานแล้ว


.....
กรรมการตรวจชิ้นงาน
23/5/62

ปริญญาานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบการจัดการการใช้งานห้องเรียนโดยใช้บล็อกเชน

CLASSROOM MANAGEMENT SYSTEM BY USING BLOCKCHAIN

ผู้จัดทำ

1. นายจิรภัทร กิจทำ 58010353
2. นายเจษฎา ธรรมสวัสดิ์ 58010211
3. นายณัชพล อุบลศักดิ์ 58010353


..... อาจารย์ที่ปรึกษา
(รศ.ดร. ปราโมทย์ วาดเขียน)


..... อาจารย์ที่ปรึกษาร่วม
(รศ.ดร. จีรสุดา โกษียากรณ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์นี้จะไม่สามารถดำเนินการได้อย่างลุล่วง หากขาดคำแนะนำ ให้ความช่วยเหลือทั้งแนวทางในการทำงาน ให้คำปรึกษาทางทฤษฎีและปฏิบัติ โดยบุคคลที่จะกล่าวถึงคือ รศ.ดร. ปราโมทย์ วาดเขียน รศ.ดร. จีรสุดา โกษิยาภรณ์ และ ผศ.ดร. ธรรมนูญ พุทธรักษ์ ซึ่งเป็นผู้คอยอบรมและแนะนำตลอดการดำเนินงานปริญญาานิพนธ์นี้ จึงขอขอบพระคุณอาจารย์ทั้งสามท่าน ด้วยความเคารพอย่างสูง

นอกจากนี้คณะผู้จัดทำ ขอขอบพระคุณ นาย สนั่นฟ้า สิงโตทอง ซึ่งเป็นผู้ให้คำแนะนำในการทำงาน ให้ความรู้เกี่ยวกับเนื้อหาที่เกี่ยวข้องและให้คำปรึกษาที่ติดต่อมา ท้ายสุดคณะผู้จัดทำขอขอบพระคุณ บิดา มารดา

นายจิรภัทร กิจทำ

นายเจษฎา ธรรมสวัสดิ์

นายณัชพล อุบลศักดิ์

ผู้จัดทำ

ระบบการจัดการการใช้งานห้องเรียนโดยการใช้บล็อกเชน
CLASSROOM MANAGEMENT SYSTEM BY USING
BLOCKCHAIN

โดย	นายจิรภัทร กิจท่า	58010187
	นายเจษฎา ธรรมสวัสดิ์	58010211
	นายณัฏพล อุบลศักดิ์	58010353

อาจารย์ที่ปรึกษา รศ.ดร.ปราโมทย์ วาดเขียน
รศ.ดร.จิรสุตา โกษียาภรณ์

บทคัดย่อ

ด้วยเทคโนโลยี Blockchain กำลังได้รับความนิยมและมีแนวโน้มที่จะถูกนำมาพัฒนาและมีการใช้งานในอนาคต ปริณยานิพนธ์นี้จึงได้นำทฤษฎีของเทคโนโลยี Blockchain มาประยุกต์ในเรื่องของการจัดเก็บข้อมูลเพื่อใช้ในการออกแบบระบบการจัดการการใช้งานห้องเรียน ในปริณยานิพนธ์นี้ Hyperledger Fabric Blockchain ได้ถูกนำมาใช้งานในการเก็บข้อมูลและการจัดการระบบซึ่งรวมถึงการจองห้องเรียนและการตรวจสอบการเข้าเรียนของนักศึกษา ซึ่งด้วยการใช้เทคโนโลยี Blockchain จะทำให้การเก็บและการจัดการข้อมูลนั้นมีความน่าเชื่อถือ ปลอดภัยจากการปลอมแปลงข้อมูล องค์ประกอบหลักของงานที่นำเสนอจะประกอบด้วย 3 ส่วน กล่าวคือ ส่วนที่หนึ่ง คือ Blockchain ซึ่งจะช่วยในการจัดการและจัดเก็บข้อมูล ส่วนที่สอง คือ เว็บไซต์ ซึ่งจะเป็นส่วนที่ทำหน้าที่เชื่อมต่อระหว่างผู้ใช้งานกับข้อมูลและ Blockchain และ ส่วนที่สาม คือ ส่วนการจัดการการใช้งานห้องเรียน ซึ่งจะมีการจัดการทั้งเชิงข้อมูลและเชิงกายภาพ โดยในเชิงข้อมูลนั้น QR code จะถูกนำมาใช้ในการตรวจสอบการเข้าเรียน ส่วนในเชิงกายภาพชุดประตูแบบแถบแม่เหล็กไฟฟ้าจะเป็นอุปกรณ์ที่ทำหน้าที่การคัดกรองการเข้าห้องเรียน

ABSTRACT

Nowsaday, Blockchain is a well received attention technology and it trends to be developed for many applications in the future. In this thesis, the concept of Blockchain technology is thus applied in data management for using in classroom management system. For a Blockchain framework, Hyperledger Fabric is used for collecting data and managing system including classroom reservation and classroom attending of students. With Blockchain technology, it makes collecting and managing data reliability and security from data deception. The proposed system is consisted of 3 parts. The first part is Blockchain which is collecting and managing data. The second part is website which is connecting between users and data and Blockchain. Finally, the last part is classroom management which includes managing in data and object aspects. For data management, QR code is used for verification of classroom attendants whereas electromagnetic door lock set is an device (object) used for screening of classroom usage.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VII
บทที่ 1	
บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	
2.1 Blockchain	2
2.1.1 Distributed Ledger	3
2.1.2 Smart Contract	3
2.1.3 Data Privacy	3
2.1.4 Consensus Protocol	4
2.2 Hyperledger Fabric Blockchain	4
2.2.1 Hyperledger Composer	5
2.2.2 Hyperledger Composer Playground	9
2.3 Docker	9
2.3.1 Docker Image	9
2.3.2 Docker Container	10
2.3.3 Docker Registry	10
2.3.4 Docker Compose	10

สารบัญ (ต่อ)

	หน้า
2.4 Laravel PHP Framework	10
2.5 Relay	11
บทที่ 3 การออกแบบและการจัดทำปริญญานิพนธ์	
3.1 การออกแบบ	13
3.1.1 การออกแบบในส่วนของ Hyperledger Fabric Blockchain	14
3.1.2 การออกแบบในส่วนของ Webpage	26
3.1.3 การออกแบบในส่วนควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า	34
3.2 เครื่องมือที่ใช้ในการทดลอง	36
3.2.1 Oscilloscope	36
3.2.2 Raspberry Pi 3 Model B	37
3.2.3 กลอนแม่เหล็กไฟฟ้า	38
3.3 การจัดเก็บผลการทดลอง	38
3.3.1 การจัดเก็บผลการทดลองในส่วนของ Hyperledger Composer	38
3.3.2 การจัดเก็บผลการทดลองในส่วนของ Webpage	39
3.3.3 การจัดเก็บผลการทดลองในส่วนควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า	40
3.3.4 การจัดเก็บผลการทดลองในส่วนการ Join Class ผ่านการสแกน QR Code	40

สารบัญ (ต่อ)

	หน้า
บทที่ 4	ผลการทดลอง
4.1	ผลการทดลองในส่วนของ Hyperledger Composer 41
4.1.1	ผลการทดสอบการจัดเก็บข้อมูลของ Participants และ Assets 41
4.1.2	ผลการทดสอบการทำงานของ Transactions 52
4.1.3	ผลการทดสอบการทำงานของ Access Control 84
4.2	ผลการทดลองในส่วนของ Webpage 92
4.2.1	ผลการทดสอบในส่วนของการแสดงผล Admin Webpage 92
4.2.2	ผลการทดสอบในส่วนของการแสดงผล Student Webpage 106
4.2.3	ผลการทดสอบในส่วนของการแสดงผล Teacher Webpage 115
4.3	ผลการทดลองในส่วนควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า 126
4.3.1	ผลการทดสอบวงจรควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า 126
4.3.2	ผลการทดสอบการควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้าผ่านไมโครคอนโทรลเลอร์ 127
4.3.3	ผลการทดสอบการทำงานของสวิตซ์ในการควบคุมการทำงานของไมโครคอนโทรลเลอร์ 128
4.4	ผลการทดลองในส่วนการตรวจสอบการเข้าชั้นเรียนผ่านการสแกน QR Code 129
บทที่ 5	สรุปผลและข้อเสนอแนะ
5.1	สรุปผล 134
5.2	ข้อเสนอแนะ 134
บรรณานุกรม	136

สารบัญรูป

รูปที่	หน้า
2.1 โครงสร้างของ Hyperledger Composer	7
2.2 การทำงานของรีเลย์ในการควบคุมการเปิด-ปิดอุปกรณ์อิเล็กทรอนิกส์	12
3.1 บล็อกไดอะแกรมแสดงภาพรวมการทำงานของระบบ	13
3.2 บล็อกไดอะแกรมแสดงการออกแบบในส่วนของ Model File	17
3.3 โฟลว์ชาร์ตแสดงการทำงานของ Transactions ในการจองห้องเรียน	18
3.4 โฟลว์ชาร์ตแสดงการทำงานของ Transactions ในการแก้ไขการจอง	19
3.5 โฟลว์ชาร์ตแสดงการทำงานของ Transactions ในการยกเลิกการจอง	20
3.6 โฟลว์ชาร์ตแสดงการทำงานของ Transactions ในการล็อกและปลดล็อกประตู	21
3.7 โฟลว์ชาร์ตแสดงการทำงานของ Transactions ในการเพิ่ม token	22
3.8 หน้าแสดงผลในการปรับใช้เครือข่ายธุรกิจของ Composer Playground	23
3.9 หน้าแสดงผลในการทดสอบเครือข่ายธุรกิจของ Composer Playground	23
3.10 หน้าแสดงผลสำหรับเลือกการ์ดเครือข่ายธุรกิจของ Composer Playground	25
3.11 หน้าแสดงผลของ Composer REST Server	25
3.12 บล็อกไดอะแกรมการทำงานส่วนของ Hyperledger Fabric	26
3.13 โฟลว์ชาร์ตแสดงการทำงานของ Admin Webpage	28
3.14 โฟลว์ชาร์ตแสดงการทำงานของ Teacher Webpage	30
3.15 โฟลว์ชาร์ตแสดงการทำงานของ Student Webpage	33
3.16 แผนผังวงจรควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า	34
3.17 บล็อกไดอะแกรมแสดงการทำงานของไมโครคอนโทรลเลอร์	36
3.18 Oscilloscope	37
3.19 Raspberry Pi 3 Model B	37
3.20 กลอนแม่เหล็กไฟฟ้า	38

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.1 Participants ของ Positions ใช้ในการกำหนดบทบาทของ Participants อื่นๆ	41
4.2 หน้าสำหรับกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง Participant ของ Admins	42
4.3 หน้าแสดงผลการจัดเก็บข้อมูลของ Admin01	42
4.4 หน้าสำหรับแก้ไขข้อมูลเพื่อทำการอัปเดตข้อมูลของ Admin01	43
4.5 หน้าแสดงผลข้อมูลของ Admin01 หลังทำการอัปเดตข้อมูล	43
4.6 หน้าสำหรับกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง Participant ของ Students	44
4.7 หน้าแสดงผลการจัดเก็บข้อมูลของ Student01	44
4.8 หน้าสำหรับแก้ไขข้อมูลเพื่อทำการอัปเดตข้อมูลของ Student01	45
4.9 หน้าแสดงผลข้อมูลของ Student01 หลังทำการอัปเดตข้อมูล	45
4.10 หน้าสำหรับกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง Participant ของ Teachers	46
4.11 หน้าแสดงผลการจัดเก็บข้อมูลของ Teacher01	46
4.12 หน้าสำหรับแก้ไขข้อมูลเพื่อทำการอัปเดตข้อมูลของ Teacher01	47
4.13 หน้าแสดงผลข้อมูลของ Teacher01 หลังทำการอัปเดตข้อมูล	47
4.14 หน้าสำหรับกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง Asset ของ StudentRoom	48
4.15 หน้าแสดงผลข้อมูลของ StudentRoom ห้อง S101	48
4.16 หน้าสำหรับกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง Asset ของ TeacherRoom	49
4.17 หน้าแสดงผลข้อมูลของ TeacherRoom ห้อง T101	49
4.18 หน้าสำหรับกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง Asset ของ TSRoom	50
4.19 หน้าแสดงผลข้อมูลของ TSRoom ห้อง TS101	50

สารบัญรูป (ต่อ)

รูปที่	หน้า	
4.20	หน้าสำหรับกรอกข้อมูลที่ใช้ในการสร้าง Asset ของ Classroom	51
4.21	หน้าแสดงผลการจัดเก็บข้อมูลของ Class01	51
4.22	หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction ของการจอง StudentRoom	53
4.23	หน้าแสดงผลการจัดเก็บข้อมูลการจองของห้อง S101	53
4.24	หน้าแสดงผลการจัดเก็บข้อมูลลำดับการจองของ Student01	54
4.25	หน้าแสดงผลเมื่อไม่สามารถทำการจอง StudentRoom ในวันเวลาที่ระบุได้	54
4.26	หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction แก่ไขการจองของ StudentRoom	55
4.27	หน้าแสดงผลการแก้ไขข้อมูลการจองของห้อง S101	56
4.28	หน้าแสดงผลเมื่อไม่สามารถเลื่อนการจอง StudentRoom ไปยังวันเวลาที่ระบุได้	56
4.29	หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction การยกเลิกการจอง StudentRoom	57
4.30	หน้าแสดงผลการลบข้อมูลการจองของห้อง S101	58
4.31	หน้าแสดงผลการลบข้อมูลลำดับการจองของ Student01	58
4.32	หน้าแสดงผลการทำงานเมื่อมี token ไม่เพียงพอสำหรับทำการจอง	59
4.33	หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction ของการเพิ่ม token	60
4.34	หน้าแสดงผลของการเพิ่ม token ของ Student01	60
4.35	หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ในการปลดล็อคประตูของ StudentRoom	61
4.36	หน้าแสดงผลการปลดล็อคประตูของห้อง S101	62
4.37	หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ในของการล็อคประตูของ StudentRoom	62

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.38 หน้าแสดงผลการลือคประตูของห้อง S101	63
4.39 หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction การจองของ TeacherRoom	64
4.40 หน้าแสดงผลการจัดเก็บข้อมูลการจองของห้อง T101	64
4.41 หน้าแสดงผลการจัดเก็บข้อมูลลำดับการจองของ Teacher01	65
4.42 หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction แก้ไขการจองของ TeacherRoom	66
4.43 หน้าแสดงผลการแก้ไขข้อมูลการจองของห้อง T101	66
4.44 หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction การยกเลิกการจอง TeacherRoom	67
4.45 หน้าแสดงผลการลบข้อมูลการจองของห้อง T101	67
4.46 หน้าแสดงผลการลบข้อมูลลำดับการจองของ Teacher01	68
4.47 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ในการปลดลือคประตูของ TeacherRoom	69
4.48 หน้าแสดงผลการปลดลือคประตูของห้อง T101	69
4.49 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ในการลือคประตูของ TeacherRoom	70
4.50 หน้าแสดงผลการลือคประตูของห้อง T101	70
4.51 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ในการจอง TSRoom ของ Students	71
4.52 หน้าแสดงผลการจัดเก็บข้อมูลการจองห้อง TS101 ของ Student02	72
4.53 หน้าแสดงผลการจัดเก็บข้อมูลลำดับการจองของ Student02	72
4.54 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction แก้ไขการจอง TSRoom ของ Students	73
4.55 หน้าแสดงผลการแก้ไขข้อมูลการจองห้อง TS101 ของ Student02	73

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.56 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ยกเลิกการจอง TSRoom ของ Students	74
4.57 หน้าแสดงผลการลบข้อมูลการจองห้อง TS101 ของ Student02	75
4.58 หน้าแสดงผลการลบข้อมูลลำดับการจองของ Student02	75
4.59 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ในการจอง TSRoom ของ Teachers	76
4.60 หน้าแสดงผลการจัดเก็บข้อมูลการจองห้อง TS101 ของ Teacher02	76
4.61 หน้าแสดงผลการจัดเก็บข้อมูลลำดับการจองของ Teacher02	77
4.62 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction แก่การจอง TSRoom ของ Teachers	78
4.63 หน้าแสดงผลการแก้ไขข้อมูลการจองห้อง TS101 ของ Teacher02	78
4.64 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ยกเลิกการจอง TSRoom ของ Teachers	79
4.65 หน้าแสดงผลการลบข้อมูลการจองห้อง TS101 ของ Teacher02	80
4.66 หน้าแสดงผลการลบข้อมูลลำดับการจองของ Teacher02	80
4.67 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ในการปลดล็อคประตูของ TSRoom	81
4.68 หน้าแสดงผลการปลดล็อคประตูของห้อง TS101	81
4.69 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ในการล็อคประตูของ TSRoom	82
4.70 หน้าแสดงผลการล็อคประตูของห้อง TS101	82
4.71 หน้าระบุข้อมูลสำหรับส่ง Transaction ในการจัดเก็บ ID นักศึกษาที่เข้าชั้นเรียน	83
4.72 หน้าแสดงผลการจัดเก็บ ID ของนักศึกษาที่เข้าร่วมชั้นเรียน	84
4.73 หน้าแสดงผลประวัติการทำงานทั้งหมดของระบบ	84

สารบัญรูป (ต่อ)

รูปที่	หน้า	
4.74	หน้า ID Registry สำหรับสร้างและปรับเปลี่ยนการใช้งาน ID ของ เครือข่าย	85
4.75	หน้าสำหรับเลือกการ์ดเครือข่ายธุรกิจในการใช้งานระบบ	85
4.76	หน้าแสดงผล Participants ของ Students ผ่านการ์ดเครือข่ายของ นักศึกษา	86
4.77	หน้าแสดงผล Participants ของ Admins ผ่านการ์ดเครือข่ายของ นักศึกษา	86
4.78	หน้าแสดงผล Participants ของ Teachers ผ่านการ์ดเครือข่ายของ นักศึกษา	87
4.79	หน้าแสดงผล Assets ของ StudentRoom ผ่านการ์ดเครือข่ายของ นักศึกษา	87
4.80	หน้าแสดงผล Assets ของ TSRoom ผ่านการ์ดเครือข่ายของนักศึกษา	88
4.81	หน้าแสดงผล Assets ของ TeacherRoom ผ่านการ์ดเครือข่ายของ นักศึกษา	88
4.82	หน้าแสดงผล Participants ของ Teachers ผ่านการ์ดเครือข่ายของ อาจารย์	89
4.83	หน้าแสดงผล Participants ของ Admins ผ่านการ์ดเครือข่ายของอาจารย์	89
4.84	หน้าแสดงผล Participants ของ Students ผ่านการ์ดเครือข่ายของ อาจารย์	90
4.85	หน้าแสดงผล Assets ของ StudentRoom ผ่านการ์ดเครือข่ายของ อาจารย์	90
4.86	หน้าแสดงผล Assets ของ TeacherRoom ผ่านการ์ดเครือข่ายของ อาจารย์	91
4.87	หน้าแสดงผล Assets ของ TSRoom ผ่านการ์ดเครือข่ายของอาจารย์	91
4.88	หน้าแสดงผล Assets ของ Classroom ผ่านการ์ดเครือข่ายของอาจารย์	92

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.89 การ login เข้าสู่ระบบ	92
4.90 การแสดงผลหน้า Admin Dashboard	93
4.91 การแสดงผลข้อมูลผู้ใช้งานในส่วนของ Admin Profile	94
4.92 การแก้ไขข้อมูลส่วนตัวผู้ใช้งานของ Admin Profile	94
4.93 การแสดงผลในส่วนของรายชื่อบัญชีผู้ใช้งานของ Student User	95
4.94 การแสดงผลในส่วนของรายชื่อบัญชีผู้ใช้งานของ Teacher User	96
4.95 การแสดงผลข้อมูลบัญชีผู้ใช้งานของ Student User	96
4.96 การแก้ไขข้อมูลของบัญชีผู้ใช้งานของ Student User	97
4.97 รายการห้องเรียนที่บัญชีผู้ใช้ Student User ได้ทำการจองไว้	97
4.98 การยกเลิกการจองห้องเรียนผ่าน Admin User	98
4.99 การปลดล็อคห้องเรียนผ่าน Admin User	98
4.100 การนับถอยหลังของการเปิดประตูในกรณีที่ห้องเรียนกำลังถูกใช้งาน	99
4.101 การแสดงผลการล็อคห้องเรียนอีกครั้งหลังผ่านไป 15 นาที	99
4.102 การ Add Token ให้กับ Student User ผ่าน Admin User	100
4.103 การกำหนดจำนวน token ที่ต้องการเพิ่มให้กับ Student User	100
4.104 การแสดงผลหลังทำการ Add Token ให้กับ Student User	101
4.105 การแสดงผลข้อมูลบัญชีผู้ใช้งานของ Teacher User	101
4.106 การแก้ไขข้อมูลบัญชีผู้ใช้งานของ Teacher User	102
4.107 การแสดงผลห้องเรียนสำหรับ Student Room	102
4.108 การแสดงผลห้องเรียนสำหรับ Teacher Room	103
4.109 การแสดงผลห้องเรียนสำหรับ Teacher and Student Room	103
4.110 การปิดการใช้งานห้องเรียน	104
4.111 การเปิดการใช้งานห้องเรียน	104
4.112 การเพิ่มห้องเรียน	104
4.113 ประวัติการใช้งานของ Admin User	105

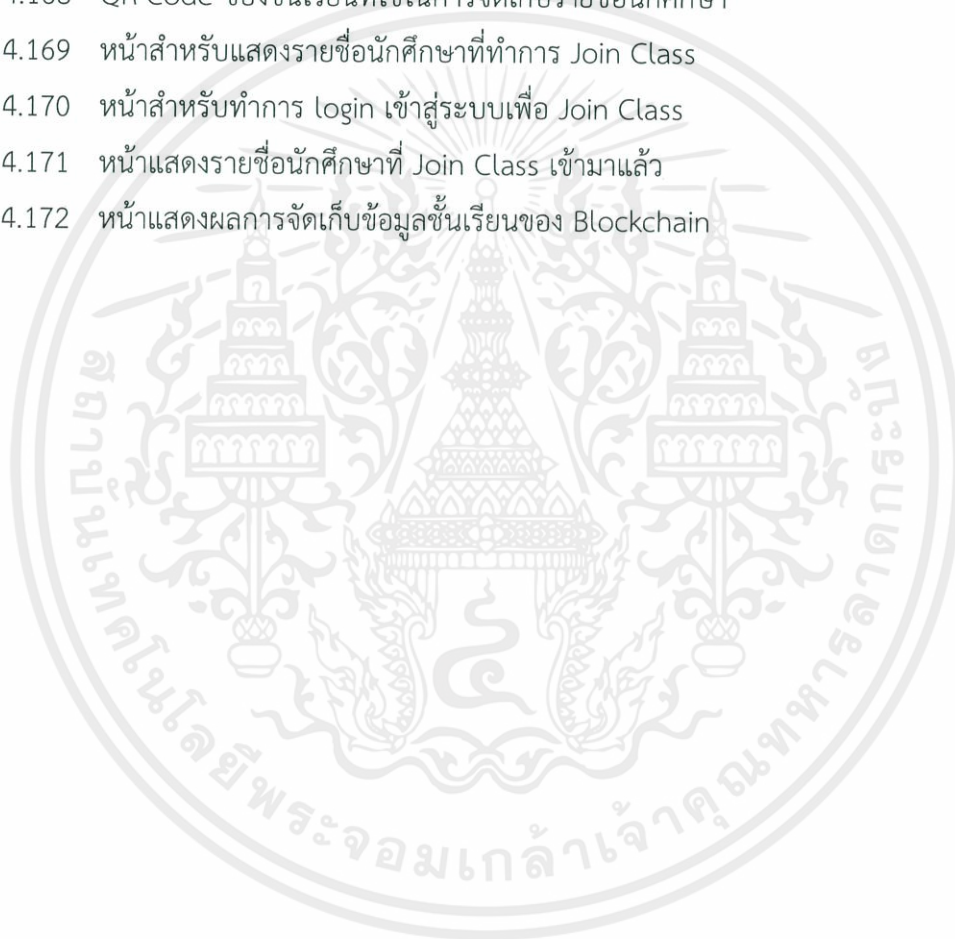
สารบัญรูป (ต่อ)

รูปที่	หน้า
4.114 การลงทะเบียนผู้ใช้งาน Student User	106
4.115 การแสดงผลหน้า Student Dashboard	107
4.116 ข้อมูลของห้องเรียนที่ได้ทำการจองไว้บน Student Dashboard	108
4.117 การยกเลิกการจองห้องเรียน	108
4.118 การแสดงผลหลังจากยกเลิกการจองห้องเรียนบน Student Dashboard	108
4.119 การแสดงผลห้องเรียนที่กำลังใช้งานบน Student Dashboard	109
4.120 ระบบทำการจับเวลา 15 นาที เมื่อทำการปลดล็อคประตู	109
4.121 ระบบทำการล็อคประตูหลังจากหมดเวลา 15 นาที	109
4.122 การแสดงผลข้อมูลผู้ใช้งานของ Student Profile	110
4.123 การแก้ไขข้อมูลผู้ใช้งานของ Student Profile	110
4.124 รายการห้องเรียนของ Student Room	111
4.125 รายการห้องเรียนของ Teacher and Student Room	111
4.126 ปฏิทินสำหรับเลือกวันที่ต้องการจองห้องเรียน Student Room	112
4.127 การเลือกช่วงเวลาที่ต้องการจองห้องเรียน Student Room	112
4.128 การยืนยันการจองห้องเรียน Student Room	112
4.129 การแสดงผลห้องเรียน Student Room ที่ได้จองไว้บนหน้า Dashboard	113
4.130 การยกเลิกการจองห้องเรียน Student Room	113
4.131 การแสดงผลหลังจากยกเลิกการจองห้องเรียน Student Room	113
4.132 การแสดงผลการเลือก QR Code ของชั้นเรียนที่ต้องการเข้าร่วม	114
4.133 การแสดงผลการเข้าร่วมชั้นเรียน	114
4.134 ประวัติการใช้งานของ Student User	114
4.135 การลงทะเบียนผู้ใช้งาน Teacher User	115
4.136 การแสดงผลหน้า Teacher Dashboard	116
4.137 ข้อมูลของห้องเรียนที่ได้ทำการจองไว้บน Teacher Dashboard	116
4.138 การยกเลิกการจองห้องเรียน	117

สารบัญรูป (ต่อ)

รูปที่	หน้า	
4.139	การแสดงผลหลังจากยกเลิกการจองห้องเรียนบน Teacher Dashboard	117
4.140	การแสดงผลห้องเรียนที่กำลังใช้งานบน Teacher Dashboard	117
4.141	ระบบทำการจับเวลา 15 นาที เมื่อทำการปลดล็อคประตู	118
4.142	ระบบทำการล็อคประตูหลังจากหมดเวลา 15 นาที	118
4.143	การแสดงผลข้อมูลผู้ใช้งานของ Teacher Profile	118
4.144	การแก้ไขข้อมูลผู้ใช้งานของ Teacher Profile	119
4.145	รายการห้องเรียนของ Teacher Room	120
4.146	ปฏิทินสำหรับเลือกวันที่ต้องการจองห้องเรียน Teacher Room	120
4.147	การเลือกช่วงเวลาที่ต้องการจองห้องเรียน Teacher Room	120
4.148	การยืนยันการจองห้องเรียน Teacher Room	121
4.149	การแสดงผลห้องเรียน Teacher Room ที่ได้จองไว้บนหน้า Dashboard	121
4.150	รายชื่อห้องเรียนของ Teacher and Student Room	122
4.151	การเลือกรายชื่อชั้นเรียนที่ต้องการ	122
4.152	การยืนยันการจองห้องเรียนแบบสร้างชั้นเรียน	122
4.153	ข้อมูลของชั้นเรียน	123
4.154	QR Code ที่ใช้ในการสแกนเพื่อ login เข้าชั้นเรียน	123
4.155	หน้าแสดงรายชื่อ Student User ที่จัดเก็บผ่าน QR Code	124
4.156	การแสดงผลรายชื่อชั้นเรียนที่ได้สร้างไว้	124
4.157	การเลือกรายชื่อชั้นเรียนเพื่อ Add Token	125
4.158	การเลือกรายชื่อ Student User จากชั้นเรียน	125
4.159	การ Add Token ให้กับ Student User	125
4.160	การแสดงผลหลังจากทำการ Add Token	126
4.161	ประวัติการใช้งานของ Teacher User	126
4.162	ผลการทดลองการทำงานของวงจรควบคุมกลอนแม่เหล็กไฟฟ้า	127

4.163	ผลการทดลองการทำงานวงจรควบคุมกลอนแม่เหล็กไฟฟ้าผ่านไมโครคอนโทรลเลอร์	128
4.164	ผลการทดลองการทำงานของปุ่มสวิตซ์ในการควบคุมไมโครคอนโทรลเลอร์	129
4.165	หน้าสำหรับกรอกข้อมูลในการ Create Class	130
4.166	หน้าแสดงผลชั้นเรียนที่ได้ทำการสร้างไว้	130
4.167	รายละเอียดของชั้นเรียน	131
4.168	QR Code ของชั้นเรียนที่ใช้ในการจัดเก็บรายชื่อนักศึกษา	131
4.169	หน้าสำหรับแสดงรายชื่อนักศึกษาที่ทำการ Join Class	132
4.170	หน้าสำหรับทำการ login เข้าสู่ระบบเพื่อ Join Class	132
4.171	หน้าแสดงรายชื่อนักศึกษาที่ Join Class เข้ามาแล้ว	133
4.172	หน้าแสดงผลการจัดเก็บข้อมูลชั้นเรียนของ Blockchain	133



ระบบการจัดการการใช้งานห้องเรียนโดยการใช้บล็อกเชน
CLASSROOM MANAGEMENT SYSTEM BY USING BLOCKCHAIN



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการจัดการการใช้งานห้องเรียนโดยการใช้บล็อกเชน
CLASSROOM MANAGEMENT SYSTEM BY USING BLOCKCHAIN

โดย

นายจิรภัทร กิจท่า	58010187
นายเจษฎา ธรรมสวัสดิ์	58010211
นายณัชพล อุบลศักดิ์	58010353

อาจารย์ที่ปรึกษา

รศ.ดร. ปราโมทย์ วาดเขียน

รศ.ดร. จีรสุดา โกษิยาภรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

เทคโนโลยี Blockchain เป็นเทคโนโลยีการจัดการข้อมูลที่กำลังได้รับความสนใจในปัจจุบันและมีแนวโน้มว่าจะถูกนำไปใช้งานเป็นจำนวนมากในอนาคต เนื่องจาก Blockchain นั้นมีลักษณะเด่นในเรื่องของการป้องกันการปลอมแปลงข้อมูลและสามารถจัดเก็บข้อมูลได้เป็นจำนวนมาก โดยเฉพาะเรื่องของการเก็บข้อมูลของ Blockchain นั้นมีความน่าเชื่อถือสูง เนื่องจากเป็นการเก็บข้อมูลแบบถาวรและการทำงานของ Blockchain ที่จำเป็นต้องใช้ข้อมูลที่มีความสอดคล้องกัน ซึ่งในปัจจุบัน Blockchain ถูกนำไปใช้ในระบบการธนาคาร ผู้จัดทำจึงต้องการศึกษาการทำงานของเทคโนโลยี Blockchain จึงได้นำเสนอปริญญาานิพนธ์ในหัวข้อเรื่อง ระบบการจองห้องเรียนโดยใช้เทคโนโลยี Blockchain ซึ่งนอกจากผู้จัดทำจะได้รับความรู้ในเรื่องของทฤษฎีเทคโนโลยี Blockchain แล้ว ระบบการจองห้องเรียนดังกล่าวก็จะเป็นระบบหนึ่งที่ช่วยอำนวยความสะดวกในการจองห้องเรียนให้แก่อาจารย์และนักศึกษาได้

1.2 วัตถุประสงค์

- 1) เพื่อศึกษาหลักการและทฤษฎีของเทคโนโลยี Blockchain
- 2) เพื่อศึกษาการสร้างระบบฐานข้อมูลโดยใช้ Hyperledger Fabric Blockchain
- 3) เพื่อศึกษาการสร้างเว็บไซต์โดยใช้โปรแกรม Laravel Framework
- 4) เพื่อศึกษาการจัดการระบบการจองห้องเรียนผ่านเว็บไซต์
- 5) เพื่อศึกษาการตรวจสอบการเข้าเรียนของนักศึกษาโดยใช้ QR Code
- 6) เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์ร่วมกับระบบฐานข้อมูลและกลอนแม่เหล็กไฟฟ้า

1.3 ขอบเขตของปริญญาานิพนธ์

- 1) สร้างระบบฐานข้อมูลโดยใช้ Hyperledger Fabric Blockchain
- 2) สร้างเว็บไซต์รองรับระบบการจองห้องเรียนด้วยโปรแกรม Laravel Framework
- 3) สร้างการตรวจสอบการเข้าเรียนของนักศึกษาโดยใช้ QR code
- 4) ไมโครคอนโทรลเลอร์สามารถเชื่อมต่อกับระบบฐานข้อมูลและสามารถควบคุม

การทำงานของกลอนแม่เหล็กไฟฟ้าได้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 Blockchain

Blockchain คือ รูปแบบการจัดเก็บข้อมูลแบบไม่รวมศูนย์ที่มีความน่าเชื่อถือและมีความปลอดภัยสูง โดยการกระจายข้อมูลไปยังทุกคนในเครือข่ายเพื่อให้ทุกคนมีข้อมูลชุดเดียวกัน เรียกว่า Distributed Ledger ทุกคนในเครือข่ายจะเชื่อมต่อถึงกันแบบ peer-to-peer สามารถอัปเดตข้อมูลถึงกันได้ ทำให้ข้อมูลของทุกคนเป็นข้อมูลชุดล่าสุดอยู่ตลอด และทุกคนจะได้รับ Private key เพื่อใช้ในการยืนยันตัวตนในการเข้าถึงและจัดการข้อมูลของตัวเอง และ Public key ใช้ระบุตัวตนเพื่อใช้ในการตรวจสอบของผู้อื่นๆภายในเครือข่าย ทุกคนในเครือข่ายจะมีส่วนร่วมในการตรวจสอบความถูกต้องของข้อมูล โดยการกำหนด Consensus ซึ่งเป็นข้อตกลงร่วมกันของทุกคนในเครือข่ายเพื่อใช้ในการยืนยันความถูกต้องของข้อมูล เช่น ระบบ Proof of Work ของ Bitcoin ถ้าหากข้อมูลถูกต้องจะเก็บข้อมูลลงใน block พร้อมกับระบบจะสุ่มค่า Hash ขึ้นมาค่าหนึ่ง และเก็บค่า Hash และค่า Hash ของ block ก่อนหน้าไปพร้อมกับข้อมูลด้วย เพื่อใช้ในการอ้างอิงตำแหน่งของ block จากนั้นจึงกระจาย block ไปยังทุกคนในเครือข่ายเพื่อนำไปต่อกับ block ก่อนหน้าโดยจะเรียงต่อกันไปเรื่อยๆ เรียกว่า chain เพราะทุก block ที่อยู่ใน chain นั้นมีการอ้างอิงถึง block ก่อนหน้า ทำให้ไม่สามารถแก้ไขข้อมูลในอดีตได้ การแก้ไขข้อมูลจะต้องทำใน block ถัดไป แต่ประวัติข้อมูลในอดีตจะยังคงอยู่ ระบบจึงมีความปลอดภัยต่อการแก้ไขและปลอมแปลงข้อมูล

ในด้านธุรกิจมีการนำเทคโนโลยี Blockchain ไปประยุกต์ใช้กับการทำธุรกรรมต่างๆ เช่น Bitcoin ที่เป็นที่ยู้งักกันมากที่สุด ทำให้ผู้ใช้สามารถทำธุรกรรมได้โดยไม่ต้องพึ่งพาการเชื่อถือบุคคลที่สามหรือคนกลาง ด้วยการใช้ Smart Contract ในการกำหนดเงื่อนไขและขั้นตอนการทำธุรกรรมโดยอัตโนมัติไว้ล่วงหน้า โดยที่จะต้องมีการตกลงกันของคู่สัญญาทั้งสองฝ่าย ก็จะสามารถทำธุรกรรมได้โดยตรงผ่าน Blockchain ทำให้องค์กรธุรกิจสามารถสร้างแอปพลิเคชันที่มีความน่าเชื่อถือ ตรวจสอบได้ และมีความโปร่งใสในการทำงานร่วมกัน Blockchain นั้นยังสามารถนำไปประยุกต์ใช้ได้กับงานด้านอื่นๆเช่นกัน ขึ้นอยู่กับการกำหนดเงื่อนไขและกลไกการทำงานของ Smart Contract

2.1.1 Distributed Ledger Technology

Distributed Ledger Technology เป็นเทคโนโลยีที่ทำการกระจายข้อมูลไปยังทุกคนในเครือข่าย ทำให้ทุกคนในเครือข่ายมีสำเนาข้อมูลที่เหมือนกัน และสามารถตรวจสอบข้อมูลได้โดยไม่ต้องมีคนกลางเหมือนกับระบบฐานข้อมูลแบบรวมศูนย์ (Centralized Database System) ที่จะเก็บข้อมูลทั้งหมดไว้ที่ศูนย์กลางเพียงอย่างเดียว Distributed Ledger มีวัตถุประสงค์เพื่อเพิ่มความน่าเชื่อถือของข้อมูล เพิ่มความปลอดภัยจากการปลอมแปลงแก้ไขข้อมูล เพิ่มความเร็วในการประมวลผล เพิ่มประสิทธิภาพในการทำงาน การทำงานมีความเป็นอิสระต่อกัน และสามารถขยายเครือข่ายได้ง่าย

2.1.2 Smart Contract

โดยปกติแล้วการทำสัญญาเงื่อนไขระหว่างสองกลุ่มนั้นจำเป็นต้องอาศัยตัวกลางอย่างสำนักงานกฎหมาย หรือการถ่ายโอนเงินก็ต้องทำผ่านธนาคาร ซึ่ง Smart Contract ก็คือสัญญาดิจิทัลที่จะทำให้ธุรกรรมระหว่างกลุ่มคน 2 กลุ่มขึ้นไปสามารถทำงานร่วมกันได้ด้วยความเชื่อใจโดยไม่ต้องอาศัยคนกลาง โดยการกำหนดเงื่อนไขและกลไกการทำธุรกรรมขึ้นมาพร้อมกัน

เมื่อผู้มีส่วนเกี่ยวข้องได้ร่วมกันร่าง Smart Contract ขึ้นมาและนำไปติดตั้งไว้กับ Blockchain ผู้ใช้จะสามารถทำธุรกรรมทุกอย่างที่ต้องการผ่าน Blockchain โดย Blockchain จะทำหน้าที่ดำเนินการทุกอย่างให้เป็นไปตามที่ตกลงกันใน Smart Contract และธุรกรรมทุกอย่างก็จะถูกบันทึกลงบน Blockchain แบบถาวร ทำให้เกิดความโปร่งใสและสร้างความเชื่อมั่นให้กับผู้มีส่วนเกี่ยวข้องทุกฝ่าย

2.1.3 Data Privacy

ในส่วนของความเป็นส่วนตัวของข้อมูลนั้น Blockchain จะทำการสร้าง Private Key ของแต่ละคนในเครือข่าย โดยจะสามารถใช้ Private key ของตัวเองในการยืนยันตัวตน เพื่อใช้งานการทำธุรกรรมต่างๆของตัวเองเท่านั้น และมี Public key ของแต่ละคนเพื่อใช้ในการตรวจสอบข้อมูลของบุคคลนั้นๆในเครือข่ายธุรกิจ และใช้ในการติดต่อธุรกรรมระหว่างกัน หากต้องการจะตรวจสอบข้อมูลทางธุรกรรมของคนอื่นในเครือข่ายธุรกิจ จำเป็นต้องรู้ Public key ของคนๆนั้น แต่ไม่สามารถดำเนินการทำธุรกรรมใดๆกับข้อมูลชุดนั้นได้หากไม่มี Private key ของข้อมูลชุดนั้น

2.1.4 Consensus Protocol

ในเครือข่ายที่เป็นระบบแบบรวมศูนย์นั้น เมื่อมีการทำธุรกรรมเกิดขึ้นจะมีคนกลางทำหน้าที่ตรวจสอบและยืนยันความถูกต้องของการทำธุรกรรม แต่ในระบบแบบกระจายศูนย์อย่าง Blockchain ระบบจะไม่มีคนกลางคอยยืนยันการทำธุรกรรมให้ ดังนั้นหน้าที่ในการตรวจสอบและยืนยันการทำธุรกรรมใดๆนั้นจึงเป็นของทุกคนในเครือข่าย โดยจะต้องมีการกำหนดกฎเกณฑ์ที่ใช้ในการยืนยันความถูกต้องของข้อมูลและทุกคนในเครือข่ายจะต้องยึดถือและปฏิบัติตามกันอย่างเคร่งครัด หากไม่ปฏิบัติตามก็จะถูกตัดออกจากเครือข่ายทันที

2.2 Hyperledger Fabric Blockchain

Hyperledger Fabric Blockchain นั้นคือซอฟต์แวร์ที่ถูกออกแบบมาให้เป็น Distributed Ledger ระหว่างองค์กรและแต่ละองค์กรก็มีความลับบางอย่างที่ไม่สามารถให้ผู้ที่ไม่เกี่ยวข้องรับรู้ หรืออาจเรียกได้ว่าเป็น Private Blockchain เช่น บริษัทขายส่งที่มีสัญญาของราคากับผู้ซื้อแต่ละคนไม่เหมือนกัน และผู้ซื้อแต่ละคนอาจไม่จำเป็นต้องเห็นปริมาณสินค้าทั้งหมดที่มี ส่วนผู้ตรวจสอบของรัฐบาลอาจต้องการตรวจสอบปริมาณของสินค้าที่มี แต่ไม่จำเป็นต้องเห็นราคา โดย Hyperledger Fabric จะใช้แนวคิดของ Channel ซึ่งเป็นช่องทางในการส่งข้อมูล ทุกคนที่อยู่ใน Channel เดียวกันจะเห็นข้อมูลและทำงานกับข้อมูลได้เหมือนกัน แต่คนที่อยู่นอก Channel ถึงแม้จะเข้าถึง Blockchain ได้แต่ก็ไม่สามารถถอดรหัสข้อมูลได้ นอกจากนี้ Fabric ยังสามารถเปลี่ยนแปลง Consensus และระบบ Login ของแต่ละองค์กรได้อย่างง่ายดาย

ข้อดีของ Fabric คือ การออกแบบให้มี Channel ซึ่งสามารถกำหนดการเห็นธุรกรรมที่เกิดขึ้นเฉพาะภายในกลุ่มสมาชิกที่ถูกระบุเท่านั้น ทำให้ข้อมูลมีความเป็นส่วนตัวสูงและยังรองรับการเขียนโปรแกรมเพื่อเรียกใช้ได้หลายภาษา นอกเหนือจากภาษา Go ที่เป็นภาษาแรกที่รองรับ โดยเฉพาะภาษาที่ได้รับความนิยมอย่าง Java และ JavaScript และยังมีเครื่องมือชื่อ Hyperledger Composer ที่เพิ่มความสะดวกในการกำหนดสมาชิกและบทบาทของแต่ละสมาชิกที่อยู่ภายในเครือข่าย ด้วยโครงสร้าง Model ของข้อมูล และฟังก์ชันการทำงานของ Smart Contract เพื่อช่วยลดการเขียนโปรแกรม นอกจากนี้ Fabric ยังสะดวกในการสร้างระบบการสมัครสมาชิกเข้ามายังเครือข่ายได้ผ่านทาง Membership Service Provider (MSP) ในร่องขอการยืนยันความ เป็นสมาชิกของเครือข่าย

2.2.1 Hyperledger Composer

Hyperledger Composer เป็นชุดเครื่องมือและซอฟต์แวร์ที่เปิดให้ใช้บริการโดยไม่เสียค่าใช้จ่าย และเป็นที่ยอมรับอย่างกว้างขวาง เพื่อช่วยให้การพัฒนาแอปพลิเคชันของ Blockchain ทำได้ง่ายยิ่งขึ้น โดยมีเป้าหมายหลักเพื่อลดระยะเวลาในการทำงาน และทำให้การใช้งาน Blockchain ร่วมกับระบบทางธุรกิจต่างๆ ทำได้ง่ายยิ่งขึ้น สามารถใช้ Composer ในการพัฒนาสำหรับการใช้งานในรูปแบบต่างๆ ได้อย่างรวดเร็ว และปรับใช้ผลการทำงานของ Blockchain ที่สร้างขึ้นได้ภายในไม่กี่สัปดาห์แทนที่จะใช้เวลาเป็นเดือนๆ Composer นั้นสามารถใช้ในการออกแบบเครือข่ายทางธุรกิจ (Business Networks) และสามารถใช้งานระบบต่างๆ ร่วมกับเทคโนโลยี Blockchain ได้

Hyperledger Composer รองรับโครงสร้างพื้นฐานและการทำงานของ Blockchain ที่มีอยู่ใน Hyperledger Fabric ซึ่งรองรับการใช้งาน Consensus Protocols เพื่อทำให้มั่นใจได้ว่าการทำธุรกรรมได้ผ่านการตรวจสอบตามเงื่อนไขและข้อตกลงที่กำหนดโดยสมาชิกภายในเครือข่ายธุรกิจ

2.2.1.1 Blockchain State Storage

ธุรกรรมทั้งหมดที่ส่งผ่านเครือข่ายทางธุรกิจจะถูกเก็บเป็นบัญชีแยกประเภท (Distributed Ledger) ไว้บน Blockchain และสถานะปัจจุบันของสินทรัพย์และสมาชิกจะถูกเก็บไว้ในฐานข้อมูลสถานะ (Blockchain State Storage) ซึ่ง Blockchain จะกระจายบัญชีแยกประเภทและฐานข้อมูลสถานะให้กับ peer ทั้งหมด และทำให้มั่นใจว่าการอัปเดตบัญชีและฐานข้อมูลนั้นสอดคล้องกันกับ peer ทั้งหมดโดยใช้ Consensus Algorithm

2.2.1.2 Connection Profiles

Hyperledger Composer ใช้ Connection Profiles ในการกำหนดระบบที่จะทำการเชื่อมต่อ ข้อมูลการเชื่อมต่อเป็นไฟล์ JSON และเป็นส่วนหนึ่งของการ์ดเครือข่ายธุรกิจ (Business Network card) ซึ่งข้อมูลเหล่านี้มักจะได้รับจากผู้สร้างระบบที่จะทำการเชื่อมต่อ และจะใช้ในการสร้างการ์ดเครือข่ายธุรกิจเพื่อให้สามารถเชื่อมต่อกับระบบได้

2.2.1.3 Assets

Assets นั้นจะเป็นสินทรัพย์ สินค้าหรือบริการที่มีตัวตนหรือไม่มีตัวตนก็ได้ ซึ่งถูกลงทะเบียนไว้ในเครือข่ายธุรกิจ Assets นั้นสามารถแสดงเกือบทุกอย่างในเครือข่ายธุรกิจ ตัวอย่างเช่น บ้านที่ต้องการขาย รายการขาย โฉนดที่ดิน และเอกสารการประกัน อาจเป็นสินทรัพย์ทั้งหมดในหนึ่งเครือข่ายธุรกิจ Assets จะต้องมีเอกลักษณ์เฉพาะในการระบุตัวตน นอกเหนือจาก

นั้นยังสามารถกำหนดคุณสมบัติได้ตามที่ต้องการ และ Assets อาจมีความเกี่ยวข้องกับ Assets หรือ Participants อื่นก็ได้

2.2.1.4 Participants

Participants คือ สมาชิกของเครือข่ายธุรกิจ อาจเป็นเจ้าของสินทรัพย์และเสนอการทำธุรกรรม ผู้ต้องการทำธุรกรรม หรือผู้เกี่ยวข้องกับการทำธุรกรรมก็ได้ ซึ่งจะต้องมีการกำหนดประเภทของ Participants และเช่นเดียวกับ Assets คือจะต้องมีการระบุตัวตน และสามารถกำหนดคุณสมบัติต่างๆได้ตามที่ต้องการ โดยหนึ่ง Participants สามารถมีได้มากกว่าหนึ่ง Identity

2.2.1.5 Transactions

Transactions เป็นกลไกที่ Participants ทำการโต้ตอบกับ Assets โดยสามารถกำหนดเงื่อนไขและการทำงานของ Transactions ให้แสดงผลได้ตามที่ต้องการ เช่น การที่สมาชิกทำการเสนอราคาประมูลสินทรัพย์ในการประมูล หรือผู้ประมูลทำการปิดการประมูลจะทำการโอนกรรมสิทธิ์ของสินทรัพย์ให้กับผู้ที่เสนอราคาสูงสุดโดยอัตโนมัติ

2.2.1.6 Identities

Identity คือ ไบรรับรองทางดิจิทัลและรหัสส่วนบุคคล (Private key) Identity จะใช้ในการทำธุรกรรมบนเครือข่ายธุรกิจ และต้องมีการจับคู่กับ Participants บนเครือข่าย โดย Identity นั้นจะถูกเก็บอยู่ใน Business Network card และถ้า Identity นั้นได้ทำการจับคู่กับ Participant ก็จะทำให้ผู้ใช้สามารถทำธุรกรรมบนเครือข่ายธุรกิจได้เหมือนกับ Participant นั้น

2.2.1.7 Business Network Cards

Business Network Cards เป็นการรวมกันของ Identity, Connection Profile และ Metadata ซึ่ง Metadata จะทำการเลือกชื่อของเครือข่ายธุรกิจที่จะเชื่อมต่อด้วย Business Network Cards ลดความซับซ้อนของขั้นตอนการเชื่อมต่อกับเครือข่ายธุรกิจ และขยายแนวความคิดเกี่ยวกับการกำหนดตัวตนภายนอกเครือข่ายธุรกิจไปสู่แนวคิดกระเป๋าตังค์ (Wallet) ที่เกี่ยวข้องกับเครือข่ายธุรกิจบางประเภท

2.2.1.8 Events

Events มีการกำหนดไว้ในเครือข่ายธุรกิจเช่นเดียวกับ Assets และ Participants เมื่อมีการกำหนด Events จะสามารถส่งข้อมูลโดยผ่านการประมวลผลของฟังก์ชัน

ทางธุรกรรม (Transaction Functions) เพื่อที่จะบอกกับระบบภายนอกว่า มีบางสิ่งที่สำคัญเกิดขึ้นกับบัญชี แอปพลิเคชันภายนอกสามารถเข้าถึง Events ผ่านทาง Composer-client API

2.2.1.9 Access Control

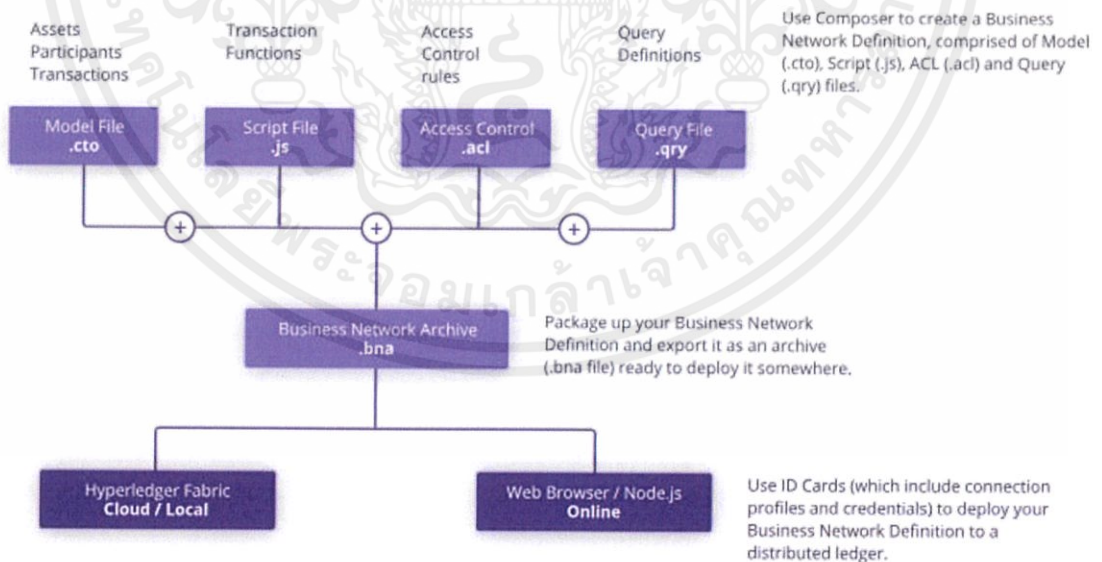
เครือข่ายทางธุรกิจอาจมีการกำหนดกฎควบคุมการเข้าถึงข้อมูล ช่วยให้สามารถควบคุมและกำหนดเงื่อนไขที่ Participants ใดจะสามารถเข้าถึง Assets ได้ได้ในเครือข่ายธุรกิจ

2.2.1.10 Queries

Queries ใช้ในการส่งคืนข้อมูลที่เกี่ยวข้องกับ Blockchain world-state Queries จะถูกกำหนดไว้ในเครือข่ายธุรกิจ รวมถึงตัวแปรสำหรับการปรับแต่งพื้นฐาน โดยการใช้ Queries จะทำให้ง่ายในการดึงข้อมูลออกมาจากเครือข่าย Blockchain โดย Queries จะทำงานผ่าน Hyperledger Composer API

2.2.1.11 Historian Registry

Historian เป็นการจดบันทึกที่เฉพาะเจาะจง ซึ่งบันทึกการทำธุรกรรมที่สำเร็จ รวมทั้ง Participants และ Identities ที่ทำธุรกรรมนั้น Historian ทำการจัดเก็บธุรกรรมในรูปแบบ Assets ของ HistorianRecord ที่ถูกกำหนดไว้ใน Namespace ของระบบ Hyperledger Composer



รูปที่ 2.1 โครงสร้างของ Hyperledger Composer

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน Hyperledger Composer ในการออกแบบเครือข่ายทางธุรกิจ มีโครงสร้างดังรูปที่ 2.1 ประกอบด้วย 4 ส่วน Model File คือ ส่วนที่กำหนดบทบาท ประเภทและรูปแบบการจัดเก็บข้อมูลหรือส่งข้อมูลของ Participants, Assets และ Transactions Script File คือ ส่วนที่ประกอบด้วยฟังก์ชันทางธุรกรรม ทำหน้าที่กำหนดเงื่อนไขและผลการทำงานของแต่ละ Transactions ซึ่งเปรียบเสมือนการกำหนด Smart Contract ให้กับ Blockchain Access Control คือ ส่วนที่กำหนดกฎควบคุมการเข้าถึงข้อมูลของแต่ละ Participants และ Query file คือ ส่วนที่ช่วยในการค้นหาและดึงข้อมูลที่เก็บไว้จาก Blockchain เมื่อรวมทั้ง 4 ส่วนไว้ด้วยกันจะได้เป็น Business Network Archive (.bna file) และจะสามารถนำไปใช้งานได้ทั้งกับ Hyperledger Fabric และผ่านเว็บเบราว์เซอร์

ตัวอย่างเครือข่ายทางธุรกิจที่ดำเนินการโดยนายหน้า สามารถออกแบบเครือข่ายธุรกิจได้ดังนี้

- Assets: บ้าน และรายการสินค้า
- Participants: ผู้ซื้อ และเจ้าของบ้าน
- Transactions: การซื้อหรือขายบ้าน และการสร้างและยกเลิกรายการสินค้า

Participants สามารถเข้าถึงการทำธุรกรรมได้จำกัดตามบทบาทของตนเองว่าเป็นผู้ซื้อ ผู้ขาย หรือนายหน้า นายหน้าสามารถสร้างแอปพลิเคชันสำหรับให้ผู้ใช้เลือกซื้อบ้านจากรายการบ้านที่เปิดขายและเสนอขายบ้าน เครือข่ายทางธุรกิจนี้ยังสามารถใช้งานร่วมกับระบบคลังสินค้าในการเพิ่มบ้านหลังใหม่เป็นไปยั้ง Assets และการนำบ้านที่ขายแล้วออกจากรายการสินค้า และกลุ่มคนอื่นที่เกี่ยวข้องสามารถลงทะเบียนเป็นสมาชิกได้ เช่น กรมที่ดินสามารถติดต่อกับผู้ซื้อเพื่อทำการโอนกรรมสิทธิ์ของที่ดินบริเวณนั้น

Hyperledger Composer เป็นรูปแบบการเขียนโปรแกรมที่ประกอบด้วยภาษาที่ใช้ ออกแบบ และชุด APIs เพื่อกำหนดและปรับใช้เครือข่ายธุรกิจและแอปพลิเคชันที่อนุญาตให้ Participants ทำการส่ง Transactions ในการแลกเปลี่ยน Assets ได้อย่างรวดเร็ว โดยสามารถใช้งาน Hyperledger Composer ได้ผ่าน UI เบรเวาร์เซอร์ที่ชื่อว่า Hyperledger Composer Playground โดย Playground นั้นใช้ได้ทั้งแบบไม่จำเป็นต้องติดตั้ง (Hosted version) หรือแบบทำการติดตั้ง เหมาะสำหรับการแก้ไขและทดสอบเครือข่ายธุรกิจแบบออฟไลน์

2.2.2 Hyperledger Composer Playground

Hyperledger Composer Playground เป็นแอปพลิเคชันเบราว์เซอร์ที่มีส่วนติดต่อผู้ใช้งาน (User Interface) สำหรับการกำหนดค่า การปรับใช้ และการทดสอบเครือข่ายธุรกิจ ใช้งานร่วมกันกับ Hyperledger Composer ในการออกแบบเครือข่ายธุรกิจ โดยสามารถทดสอบโค้ดที่ทำการออกแบบได้ผ่านทางหน้าแสดงผลของ Playground สามารถใช้งานได้แบบออนไลน์ผ่านเว็บเบราว์เซอร์ หรือแบบติดตั้งลงบนเครื่องเพื่อใช้งานแบบออฟไลน์ คุณสมบัติพิเศษของ Playground คือ ทำให้ผู้ใช้สามารถจัดการความปลอดภัยของเครือข่ายธุรกิจ เพิ่มสมาชิกใหม่เข้าสู่เครือข่ายธุรกิจ และเชื่อมต่อกับหลายเครือข่ายธุรกิจได้

2.3 Docker

เนื่องจาก Hyperledger Fabric มีการทำงานแยกเป็นหลายส่วน จึงจำเป็นต้องใช้งานหลายซอฟต์แวร์บนเครื่องเดียวกัน ดังนั้นจึงใช้ Docker เพื่อเข้ามาช่วยในการจัดการซอฟต์แวร์ของระบบ Docker คือ Software Container ตัวหนึ่ง ซึ่งมีการทำงานในลักษณะจำลองสภาพแวดล้อมเฉพาะขึ้นมาบนเครื่องเซิร์ฟเวอร์ ทำให้ซอฟต์แวร์ทำงานได้โดยไม่รบกวนกับซอฟต์แวร์ตัวอื่นที่อยู่บนระบบปฏิบัติการเดียวกัน Docker จะทำการติดตั้งโปรแกรมลงบน Container เพื่อนำไปใช้ในการ run Service ของระบบที่ต้องการ โดยจะทำการกำหนดค่าพื้นฐานต่างๆของซอฟต์แวร์ที่ใช้งานในแต่ละ Service ไว้ให้เรียบร้อยแล้ว Docker นั้นถูกพัฒนาขึ้นมาให้สามารถจัดการ Container ได้ง่าย มีขนาดเล็ก และใช้แนวคิด build, ship, run ที่ทำให้การสร้าง Container นั้นเร็วขึ้นมาก Docker นั้น เป็นที่นิยมกันอย่างแพร่หลาย เนื่องจากสามารถใช้งานได้ง่าย และสามารถตอบสนองความต้องการของผู้พัฒนาโปรแกรม (Developer) และผู้ดูแลระบบ (System Admin) ได้อย่างดี นอกจาก Docker ยังมี Software Container อื่นที่มีการใช้งานกัน เช่น LXC (Linux Container), Solaris Containers, OpenVZ แต่ไม่เป็นที่แพร่หลายมากนัก เนื่องจากมีความยุ่งยากในการใช้งาน

2.3.1 Docker Image

Docker Image เป็นเหมือนตัวต้นแบบของ Container ซึ่งภายในจะประกอบด้วยแอปพลิเคชันต่างๆ ที่มีการติดตั้งไว้เพื่อใช้งานสำหรับ Service นั้นๆ รวมทั้งมีการกำหนดการตั้งค่าที่จำเป็นต่างๆไว้เรียบร้อยแล้ว สามารถนำมาใช้งานได้โดย pull Docker Image จาก Registry และผู้ใช้งานยังสามารถสร้าง Docker Image สำหรับใช้งานเองได้อีกด้วย

2.3.2 Docker Container

Docker Container สามารถมองได้เสมือนกล่อง ซึ่งนำ Docker Image มาติดตั้ง เพื่อให้สามารถใช้งาน Service ที่ต้องการจาก Image นั้นๆได้ โดยในแต่ละ Container จะมีการใช้งาน RAM, CPU, Configuration files ต่างๆเป็นของตัวเอง และยังสามารถสั่ง start, stop Container นั้นๆได้อีกด้วย

2.3.3 Docker Registry

ผู้ใช้สามารถสร้าง Docker Image แล้วนำไปเก็บรวบรวมไว้บน Docker Registry โดย Docker Registry ในปัจจุบันก็มีให้เลือกใช้งานได้หลากหลายโดยมี Docker Hub เป็น Docker Registry หลักในการเรียกใช้ Docker Image และนอกจากนี้ยังมีผู้ให้บริการ Docker Registry รายอื่นๆด้วย เช่น Gitlab, Quay.io, Google Cloud เป็นต้น

2.3.4 Docker Compose

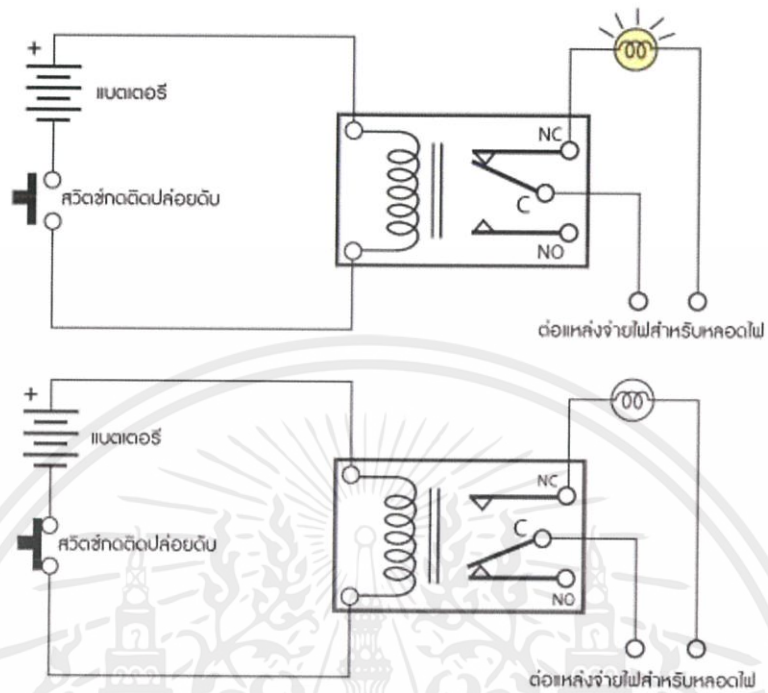
Docker Compose เป็นซอฟต์แวร์สำหรับอำนวยความสะดวกในการใช้งาน Container โดยทำให้ผู้ใช้สามารถสั่ง run หลาย Container ได้พร้อมๆกัน รวมไปถึงเชื่อมโยง Container ให้สามารถทำงานร่วมกันได้ Docker Compose นั้นสามารถ start, stop, restart, log, build Docker Container ได้โดยใช้คำสั่งบรรทัดเดียว และทำได้หลาย Container พร้อมๆกัน หรืออาจมองว่าเป็นหนึ่ง Service ได้เลย Docker Compose เป็นเครื่องมือที่สำคัญมากๆหากจะใช้ Docker ในการพัฒนาซอฟต์แวร์ เพราะเป็นการนำเอาข้อดีของ Docker มาทำให้สามารถทำงานได้เพียงเขียนคำสั่งไม่กี่คำสั่ง

2.4 Laravel PHP Framework

Laravel Framework คือ PHP Framework ที่เปิดให้สามารถใช้งานได้โดยไม่เสียค่าใช้จ่าย พัฒนาขึ้นโดย Taylor Otwell เพื่อใช้สำหรับพัฒนาเว็บแอปพลิเคชันในรูปแบบ MVC (Model Views Controller) ซึ่งรองรับการใช้งานในการออกแบบเว็บแอปพลิเคชันรวมถึงส่วนติดต่อผู้ใช้งาน (User Interface) Laravel ออกแบบมาเพื่อให้เราสามารถเขียนโค้ดได้อย่างเป็นระเบียบ สามารถอ่านและทำความเข้าใจได้ง่าย และ Source Code ของ Laravel ได้ถูกเก็บไว้บน Host ของ Github ภายใต้ลิขสิทธิ์ของ MIT ในปัจจุบันมีการนำมาใช้งานอย่างแพร่หลายและมีการพัฒนาอย่างต่อเนื่อง โดย Laravel PHP Framework ปัจจุบันอยู่ที่ Version 5.8

2.5 Relay

รีเลย์เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็กเพื่อใช้ในการเปลี่ยนสถานะของเอาต์พุต โดยการป้อนกระแสไฟฟ้าให้กับส่วนของขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งเราสามารถนำรีเลย์ไปประยุกต์ใช้ในการควบคุมวงจรต่าง ๆ ในงานอิเล็กทรอนิกส์ รีเลย์ประกอบด้วยส่วนสำคัญ 2 ส่วน คือ ส่วนของขดลวดเหนี่ยวนำกระแสตัว (Coil) ทำหน้าที่สร้างสนามแม่เหล็กไฟฟ้า ทำงานโดยการรับแรงดันไฟจากภายนอกต่อคร่อมที่ขดลวดเหนี่ยวนำ และส่วนของหน้าสัมผัส (Contact) ทำหน้าที่เหมือนสวิตช์จ่ายกระแสไฟให้กับอุปกรณ์ที่เราต้องการ โดยปกติเมื่อขดลวดไม่ได้รับแรงดันไฟแกนโลหะด้านในจะสัมผัสกับหน้าสัมผัสข้างหนึ่ง และเมื่อขดลวดได้รับแรงดันไฟตามค่าที่กำหนดในแต่ละรุ่น จะเกิดสนามแม่เหล็กไฟฟ้าดึงดูดให้แกนโลหะด้านในไปกระทบกับหน้าสัมผัสอีกข้างหนึ่ง ทำให้สามารถสลับการจ่ายเอาต์พุตระหว่างหน้าสัมผัสทั้งสองผ่านการจ่ายแรงดันไฟให้กับขดลวดเหนี่ยวนำ โดยจุดต่อทั้ง 3 จุดในรีเลย์ประกอบด้วย จุดต่อ NC ย่อมาจาก Normal Close หากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะติดกัน โดยทั่วไปมักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา จุดต่อ NO ย่อมาจาก Normal Open หากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะไม่ติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการควบคุมการเปิด-ปิด และจุดต่อ C ย่อมาจาก Common คือจุดร่วมที่ต่อมาจากแหล่งจ่ายไฟ มีการทำงานดังรูปที่ 2.2



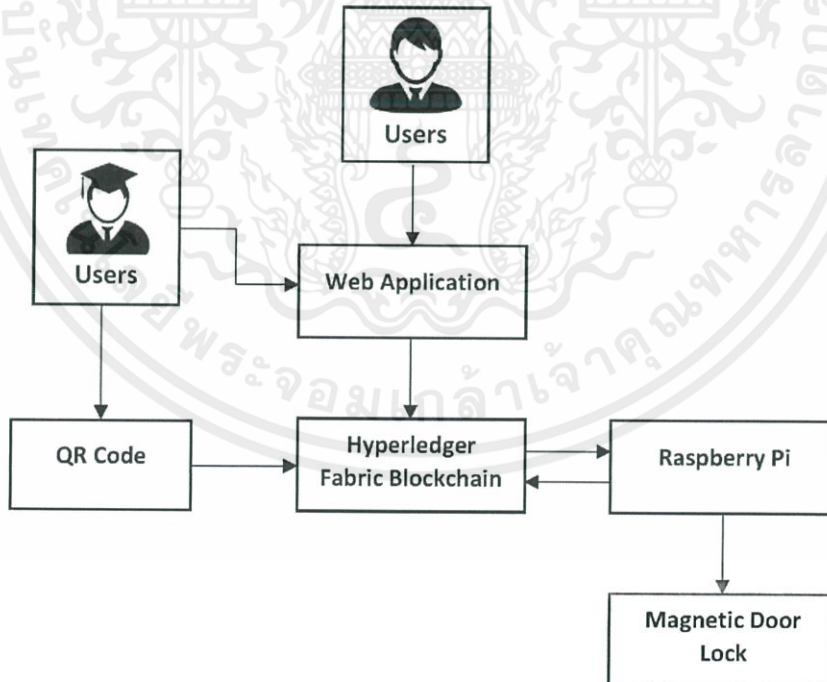
รูปที่ 2.2 การทำงานของรีเลย์ในการควบคุมการเปิด-ปิดอุปกรณ์อิเล็กทรอนิกส์

บทที่ 3

การออกแบบและการจัดทำปฏิญานิพนธ์

3.1 การออกแบบ

ปฏิญานิพนธ์นี้จะนำเสนอระบบการจัดการห้องเรียนโดยประยุกต์ใช้งานร่วมกับเทคโนโลยีการจัดเก็บข้อมูลของบล็อกเชน ซึ่งในบทนี้จะนำเสนอในส่วนของการออกแบบระบบ ภาพรวมการทำงานของระบบสามารถแสดงได้ดังรูปที่ 3.1 โดยการทำงานของระบบจะแบ่งออกเป็นสามส่วน ส่วนแรกคือส่วนของ Hyperledger Fabric Blockchain ซึ่งประกอบด้วย การติดตั้งเครื่องมือที่จำเป็น และการออกแบบระบบผ่านทาง Hyperledger Composer เพื่อใช้ในการจัดการห้องเรียน ส่วนที่สองคือส่วนติดต่อผู้ใช้ผ่านทางเว็บแอปพลิเคชัน ส่วนที่สามคือส่วนควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า โดยใช้การทำงานของวงจรรีเลย์และทรานซิสเตอร์ร่วมกับไมโครคอนโทรลเลอร์



รูปที่ 3.1 บล็อกไดอะแกรมแสดงภาพรวมการทำงานของระบบ

3.1.1 การออกแบบในส่วนของ Hyperledger Fabric Blockchain

การออกแบบระบบในส่วนแรกจะทำการติดตั้ง Hyperledger Fabric และเครื่องมือที่จำเป็นต้องใช้ในการทำงานของระบบ รวมถึงเครื่องมือที่ใช้ในการออกแบบและทดสอบระบบ โดยจะใช้ Hyperledger Composer ซึ่งเป็นซอฟต์แวร์ที่ช่วยในการออกแบบ Smart Contract ที่ใช้ในการกำหนดเงื่อนไขและการทำงานของระบบ และใช้ Composer Playground ในการทดสอบผลการทำงานของระบบที่ทำการออกแบบ และสามารถใช้งานร่วมกับเว็บแอปพลิเคชันผ่านทาง RESTful APIs ของ Composer REST Server และยังสามารถใช้ APIs ในการควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้าร่วมกับไมโครคอนโทรลเลอร์

3.1.1.1 การติดตั้งเครื่องมือและซอฟต์แวร์ที่ใช้ในการทำงาน

ขั้นตอนที่ 1 ติดตั้ง Command-line Interface (CLI) บางเครื่องมือที่อยู่ใน CLI มีประโยชน์ในการใช้พัฒนา Composer เครื่องมือที่สำคัญที่สุดอย่างหนึ่ง คือ Composer CLI ซึ่งประกอบด้วยสิ่งที่จำเป็นต้องใช้ในการดำเนินการทั้งหมดจึงติดตั้งเป็นอันดับแรก ติดตั้ง Hyperledger Composer Generator ซึ่งมีประโยชน์อย่างมากในการสร้างแอปพลิเคชัน และติดตั้ง Yeoman เครื่องมือสำหรับสร้างแอปพลิเคชันที่จะใช้งานร่วมกับ Hyperledger Composer Generator สุดท้ายติดตั้ง Composer REST Server ใช้ในการใช้งาน REST Server บนเครื่องเพื่อแสดงเครือข่ายธุรกิจเป็น RESTful APIs

```
npm install -g composer-cli@0.19
```

```
npm install -g generator-hyperledger-composer@0.19
```

```
npm install -g yo
```

```
npm install -g composer-rest-server@0.19
```

ขั้นตอนที่ 2 ติดตั้ง Composer Playground สามารถติดตั้งแอปพลิเคชัน เบราวเซอร์ Playground ให้ทำงานบนเครื่องได้ เพื่อใช้ในการแก้ไขและทดสอบผลการทำงานของเครือข่ายธุรกิจ

```
npm install -g composer-playground@0.19
```

ขั้นตอนที่ 3 ติดตั้ง Integrate Development Environment (IDE) ถึงแม้ว่าแอปพลิเคชันเบราวเซอร์จะสามารถใช้งานได้กับโค้ดของเครือข่ายธุรกิจ แต่จะมีความสะดวกมากกว่าหากทำงานผ่าน IDE ซึ่งในที่นี้จะใช้ Visual Studio Code โดยติดตั้ง VSCode จาก URL <https://code.visualstudio.com/download> แล้วเปิด VSCode ไปที่ส่วนขยาย ค้นหาและติดตั้งส่วนขยายของ Hyperledger Composer จาก Marketplace

3.1.1.2 การติดตั้ง Hyperledger Fabric

ขั้นตอนที่ 1 เข้าไปที่พื้นที่ที่ต้องการติดตั้ง ~/fabric-dev-servers โหลดไฟล์ .tar.gz ที่จะประกอบด้วยเครื่องมือที่จำเป็นสำหรับติดตั้ง Hyperledger Fabric คำสั่ง curl จะเป็นการโหลดไฟล์จาก URL แล้วนำมาแตกไฟล์ด้วยคำสั่ง tar -xvf หากไฟล์ที่โหลดเป็นไฟล์ .zip ก็ใช้คำสั่ง unzip ในการแตกไฟล์แทน

```
mkdir ~/fabric-dev-servers && cd ~/fabric-dev-servers
curl -O https://raw.githubusercontent.com/hyperledger/composer-tools/master/packages/fabric-dev-servers/fabric-dev-servers.tar.gz
tar -xvf fabric-dev-servers.tar.gz
```

ขั้นตอนที่ 2 โหลด Image ของ Docker ของ Hyperledger Fabric v1.1 runtime สามารถดู Image ของ Docker ได้จาก Docker Hub หรือใช้คำสั่ง docker images

```
cd ~/fabric-dev-servers
export FABRIC_VERSION=hlfv11
./downloadFabric.sh
```

ขั้นตอนที่ 3 การเริ่มการใช้งาน Hyperledger Fabric เมื่อเริ่มต้นการทำงานครั้งใหม่จำเป็นต้องเริ่มการใช้งาน Fabric (start Docker) และสร้างการ์ด PeerAdmin เป็นผู้ดูแลของระบบไว้จัดการภายในเครือข่าย สามารถหยุดการทำงานของ Fabric โดยใช้คำสั่ง ~/fabric-dev-servers/stopFabric.sh หรือหยุดการทำงานของ Image โดยใช้คำสั่ง ~/fabric-

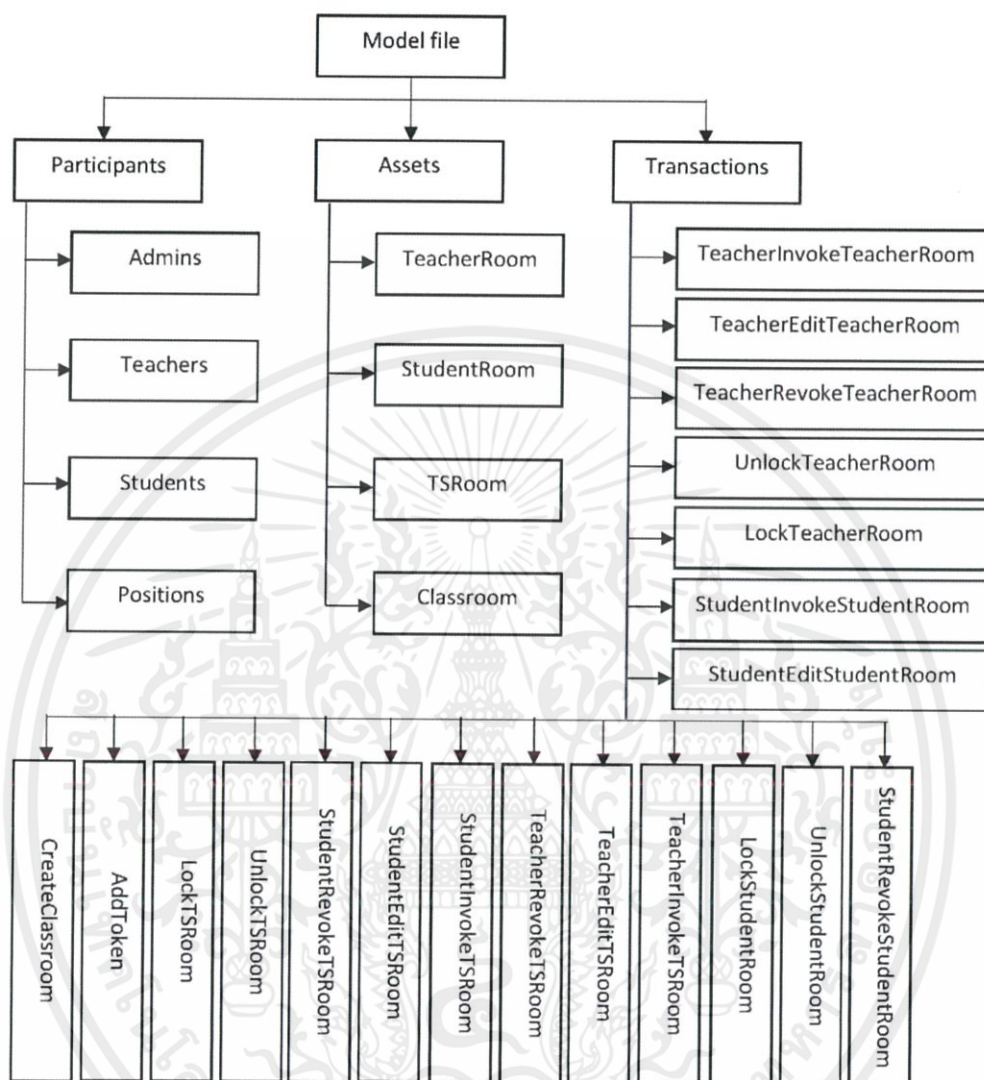
dev-servers/teardownFabric.sh ถ้าหยุดการทำงานของ Image ครั้งต่อไปที่เริ่มการทำงาน จึงจำเป็นต้องสร้างการ์ด PeerAdmin ใหม่เหมือนครั้งแรก

```
cd ~/fabric-dev-servers
export FABRIC_VERSION=hlfv11
./startFabric.sh
./createPeerAdminCard.sh
```

3.1.1.3 การออกแบบการทำงานของระบบผ่าน Hyperledger Composer การออกแบบ Hyperledger Composer จะประกอบด้วย 4 ส่วน คือ Model File, Script File, Access Control และ Query File ในส่วนของ Model File จะประกอบด้วย Participants, Assets และ Transactions ดังแสดงในรูปที่ 3.2 โดย Participants จะเป็นส่วนของผู้ใช้แบ่งออกเป็น Admins จะกำหนดรูปแบบการจัดเก็บข้อมูลบัญชีผู้ใช้ของผู้ดูแลระบบ Teachers จะกำหนดรูปแบบการจัดเก็บข้อมูลบัญชีผู้ใช้ของอาจารย์ Students จะกำหนดรูปแบบการจัดเก็บข้อมูลบัญชีผู้ใช้ของนักเรียน นักศึกษา และ Positions จะทำการกำหนดบทบาทของผู้ใช้ โดย Participants อื่นจะทำการจัดเก็บข้อมูลบทบาทโดยอ้างอิงจาก Participants ของ Positions ว่าเป็นผู้ดูแลระบบ อาจารย์ หรือ นักศึกษา

ส่วนของ Assets เป็นส่วนของห้องเรียนแบ่งเป็น TeacherRoom จะกำหนดรูปแบบการเก็บข้อมูลของห้องที่สามารถใช้งานได้เฉพาะอาจารย์ เช่น ห้องประชุม เป็นต้น StudentRoom จะกำหนดรูปแบบการจัดเก็บข้อมูลของห้องที่สามารถใช้งานได้เฉพาะนักศึกษา เช่น ห้องกิจกรรม TSRoom จะกำหนดรูปแบบการจัดเก็บข้อมูลของห้องที่สามารถใช้งานได้ทั้งอาจารย์และนักเรียน และ Classroom จะใช้สำหรับให้อาจารย์สามารถสร้างรายชื่อนักศึกษาในแต่ละรายวิชาเพื่อใช้ในการตรวจสอบการเข้าเรียน

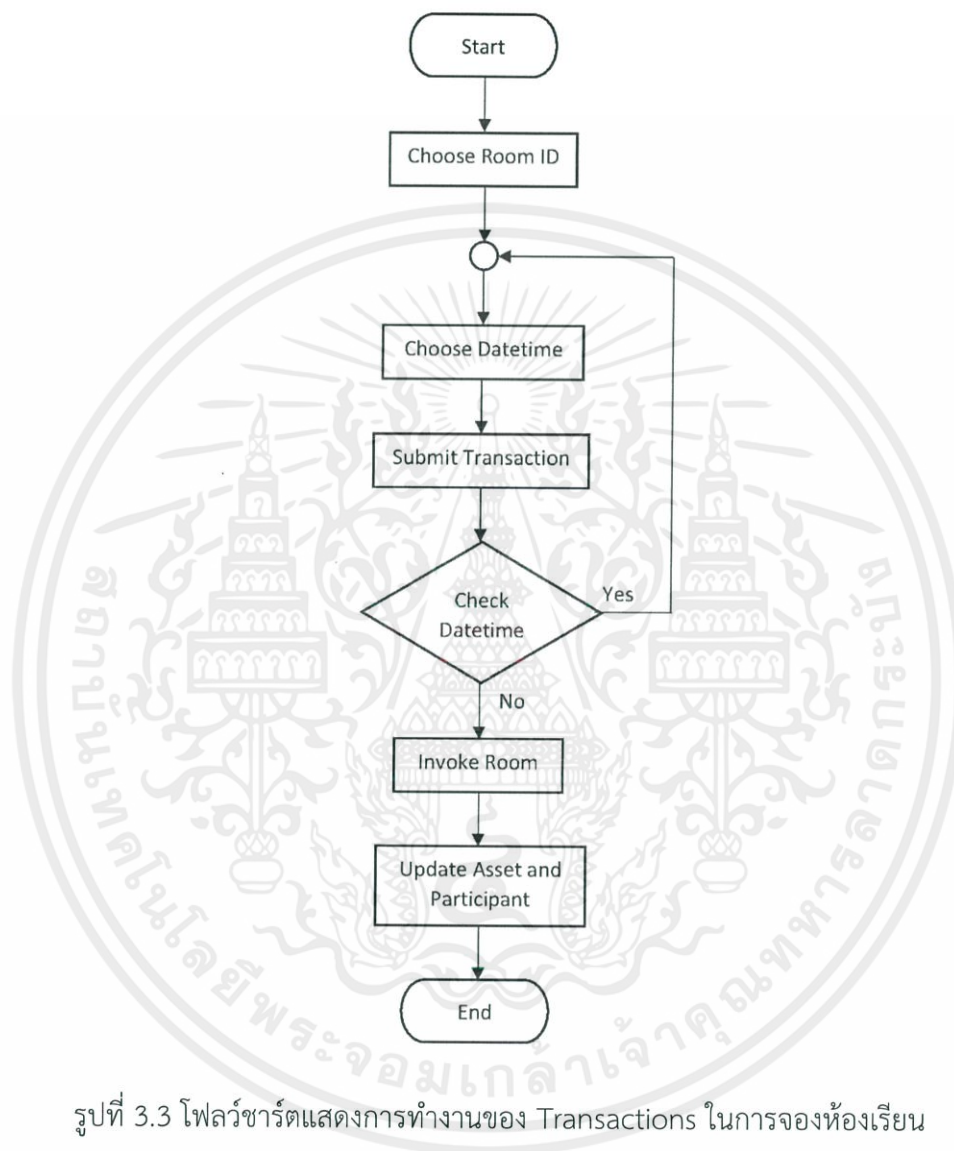
ส่วนของ Transactions จะเป็นส่วนของการดำเนินการทั้งหมดที่ Participants สามารถใช้งานได้ ซึ่งจะประกอบด้วย การจอง การยกเลิกการจอง การแก้ไขการจอง การเพิ่มเหรียญสำหรับใช้ในการจอง การปลดล็อคห้องและการล็อคห้อง โดยจะทำการกำหนดข้อมูลที่จำเป็นจะต้องใช้ในแต่ละการดำเนินการ และจะต้องสอดคล้องกับการออกแบบ Script file



รูปที่ 3.2 บล็อกไดอะแกรมแสดงการออกแบบในส่วนของ Model File

ในส่วนของ Script File จะเป็นส่วนของการกำหนดเงื่อนไขและการทำงานของ Transactions โดยจะต้องนำข้อมูลที่ส่งมากับ Transactions ไปใช้ในการตรวจสอบเงื่อนไขและการแสดงผลการจัดเก็บข้อมูลของ Assets โดยแบ่งเป็นการทำงานของ Transactions ในการจองห้องเรียนแสดงดังรูปที่ 3.3 ระบบจะทำการรับค่า ID ของห้องและวันที่ต้องการทำการจอง จากนั้นทำการส่ง Transaction เพื่อทำการจอง ระบบจะทำการตรวจสอบเงื่อนไขการทำงานและ

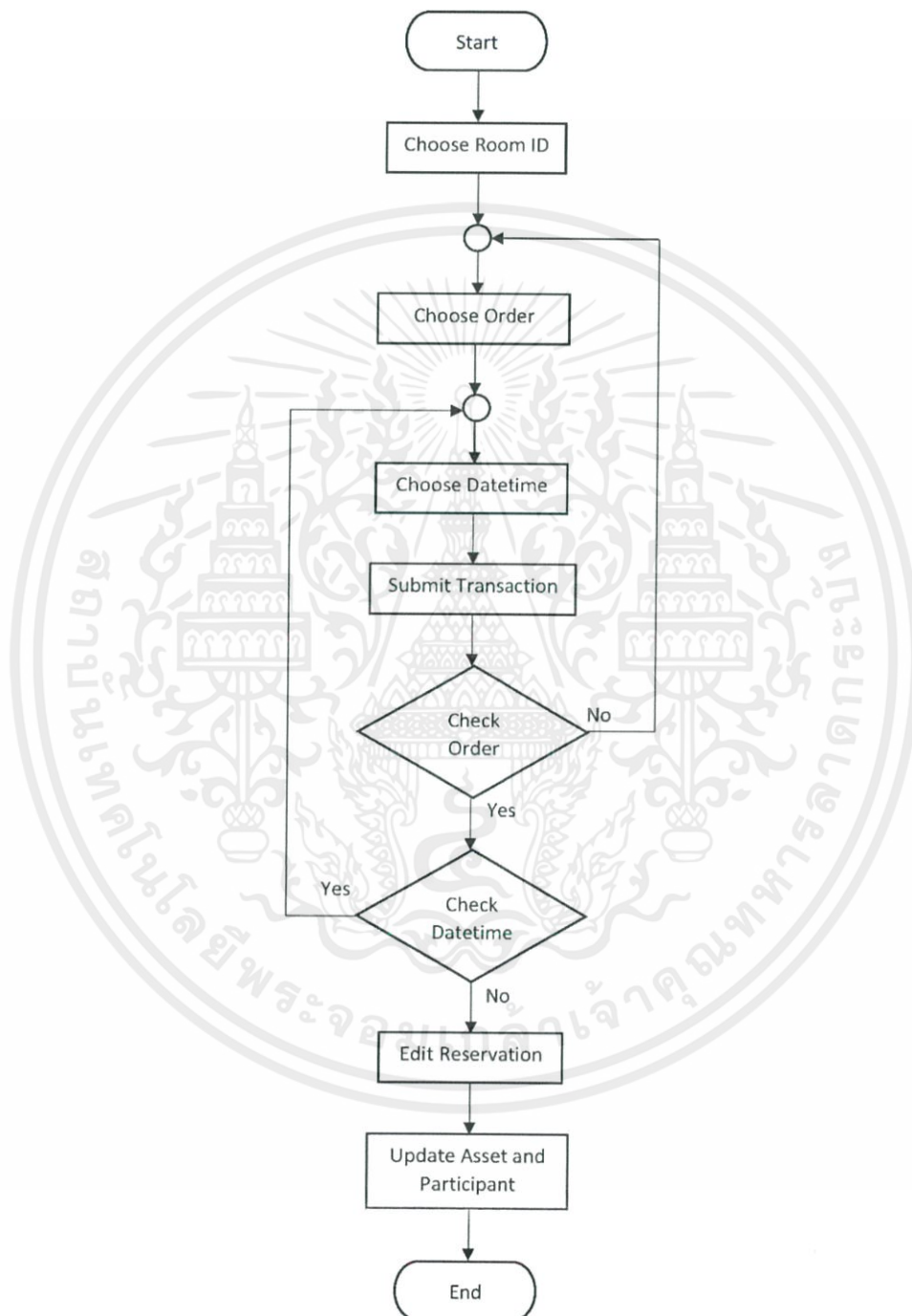
วันเวลาที่ต้องการจอง หากในระบบไม่มีข้อมูลการจองของวันและเวลาดังกล่าว ระบบจะทำการจองห้องและจัดเก็บข้อมูลการจองลงไปยัง Asset ของห้อง และ Participant ของผู้ใช้



รูปที่ 3.3 โฟลว์ชาร์ตแสดงการทำงานของ Transactions ในการจองห้องเรียน

การทำงานของ Transactions ในการแก้ไขการจองแสดงดังรูปที่ 3.4 ระบบจะรับค่า ID และลำดับการจองของห้อง พร้อมทั้งระบุวันเวลาที่ต้องการเลื่อนการจอง จากนั้นทำการส่ง Transactions เพื่อทำการแก้ไขการจอง ระบบจะทำการตรวจสอบลำดับการจอง และวันเวลาที่ต้องการจอง หากลำดับการจองถูกต้องและไม่มีการจัดเก็บข้อมูลการจองในวันเวลาดังกล่าว

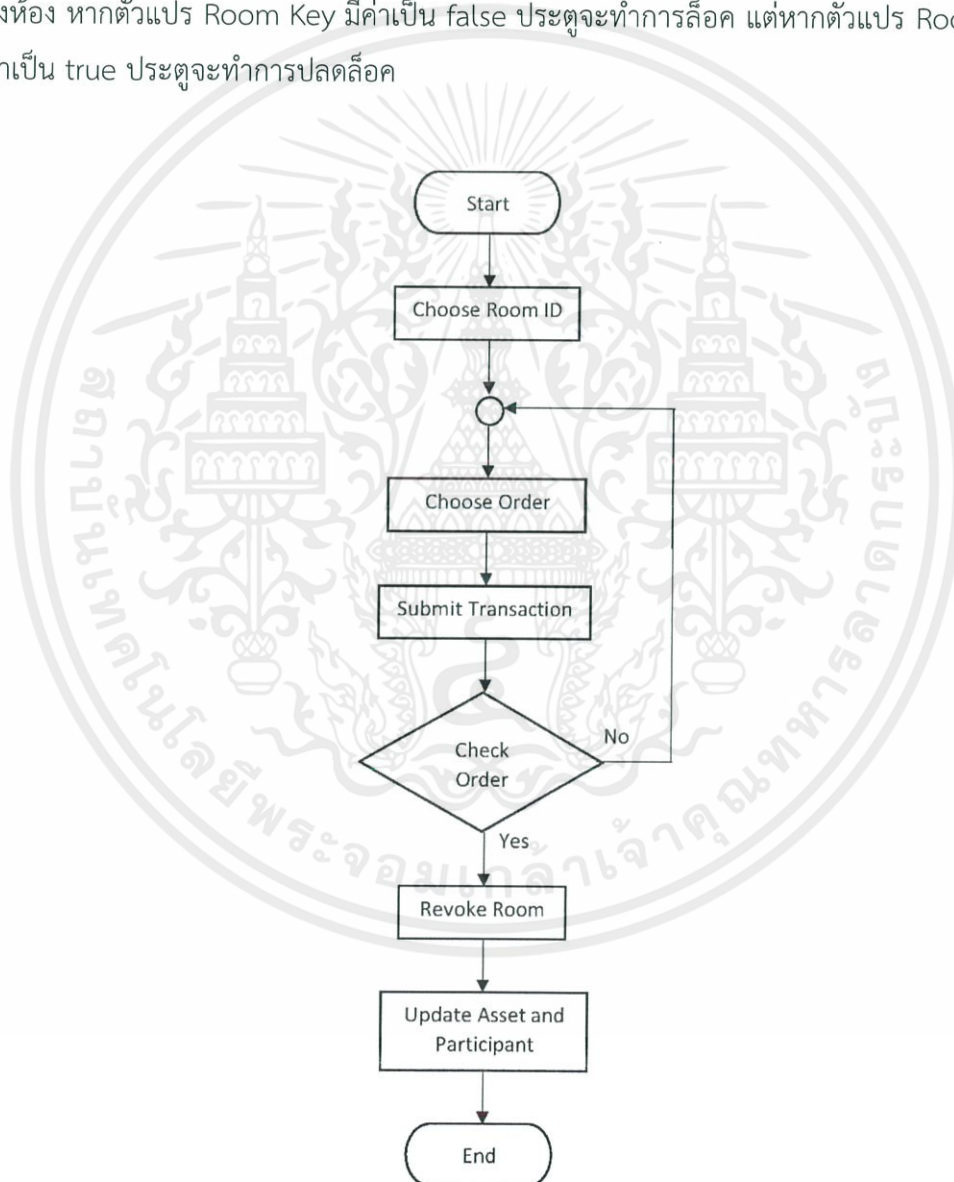
ระบบจะทำการแก้ไขข้อมูลวันเวลาที่ทำการจอง และทำการอัปเดตข้อมูลไปยัง Asset ของห้อง และ Participant ของผู้ใช้



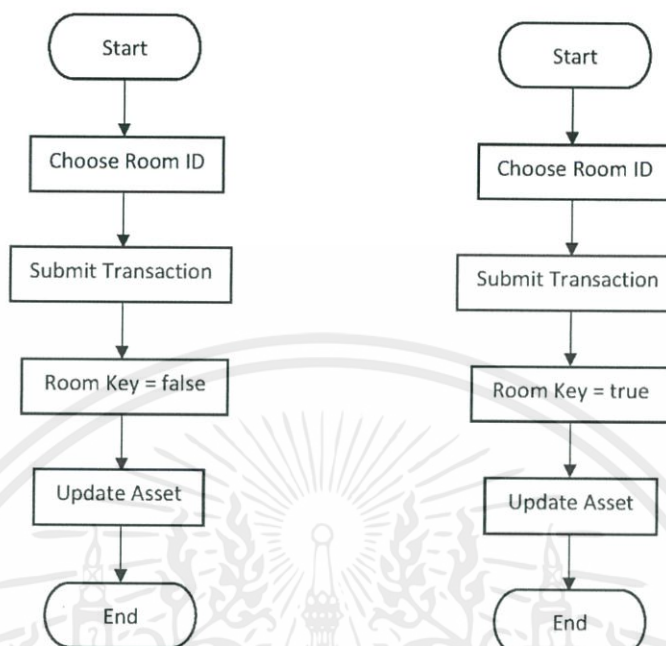
รูปที่ 3.4 โฟลว์ชาร์ตแสดงการทำงานของ Transactions ในการแก้ไขการจอง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Transactions ในการยกเลิกการจองแสดงดังรูปที่ 3.5 ระบบจะทำการรับค่า ID และลำดับการจองของห้อง จากนั้นทำการส่ง Transaction เพื่อยกเลิกการจอง ระบบจะทำการตรวจสอบลำดับการจอง หากลำดับการจองถูกต้อง ระบบจะทำการยกเลิกการจองและลบข้อมูลการจองออกจาก Asset ของห้องและ Participant ของผู้ใช้ รวมถึงการทำงานของ Transactions ในการล๊อคประตูและปลดล๊อคประตูดังรูปที่ 3.6 ระบบจะทำการรับค่า ID ของห้อง และทำการส่ง Transaction เพื่อเปลี่ยนแปลงค่าตัวแปร Room Key ที่อยู่ใน Asset ของห้อง หากตัวแปร Room Key มีค่าเป็น false ประตูจะทำการล๊อค แต่หากตัวแปร Room Key มีค่าเป็น true ประตูจะทำการปลดล๊อค



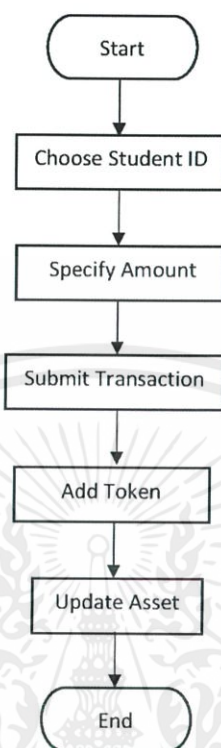
รูปที่ 3.5 โฟลว์ชาร์ตแสดงการทำงานของ Transactions ในการยกเลิกการจอง



รูปที่ 3.6 โฟลว์ชาร์ตแสดงการทำงานของ Transactions ในการล็อกและปลดล็อกประตู

Transaction ในการเพิ่มเหรียญ token ที่ใช้ในการจองมีการทำงานดังรูปที่ 3.7 ระบบจะรับค่าข้อมูล ID ผู้ใช้ของ Students และจำนวนของ token ที่ต้องการเพิ่ม จากนั้นทำการส่ง Transaction ระบบจะทำการเพิ่ม token ให้กับ Participant ของ Students

ในส่วนของ Access Control จะเป็นการกำหนดสิทธิ์การเข้าถึงข้อมูลของแต่ละ Positions ซึ่งจะทำให้แต่ละ Participants สามารถเข้าถึงข้อมูลได้ไม่เหมือนกัน โดยที่ผู้ดูแลระบบจะสามารถเข้าถึงข้อมูลได้ทั้งหมด และในส่วนของ Query File จะเป็นส่วนที่ช่วยในการค้นหาและดึงข้อมูลจากระบบได้ง่ายขึ้น

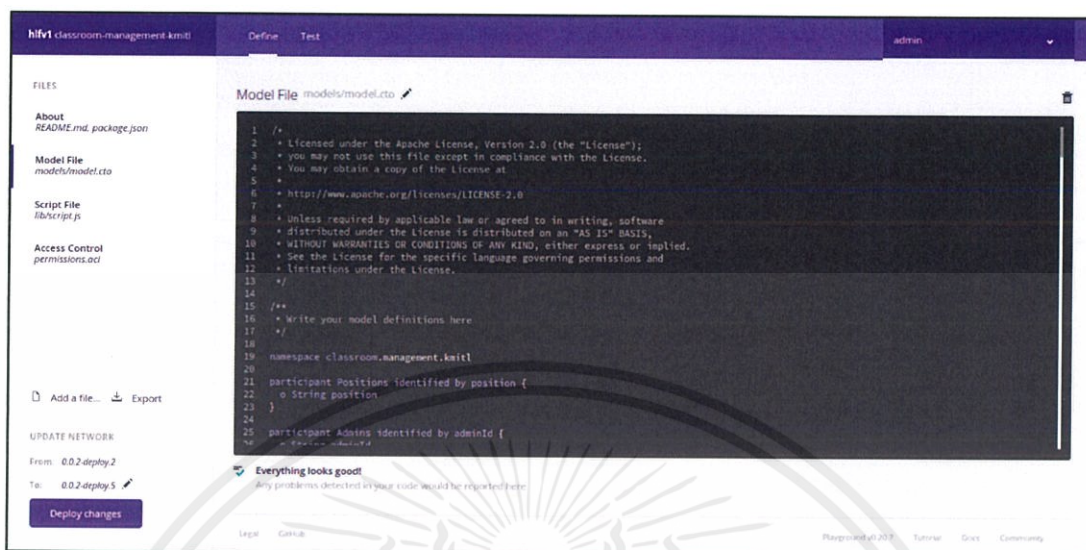


รูปที่ 3.7 โฟลว์ชาร์ตแสดงการทำงานของ Transactions ในการเพิ่ม token

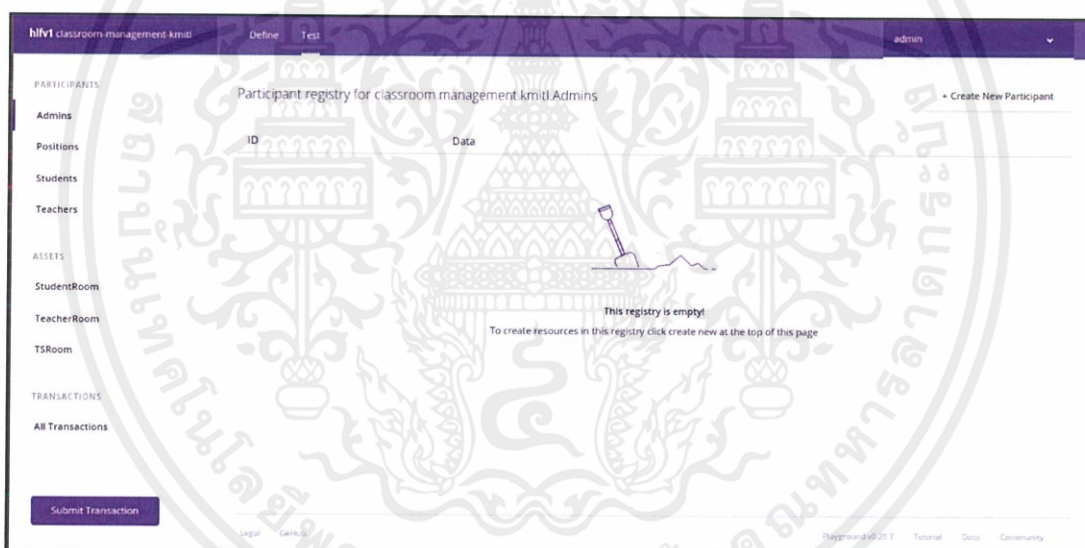
3.1.1.4 การใช้งานแอปพลิเคชันเว็บไซต์ของ Playground

การใช้งานแอปพลิเคชัน Playground บนเว็บเบราว์เซอร์เพื่อทดสอบผลการทำงานของเครือข่ายธุรกิจ ควบคู่ไปกับการออกแบบการทำงานของระบบ โดยสามารถนำโค้ดที่ทำการออกแบบในแต่ละส่วนมาทำการปรับใช้กับระบบ ดังรูปที่ 3.8 และสามารถทดสอบการทำงานของระบบที่ได้ทำการออกแบบผ่านทางหน้าแสดงผลของ Playground ดังรูปที่ 3.9

composer-playground



รูปที่ 3.8 หน้าแสดงผลในการปรับใช้เครือข่ายธุรกิจของ Composer Playground



รูปที่ 3.9 หน้าแสดงผลในการทดสอบเครือข่ายธุรกิจของ Composer Playground

3.1.1.5 การติดตั้ง Hyperledger Composer ลงบน Fabric

ขั้นตอนที่ 1 สร้างไฟล์เก็บข้อมูลเครือข่ายธุรกิจ เมื่อทำการกำหนดเครือข่ายธุรกิจแล้ว จะต้องจัดเก็บไว้ในไฟล์ที่เก็บข้อมูลเครือข่ายธุรกิจ (Business Network Archive (.bna)) โดยจะทำการสร้างไฟล์ classroom-management-kmitl@0.0.1.bna

```
composer archive create -t dir -n .
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 การปรับใช้เครือข่ายธุรกิจ หลังจากทำการสร้างไฟล์ .bna เครือข่ายธุรกิจสามารถปรับใช้ได้กับ Hyperledger Fabric โดยปกติจำเป็นต้องมีข้อมูลจากผู้ดูแล Fabric เพื่อใช้ในการสร้าง Identity ของ PeerAdmin ให้มีสิทธิพิเศษในการติดตั้ง chaincode ไปยัง peer รวมถึงการเริ่มการทำงาน chaincode บน Composer

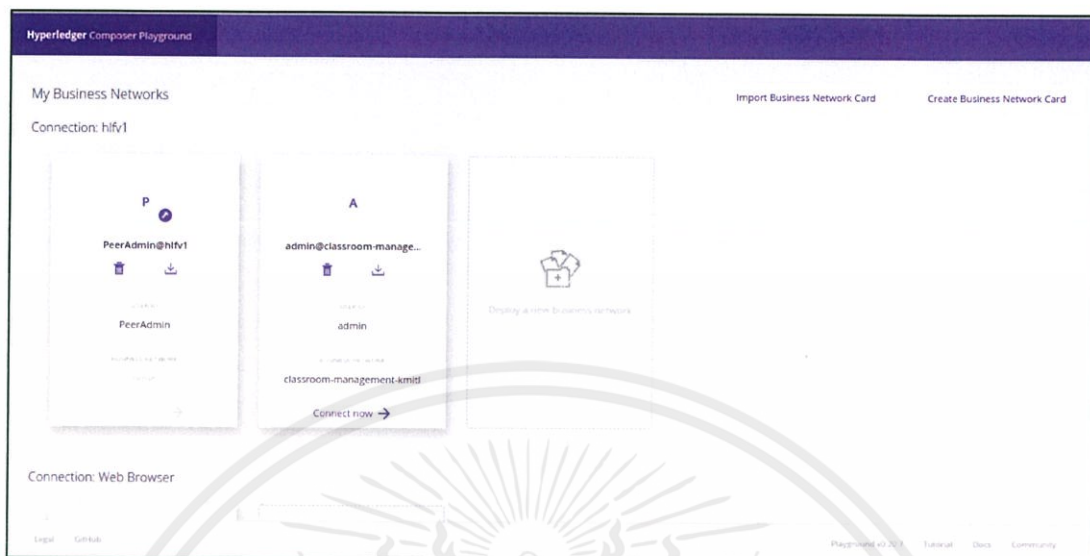
การปรับใช้ระบบร่วมกับเครือข่ายธุรกิจของ Hyperledger Fabric จำเป็นต้องมีการติดตั้ง Hyperledger Composer ลงบน peer เครือข่ายธุรกิจจึงจะเริ่มทำงานได้ และจะต้องสร้าง Participant, Identity และการ์ด ขึ้นมาใหม่เพื่อเป็นผู้ดูแลเครือข่ายธุรกิจ โดยการติดตั้งเครือข่ายธุรกิจจะต้องใช้ไฟล์ .bna ที่ได้สร้างไว้ ซึ่งทางผู้จัดทำใช้ classroom-management-kmitl@0.0.1.bna และเริ่มการทำงานของเครือข่ายธุรกิจ โดยกำหนดชื่อเครือข่าย และข้อมูลที่จำเป็น จากนั้นจึงทำการนำการ์ดผู้ดูแลเครือข่ายเข้าสู่ระบบ จะได้การ์ด admin@classroom-management-kmitl เพื่อใช้ในการจัดการระบบ และแสดงบนเครือข่ายผ่าน Composer Playground ดังรูปที่ 3.10

```
composer network install --card PeerAdmin@hlfv1 --archiveFile tutorial-
network@0.0.1.bna
```

```
composer network start --networkName tutorial-network --networkVersion 0.0.1 --
networkAdmin admin --networkAdminEnrollSecret adminpw --card PeerAdmin@hlfv1 --
file networkadmin.card
```

```
composer card import --file networkadmin.card
```

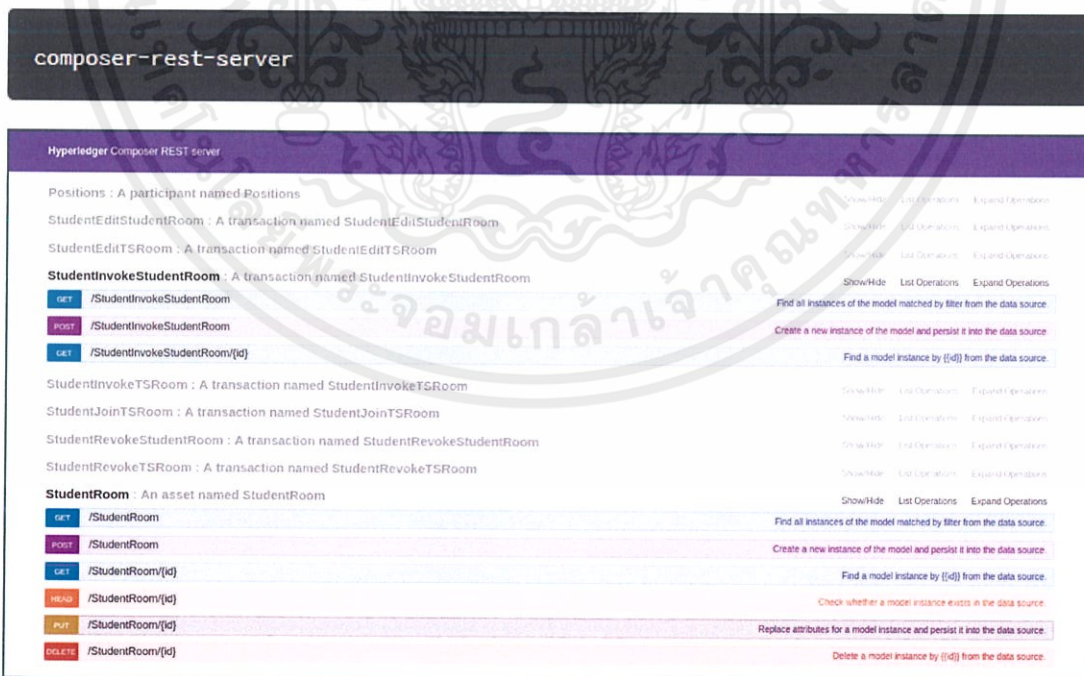
```
composer network ping --card admin@tutorial-network
```



รูปที่ 3.10 หน้าแสดงผลสำหรับเลือกการ์ดเครือข่ายธุรกิจของ Composer Playground

3.1.1.6 สร้าง REST Server

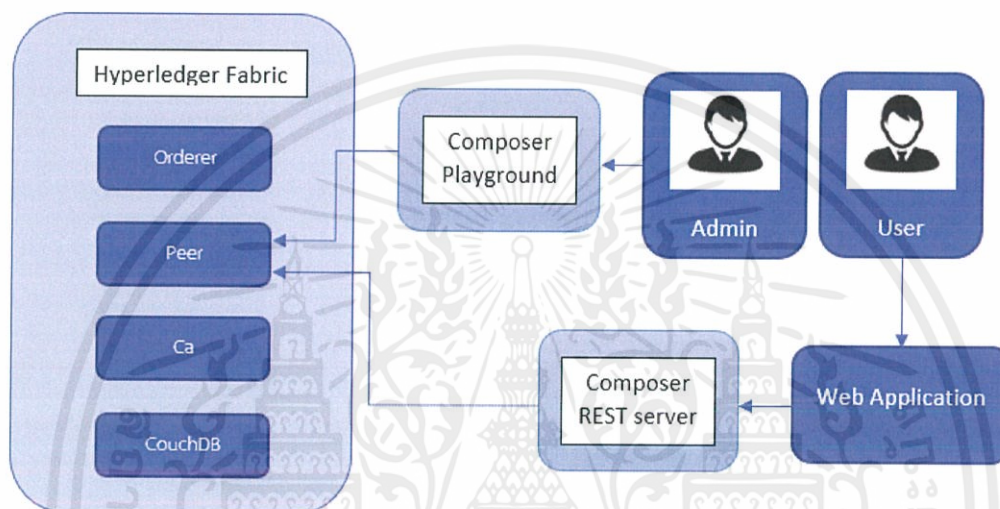
Hyperledger Composer สามารถสร้าง RESTful APIs ของเครือข่ายธุรกิจขึ้นมา ซึ่งมีการทำงานร่วมกับเว็บแอปพลิเคชันผ่าน Smart Contract จะได้ APIs สำหรับเชื่อมต่อกับหน้าเว็บ ดังรูปที่ 3.11



รูปที่ 3.11 หน้าแสดงผลของ Composer REST Server

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานในส่วนของ Hyperledger Fabric นั้น สามารถแสดงได้ดังรูปที่ 3.12 โดยผู้ดูแลระบบจะสามารถใช้งาน Composer Playground ในการเข้าถึงระบบที่ติดตั้งอยู่บน peer และสามารถทำการทดสอบและแก้ไขระบบได้ตามที่ต้องการ ในส่วนของผู้ใช้งานสามารถเข้าถึงระบบได้ผ่านทางเว็บแอปพลิเคชันที่เชื่อมต่อกับ Composer REST Server และสามารถใช้งานระบบการจองห้องเรียนได้เช่นเดียวกัน



รูปที่ 3.12 บล็อกไดอะแกรมการทำงานส่วนของ Hyperledger Fabric

3.1.2 การออกแบบในส่วนของ Webpage

ในส่วนของการออกแบบ Webpage ได้ทำการออกแบบผ่านโปรแกรม Laravel Framework ซึ่งเป็นเครื่องมือในการช่วยออกแบบหน้าเว็บ โดยได้ทำการแบ่งทั้งหมดออกเป็น 3 ส่วน ได้แก่

3.1.2.1 การออกแบบ Admin Webpage

ผู้จัดทำได้ทำการออกแบบเว็บในส่วนของ Admin ตามรูปที่ 3.13 โดยเริ่มจากการออกแบบหน้า login เมื่อผู้ใช้ทำการ login หากผู้ใช้มีบัญชีข้อมูลผู้ใช้งาน (User Account) อยู่ภายในระบบแล้ว ระบบจะทำการตรวจสอบความถูกต้องของ password แล้วจึงทำการ login เข้าสู่หน้าหลัก (Dashboard) แต่หากผู้ใช้งานไม่มีบัญชีข้อมูลผู้ใช้งาน จะต้องขออนุญาตจากทางแอดมินคนอื่นที่มีอยู่ภายในระบบอยู่แล้ว เพื่อทำการ register (ในการ register ของบัญชีข้อมูลผู้ใช้งานแบบ Admin จะเป็นการ register ผ่านทาง Hyperledger Fabric โดยตรงซึ่งจะต่างจากการ Register ของ Teacher และ Student) หากเป็นกรณีที่เป็นผู้ใช้งานคนแรก หรือทางคณะ

ผู้จัดทำสามารถ register ได้เลยโดยไม่ต้องร้องขออนุญาต ในส่วนของ Admin Dashboard จะประกอบไปด้วย 4 ส่วน ได้แก่

1) การออกแบบในส่วนของคุณสมบัติส่วนตัวของ Admin (Profile)

ในหน้าของคุณสมบัติส่วนตัวของผู้ใช้งาน จะเป็นหน้าที่แสดงข้อมูลส่วนตัวของผู้ใช้งานของ Admin ซึ่งผู้ใช้งานสามารถทำการแก้ไขข้อมูลส่วนตัวได้ตลอดเวลา

2) การออกแบบในส่วนของคุณสมบัติผู้ใช้อื่นๆ (User)

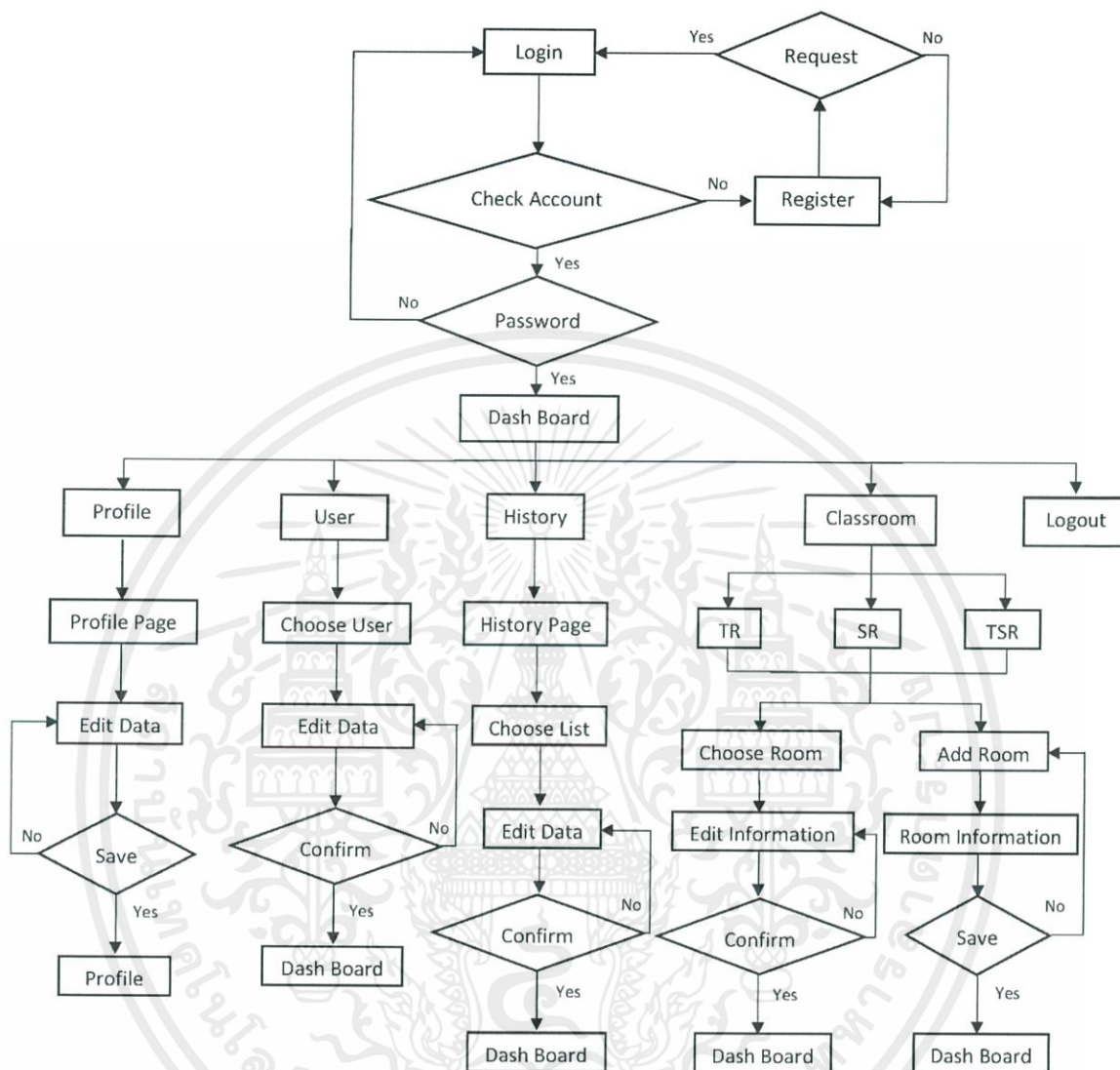
ในส่วนนี้เป็นส่วนที่ใช้ในการแก้ไขข้อมูลผู้ใช้งานภายในระบบ และยกเลิกการจองของผู้ใช้งานในกรณีฉุกเฉินหรือเกิดความผิดพลาด รวมถึงการเพิ่มเหรียญสำหรับการจองห้องเรียนในกรณีที่ต้องมีการใช้เหรียญให้แก่ผู้ใช้งานในระบบ

3) การออกแบบในส่วนของประวัติการใช้งานของบัญชีผู้ใช้ (History)

ในส่วนนี้เป็นส่วนที่แสดงประวัติการใช้งานของผู้ใช้งาน โดยจะแสดงการใช้งานทุกรูปแบบซึ่งประกอบไปด้วยการเพิ่มเหรียญสำหรับการใช้งานแก่ผู้ใช้อื่น การยกเลิกการจองห้องเรียนในกรณีต่างๆ และ การปิดการใช้งานห้อง หรือ การเพิ่มห้องสำหรับการใช้งาน นอกจากนี้ยังสามารถดูรายละเอียดและปรับแก้การใช้งานบางอย่างในกรณีที่เกิดความผิดพลาด เช่น การเพิ่มเหรียญผิดบัญชี หรือ การยกเลิกการปิดการใช้งานห้องเรียน

4) การออกแบบในส่วนของคุณสมบัติห้องเรียน (Classroom)

ในส่วนนี้เป็นส่วนที่ใช้ในการแก้ไขข้อมูลห้องเรียนภายในระบบ รวมถึงการเพิ่มและปิดการใช้งานห้องเรียนในระบบ โดยจะถูกแบ่งห้องออกเป็น 3 ประเภท คือ Teacher Room, Student Room, Teacher and Student Room โดยเมื่อกดเลือกห้องจะสามารถเลือกที่จะเพิ่มห้องเรียนหรือปิดการใช้งานห้องเรียนได้



รูปที่ 3.13 โฟลว์ชาร์ตแสดงการทำงานของ Admin Webpage

3.1.2.2 การออกแบบ Student Webpage

ส่วนต่อมาในการออกแบบ Webpage ของ Student ทางคณะผู้จัดทำได้ออกแบบให้มีความสะดวกต่อการใช้งานของผู้ใช้งาน แสดงดังรูปที่ 3.14 โดยเริ่มจากการ login ในกรณีที่ผู้ใช้มีบัญชีผู้ใช้อยู่แล้ว จะสามารถ login โดยจะมีการตรวจสอบ password หากตรวจสอบแล้วถูกต้องจะเข้าสู่ Dashboard ส่วนในกรณีที่ผู้ใช้ไม่มีบัญชีผู้ใช้ ให้ทำการ register แล้วจึง login เข้าสู่ Dashboard ซึ่งภายในแบ่งออกเป็น 4 ส่วน ได้แก่

1) การออกแบบในส่วนของคุณข้อมูลส่วนตัวของ Student (Profile)

ส่วนแรกเป็นส่วนของคุณข้อมูลส่วนตัวผู้ใช้งาน จะเป็นส่วนที่แสดง และแก้ไข ข้อมูลส่วนตัวของผู้ใช้งาน ในส่วนนี้จะมีความคล้ายคลึงกับส่วนของ Admin Profile แต่จะมีรายละเอียดของคุณข้อมูลที่แตกต่างกันออกไป

2) การออกแบบในส่วนของคุณข้อมูลห้องเรียน (Classroom)

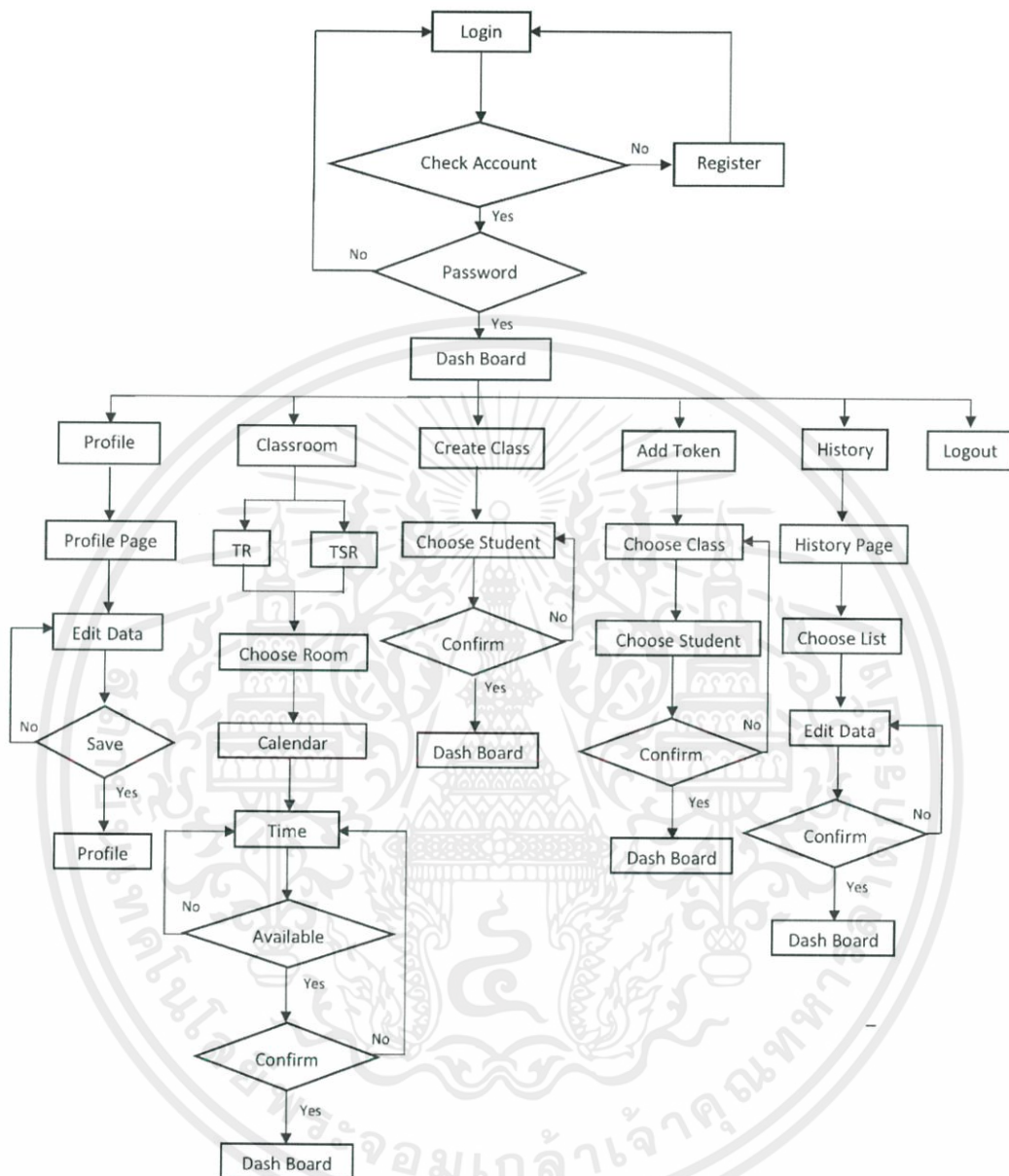
ในส่วนนี้จะมีความแตกต่างจากส่วนของ Admin กล่าวคือ ในส่วนของคุณ Student จะถูกใช้สำหรับการจองห้องเรียนโดยเฉพาะ โดยห้องเรียนจะถูกแบ่งออกเป็น 2 แบบ คือ Student Room และ Teacher and Student Room เมื่อผู้ใช้งานทำการเลือกแบบใดแบบหนึ่ง จะพบรายชื่อห้องเรียน ซึ่งจะมีข้อมูลของห้องนั้นระบุอยู่ จากนั้นผู้ใช้งานสามารถทำการเลือกวัน และเวลาในการทำการจองในห้องที่สามารถจองได้ รวมถึงการยกเลิกการจองในห้องที่ได้ทำการจองไว้แล้ว เมื่อทำการจองเสร็จสิ้นผู้ใช้งานจะสามารถใช้งานได้โดยทำการ login แล้วกด Unlock ห้องเรียนเมื่อถึงเวลาที่ทำการจองห้องเรียน และจะสามารถใช้งานได้จนกระทั่งหมดเวลาการจอง โดยห้องที่สามารถจองได้คือห้องที่ไม่มีผู้ใช้งานอื่นมาทำการจองในเวลานั้น และผู้ใช้งานไม่สามารถจองห้องในเวลาที่มีการจองไว้ก่อนแล้วจากผู้ใช้งานอื่น ในกรณีที่ผู้ใช้งานต้องการจะจองห้อง Teacher and Student Room จำเป็นจะต้องมี token หรือ เหรียญ ที่ใช้สำหรับการจองห้องเรียน โดยเหรียญนี้จะได้รับจาก Teacher หรือ Admin เท่านั้น

3) การออกแบบในส่วนของการเข้าร่วมชั้นเรียน (Join Class)

ส่วนนี้จะเป็นส่วนที่ทำงานร่วมกับส่วนของคุณ Teacher Webpage ซึ่งจะกล่าวต่อไปในหัวข้อ 3.1.2.3 โดยในส่วนนี้จะเป็นการสแกน QR Code ที่ได้รับจาก Teacher ซึ่งได้ทำการสร้างชั้นเรียนผ่านการจองห้องของคุณ Teacher User เมื่อผู้ใช้งานทำการสแกนระบบจะทำการบันทึกการเข้าร่วมชั้นเรียน เพื่อยืนยันการเข้าเรียนของผู้ใช้งาน รวมถึงการร้องขอการเข้าร่วมในกลุ่มชั้นเรียนด้วย

4) การออกแบบในส่วนของคุณประวัติการใช้งานของผู้ใช้งาน (History)

ในส่วนของคุณการออกแบบของคุณประวัติการใช้งานของผู้ใช้งานจะมีความคล้ายคลึงกับในส่วนของคุณ Admin แต่จะมีการเพิ่มเติมของคุณการจองห้องเรียน ซึ่งจะแสดงทั้งส่วนที่ใช้งานไปแล้ว หรือยกเลิกการจองไปแล้ว รวมถึงห้องที่รอเวลาการจองและห้องที่กำลังใช้งานอยู่นอกจากนี้ยังแสดงในส่วนของคุณ Join Class ด้วย



รูปที่ 3.14 โฟลว์ชาร์ตแสดงการทำงานของ Teacher Webpage

3.1.2.3 การออกแบบ Teacher Webpage

ในส่วนสุดท้ายซึ่งเป็นการออกแบบ Teacher Webpage ซึ่งจะมีขั้นตอนการ login ที่เหมือนกับ Student Webpage แต่อาจมีรายละเอียดที่อยู่ภายใน Register Page ที่ต่างกันเล็กน้อย และในส่วนของ Dashboard ได้ทำการแบ่งเป็น 4 ส่วน แสดงดังรูปที่ 3.15 ได้แก่

1) การออกแบบในส่วนของคุณข้อมูลส่วนตัวของ Student (Profile)

ส่วนแรกเป็นส่วนของคุณข้อมูลส่วนตัวผู้ใช้งาน จะเป็นส่วนที่แสดง และแก้ไข ข้อมูลส่วนตัวของผู้ใช้งาน ในส่วนนี้จะมีความคล้ายคลึงกับส่วนของ Admin Profile และ Student Profile แต่จะมีรายละเอียดของคุณข้อมูลที่แตกต่างกันออกไป

2) การออกแบบในส่วนของคุณข้อมูลห้องเรียน (Classroom)

ในส่วนนี้จะมีความคล้ายคลึงกับในส่วนของคุณ Student ที่จะถูกใช้สำหรับการจองห้องเรียนโดยเฉพาะ โดยห้องเรียนจะถูกแบ่งออกเป็น 2 แบบ คือ Teacher Room และ Teacher and Student Room เมื่อผู้ใช้งานทำการเลือกแบบใดแบบหนึ่งจะพบรายชื่อห้องเรียน ซึ่งจะมีข้อมูลของห้องนั้นระบุอยู่ จากนั้นผู้ใช้งานสามารถทำการเลือกวันและเวลาในการทำการจองในห้องที่สามารถจองได้ รวมถึงการยกเลิกการจองในห้องที่ได้ทำการจองไว้แล้ว เมื่อทำการจองเสร็จสิ้นผู้ใช้งานจะสามารถใช้งานได้โดยทำการ login แล้วกด Unlock ห้องเรียนเมื่อถึงเวลาที่ทำการจองห้องเรียน และจะสามารถใช้งานได้จนกระทั่งหมดเวลาการจอง โดยห้องที่สามารถจองได้คือห้องที่ไม่มีผู้ใช้งานอื่นมาทำการจองในเวลานั้น และผู้ใช้งานไม่สามารถจองห้องในเวลาที่มีการจองไว้ก่อนแล้วจากผู้ใช้งานอื่น ในส่วนที่เพิ่มมาสำหรับกรณีที่ผู้ใช้งานเลือกห้องเรียนที่อยู่ในรายชื่อ Teacher and Student Room ผู้ใช้งานจะสามารถสร้าง Class ซึ่งจะเป็นการทำงานที่สอดคล้องกับการ Join Class ของ Student ที่กล่าวไว้ในหัวข้อ 3.1.2.2 โดยผู้ใช้งานสามารถเลือกกลุ่มการเรียนที่ต้องการจากรายชื่อที่ได้จากการ Create Class พร้อมระบุรายละเอียดของชั้นเรียน จากนั้นให้ผู้ใช้งานทำการเลือกวันเวลาที่ต้องการใช้งานห้องเรียน เมื่อผู้ใช้เพิ่มเติมรายละเอียดและยืนยันการใช้งานแล้ว ผู้ใช้จะได้รับ QR Code เพื่อใช้สำหรับตรวจสอบการเข้าร่วมชั้นเรียนของ Student

3) การออกแบบในส่วนของการเพิ่มรายชื่อกลุ่มเรียน (Create Class)

ในส่วนนี้จะเป็นการออกแบบหน้าเว็บสำหรับการเพิ่มรายชื่อสำหรับการเข้าใช้งานของ Student เมื่อ Student ทำการสแกน QR Code เพื่อทำการเข้าร่วมชั้นเรียน รายชื่อของผู้ใช้งานนั้นจะถูกส่งมาและทำการเก็บรายชื่อไว้เพื่อเรียกใช้ในการสร้างชั้นเรียนครั้งต่อไป หรือผู้ใช้งานสามารถเลือกนักเรียนจากในกลุ่มรายชื่อเพื่อทำการเพิ่มเหรียญการใช้งานห้องเรียนให้นักเรียนได้

4) การออกแบบในส่วนของการเพิ่มเหรียญใช้งานห้องเรียน (Add Token)

ในส่วนนี้จะเป็นการออกแบบหน้าเว็บสำหรับการเพิ่มเหรียญการใช้งานของ Student เมื่อ Student ทำการจองห้องที่มีรายชื่ออยู่ใน Teacher and Student Room หรือกลุ่ม

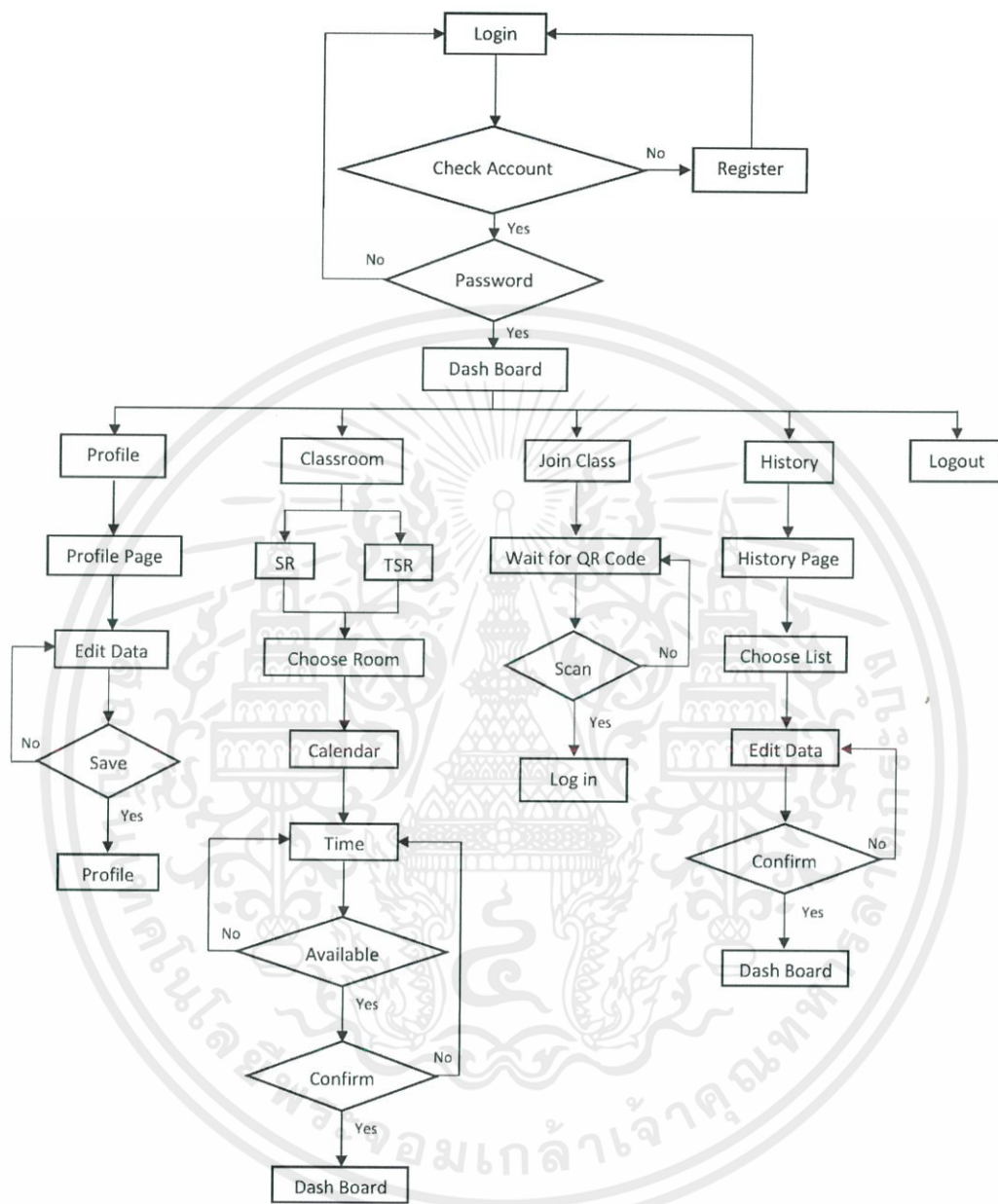
รายชื่อที่สร้างขึ้นใน Create Class โดยผู้ใช้งานสามารถเลือกนักเรียนจากในกลุ่มชั้นเรียนใดก็ได้แล้ว จึงทำการเพิ่มเหรียญให้แก่นักเรียนคนนั้น (ในส่วนนี้จะมีความคล้ายคลึงกับส่วนของ Admin ที่สามารถกดเพิ่มเหรียญให้แก่ผู้ใช้งานคนอื่นได้)

5) การออกแบบในส่วนของประวัติการใช้งานของผู้ใช้งาน (History)

ในส่วนของการออกแบบของประวัติการใช้งานของผู้ใช้งานจะมีความคล้ายคลึงกับในส่วนของ Student แต่จะมีการเปลี่ยนในส่วนของการเข้าร่วมชั้นเรียน ให้กลายเป็นการเพิ่มเหรียญแทน โดยที่ยังคงมีรายละเอียดต่างๆอยู่ภายในแต่ละการใช้งานนั้น



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 โฟลว์ชาร์ตแสดงการทำงานของ Student Webpage

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

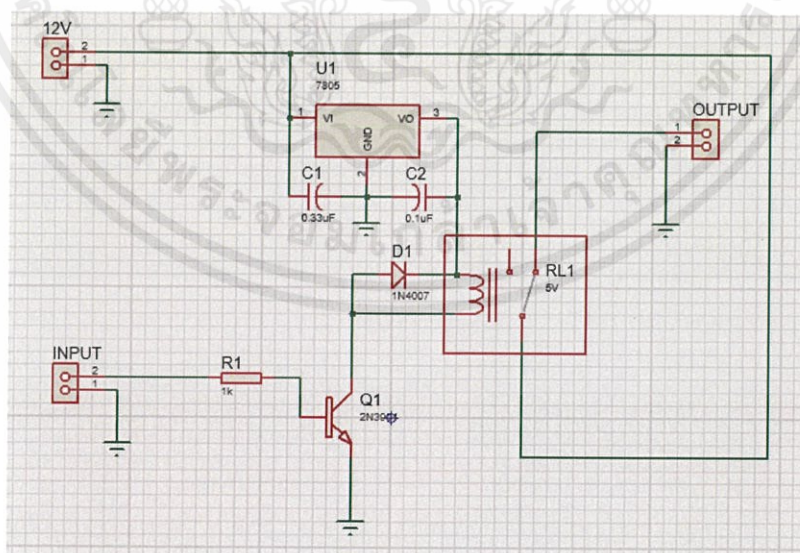
3.1.3 การออกแบบในส่วนควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า

3.1.3.1 การออกแบบวงจรควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า

วงจรที่ใช้ในการควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้าเพื่อเปิด-ปิด ประตูนั้น ผู้จัดทำได้ออกแบบวงจรการทำงานดังแสดงในรูปที่ 3.16 โดยใช้การทำงานของทรานซิสเตอร์ร่วมกับรีเลย์ โดยอาศัยการทำงานของทรานซิสเตอร์ 2 โหมด คือ โหมด Saturation และ โหมด Cut-off มาทำหน้าที่เสมือนสวิตช์ 2 สถานะ คือ สวิตช์ทำงาน และ สวิตช์ไม่ทำงาน ตามลำดับ โดยสัญญาณแรงดันไฟตรงจากขา GPIO ของ Raspberry Pi จะเป็นสัญญาณควบคุมซึ่งจะถูกป้อนเข้าที่ขาเบสของทรานซิสเตอร์ ซึ่งจะมี 2 กรณี ดังนี้

1) กรณีผู้ใช้งานทำการจองห้องทำการกดเปิดห้องเรียนผ่านหน้าเว็บของระบบเมื่อถึงเวลาที่ได้ทำการจองไว้หรือกดปุ่มสวิตช์จากภายในห้องเรียนเพื่อทำการเปิดประตู ขา GPIO ของ Raspberry Pi จะจ่ายแรงดันไฟตรง 3.3V ทำให้มีแรงดันไฟเลี้ยง 5V จ่ายให้กับรีเลย์ซึ่งจะทำให้รีเลย์ทำงานและจะทำการตัดกระแสไฟตรงแรงดัน 12V ที่จ่ายไปยังกลอนแม่เหล็กไฟฟ้า กลอนแม่เหล็กไฟฟ้าหยุดการทำงาน ทำให้ประตูสามารถถูกเปิดได้

2) กรณีที่ไม่มีการจองห้องผ่านระบบการจองห้องเรียนหรือยังไม่ได้กดเปิดห้องเรียนเพื่อทำการเปิดประตู ขา GPIO ของ Raspberry Pi จะจ่ายแรงดันไฟตรง 0V ทำให้ไม่มีแรงดันไฟเลี้ยงจ่ายให้กับรีเลย์ซึ่งจะทำให้รีเลย์ไม่ทำงาน กลอนแม่เหล็กไฟฟ้าไม่ถูกตัดกระแสไฟจึงไม่หยุดการทำงาน ทำให้ประตูไม่สามารถถูกเปิดได้

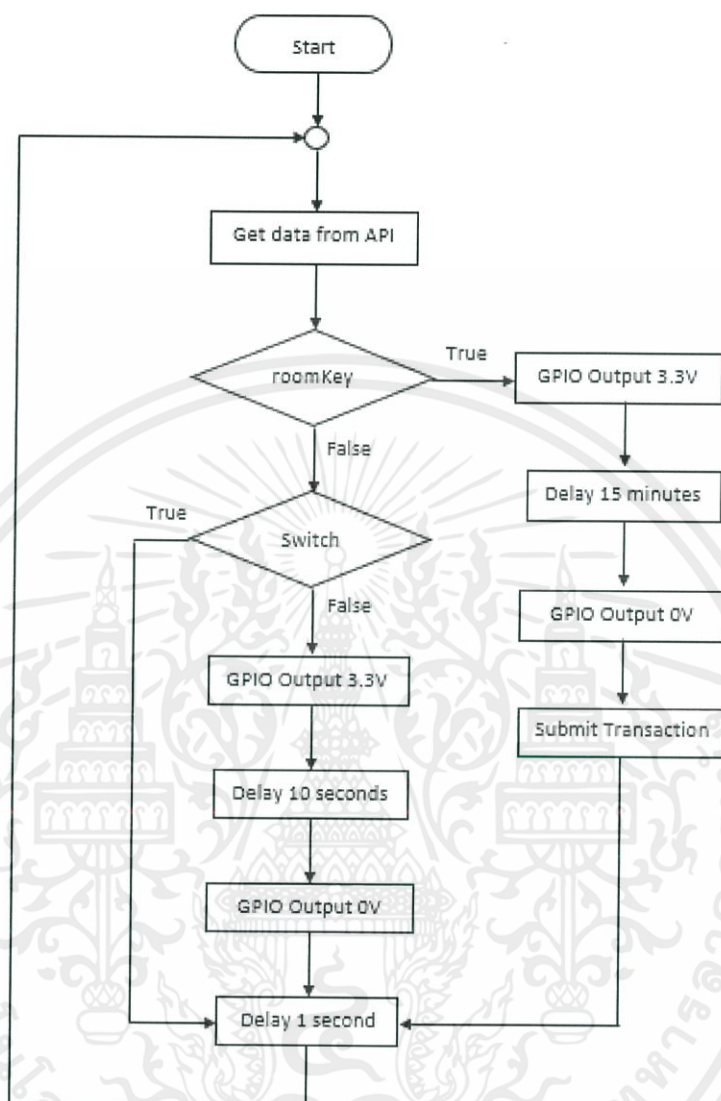


รูปที่ 3.16 แผนผังวงจรควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า

3.1.3.2 การออกแบบคำสั่งการส่งสัญญาณควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้าจากไมโครคอนโทรลเลอร์

ทำการออกแบบโค้ดภาษา Python ที่ใช้ควบคุมการส่งสัญญาณผ่านขา GPIO ของ Raspberry Pi เข้าสู่วงจรควบคุมการทำงานของกลอนแม่เหล็ก มีขั้นตอนการทำงานดังแสดงในรูปที่ 3.17 โดยออกแบบให้ Raspberry Pi ทำการรับค่าข้อมูลตัวแปร Room Key ที่เก็บอยู่ในระบบ ผ่านทาง RESTful API ทุกๆ 1 วินาที เมื่อตัวแปรที่รับมามีค่าเป็น true Raspberry Pi จะทำการจ่ายแรงดันไฟตรง 3.3V เป็นสัญญาณอินพุตของวงจรควบคุมการทำงานของประตูแม่เหล็ก เพื่อหยุดการทำงานของกลอนแม่เหล็กไฟฟ้าเป็นระยะเวลา 15 นาที หลังจากนั้นจึงหยุดการจ่ายสัญญาณ ทำให้กลอนแม่เหล็กไฟฟ้ากลับมาทำงานอีกครั้ง แล้วส่ง Transactions ของการถือคห้องเรียนไปยังระบบผ่าน API เพื่อทำการเปลี่ยนค่าตัวแปร Room Key ให้มีค่าเป็น false

นอกจากนี้ยังสามารถใช้ปุ่มสวิตช์ภายในห้องในการหยุดการทำงานของกลอนแม่เหล็กไฟฟ้าได้เช่นกัน โดยเมื่อกดสวิตช์ Raspberry Pi จะได้รับค่าอินพุตจากสวิตช์ที่ต่อกับกราวด์มีค่าเป็น false Raspberry Pi จะทำการจ่ายสัญญาณแรงดันไฟตรง 3.3V เพื่อหยุดการทำงานของกลอนแม่เหล็กไฟฟ้าเป็นระยะเวลา 10 วินาที หลังจากนั้นจึงหยุดจ่ายสัญญาณและกลอนแม่เหล็กไฟฟ้ากลับมาทำงานตามปกติ

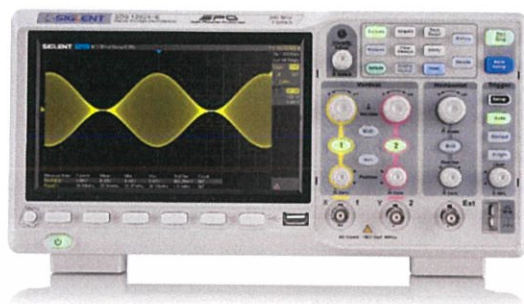


รูปที่ 3.17 บล็อกไดอะแกรมแสดงการทำงานของไมโครคอนโทรลเลอร์

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 Oscilloscope

Oscilloscope คือ เครื่องมือที่ใช้วัดและแสดงรูปของแรงดันไฟฟ้า ซึ่งสามารถวัดค่าความถี่ เฟส และคาบเวลาของสัญญาณได้ ซึ่งผลสัญญาณจะถูกแสดงออกมาเป็นกราฟผ่านหลอดภาพที่ฉาบด้วยฟอสเฟอร์ แสดงดังรูปที่ 3.18



รูปที่ 3.18 Oscilloscope

3.2.2 Raspberry Pi 3 Model B

Raspberry Pi คือ บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก สามารถใช้งานได้หลากหลายไม่ว่าจะเป็นการใช้งาน Spreadsheet การประมวลผลค่า เชื่อมต่ออินเทอร์เน็ต ส่งอีเมล หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วิดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย โดยใน Raspberry Pi 3 Model B นั้นจะมี CPU 1.2GHz 64bit Quad Core ARM Cortex-A53 (ARMv8) รองรับ Wi-Fi 2.4GHz และ Bluetooth 4.1 (BLE) พร้อมสายอากาศในตัว สามารถใช้งานกับ MicroSD card ได้สูงสุดถึง 64GB และ Micro USB Power รับไฟได้สูงสุด 2.5A แสดงดังรูปที่ 3.19



รูปที่ 3.19 Raspberry Pi 3 Model B

3.2.3 กลอนแม่เหล็กไฟฟ้า

กลอนแม่เหล็กไฟฟ้าที่ใช้กันทั่วไปหลักๆมีอยู่ 2 แบบ ซึ่งแต่ละแบบจะมีการใช้งานแตกต่างกัน แบบแรกเป็นแบบทั่วไปจะใช้การจ่ายไฟไปยังแถบแม่เหล็ก และใช้แรงดึงดูดของแม่เหล็กไฟฟ้าในการดูดแผ่นเหล็กเข้ามา เรียกว่า Magnetic Lock มีข้อดีที่ความแข็งแรง ทนทาน แต่มีข้อเสียที่สามารถติดตั้งได้เฉพาะประตูที่มีวงกบเพื่อยึดกับตัวกลอนแม่เหล็ก และเมื่อติดตั้งแล้วประตูจะสามารถเปิดได้แค่ทางเดียว แบบที่สองจะใช้การทำงานแบบสลักลิค โดยจะใช้เซนเซอร์ในการตรวจจับแผ่นแม่เหล็ก เมื่อตำแหน่งของแผ่นแม่เหล็กตรงกับขั้วสลักจึงจะติดสลักขึ้นมาล็อกทันที เรียกว่า Electric Bolt มีความทนทานน้อยกว่าแบบแรก แต่สามารถเปิดประตูได้ทั้ง 2 ด้าน โดยทั้ง 2 แบบต้องใช้ไฟเลี้ยงอุปกรณ์ในการทำงาน โดยไฟที่จ่ายนั้นอาจมาจาก Power Supply หรือ เครื่องควบคุมการทำงานของประตู แสดงดังรูปที่ 3.20



รูปที่ 3.20 กลอนแม่เหล็กไฟฟ้า

3.3 การจัดเก็บผลการทดลอง

3.3.1 การจัดเก็บผลการทดลองในส่วนของ Hyperledger Composer

3.3.1.1 การทดสอบการจัดเก็บข้อมูลของ Participants และ Assets

การทดสอบการทำงานของ Hyperledger Composer จะใช้การแสดงผลผ่าน Composer Playground ในการจัดเก็บผล โดยในการทดสอบการจัดเก็บข้อมูลของ Participants และ Assets นั้นจะทำการทดลองสร้าง Participants แต่ละประเภทและ Assets แต่ละประเภท จากนั้นทำการทดสอบการแก้ไขข้อมูลของแต่ละ Participants และบันทึกหน้าแสดงผลของ Composer Playground ในแต่ละการทดลอง สังเกตผลการทดลองของ Participants และ Assets ที่บันทึกได้

3.3.1.2 การทดสอบการใช้งาน Transactions

การทดสอบการใช้งาน Transactions จะเป็นการทดสอบผลการทำงานของแต่ละ Transactions และจะต้องมีความสอดคล้องกับเงื่อนไขที่กำหนดไว้ในส่วนของ Script file โดยจะทำการทดลองส่งแต่ละ Transactions เพื่อสังเกตและบันทึกผลการทำงานผ่านหน้าแสดงผลของ Playground หากเกิดความไม่สอดคล้องหรือเกิดข้อผิดพลาด จะสามารถทำการแก้ไขในส่วน ของ Transactions และ Script File เพื่อนำมาปรับใช้ได้ภายหลัง

3.3.1.3 การทดสอบการเข้าถึงข้อมูลของ Access Control

การทดสอบการเข้าถึงข้อมูลของแต่ละ Participants จะทำการทดสอบโดยการสร้างการ์ด teacher@classroom-management-kmitl และ student@classroom-management-kmitl ขึ้น เพื่อใช้ในการเข้าใช้งานผ่าน Positions ของอาจารย์และนักศึกษา ทดสอบการเข้าถึงข้อมูลของระบบผ่านทางฝ่ายอาจารย์และนักศึกษา จากนั้นบันทึกผลผ่านหน้าแสดงผลของ Playground

3.3.2 การจัดเก็บผลการทดลองในส่วน of Webpage

ทางคณะผู้จัดทำได้ทำการเก็บผลการทดลองโดยการสร้างเว็บไซต์แล้วทดลองเชื่อมต่อกับ Hyperledger Fabric แล้วทำการทดสอบแยกในแต่ละส่วน โดยเริ่มแบ่งจากการเก็บผลดังนี้

3.3.2.1 การทดสอบการแสดงผล Admin Webpage

ในการทดสอบครั้งนี้เราจะทำการจัดเก็บผลโดยเรียงตามโพล์ชาร์ตใน 3.1.2.1 โดยเริ่มจากส่วนของการ Login, Register, Dashboard, Profile, User, History และ Classroom ตามลำดับ โดยในแต่ละหัวข้อก็จะทำการเก็บผลย่อยลงไปอีกตามที่ได้ออกแบบไว้ รวมถึงการเก็บผลในกรณีที่เกิดผิดพลาดหรือรูปแบบการใช้งานอื่นๆเช่นกัน

3.3.2.2 การทดสอบการแสดงผล Student Webpage

ในการทดสอบครั้งนี้เราจะทำการจัดเก็บผลโดยเรียงตามโพล์ชาร์ตใน 3.1.2.2 โดยเริ่มจากส่วนของการ Login, Register, Dashboard, Profile, Classroom, Join class และ History ตามลำดับ โดยในแต่ละหัวข้อก็จะทำการเก็บผลย่อยลงไปอีกตามที่ได้ออกแบบไว้ รวมถึงการเก็บผลในกรณีที่เกิดผิดพลาดหรือรูปแบบการใช้งานอื่นๆเช่นกัน

3.3.2.3 การทดสอบการแสดงผล Teacher Webpage

ในการทดสอบครั้งนี้เราจะทำการจัดเก็บผลโดยเรียงตามโพล์ชาร์ตใน 3.1.2.3 โดยเริ่มจากส่วนของการ Login, Register, Dashboard, Profile, Classroom, Add

token และ History ตามลำดับ โดยในแต่ละหัวข้อก็จะทำการเก็บผลการทดลองตามที่ได้ออกแบบไว้ รวมถึงการเก็บผลในกรณีที่เกิดความผิดพลาดหรือรูปแบบการใช้งานอื่นๆเช่นกัน

3.3.3 การจัดเก็บผลการทดลองในส่วนควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า

3.3.3.1 ทดสอบการทำงานของวงจรควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้าตามสัญญาณควบคุมที่ได้จากขา GPIO ของ Raspberry Pi โดยจะใช้การกำหนดให้ Raspberry Pi ส่งสัญญาณออกจากขา GPIO มา 2 สถานะ คือ 3.3V และ 0V เพื่อตรวจสอบการทำงานของกลอนแม่เหล็กไฟฟ้า

3.3.3.2 ทดสอบการทำงานของกลอนแม่เหล็กไฟฟ้าตามคำสั่งการส่งสัญญาณควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้าจากไมโครคอนโทรลเลอร์ ทำการทดสอบการกดเปิดประตูผ่านทางหน้าเว็บของระบบ แล้วทำการวัดสัญญาณที่ขา GPIO ของ Raspberry Pi และสัญญาณที่ขาเอาต์พุตของรีเลย์ว่ามีการทำงานเป็นไปตามที่ออกแบบไว้ตามคำสั่งหรือไม่

3.3.3.3 ทดสอบการทำงานของปุ่มสวิตช์ในการส่งค่าอินพุตไปยัง Raspberry Pi เพื่อใช้ในการส่งสัญญาณจาก Raspberry Pi เพื่อควบคุมการทำงานของกลอนแม่เหล็ก โดยวัดสัญญาณอินพุตที่ขาของสวิตช์ และสัญญาณเอาต์พุตที่ออกจากขา GPIO ของ Raspberry Pi เพื่อตรวจสอบการทำงานของไมโครคอนโทรลเลอร์

3.3.4 การจัดเก็บผลการทดลองในส่วนของการ Join Class ผ่านการสแกน QR Code

ทดสอบการทำงานของตรวจสอบการเข้าชั้นเรียนของนักศึกษา โดยทำการสร้าง Classroom ขึ้นมาเพื่อจัดเก็บรายชื่อนักศึกษาที่เข้าชั้นเรียนไว้ในระบบ และใช้การสแกน QR Code ที่ได้รับจากการสร้าง Classroom เพื่อให้นักศึกษาทำการ login เข้าสู่ระบบ และจัดเก็บรายชื่อไว้ใน Classroom ที่สร้างขึ้นนั้น จากนั้นทำการบันทึกผลการทำงานของระบบผ่านทางหน้าเว็บและหน้าแสดงผลของ Composer Playground

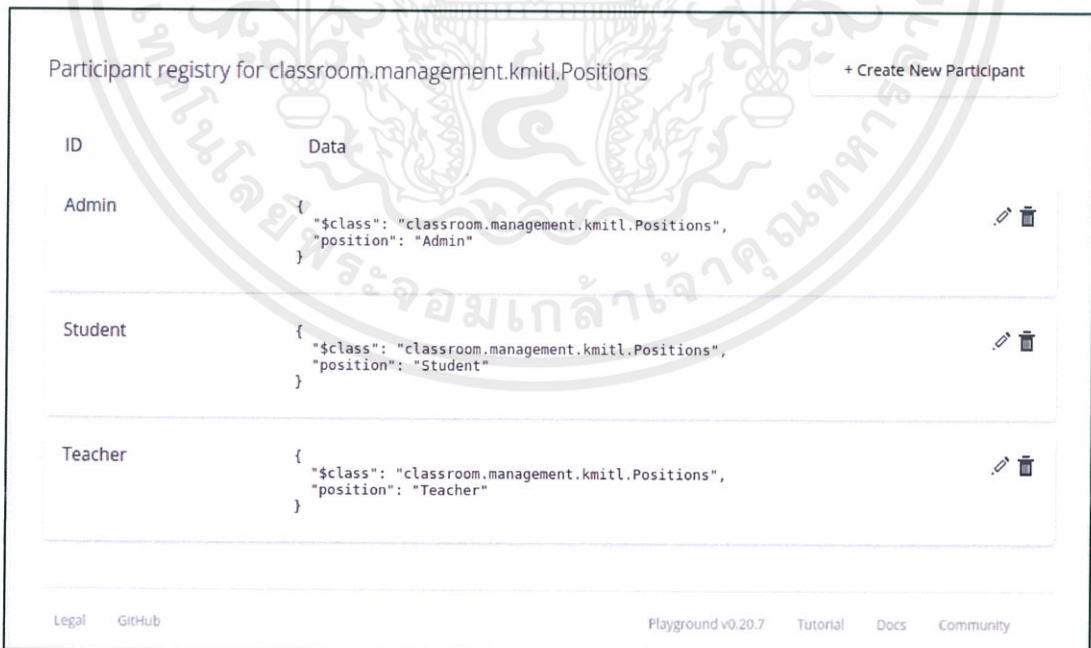
บทที่ 4







ผลการทดลอง

4.1 ผลการทดลองในส่วนของ Hyperledger Composer

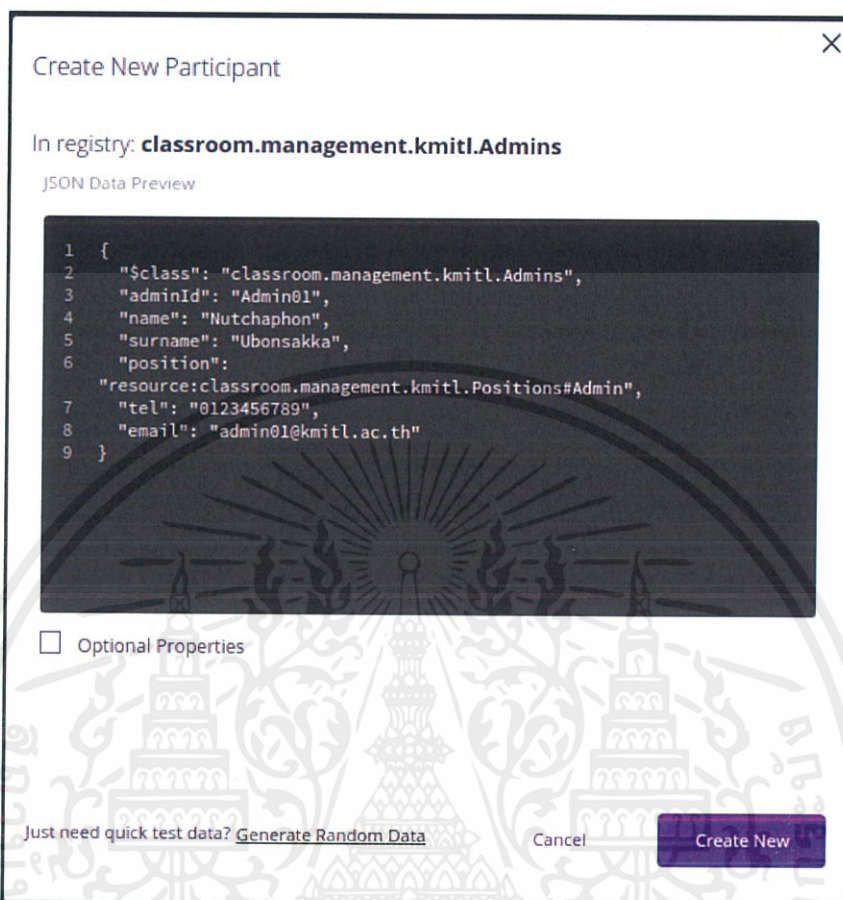
4.1.1 ผลการทดสอบการจัดเก็บข้อมูลของ Participants และ Assets

การทดลองในส่วนของการจัดเก็บข้อมูลจะทำการสร้างตัวอย่าง Participants และ Assets ขึ้นมาผ่านทาง Composer Playground และทำการกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง จากนั้นบันทึกผลที่เกิดขึ้นผ่านหน้าแสดงผลของ Playground โดยการสร้าง Participants นั้น จะต้องทำการกำหนดบทบาทของแต่ละ Participants ผ่านทางการสร้าง Positions ต่างๆขึ้นมาดัง แสดงในรูปที่ 4.1 ซึ่งจะแบ่งเป็น ผู้ดูแลระบบ (Admin), นักศึกษา (Student) และอาจารย์ (Teacher) ในรูปที่ 4.2 เป็นการทดสอบการสร้าง Participants ของ Admins โดยกำหนดให้ ID ของผู้ดูแลระบบ คือ Admin01 ทำการกรอกข้อมูลส่วนตัวที่ใช้ในการลงทะเบียนทั้งหมด ได้แก่ ชื่อ-นามสกุล, เบอร์โทรศัพท์, อีเมล และกำหนดบทบาทให้เป็นผู้ดูแลระบบ (Admin) เมื่อทำการสร้าง จะได้ Participant ของ Admins ที่ชื่อ Admint01 ขึ้นมาและจัดเก็บข้อมูลทั้งหมดที่ได้ทำการ ลงทะเบียนแสดงดังรูปที่ 4.3 จากนั้นทำการทดสอบการแก้ไขข้อมูลของ Admins โดยทำการแก้ไข ในส่วนของชื่อ-นามสกุลของ Admin01 ดังรูปที่ 4.4 และรูปที่ 4.5

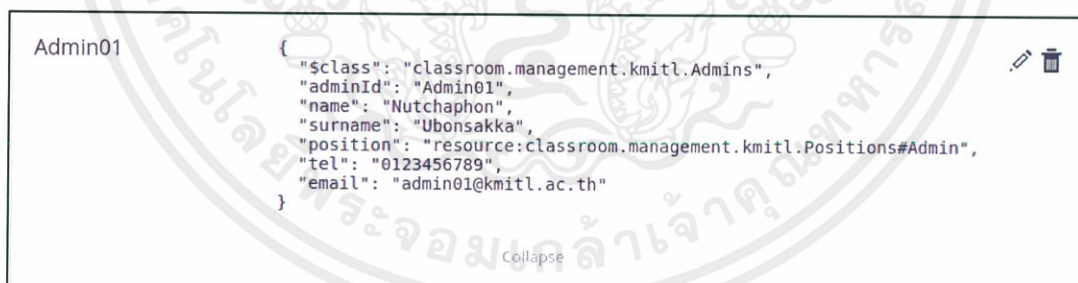


ID	Data	
Admin	<pre>{ "\$class": "classroom.management.kmitl.Positions", "position": "Admin" }</pre>	 
Student	<pre>{ "\$class": "classroom.management.kmitl.Positions", "position": "Student" }</pre>	 
Teacher	<pre>{ "\$class": "classroom.management.kmitl.Positions", "position": "Teacher" }</pre>	 

รูปที่ 4.1 Participants ของ Positions ใช้ในการกำหนดบทบาทของ Participants อื่นๆ



รูปที่ 4.2 หน้าสำหรับกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง Participant ของ Admins



รูป 4.3 หน้าแสดงผลการจัดเก็บข้อมูลของ Admin01

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Update Participant

In registry: **classroom.management.kmitl.Admins**

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.Admins",
3   "adminId": "Admin01",
4   "name": "Chetsada",
5   "surname": "Tarasawat",
6   "position":
7   "resource:classroom.management.kmitl.Positions#Admin",
8   "tel": "0123456789",
9   "email": "admin01@kmitl.ac.th"
}

```

Optional Properties

Cancel Update

รูปที่ 4.4 หน้าสำหรับแก้ไขข้อมูลเพื่อทำการอัปเดตข้อมูลของ Admin01

Admin01

```

{
  "$class": "classroom.management.kmitl.Admins",
  "adminId": "Admin01",
  "name": "Chetsada",
  "surname": "Tarasawat",
  "position": "resource:Classroom.management.kmitl.Positions#Admin",
  "tel": "0123456789",
  "email": "admin01@kmitl.ac.th"
}

```

Collapse

รูปที่ 4.5 หน้าแสดงผลข้อมูลของ Admin01 หลังทำการอัปเดตข้อมูล

จากนั้นทำการทดสอบการสร้าง Participants ของ Students ในรูปที่ 4.6 โดยกำหนดให้ ID ของนักศึกษา คือ Student01 จากนั้นทำการกรอกข้อมูลส่วนตัว ชื่อ-นามสกุล, คณะ, ภาควิชา, เบอร์โทรศัพท์, อีเมล และกำหนดบทบาทให้เป็นนักศึกษา (Student) นอกจากนี้ยังสามารถกำหนด token ที่ใช้ในการจองห้องเรียนสำหรับนักศึกษา ซึ่งโดยปกติจะกำหนดให้นักศึกษาแต่ละคนมี token เริ่มต้นอยู่ที่ 2 เหรียญ และเมื่อทำการสร้างจะได้ Participant ของ Students ที่

ชื่อ Student01 ดังแสดงในรูปที่ 4.7 แล้วทำการทดสอบการแก้ไขข้อมูลของ Students โดยทำการแก้ไขในส่วนของชื่อ-นามสกุลของ Student01 ดังแสดงในรูปที่ 4.8 และรูปที่ 4.9

Create New Participant

In registry: **classroom.management.kmitl.Students**

JSON Data Preview

```

1  {
2  "$class": "classroom.management.kmitl.Students",
3  "studentId": "Student01",
4  "name": "Nutchaphon",
5  "surname": "Ubonsakka",
6  "position":
7  "resource:classroom.management.kmitl.Positions#Student",
8  "faculty": "Engineering",
9  "major": "Telecommunication",
10 "year": "4",
11 "tel": "0123456789",
12 "email": "student01@kmitl.ac.th",
13 "token": 2
  }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Create New

รูปที่ 4.6 หน้าสำหรับกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง Participant ของ Students

```

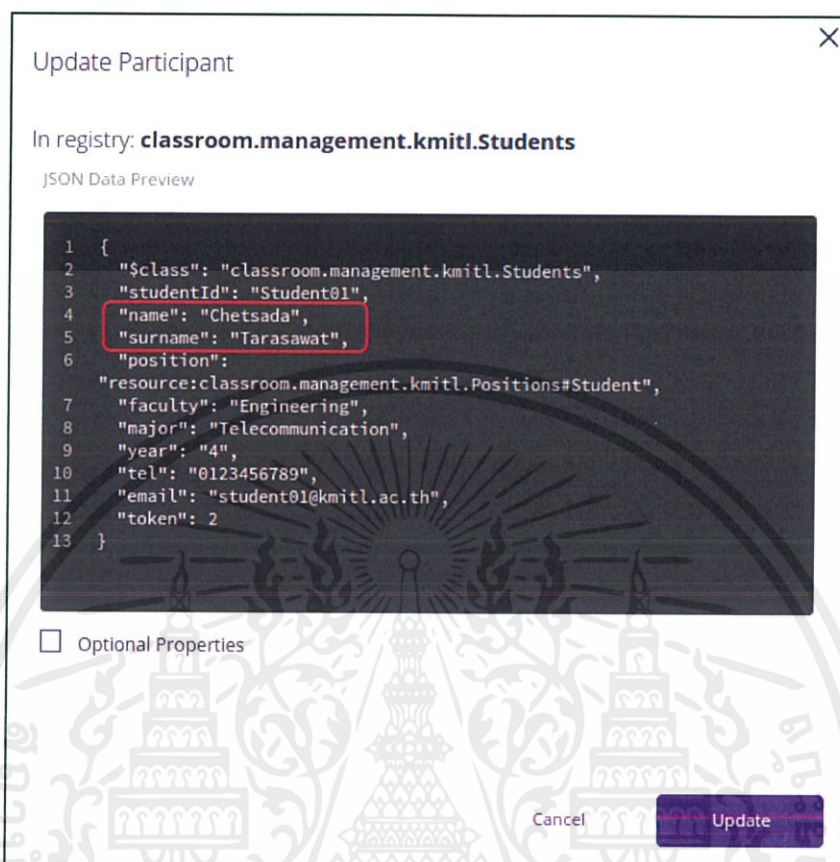
Student01
{
  "$class": "classroom.management.kmitl.Students",
  "studentId": "Student01",
  "name": "Nutchaphon",
  "surname": "Ubonsakka",
  "position": "resource:classroom.management.kmitl.Positions#Student",
  "faculty": "Engineering",
  "major": "Telecommunication",
  "year": "4",
  "tel": "0123456789",
  "email": "student01@kmitl.ac.th",
  "token": 2
}

```

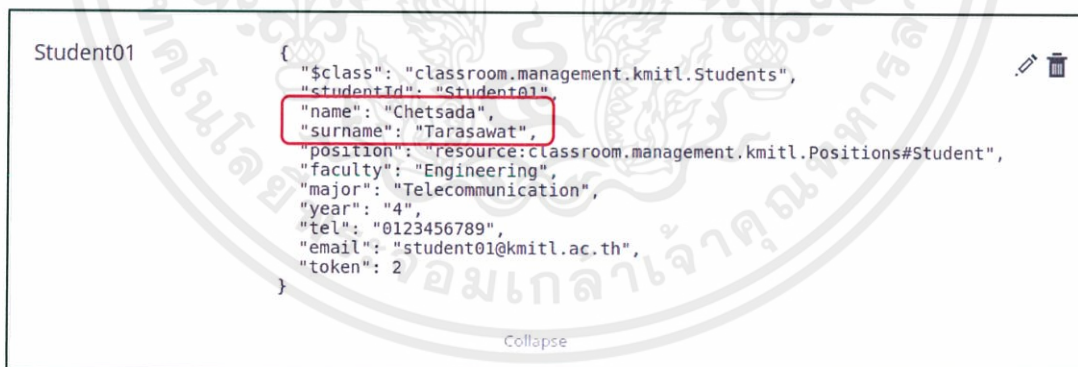
Collapse

รูปที่ 4.7 หน้าแสดงผลการจัดเก็บข้อมูลของ Student01

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 หน้าสำหรับแก้ไขข้อมูลเพื่อทำการอัปเดตข้อมูลของ Student01

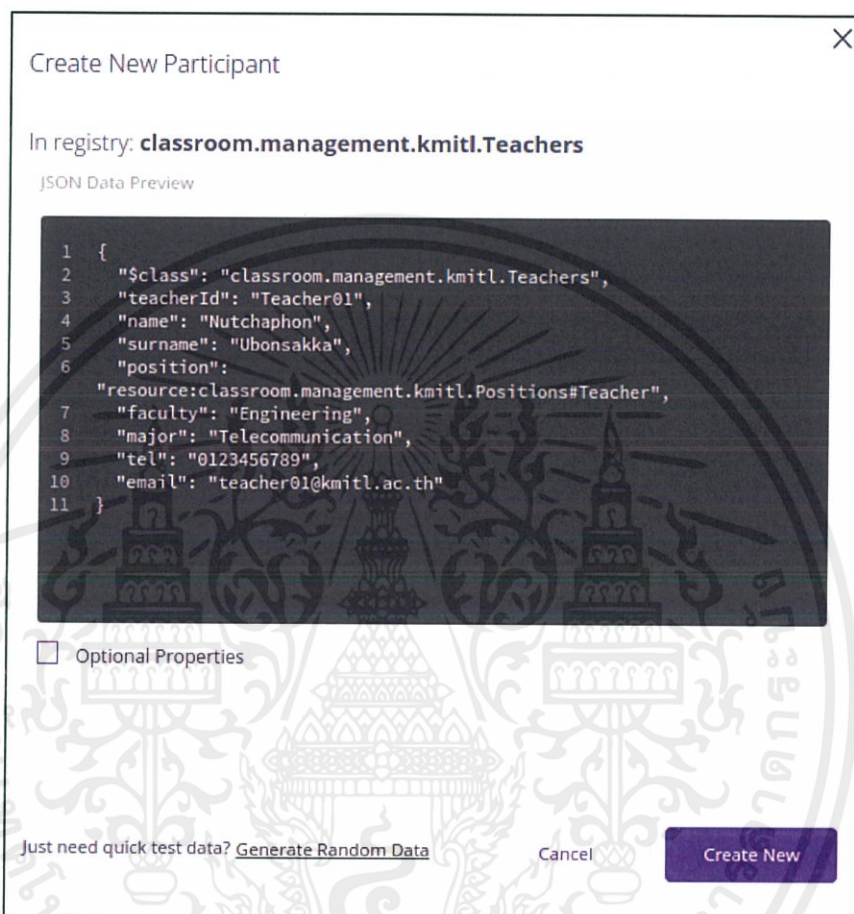


รูปที่ 4.9 หน้าแสดงผลข้อมูลของ Student01 หลังทำการอัปเดตข้อมูล

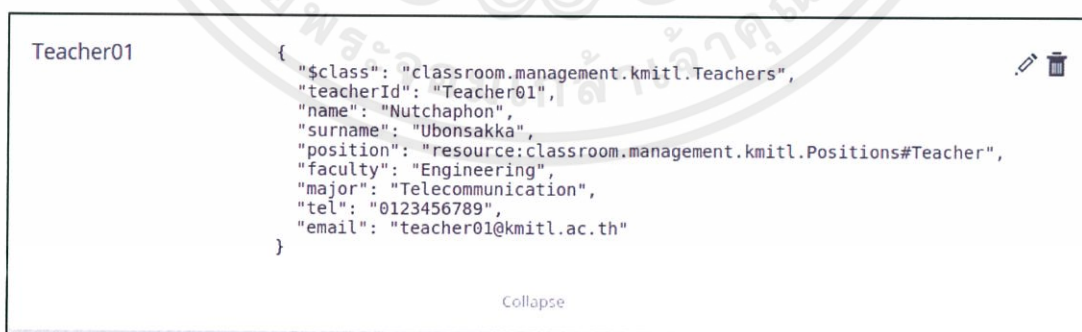
ทำการทดสอบการสร้าง Participants ของ Teachers โดยกำหนดให้ ID ของอาจารย์ คือ Teacher01 ทำการกรอกข้อมูลส่วนตัว ชื่อ-นามสกุล, คณะ, ภาควิชา, เบอร์โทรศัพท์, อีเมล และกำหนดบทบาทให้เป็นอาจารย์ (Teacher) ดังรูปที่ 4.10 จะได้ Participant ของ Teachers ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อ Teacher01 ดังแสดงในรูปที่ 4.11 จากนั้นทำการทดสอบการแก้ไขข้อมูลของ Teachers โดยจะทำการแก้ไขที่ส่วนของชื่อ-นามสกุลของ Teacher01 ดังแสดงในรูปที่ 4.12 และรูปที่ 4.13

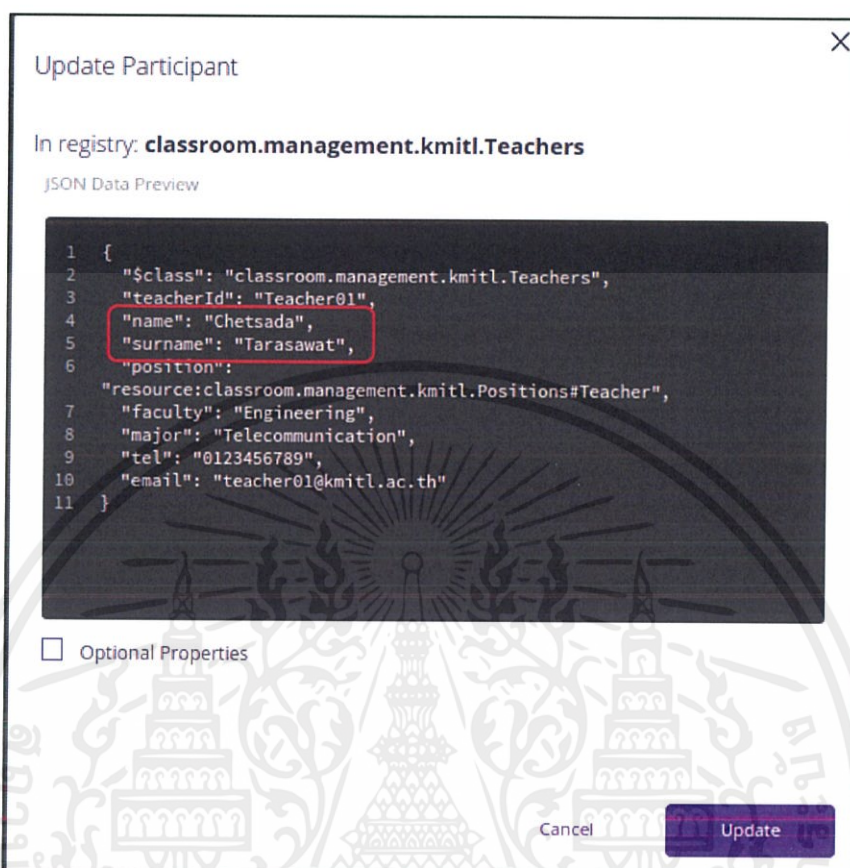


รูปที่ 4.10 หน้าสำหรับกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง Participant ของ Teachers

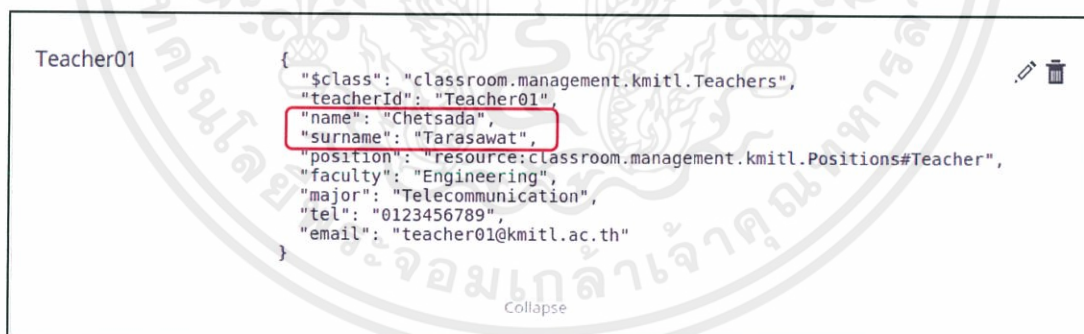


รูปที่ 4.11 หน้าแสดงผลการจัดเก็บข้อมูลของ Teacher01

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



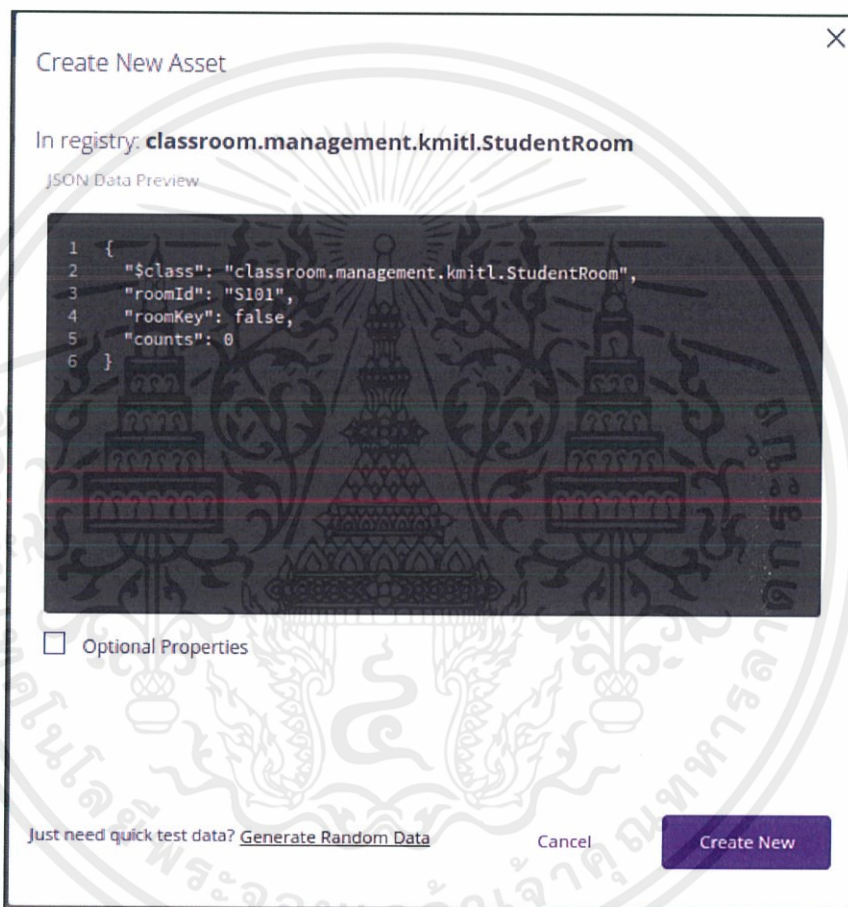
รูปที่ 4.12 หน้าสำหรับแก้ไขข้อมูลเพื่อทำการอัปเดตข้อมูลของ Teacher01



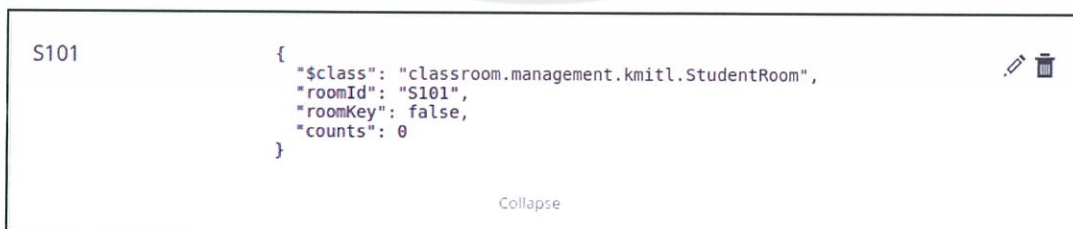
รูปที่ 4.13 หน้าแสดงผลข้อมูลของ Teacher01 หลังทำการอัปเดตข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการสร้าง Assets นั้นจะแบ่งเป็น StudentRoom, TeacherRoom, TSRoom และ Classroom รูปที่ 4.14 แสดงการทดสอบการสร้าง Asset ของ StudentRoom โดยกำหนดให้ ID ของห้อง คือ S101 กำหนดให้ตัวแปร Room Key ที่ใช้ควบคุมการเปิดห้องมีค่าเริ่มต้นเป็น false และการนับจำนวนผู้ใช้งานเริ่มต้นที่ 0 คน เมื่อทำการสร้างจะได้ Asset ของ StudentRoom ที่ชื่อ S101 ดังรูปที่ 4.15



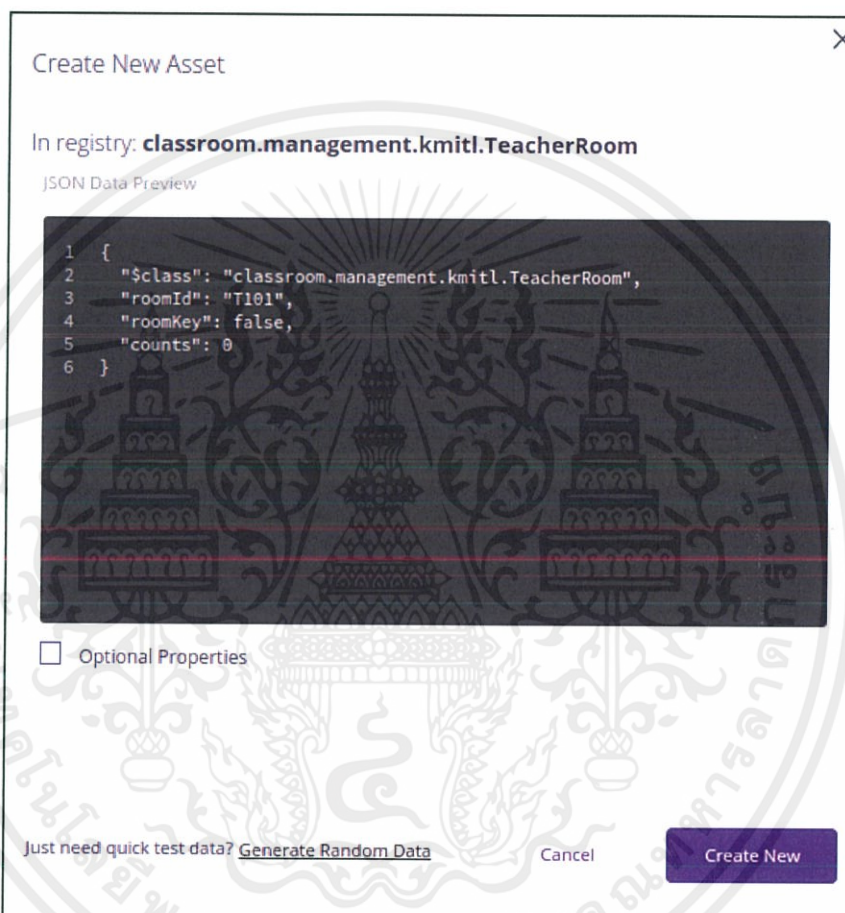
รูปที่ 4.14 หน้าสำหรับกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง Asset ของ StudentRoom



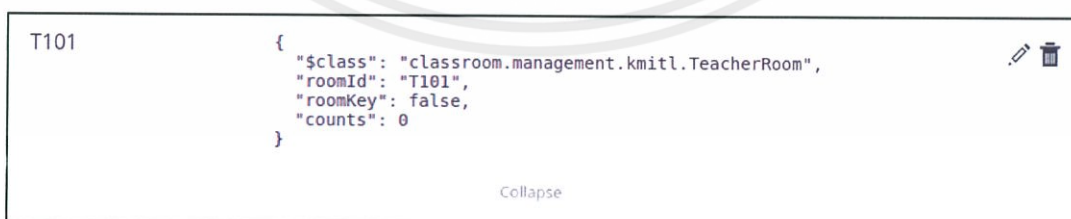
รูปที่ 4.15 หน้าแสดงผลข้อมูลของ StudentRoom ห้อง S101

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบการสร้าง Asset ของ TeacherRoom โดยกำหนด ID ของห้อง คือ T101 กำหนดให้ตัวแปร Room Key ที่ใช้ควบคุมการเปิดห้องมีค่าเริ่มต้นเป็น false และการนับจำนวนผู้ใช้งานเริ่มต้นที่ 0 คน ดังรูปที่ 4.16 เมื่อทำการสร้างจะได้ Asset ของ TeacherRoom ที่ชื่อ T101 ดังรูปที่ 4.17



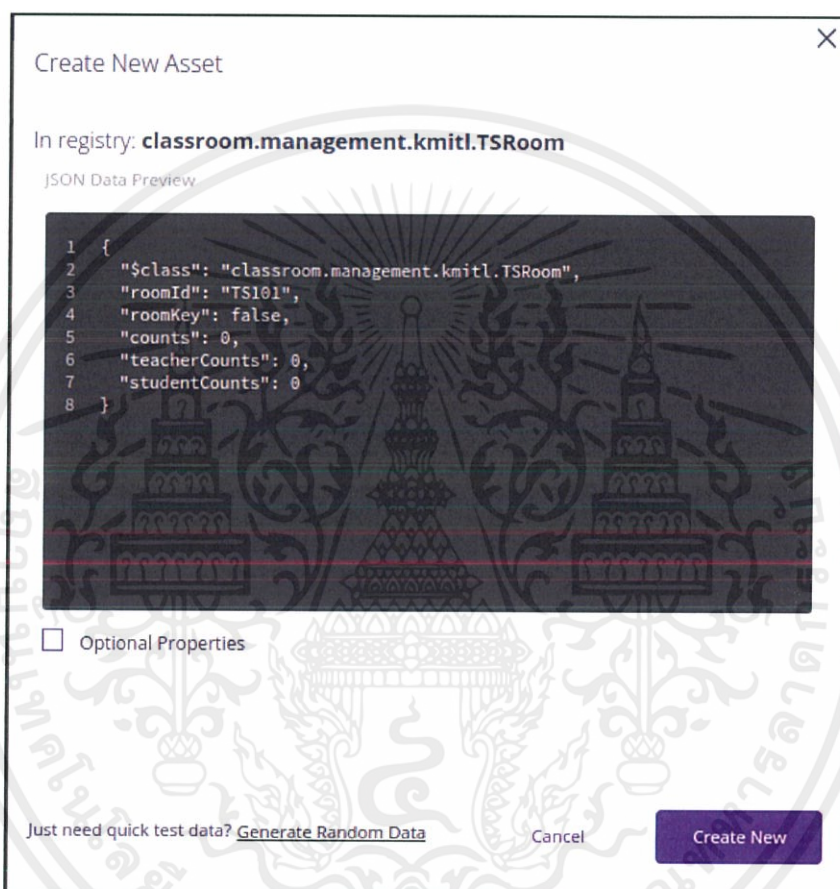
รูปที่ 4.16 หน้าสำหรับกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง Asset ของ TeacherRoom



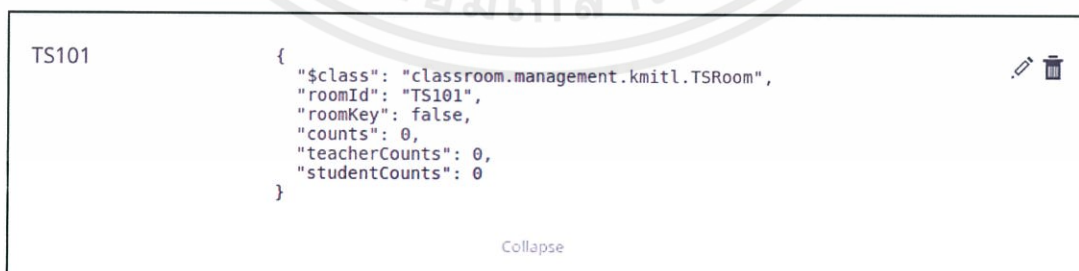
รูปที่ 4.17 หน้าแสดงผลข้อมูลของ TeacherRoom ห้อง T101

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบการสร้าง Asset ของ TSRoom โดยกำหนด ID ของห้อง คือ TS101 กำหนดให้ตัวแปร Room Key ที่ใช้ควบคุมการเปิดห้องมีค่าเริ่มต้นเป็น false และการนับจำนวนผู้ใช้งานเริ่มต้นที่ 0 คน โดยจะมีการนับจำนวนอาจารย์ และนักศึกษาที่ใช้งานแยกจากกัน ดังรูปที่ 4.18 เมื่อทำการสร้างจะได้ Asset ของ TeacherRoom ที่ชื่อ TS101 ดังรูปที่ 4.19



รูปที่ 4.18 หน้าสำหรับกรอกข้อมูลที่จำเป็นเพื่อทำการสร้าง Asset ของ TSRoom



รูปที่ 4.19 หน้าแสดงผลข้อมูลของ TSRoom ห้อง TS101

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบการสร้าง Asset ของ Classroom เพื่อใช้ในการจัดเก็บรายชื่อนักศึกษาในแต่ละรายวิชา โดยกำหนดให้ ID คือ Class01 และทำการกรอกข้อมูลคณะ, ภาควิชา, รายวิชา, กลุ่มเรียน, วันและเวลาทำการสอน และจำนวนนักศึกษาในรายวิชา ดังแสดงในรูปที่ 4.20 เมื่อทำการสร้างจะได้ Asset ของ Classroom ที่ชื่อ Class01 ดังรูปที่ 4.21

Create New Asset

In registry: **classroom.management.kmitl.Classroom**

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.Classroom",
3   "classId": "Class01",
4   "faculty": "Engineering",
5   "major": "Telecommunication",
6   "subject": "Telecom Project",
7   "section": "1",
8   "date": "30/03/2019",
9   "time": "9.00-12.00",
10  "counts": 0
11 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Create New

รูปที่ 4.20 หน้าสำหรับกรอกข้อมูลที่ใช้ในการสร้าง Asset ของ Classroom

Class01

```

{
  "$class": "classroom.management.kmitl.Classroom",
  "classId": "Class01",
  "faculty": "Engineering",
  "major": "Telecommunication",
  "subject": "Telecom Project",
  "section": "1",
  "date": "30/03/2019",
  "time": "9.00-12.00",
  "counts": 0
}

```

Collapse

รูปที่ 4.21 หน้าแสดงผลการจัดเก็บข้อมูลของ Class01

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลการทดสอบการทำงานของ Transactions

การทดลองในส่วนนี้จะทำการทดสอบการทำงานของระบบโดยการส่ง Transactions และบันทึกผลการทำงานผ่านทาง Composer Playground ในรูปที่ 4.22 แสดงการทดสอบการทำงานในการจอง StudentRoom ของ Students (StudentInvokeStudentRoom) โดยทำการระบุข้อมูลที่จำเป็นต้องส่งไปกับ Transaction ได้แก่ ID ของผู้ใช้, ID ของห้องที่ต้องการจอง พร้อมระบุวันและเวลาที่ต้องการทำการจอง เมื่อทำการส่ง Transaction ระบบจะทำการจัดเก็บข้อมูลการจองตามที่ได้ออกแบบไว้ แสดงผลการจัดเก็บข้อมูลการจองของห้อง S101 ดังรูปที่ 4.23 โดยระบบจะทำการจัดเก็บลำดับการจอง, ID ผู้ใช้, ข้อมูลวันและเวลาที่ต้องการทำการจอง รวมถึงบันทึกวันเวลาที่เกิด Transaction ในรูปแบบ timestamp และ Transaction ID ของการจอง ซึ่งเปรียบเสมือนค่า Hash ที่ใช้ในการอ้างอิงตำแหน่งกับ block ต่อไป และแสดงผลการจัดเก็บข้อมูลการจองในส่วนของ Student01 ดังรูปที่ 4.24 โดยทำการจัดเก็บ ID ของห้องที่จอง และลำดับการจองของห้องนั้น พร้อมทั้งหักจำนวน token ที่ใช้ในการจองห้องออกไป ซึ่งหากทำการจองในวันและเวลาที่มีการบันทึกไว้แล้ว ระบบจะไม่สามารถทำการจองได้และส่งข้อความแจ้งเตือนความผิดพลาด ดังแสดงในรูปที่ 4.25

Submit Transaction ✕

Transaction Type StudentInvokeStudentRoom ▼

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.StudentInvokeStudentRoom",
3   "student":
4   "resource:classroom.management.kmitl.Student:#Student01",
5   "roomId": "resource:classroom.management.kmitl.StudentRoom#S101",
6   "structure": {
7     "$class": "classroom.management.kmitl.Structure",
8     "date": "30/03/2019",
9     "time": "9.00-12.00"
10  }
11 }
```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.22 หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction ของการจอง StudentRoom

S101 ✎ ✕

```

{
  "$class": "classroom.management.kmitl.StudentRoom",
  "roomId": "S101",
  "roomKey": false,
  "counts": 1,
  "matching": [
    0
  ],
  "student": [
    "resource:classroom.management.kmitl.Students#Student01"
  ],
  "structure": [
    {
      "$class": "classroom.management.kmitl.Structure",
      "date": "30/03/2019",
      "time": "9.00-12.00"
    }
  ],
  "timestamp": "\"2019-03-24T16:44:02.633Z\"",
  "transactionId": "7940ec168d4a52185dcd2e1d34315249a374c38d681c4ca8f",
  "transaction_p": "7940ec168d4a52185dcd2e1d34315249a374c38d681c4ca8f"
}
```

Collapse

รูปที่ 4.23 หน้าแสดงผลการจัดเก็บข้อมูลการจองของห้อง S101

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Student01      {
                 {"$class": "classroom.management.kmitl.Students",
                  "studentId": "Student01",
                  "name": "Chetsada",
                  "surname": "Tarasawat",
                  "position": "resource:classroom.management.kmitl.Positions#Student",
                  "faculty": "Engineering",
                  "major": "Telecommunication",
                  "year": "4",
                  "tel": "0123456789",
                  "email": "student01@kmitl.ac.th",
                  "token": 1,
                  "room": [
                    "S101#0"
                  ]
                 }

```

Collapse

รูปที่ 4.24 หน้าแสดงผลการจัดเก็บข้อมูลลำดับการจองของ Student01

Submit Transaction ✕

Transaction Type: StudentInvokeStudentRoom

JSON Data Preview

```

1  {
2    "$class": "classroom.management.kmitl.StudentInvokeStudentRoom",
3    "student":
4    "resource:classroom.management.kmitl.Students#Student01",
5    "roomId": "resource:classroom.management.kmitl.StudentRoom#S101",
6    "structure": {
7      "$class": "classroom.management.kmitl.Structure",
8      "date": "30/03/2019",
9      "time": "9.00-12.00"
10   }

```

Optional Properties

Error: Error trying invoke business network with transaction id 371383eddb7ab57586200fae5a962f6dccb33cdc8eee7a6c801ca98fb28bba. Error: No valid responses from any peers. Response from attempted peer comms was an error: Error: transaction returned with failure: Error: ##### This datetime has already exist. #####

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.25 หน้าแสดงผลเมื่อไม่สามารถทำการจอง StudentRoom ในวันเวลาที่ระบุได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการทดสอบการทำงานของ Transaction ในการแก้ไขข้อมูลวันและเวลาที่ ต้องการทำการจอง StudentRoom ของ Students (StudentEditStudentRoom) ในรูปที่ 4.26 โดยทำการระบุข้อมูล ID ผู้ใช้, ID ของห้อง และลำดับการจอง พร้อมระบุวันและเวลาที่ต้องการจะ เลื่อนการจอง เมื่อทำการส่ง Transaction ระบบจะทำการแก้ไขข้อมูลวันและเวลาที่ การจอง ของห้อง S101 ดังแสดงในรูปที่ 4.27 และบันทึก timestamp, Transaction ID ของการแก้ไขการจอง และบันทึก Transaction ID ของ Transaction ก่อนหน้าเพื่อเป็นการอ้างอิงตำแหน่งของ block และเช่นเดียวกับการจองห้อง หากวันและเวลาที่ต้องการเลื่อนนั้นตรงกับวันและเวลาที่มีการ จองอยู่ในระบบแล้ว จะไม่สามารถทำการจองได้ ดังรูปที่ 4.28

Submit Transaction

Transaction Type StudentEditStudentRoom

JSON Data Preview

```

1  {
2    "$class": "classroom.management.kmitl.StudentEditStudentRoom",
3    "student":
4    "resource:classroom.management.kmitl.Students#Student01",
5    "roomId": "resource:classroom.management.kmitl.StudentRoom#S101",
6    "counts": 0,
7    "structure": {
8      "$class": "classroom.management.kmitl.Structure",
9      "date": "30/03/2019",
10     "time": "13.00-16.00"
11  }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.26 หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction แก้ไขการจองของ StudentRoom

```

S101
{
  "$class": "classroom.management.kmitl.StudentRoom",
  "roomId": "S101",
  "roomKey": false,
  "counts": 1,
  "matching": [
    0
  ],
  "student": [
    "resource:classroom.management.kmitl.Students#Student01"
  ],
  "structure": [
    {
      "$class": "classroom.management.kmitl.Structure",
      "date": "30/03/2019",
      "time": "13.00-16.00"
    }
  ],
  "timestamp": "\"2019-03-24T16:45:49.264Z\"",
  "transactionId": "a88867790d47d130ff151dbbe55309a36e546d71139758a21",
  "transaction_p": "7940ec168d4a52185dcd2e1d34315249a374c38d681c4ca8f"
}

```

รูปที่ 4.27 หน้าแสดงผลการแก้ไขข้อมูลการจองของห้อง S101

Submit Transaction

Transaction Type: StudentEditStudentRoom

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.StudentEditStudentRoom",
3   "student":
4     "resource:classroom.management.kmitl.Students#Student01",
5   "roomId": "resource:classroom.management.kmitl.StudentRoom#S101",
6   "counts": 0,
7   "structure": {
8     "$class": "classroom.management.kmitl.Structure",
9     "date": "30/03/2019",
10    "time": "13.00-16.00"
11  }

```

Optional Properties

Error: Error trying invoke business network with transaction id 94e61c61b362c517763c58e648c8f6cb3ef954ce40a02a6b060a12e782d1cab8. Error: No valid responses from any peers. response from attempted peer comms was an error: Error: transaction returned with failure: Error: ##### This datetime has already exist. #####

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.28 หน้าแสดงผลเมื่อไม่สามารถเลื่อนการจอง StudentRoom ไปยังวันเวลาที่ระบุได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบการทำงานของ Transaction ในการยกเลิกการจอง StudentRoom ของ Students (StudentRevokeStudentRoom) โดยทำการระบุ ID ผู้ใช้, ID ของห้อง และลำดับการจอง ดังรูปที่ 4.29 เมื่อทำการส่ง Transaction ระบบจะลบข้อมูลลำดับการจอง, ID ของผู้จอง, ข้อมูลวันและเวลาที่เคยทำการจองออกจากห้อง S101 ดังรูปที่ 4.30 และทำการบันทึก timestamp และ Transaction ID ทั้งเก่าและใหม่ รวมถึงทำการลบข้อมูลห้องที่กำลังจองออกจาก Student01 ดังรูปที่ 4.31

Submit Transaction

Transaction Type StudentRevokeStudentRoom

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.StudentRevokeStudentRoom",
3   "student":
4     "resource:classroom.management.kmitl.Students#Student01",
5   "roomId": "resource:classroom.management.kmitl.StudentRoom#S101",
6   "counts": 0
7 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.29 หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction การยกเลิกการจอง StudentRoom

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

S101
{
  "$class": "classroom.management.kmitl.StudentRoom",
  "roomId": "S101",
  "roomKey": false,
  "counts": 1,
  "matching": [],
  "student": [],
  "structure": [],
  "timestamp": "\"2019-03-24T16:48:03.921Z\"",
  "transactionId": "6ca66a13cc1b19ca69b81c6062d1db526a8b63713bab22d32ct",
  "transaction_p": "a88867790d47d130ff151dbbe55309a36e546d71139758a21cf"
}

```

Collapse

รูปที่ 4.30 หน้าแสดงผลการลบข้อมูลการจองของห้อง S101

```

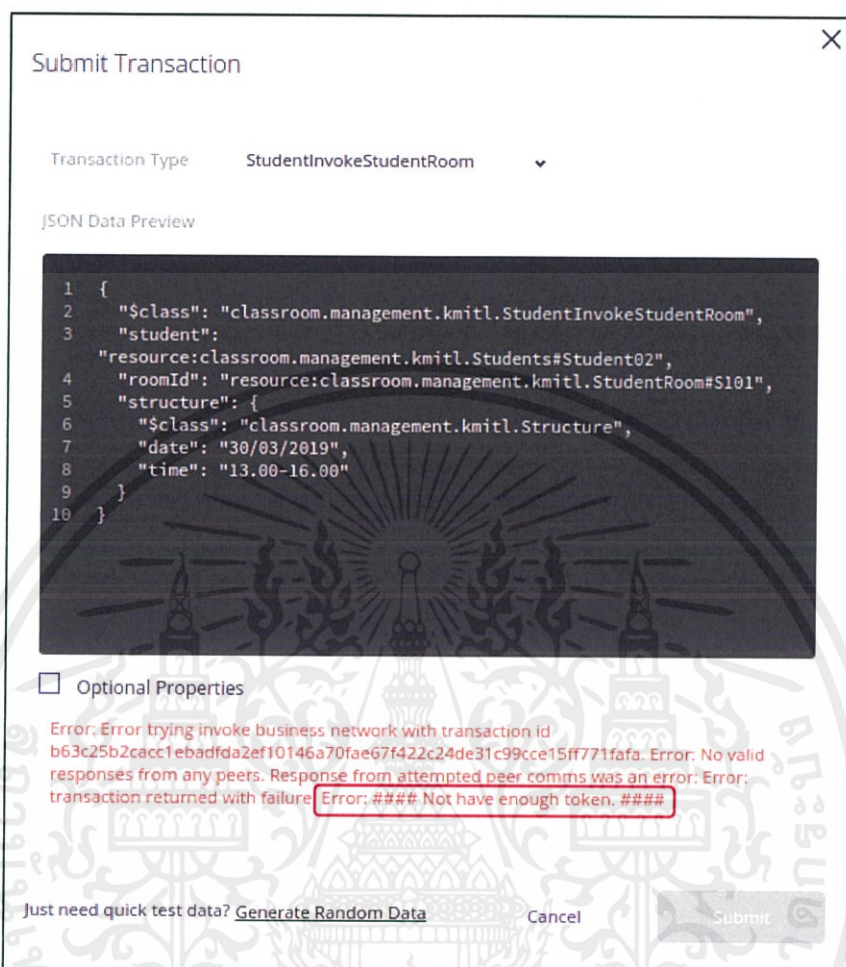
Student01
{
  "$class": "classroom.management.kmitl.Students",
  "studentId": "Student01",
  "name": "Chetsada",
  "surname": "Tarasawat",
  "position": "resource:classroom.management.kmitl.Positions#Student",
  "faculty": "Engineering",
  "major": "Telecommunication",
  "year": "4",
  "tel": "0123456789",
  "email": "student01@kmitl.ac.th",
  "token": 1,
  "room": []
}

```

Collapse

รูปที่ 4.31 หน้าแสดงผลการลบข้อมูลลำดับการจองของ Student01

หากนักศึกษาไม่มี token ที่ใช้สำหรับทำการจองจะไม่สามารถทำการจองได้ และระบบจะส่งข้อความแจ้งเตือน ดังรูปที่ 4.32 จำเป็นจะต้องทำการเพิ่ม token โดยอาจารย์หรือผู้ดูแลระบบ ในรูปที่ 4.33 แสดงการทดสอบการทำงานของ Transaction ในการเพิ่ม token ที่ใช้ในการจองห้องเรียนของนักศึกษา (AddToken) โดยทำการระบุ ID ของ Students ที่ต้องการเพิ่ม token และจำนวน token ที่ต้องการเพิ่ม และระบบจะทำการเพิ่ม token ให้กับ Students ดังแสดงในรูปที่ 4.34



รูปที่ 4.32 หน้าแสดงผลการทำงานเมื่อมี token ไม่เพียงพอสำหรับการจอง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Submit Transaction

Transaction Type AddToken

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.AddToken",
3   "student":
4   "resource:classroom.management.kmitl.Students#Student01",
5   "token": 1
6 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.33 หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction ของการเพิ่ม token

```

Student01 {
  "$class": "classroom.management.kmitl.Students",
  "studentId": "Student01",
  "name": "Chetsada",
  "surname": "Tarasawat",
  "position": "resource:classroom.management.kmitl.Positions#Student",
  "faculty": "Engineering",
  "major": "Telecommunication",
  "year": "4",
  "tel": "0123456789",
  "email": "student01@kmitl.ac.th",
  "token": 2,
  "room": []
}

```

Collapse

รูปที่ 4.34 หน้าแสดงผลของการเพิ่ม token ของ Student01

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบการทำงานของ Transaction ในการปลดล็อคประตูของ StudentRoom (UnlockStudentRoom) แสดงดังรูปที่ 4.35 โดยทำการระบุ ID ของห้องที่ต้องการ ระบบจะทำการเปลี่ยนค่าตัวแปร Room Key ของห้องให้มีค่าเป็น true ดังรูปที่ 4.36 เพื่อใช้งานร่วมกับไมโครคอนโทรลเลอร์ในการควบคุมกลอนแม่เหล็กไฟฟ้าให้หยุดการทำงาน และทดสอบการทำงานของ Transaction ในการล็อคประตูของ StudentRoom (LockStudentRoom) ในรูปที่ 4.37 โดยทำการระบุ ID ของห้องเช่นเดียวกัน ระบบจะทำการเปลี่ยนค่าตัวแปร Room Key ให้มีค่าเป็น false ดังรูปที่ 4.38 เพื่อให้กลอนแม่เหล็กไฟฟ้ากลับมาทำงานตามปกติ

Submit Transaction

Transaction Type: UnlockStudentRoom

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.UnlockStudentRoom",
3   "roomId": "resource:classroom.management.kmitl.StudentRoom:S101"
4 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.35 หน้าสำหรับระบุข้อมูลที่ส่ง Transaction ในการปลดล็อคประตูของ StudentRoom

```

S101
{
  "$class": "classroom.management.kmitl.StudentRoom",
  "roomId": "S101",
  "roomKey": true,
  "counts": 1,
  "matching": [],
  "student": [],
  "structure": [],
  "timestamp": "\"2019-03-24T16:50:55.122Z\"",
  "transactionId": "fa8368e770d3e7fe6bed50f0a0688195f369f9893a76ed1a1af",
  "transaction_p": "6ca66a13cc1b19ca69b81c6062d1db526a8b63713bab22d32ct"
}

```

Collapse

รูปที่ 4.36 หน้าแสดงผลการปลดล็อคประตูของห้อง S101

Submit Transaction
✕

Transaction Type: LockStudentRoom

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.LockStudentRoom",
3   "roomId": "resource:classroom.management.kmitl.StudentRoom#S101"
4 }

```

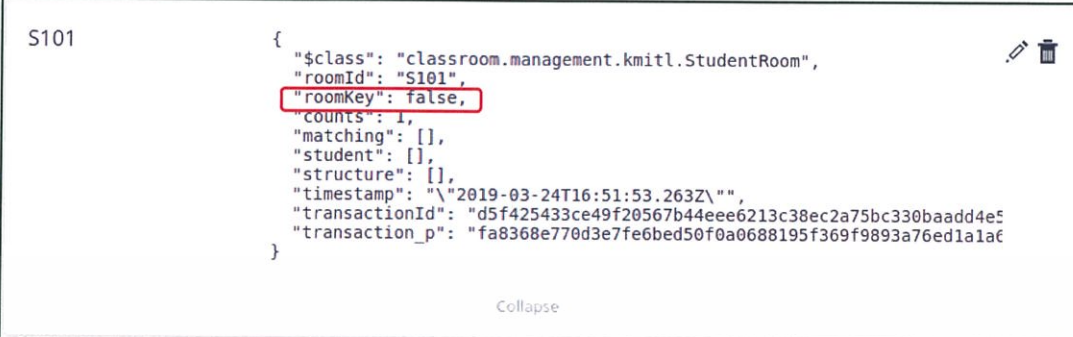
Optional Properties

Just need quick test data? [Generate Random Data](#)

Cancel Submit

รูปที่ 4.37 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ในการล็อคประตูของ StudentRoom

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

S101      {
           {"$class": "classroom.management.kmitl.StudentRoom",
            "roomId": "S101",
            "roomKey": false,
            "counts": 1,
            "matching": [],
            "student": [],
            "structure": [],
            "timestamp": "\2019-03-24T16:51:53.263Z",
            "transactionId": "d5f425433ce49f20567b44eee6213c38ec2a75bc330baadd4e5",
            "transaction_p": "fa8368e770d3e7fe6bed50f0a06688195f369f9893a76ed1a1af"}
           }
           Collapse
  
```

รูปที่ 4.38 หน้าแสดงผลการลือคประตูของห้อง S101

ในรูปที่ 4.39 แสดงการทดสอบการทำงานในการจอง TeacherRoom ของ Teachers (TeacherInvokeTeacherRoom) โดยทำการระบุข้อมูล ID ของผู้ใช้, ID ของห้องที่ต้องการจอง พร้อมระบุวันและเวลาที่ต้องการทำการจอง เมื่อทำการส่ง Transaction ระบบจะทำการจัดเก็บ ลำดับการจอง, ID ผู้ใช้, ข้อมูลวันและเวลาที่ต้องการการจอง รวมถึงจัดเก็บ timestamp และ Transaction ID ของการจอง ลงในข้อมูลของห้อง T101 ดังแสดงในรูปที่ 4.40 และในส่วนของ Teacher01 จะทำการจัดเก็บ ID ของห้องที่จอง และลำดับการจองของห้องนั้น ดังรูปที่ 4.41

Submit Transaction ✕

Transaction Type TeacherInvokeTeacherRoom ▼

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.TeacherInvokeTeacherRoom",
3   "teacher":
4   "resource:classroom.management.kmitl.Teachers#Teacher01",
5   "roomId": "resource:classroom.management.kmitl.teacherroom#T101",
6   "structure": {
7     "$class": "classroom.management.kmitl.Structure",
8     "date": "30/03/2019",
9     "time": "9.00-12.00"
10  }
  }
```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.39 หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction การจองของ TeacherRoom

T101 ✎ ✕

```

{
  "$class": "classroom.management.kmitl.TeacherRoom",
  "roomId": "T101",
  "roomKey": false,
  "counts": 1,
  "matching": [
    0
  ],
  "teacher": [
    "resource:classroom.management.kmitl.Teachers#Teacher01"
  ],
  "structure": [
    {
      "$class": "classroom.management.kmitl.Structure",
      "date": "30/03/2019",
      "time": "9.00-12.00"
    }
  ],
  "timestamp": "\"2019-03-24T16:54:30.597Z\"",
  "transactionId": "79d00d1ec19a4455f4a08880a74b8b1e356a2bcb1c1078c19",
  "transaction_p": "79d00d1ec19a4455f4a08880a74b8b1e356a2bcb1c1078c19"
}
```

Collapse

รูปที่ 4.40 หน้าแสดงผลการจัดเก็บข้อมูลการจองของห้อง T101

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Teacher01      {
                 {"$class": "classroom.management.kmitl.Teachers",
                  "teacherId": "Teacher01",
                  "name": "Chetsada",
                  "surname": "Tarasawat",
                  "position": "resource: classroom.management.kmitl.Positions#Teacher",
                  "faculty": "Engineering",
                  "major": "Telecommunication",
                  "tel": "0123456789",
                  "email": "teacher01@kmitl.ac.th",
                  "room": [
                    "T101#0"
                  ]
                 }

```

Collapse

รูปที่ 4.41 หน้าแสดงผลการจัดเก็บข้อมูลลำดับการจองของ Teacher01

ทดสอบการทำงานของ Transaction ในการแก้ไขวันและเวลาที่ต้องการทำการจอง TeacherRoom ของ Teachers (TeacherEditTeacherRoom) แสดงดังรูปที่ 4.42 โดยทำการระบุ ID ผู้ใช้, ID ห้อง และลำดับการจอง พร้อมระบุวันและเวลาที่ต้องการจะเลื่อนการจอง เมื่อทำการส่ง Transaction ระบบจะทำการแก้ไขข้อมูลวันและเวลาที่ต้องการจองของห้อง T101 ดังแสดงในรูปที่ 4.43 รวมทั้งบันทึก timestamp, Transaction ID และ Transaction ID ก่อนหน้าเพื่อใช้อ้างอิงตำแหน่งของ block

Submit Transaction

Transaction Type **TeacherEditTeacherRoom**

JSON Data Preview

```

1  {
2    "$class": "classroom.management.kmitl.TeacherEditTeacherRoom",
3    "teacher":
4    "resource:classroom.management.kmitl.Teachers#Teacher01",
5    "roomId": "resource:classroom.management.kmitl.TeacherRoom#T101",
6    "counts": 0,
7    "structure": {
8      "$class": "classroom.management.kmitl.Structure",
9      "date": "30/03/2019",
10     "time": "13.00-16.00"
11  }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.42 หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction แก้ไขการจองของ TeacherRoom

T101

```

{
  "$class": "classroom.management.kmitl.TeacherRoom",
  "roomId": "T101",
  "roomKey": false,
  "counts": 1,
  "matching": [
    0
  ],
  "teacher": [
    "resource:classroom.management.kmitl.Teachers#Teacher01"
  ],
  "structure": {
    {
      "$class": "classroom.management.kmitl.Structure",
      "date": "30/03/2019",
      "time": "13.00-16.00"
    }
  ],
  "timestamp": "\"2019-03-24T16:56:09.118Z\"",
  "transactionId": "c9dd55cca90a4cf77a9d8583c7b21df2964a58e5c66abb03c1",
  "transaction_p": "79d00d1ec19a4455f4a08880a74b8b1e356a2bcb1c1078c19"
}

```

Collapse

รูปที่ 4.43 หน้าแสดงผลการแก้ไขข้อมูลการจองของห้อง T101

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบการทำงานของ Transaction ในการยกเลิกการจอง TeacherRoom ของ Teachers (TeacherRevokeTeacherRoom) โดยทำการระบุ ID ผู้ใช้, ID ของห้อง และลำดับการจอง ดังรูปที่ 4.44 เมื่อทำการส่ง Transaction ระบบจะลบข้อมูลลำดับการจอง, ID ของผู้จอง, ข้อมูลวันและเวลาที่เคยทำการจองออกจากห้อง T101 ดังรูปที่ 4.45 และทำการบันทึก timestamp และ Transaction ID รวมถึงทำการลบข้อมูลห้องที่กำลังจองออกจาก Teacher01 ดังรูปที่ 4.46

Submit Transaction

Transaction Type: TeacherRevokeTeacherRoom

{JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.TeacherRevokeTeacherRoom",
3   "teacher":
4     "resource:classroom.management.kmitl.Teachers#Teacher01",
5   "roomId": "resource:classroom.management.kmitl.TeacherRoom#T101",
6   "counts": 0
7 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.44 หน้าสำหรับระบุข้อมูลที่ใช้ในการส่ง Transaction การยกเลิกการจอง TeacherRoom

T101

```

{
  "$class": "classroom.management.kmitl.TeacherRoom",
  "roomId": "T101",
  "roomKey": false,
  "counts": 1,
  "matching": [],
  "teacher": [],
  "structure": [],
  "timestamp": "\"2019-03-24T16:57:14.496Z\"",
  "transactionId": "a163a6d85a3e8499a9be4676f73d665429e9f12ef08c5e52d3f",
  "transaction_p": "c9dd55cca90a4cf77a9d8583c7b21df2964a58e5c66abb03c6e"
}

```

Collapse

รูปที่ 4.45 หน้าแสดงผลการลบข้อมูลการจองของห้อง T101

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Teacher01      {
                  {"$class": "classroom.management.kmitl.Teachers",
                   "teacherId": "Teacher01",
                   "name": "Chetsada",
                   "surname": "Tarasawat",
                   "position": "resource:classroom.management.kmitl.Positions#Teacher",
                   "faculty": "Engineering",
                   "major": "Telecommunication",
                   "tel": "0123456789",
                   "email": "teacher01@kmitl.ac.th",
                   "room": []
                  }
                }
  
```

Collapse

รูปที่ 4.46 หน้าแสดงผลการลบข้อมูลลำดับการจองของ Teacher01

ทดสอบการทำงานของ Transaction ในการปลดล็อคประตูของ TeacherRoom (UnlockTeacherRoom) แสดงดังรูปที่ 4.47 โดยทำการระบุ ID ของห้อง ระบบจะทำการเปลี่ยนค่าตัวแปร Room Key ของห้องให้มีค่าเป็น true ดังรูปที่ 4.48 เพื่อให้กลอนแม่เหล็กไฟฟ้าหยุดการทำงาน และสามารถเปิดประตูได้ และทดสอบการทำงานของ Transaction ในการล็อคประตูของ TeacherRoom (LockTeacherRoom) ในรูปที่ 4.49 โดยทำการระบุ ID ของห้องเช่นเดียวกัน ระบบจะทำการเปลี่ยนค่าตัวแปร Room Key ให้มีค่าเป็น false ดังรูปที่ 4.50 เพื่อให้กลอนแม่เหล็กไฟฟ้ากลับมาทำงานตามปกติ

Submit Transaction

Transaction Type: UnlockTeacherRoom

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.UnlockTeacherRoom",
3   "roomId": "resource:classroom.management.kmitl.TeacherRoom#T101"
4 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.47 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ในการปลดล็อคประตูของ TeacherRoom

T101

```

{
  "$class": "classroom.management.kmitl.TeacherRoom",
  "roomId": "T101",
  "roomKey": true,
  "counts": 1,
  "matching": [],
  "teacher": [],
  "structure": [],
  "timestamp": "\"2019-03-24T16:57:46.156Z\"",
  "transactionId": "b82dd283b01516865cfcb426d0549a40a64b7305bd8cea03cf6",
  "transaction_p": "a163a6d85a3e8499a9be4676f73d665429e9f12ef08c5e52d3f"
}

```

Collapse

รูปที่ 4.48 หน้าแสดงผลการปลดล็อคประตูของห้อง T101

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Submit Transaction

Transaction Type **LockTeacherRoom**

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.LockTeacherRoom",
3   "roomId": "resource:classroom.management.kmitl.TeacherRoom#T101"
4 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) [Cancel](#) [Submit](#)

รูปที่ 4.49 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ในการล็อคประตูของ TeacherRoom

T101

```

{
  "$class": "classroom.management.kmitl.TeacherRoom",
  "roomId": "T101",
  "roomKey": false,
  "counts": 1,
  "matching": [],
  "teacher": [],
  "structure": [],
  "timestamp": "\2019-03-24T16:58:08.238Z",
  "transactionId": "a654762b7fec64da6ba8454696d2b9c5e96e75aa7e771b3def1",
  "transaction_p": "b82dd283b01516865cfc426d0549a40a64b7305bd8cea03cfe"
}

```

Collapse

รูปที่ 4.50 หน้าแสดงผลการล็อคประตูของห้อง T101

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบการทำงานของ Transaction ในการจอง TSRoom ของ Students (StudentInvokeTSRoom) โดยทำการระบุข้อมูลที่ส่งไปกับ Transaction คือ ID ของผู้ใช้, ID ของห้องที่ต้องการจอง, วันและเวลาที่ต้องการทำการจอง ดังรูปที่ 4.51 เมื่อทำการส่ง Transaction ระบบจะทำการจัดเก็บข้อมูลลำดับการจอง, ID ผู้ใช้, วันเวลาที่ต้องการจอง, timestamp, Transaction ID ไปยังห้อง TS101 ดังรูปที่ 4.52 และทำการจัดเก็บลำดับการจองไปยัง Student02 ดังรูปที่ 4.53

Submit Transaction

Transaction Type StudentInvokeTSRoom

JSON Data Preview

```

1  {
2    "$class": "classroom.management.kmitl.StudentInvokeTSRoom",
3    "student":
4      "resource:classroom.management.kmitl.Students#Student02",
5    "roomId": "resource:classroom.management.kmitl.TSRoom#TS101",
6    "structure": {
7      "$class": "classroom.management.kmitl.Structure",
8      "date": "30/03/2019",
9      "time": "9.00-12.00"
10   }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) [Cancel](#) [Submit](#)

รูปที่ 4.51 หน้าสำหรับระบุข้อมูลที่ผู้ใช้ส่ง Transaction ในการจอง TSRoom ของ Students

```

TS101
{
  "$class": "classroom.management.kmitl.TSRoom",
  "roomId": "TS101",
  "roomKey": false,
  "counts": 1,
  "teacherCounts": 0,
  "studentCounts": 1,
  "studentMatching": [
    0
  ],
  "student": [
    "resource:classroom.management.kmitl.Students#Student02"
  ],
  "studentStructure": [
    {
      "$class": "classroom.management.kmitl.Structure",
      "date": "30/03/2019",
      "time": "9.00-12.00"
    }
  ],
  "timestamp": "\"2019-03-25T18:36:41.721Z\"",
  "transactionId": "d348788c7676165fcb55b75269c37971c9bcb8a4f53e210fa",
  "transaction_p": "d348788c7676165fcb55b75269c37971c9bcb8a4f53e210fa"
}

```

รูปที่ 4.52 หน้าแสดงผลการจัดเก็บข้อมูลการจองห้อง TS101 ของ Student02

```

Student02
{
  "$class": "classroom.management.kmitl.Students",
  "studentId": "Student02",
  "name": "Nutchaphon",
  "surname": "Ubonsakka",
  "position": "resource:classroom.management.kmitl.Positions#Student",
  "faculty": "Engineering",
  "major": "Telecommunication",
  "tel": "0123456789",
  "email": "student02@kmitl.ac.th",
  "token": 1,
  "room": [
    "TS101#0"
  ]
}

```

รูปที่ 4.53 หน้าแสดงผลการจัดเก็บข้อมูลลำดับการจองของ Student02

ทดสอบการทำงานของ Transaction ในการแก้ไขการจอง TSRoom ของ Students (StudentEditTSRoom) โดยทำการระบุข้อมูล ID ของผู้จอง, ID ของห้องที่ทำการจอง, ลำดับการจอง, วันและเวลาที่ต้องการเลื่อนการจอง ดังรูปที่ 4.54 เมื่อทำการส่ง Transaction ระบบจะทำการแก้ไขข้อมูลวันเวลาที่ทำการจองของห้อง TS101 รวมถึงบันทึก timestamp และ Transaction ID ดังแสดงในรูปที่ 4.55

Submit Transaction

Transaction Type StudentEditTSRoom

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.StudentEditTSRoom",
3   "student": "resource:classroom.management.kmitl.Teachers#0121",
4   "roomId": "resource:classroom.management.kmitl.TSRoom#TS101",
5   "counts": 0,
6   "structure": {
7     "$class": "classroom.management.kmitl.Structure",
8     "date": "30/03/2019",
9     "time": "13.00-16.00"
10  }
11 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.54 หน้าสำหรับระบุข้อมูลที่ใส่ส่ง Transaction แก้ไขการจอง TSRoom ของ Students

TS101

```

{
  "$class": "classroom.management.kmitl.TSRoom",
  "roomId": "TS101",
  "roomKey": false,
  "counts": 1,
  "teacherCounts": 0,
  "studentCounts": 1,
  "studentMatching": [
    0
  ],
  "student": [
    "resource:classroom.management.kmitl.Students#Student02"
  ],
  "studentStructure": [
    {
      "$class": "classroom.management.kmitl.Structure",
      "date": "30/03/2019",
      "time": "13.00-16.00"
    }
  ],
  "timestamp": "\"2019-03-25T18:40:34.947Z\"",
  "transactionId": "380dc96525fad463f4feff2ac345dd45b9656b6946af82aa3f",
  "transaction_p": "d348788c7676165fcb55b75269c37971c9bcb8a4f53e210fa"
}

```

Collapse

รูปที่ 4.55 หน้าแสดงผลการแก้ไขข้อมูลการจองห้อง TS101 ของ Student02

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบการทำงานของ Transaction ในการยกเลิกการจอง TSRoom ของ Students (StudentRevokeTSRoom) ทำการระบุข้อมูลที่จำเป็น ID ผู้ใช้, ID ของห้อง และลำดับการจอง ดังรูปที่ 4.56 ระบบจะทำการลบข้อมูลลำดับการจอง, ID ผู้จอง, วันที่ทำการจองออกจากห้อง TS101 รวมถึงบันทึก timestamp และ Transaction ID แสดงดังรูปที่ 4.57 และลบข้อมูลลำดับการจองออกจาก Student02 ดังรูปที่ 4.58

Submit Transaction

Transaction Type StudentRevokeTSRoom

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.StudentRevokeTSRoom",
3   "student":
4     "resource:classroom.management.kmitl.Students#Student02",
5     "roomId": "resource:classroom.management.kmitl.TSRoom#TS101",
6     "counts": 0
7 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.56 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ยกเลิกการจอง TSRoom ของ Students

```

TS101
{
  "$class": "classroom.management.kmitl.TSRoom",
  "roomId": "TS101",
  "roomKey": false,
  "counts": 1,
  "teacherCounts": 0,
  "studentCounts": 1,
  "studentMatching": [],
  "student": [],
  "studentStructure": [],
  "timestamp": "\"2019-03-25T18:44:23.248Z\"",
  "transactionId": "db842394acdfa014949e6bcf0e6f3b90bdd970e1ab8f3369b",
  "transaction_p": "380dc96525fad463f4feff2ac345dd45b9656b6946af82aa3f"
}

```

Collapse

รูปที่ 4.57 หน้าแสดงผลการลบข้อมูลการจองห้อง TS101 ของ Student02

```

Student02
{
  "$class": "classroom.management.kmitl.Students",
  "studentId": "Student02",
  "name": "Nutchaphon",
  "surname": "Ubonsakka",
  "position": "resource: classroom.management.kmitl.Positions#Student",
  "faculty": "Engineering",
  "major": "Telecommunication",
  "tel": "0123456789",
  "email": "student02@kmitl.ac.th",
  "token": 1,
  "room": []
}

```

Collapse

รูปที่ 4.58 หน้าแสดงผลการลบข้อมูลลำดับการจองของ Student02

ทดสอบการทำงานของ Transaction ในการจอง TSRoom ของ Teachers (TeacherInvokeTSRoom) ในรูปที่ 4.59 โดยทำการระบุข้อมูล ID ผู้ใช้, ID ของห้องที่ต้องการจอง และวันเวลาที่ต้องการทำการจอง เมื่อส่ง Transaction ระบบจะทำการจัดเก็บข้อมูลลำดับการจอง, ID ผู้ใช้, วันและเวลาที่ต้องการจอง, timestamp และ Transaction ID ไปยังห้อง TS101 แสดงดังรูปที่ 4.60 และทำการจัดเก็บข้อมูลลำดับการจองไปยัง Teacher02 ดังรูปที่ 4.61

Submit Transaction
✕

Transaction Type TeacherInvokeTSRoom ▼

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.TeacherInvokeTSRoom",
3   "teacher":
4   "resource:classroom.management.kmitl.Teacher#Teacher02",
5   "roomId": "resource:classroom.management.kmitl.TSRoom#TS101",
6   "structure": {
7     "$class": "classroom.management.kmitl.Structure",
8     "date": "30/03/2019",
9     "time": "9.00-12.00"
10  }
  }
```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.59 หน้าสำหรับระบุข้อมูลที่ใส่ส่ง Transaction ในการจอง TSRoom ของ Teachers

TS101
✎ 🗑

```

{
  "$class": "classroom.management.kmitl.TSRoom",
  "roomId": "TS101",
  "roomKey": false,
  "counts": 2,
  "teacherCounts": 1,
  "studentCounts": 1,
  "teacherMatching": [
    0
  ],
  "studentMatching": [],
  "teacher": [
    "resource:classroom.management.kmitl.Teachers#Teacher02"
  ],
  "student": [],
  "teacherStructure": [
    {
      "$class": "classroom.management.kmitl.Structure",
      "date": "30/03/2019",
      "time": "9.00-12.00"
    }
  ],
  "studentStructure": [],
  "timestamp": "\"2019-03-25T18:46:45.479Z\"",
  "transactionID": "a643cf76b71f88cf8320b0ad7b62856c18d98bbfd92103697f",
  "transaction_p": "db842394acdfa014949e6bcf0e6f3b90bdd970e1ab8f3369b"
}
```

Collapse

รูปที่ 4.60 หน้าแสดงผลการจัดเก็บข้อมูลการจองห้อง TS101 ของ Teacher02

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Teacher02      {
                  {"$class": "classroom.management.kmitl.Teachers",
                   "teacherId": "Teacher02",
                   "name": "Nutchaphon",
                   "surname": "Ubonsakka",
                   "position": "resource:classroom.management.kmitl.Positions#Teacher"
                   "faculty": "Engineering",
                   "major": "Telecommunication",
                   "tel": "0123456789",
                   "email": "teacher02@kmitl.ac.th",
                   "room": [
                     "TS101#0"
                   ]
                  }
                }
  
```

Collapse

รูปที่ 4.61 หน้าแสดงผลการจัดเก็บข้อมูลลำดับการจองของ Teacher02

ทำการทดสอบการทำงานของ Transaction ในการแก้ไขการจอง TSRoom ของ Teachers (TeacherEditTSRoom) โดยทำการระบุ ID ผู้จอง, ID ของห้อง, ลำดับของการจอง และวันที่ต้องการเลื่อนการจอง แสดงดังรูปที่ 4.62 ระบบจะทำการแก้ไขข้อมูลวันที่ทำการจอง และบันทึก timestamp, Transaction ID ไปยังข้อมูลของห้อง TS101 ดังรูปที่ 4.63

Submit Transaction ✕

Transaction Type TeacherEditTSRoom ▼

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.TeacherEditTSRoom",
3   "teacher":
4   "resource:classroom.management.kmitl.Teachers#Teacher02",
5   "roomId": "resource:classroom.management.kmitl.TSRoom#TS101",
6   "counts": 0,
7   "structure": {
8     "$class": "classroom.management.kmitl.Structure",
9     "date": "30/03/2019",
10    "time": "13.00-16.00"
11  }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.62 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction แก้ไขการจอง TSRoom ของ Teachers

TS101 ✎ 🗑

```

{
  "$class": "classroom.management.kmitl.TSRoom",
  "roomId": "TS101",
  "roomKey": false,
  "counts": 2,
  "teacherCounts": 1,
  "studentCounts": 1,
  "teacherMatching": [
    0
  ],
  "studentMatching": [],
  "teacher": [
    "resource:classroom.management.kmitl.Teachers#Teacher02"
  ],
  "student": [],
  "teacherStructure": [
    {
      "$class": "classroom.management.kmitl.Structure",
      "date": "30/03/2019",
      "time": "13.00-16.00"
    }
  ],
  "studentStructure": [],
  "timestamp": "\"2019-03-25T19:04:02.912Z\"",
  "transactionId": "c80e27ab6daa1c6d8cf8f55319922487972d5e5d55e1bd561c",
  "transaction_p": "a643cf76b71f88cf8320b0ad7b62856c18d98bbfd921036971"
}

```

Collapse

รูปที่ 4.63 หน้าแสดงผลการแก้ไขข้อมูลการจองห้อง TS101 ของ Teacher02

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบการทำงานของ Transaction ในการยกเลิกการจอง TSRoom ของ Teachers (TeacherRevokeTSRoom) โดยจะต้องทำการระบุ ID ของผู้จอง, ID ของห้อง และลำดับของการจอง ดังรูปที่ 4.64 ระบบจะทำการลบข้อมูลลำดับการจอง, ID ผู้จอง, วันและเวลาที่ทำการจองออกจากห้อง TS101 รวมทั้งบันทึก timestamp และ Transaction ID ดังรูปที่ 4.65 และทำการลบข้อมูลลำดับการจองออกจาก Teacher02 ดังรูปที่ 4.66

Submit Transaction

Transaction Type: TeacherRevokeTSRoom

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.TeacherRevokeTSRoom",
3   "teacher":
4     "resource:classroom.management.kmitl.Teachers#Teacher02",
5   "roomId": "resource:classroom.management.kmitl.TSRoom#TS101",
6   "counts": 0
7 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.64 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ยกเลิกการจอง TSRoom ของ Teachers

```

TS101
{
  "$class": "classroom.management.kmitl.TSRoom",
  "roomId": "TS101",
  "roomKey": false,
  "counts": 2,
  "teacherCounts": 1,
  "studentCounts": 1,
  "teacherMatching": [],
  "studentMatching": [],
  "teacher": [],
  "student": [],
  "teacherStructure": [],
  "studentStructure": [],
  "timestamp": "\"2019-03-25T19:07:10.377Z\"",
  "transactionId": "ccbb81604bbf7b3bfc21b693f3316a81f4a987bec7fa05a03",
  "transaction_p": "c80e27ab6daa1c6d8cf8f55319922487972d5e5d55e1bd561"
}

```

Collapse

รูปที่ 4.65 หน้าแสดงผลการลบข้อมูลการจองห้อง TS101 ของ Teacher02

```

Teacher02
{
  "$class": "classroom.management.kmitl.Teachers",
  "teacherId": "Teacher02",
  "name": "Nutchaphon",
  "surname": "Ubonsakka",
  "position": "resource:classroom.management.kmitl.Positions#Teacher",
  "faculty": "Engineering",
  "major": "Telecommunication",
  "tel": "0123456789",
  "email": "teacher02@kmitl.ac.th",
  "room": []
}

```

Collapse

รูปที่ 4.66 หน้าแสดงผลการลบข้อมูลลำดับการจองของ Teacher02

ทดสอบการทำงานของ Transaction ในการปลดล็อคประตูของ TSRoom (UnlockTSRoom) โดยทำการระบุ ID ของห้อง ดังรูปที่ 4.67 ระบบจะทำการเปลี่ยนค่าตัวแปร Room Key ของห้องให้มีค่าเป็น true แสดงดังรูปที่ 4.68 เพื่อให้กลอนแม่เหล็กไฟฟ้าหยุดการทำงาน และสามารถเปิดประตูได้ และทดสอบการทำงานของ Transaction ในการล็อคประตูของ TeacherRoom (LockTSRoom) แสดงในรูปที่ 4.69 โดยทำการระบุ ID ของห้องเช่นเดียวกัน ระบบจะทำการเปลี่ยนค่าตัวแปร Room Key ให้กลับเป็น false อีกครั้ง ดังรูปที่ 4.70 เพื่อให้กลอนแม่เหล็กไฟฟ้าทำงานอีกครั้ง

Submit Transaction

Transaction Type: UnlockTSRoom

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.UnlockTSRoom",
3   "roomId": "resource:classroom.management.kmitl.TSRoom#TS101"
4 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.67 หน้าสำหรับระบุข้อมูลที่ใช้ส่ง Transaction ในการปลดล็อคประตูของ TSRoom

```

TS101 {
  "$class": "classroom.management.kmitl.TSRoom",
  "roomId": "TS101",
  "roomKey": true,
  "counts": 2,
  "teacherCounts": 1,
  "studentCounts": 1,
  "teacherMatching": [],
  "studentMatching": [],
  "teacher": [],
  "student": [],
  "teacherStructure": [],
  "studentStructure": [],
  "timestamp": "\"2019-03-25T19:08:52.725Z\"",
  "transactionId": "cdf9d8925d4fa2780ecd69698ad197e734512d3a560650984",
  "transaction_p": "ccbb81604bbf7b3bfc21b693f3316a81f4a987bec7fa05a03"
}

```

Collapse

รูปที่ 4.68 หน้าแสดงผลการปลดล็อคประตูของห้อง TS101

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Submit Transaction

Transaction Type **LockTSRoom**

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.LockTSRoom",
3   "roomId": "resource:classroom.management.kmitl.TSRoom#TS101"
4 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

รูปที่ 4.69 หน้าสำหรับระบุข้อมูลที่ใส่ส่ง Transaction ในการล็อกประตูของ TSRoom

```

TS101 {
  "$class": "classroom.management.kmitl.TSRoom",
  "roomId": "TS101",
  "roomKey": false,
  "counts": 2,
  "teacherCounts": 1,
  "studentCounts": 1,
  "teacherMatching": [],
  "studentMatching": [],
  "teacher": [],
  "student": [],
  "teacherStructure": [],
  "studentStructure": [],
  "timestamp": "\2019-03-25T19:09:50.614Z",
  "transactionId": "aee58eac55a49f834df5c9b2353e81927b09b5cac29c5ba57",
  "transaction_p": "cdf9d8925d4fa2780ecd69698ad197e734512d3a560650984"
}

```

Collapse

รูปที่ 4.70 หน้าแสดงผลการล็อกประตูของห้อง TS101

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบการทำงานของ Transaction ในการจัดเก็บรายชื่อของผู้เข้าร่วมชั้นเรียน (CreateClassroom) โดยทำการส่ง ID ของนักศึกษาที่เข้าร่วมชั้นเรียนไปยัง ID ของ Classroom ที่กำหนด ดังรูปที่ 4.71 ผ่านทางการสแกน QR Code และจัดเก็บรายชื่อไว้ใน Assets ของ Classroom ที่ได้ทำการสร้างไว้ ดังรูปที่ 4.72

Submit Transaction

Transaction Type: CreateClassroom

JSON Data Preview

```

1 {
2   "$class": "classroom.management.kmitl.CreateClassroom",
3   "student":
4   "resource:classroom.management.kmitl.Students#Student01",
5   "classId":
6   "resource:classroom.management.kmitl.Classroom#Class01"
7 }

```

Optional Properties

Just need quick test data? [Generate Random Data](#) [Cancel](#) [Submit](#)

รูปที่ 4.71 หน้าระบุข้อมูลสำหรับส่ง Transaction ในการจัดเก็บ ID นักศึกษาที่เข้าชั้นเรียน

```

Class01
{
  "$class": "classroom.management.kmitl.Classroom",
  "classId": "Class01",
  "faculty": "Engineering",
  "major": "Telecommunication",
  "subject": "Telecom Project",
  "section": "01",
  "date": "30/03/2019",
  "time": "9.00-12.00",
  "counts": 2,
  "studentList": [
    "resource:classroom.management.kmitl.Students#Student01",
    "resource:classroom.management.kmitl.Students#Student02"
  ]
}

```

รูปที่ 4.72 หน้าแสดงผลการจัดเก็บ ID ของนักศึกษาที่เข้าร่วมชั้นเรียน

การดำเนินการต่างๆของระบบ เช่น การสร้าง การแก้ไขข้อมูล การลบข้อมูล และการส่ง Transactions ทั้งหมด จะถูกเก็บบันทึกไว้ในระบบและไม่สามารถทำการแก้ไขหรือปลอมแปลงได้ โดยสามารถดูประวัติการทำงานทั้งหมดของระบบได้ผ่านทาง Historian Registry ดังรูปที่ 4.73

Date, Time	Entry Type	Participant	
2019-03-26, 12:47:28	Issuelidentity	admin (NetworkAdmin)	view record
2019-03-26, 12:46:11	Issuelidentity	admin (NetworkAdmin)	view record
2019-03-26, 02:24:14	CreateClassroom	admin (NetworkAdmin)	view record
2019-03-26, 02:16:41	UpdateParticipant	admin (NetworkAdmin)	view record
2019-03-26, 02:09:50	LockTSRoom	admin (NetworkAdmin)	view record
2019-03-26, 02:08:52	UnlockTSRoom	admin (NetworkAdmin)	view record

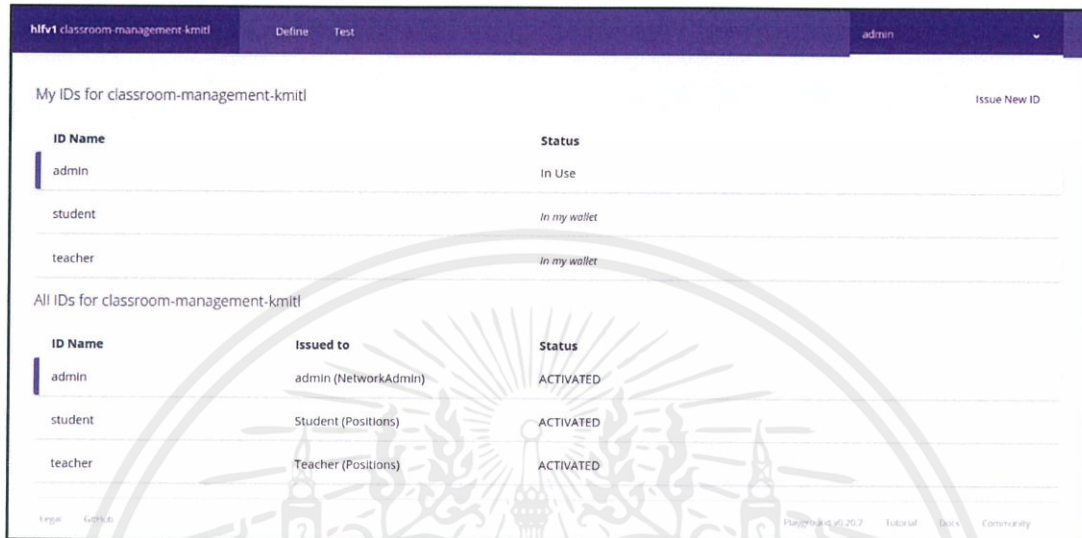
รูปที่ 4.73 หน้าแสดงผลประวัติการทำงานทั้งหมดของระบบ

4.1.3 ผลการทดสอบการทำงานของ Access Control

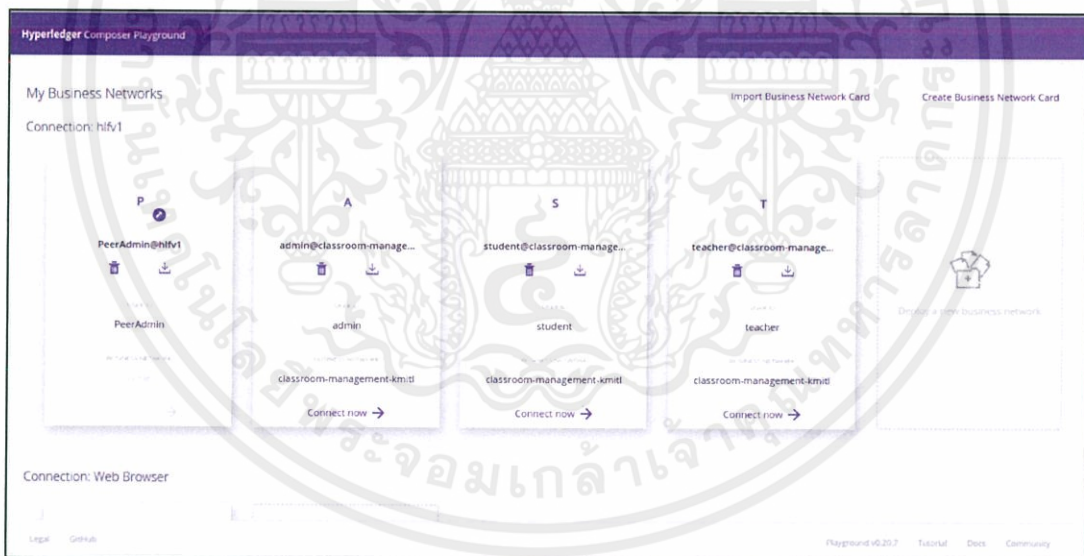
ในส่วนของการทดสอบการทำงานในการควบคุมการเข้าถึงข้อมูลของ Access Control นั้น สามารถทำได้โดยทำการสร้าง ID ผู้ใช้งานเครือข่ายของนักศึกษาและอาจารย์ขึ้นมา ซึ่งจะอ้างอิงจาก Participants ของ Positions ดังรูปที่ 4.74 จะได้การ์ดเครือข่ายของนักศึกษา คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

student@classroom-management-kmitl และการ์ดเครือข่ายของอาจารย์ คือ teacher@classroom-management-kmitl ดังแสดงในรูปที่ 4.75



รูปที่ 4.74 หน้า ID Registry สำหรับสร้างและปรับเปลี่ยนการใช้งาน ID ของเครือข่าย



รูปที่ 4.75 หน้าสำหรับเลือกการ์ดเครือข่ายธุรกิจในการใช้งานระบบ

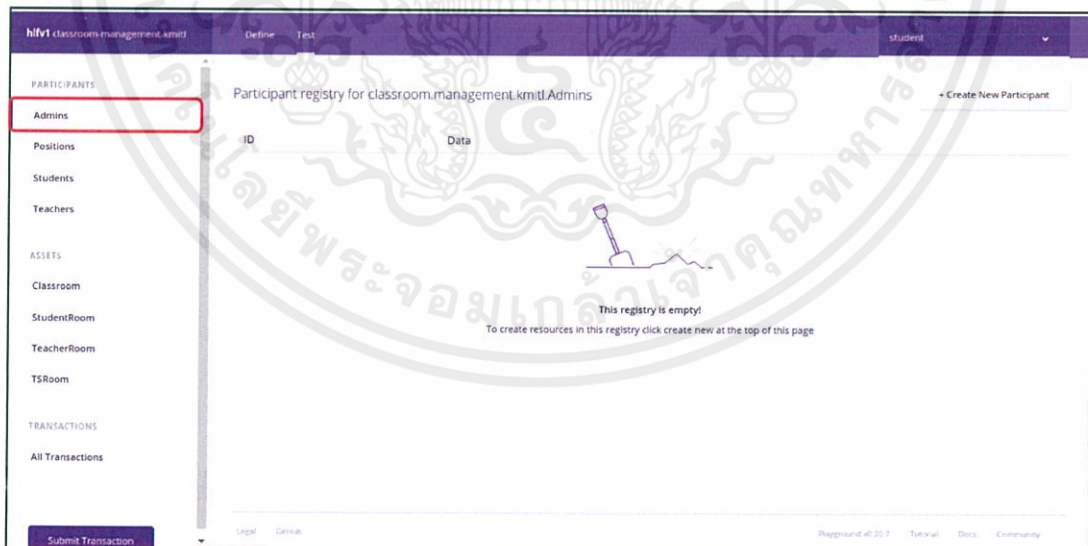
ทดสอบการเข้าถึงข้อมูลต่างๆภายในระบบผ่านการ์ดเครือข่ายของนักศึกษา และบันทึกผลผ่านหน้าแสดงผลของ Playground นักศึกษาจะสามารถเข้าถึงข้อมูล Participants ของ Students รวมถึงสามารถทำการแก้ไขและอัปเดตข้อมูลต่างๆได้ ดังรูปที่ 4.76 แต่ไม่สามารถเข้าถึงข้อมูลของ Admins และ Teachers แสดงดังรูปที่ 4.77 และ 4.78

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นักศึกษาสามารถเข้าถึงข้อมูล Assets ของ StudentRoom รวมถึงสามารถทำการส่ง Transactions ทั้งหมดที่เกี่ยวข้องกับการใช้งาน StudentRoom ของ Students ดังแสดงในรูปที่ 4.79 และสามารถเข้าถึงข้อมูล Assets ของ TSRoom และการส่ง Transactions ทั้งหมดในการใช้งาน TSRoom ของ Students ในรูปที่ 4.80 แต่ไม่สามารถใช้งาน Assets ของ TeacherRoom ได้ ดังรูปที่ 4.81

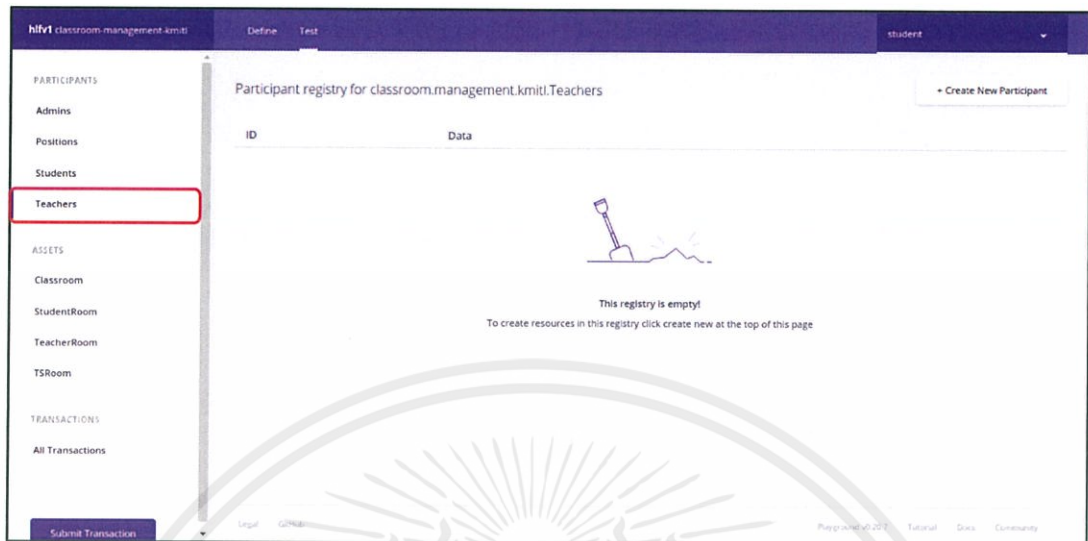


รูปที่ 4.76 หน้าแสดงผล Participants ของ Students ผ่านการ์ดเครือข่ายของนักศึกษา

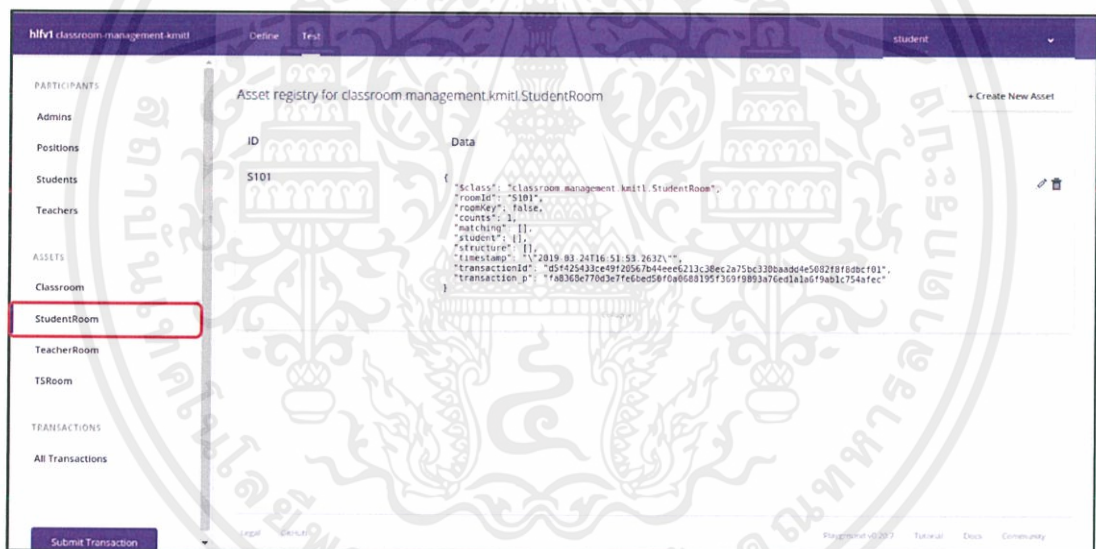


รูปที่ 4.77 หน้าแสดงผล Participants ของ Admins ผ่านการ์ดเครือข่ายของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

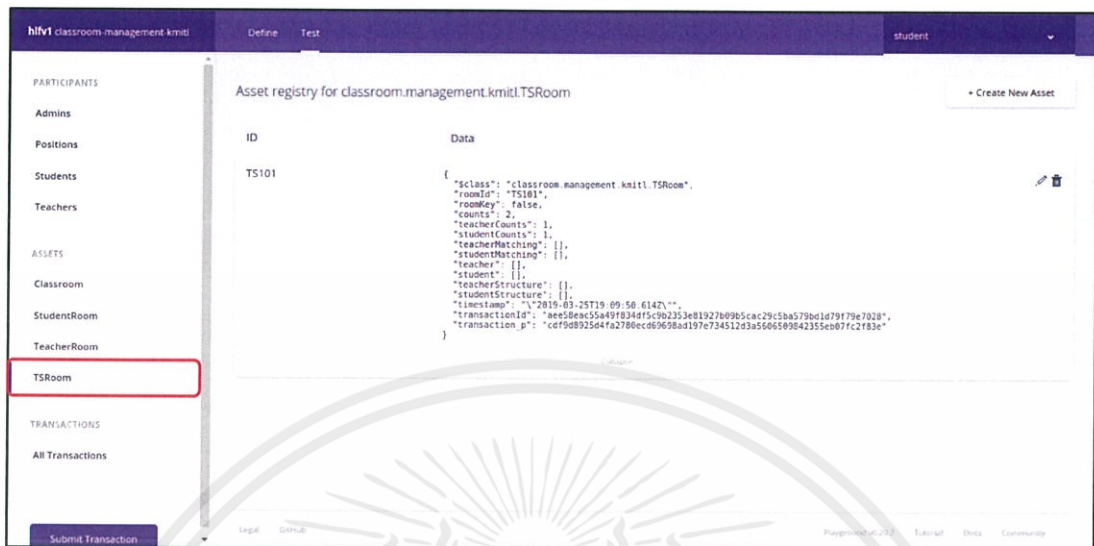


รูปที่ 4.78 หน้าแสดงผล Participants ของ Teachers ผ่านการ์ดเครือข่ายของนักศึกษา

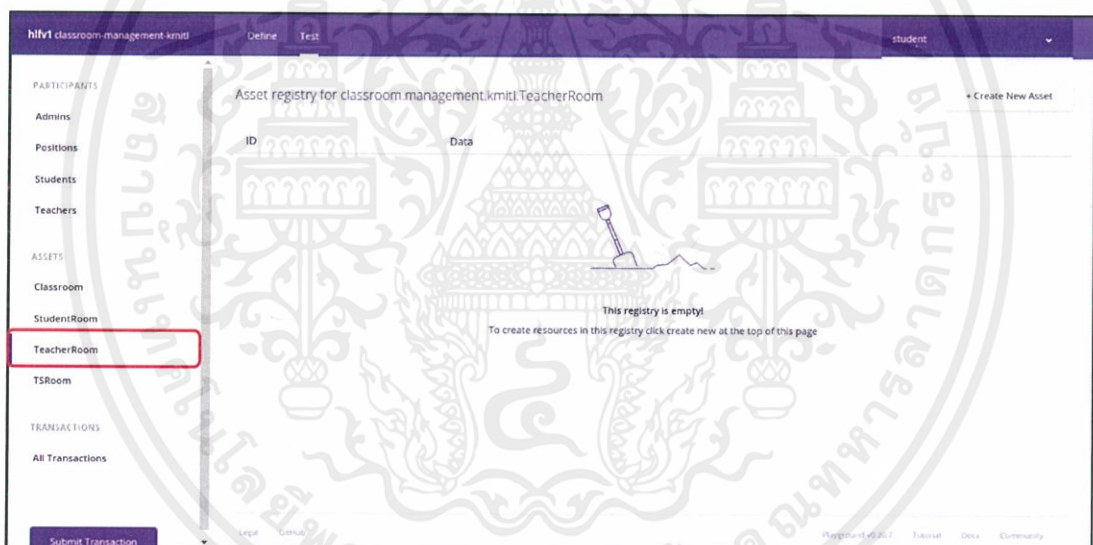


รูปที่ 4.79 หน้าแสดงผล Assets ของ StudentRoom ผ่านการ์ดเครือข่ายของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



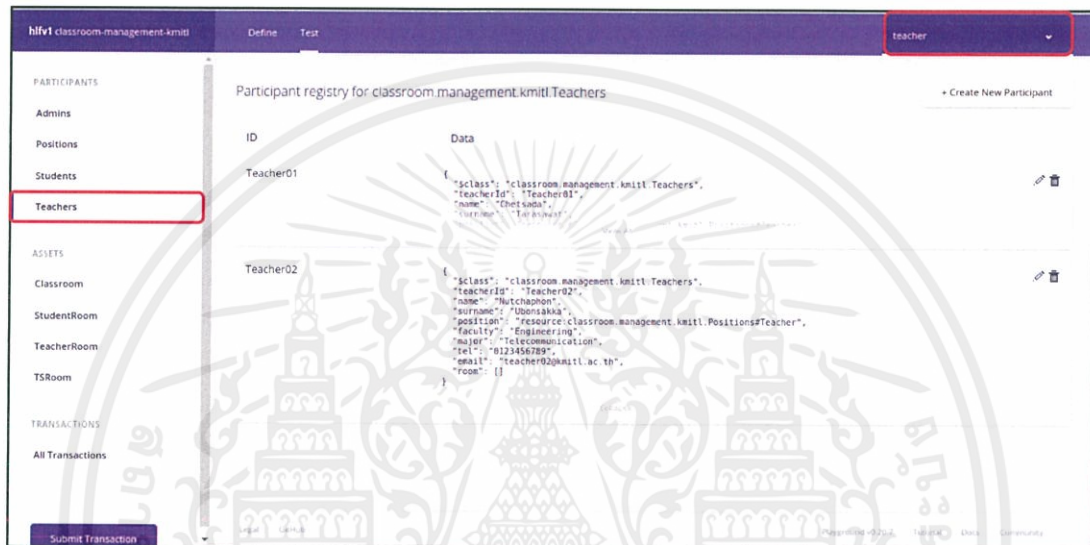
รูปที่ 4.80 หน้าแสดงผล Assets ของ TSRoom ผ่านการ์ดเครือข่ายของนักศึกษา



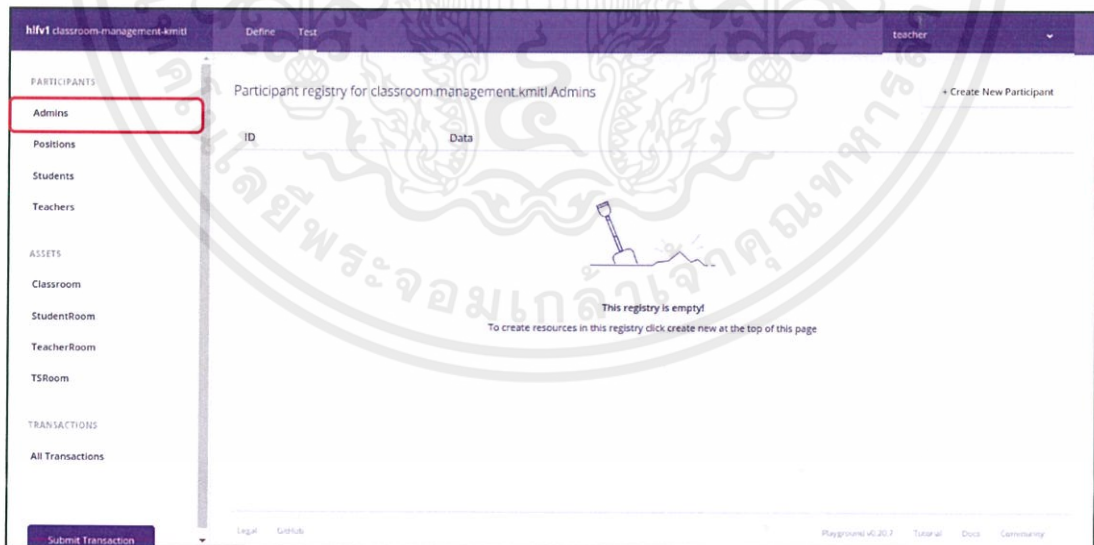
รูปที่ 4.81 หน้าแสดงผล Assets ของ TeacherRoom ผ่านการ์ดเครือข่ายของนักศึกษา

ในส่วนของการทดสอบการเข้าถึงข้อมูลต่างๆผ่านทางการ์ดเครือข่ายของอาจารย์ อาจารย์จะสามารถเข้าถึงข้อมูล Participants ของ Teachers รวมถึงสามารถทำการแก้ไขและอัปเดตข้อมูลต่างๆได้ ดังรูปที่ 4.82 และสามารถเข้าถึงข้อมูล Participants ของ Students ได้เช่นกัน แต่ไม่สามารถทำการแก้ไขข้อมูลได้ ดังรูปที่ 4.83 อาจารย์จะไม่สามารถเข้าถึงข้อมูล Participants ของ Admins ดังรูปที่ 4.84

อาจารย์สามารถเข้าถึงข้อมูล Assets ของ TeacherRoom และ TSRoom รวมถึงสามารถทำการส่ง Transactions ในการใช้งาน Assets ทั้งสองของ Teachers ดังแสดงในรูปที่ 4.85 และรูปที่ 4.86 แต่ไม่สามารถเข้าถึงข้อมูลและใช้งาน Assets ของ StudentRoom ได้ ดังรูปที่ 4.87 และสามารถเข้าถึงข้อมูล Assets ของ Classroom เพื่อใช้ในการตรวจสอบการเข้าชั้นเรียนของนักศึกษาได้ดังรูปที่ 4.88

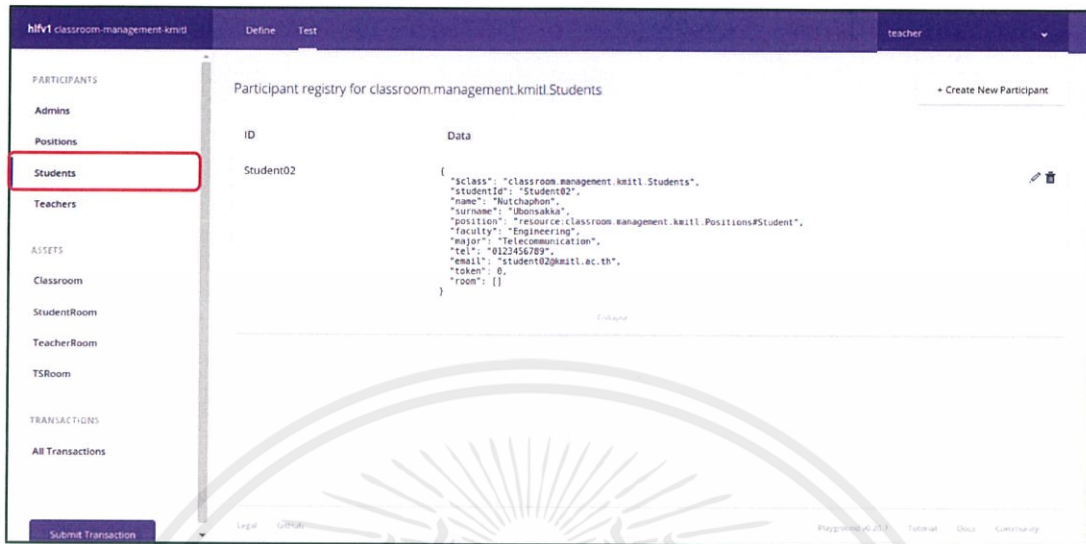


รูปที่ 4.82 หน้าแสดงผล Participants ของ Teachers ผ่านการ์ดเครือข่ายของอาจารย์

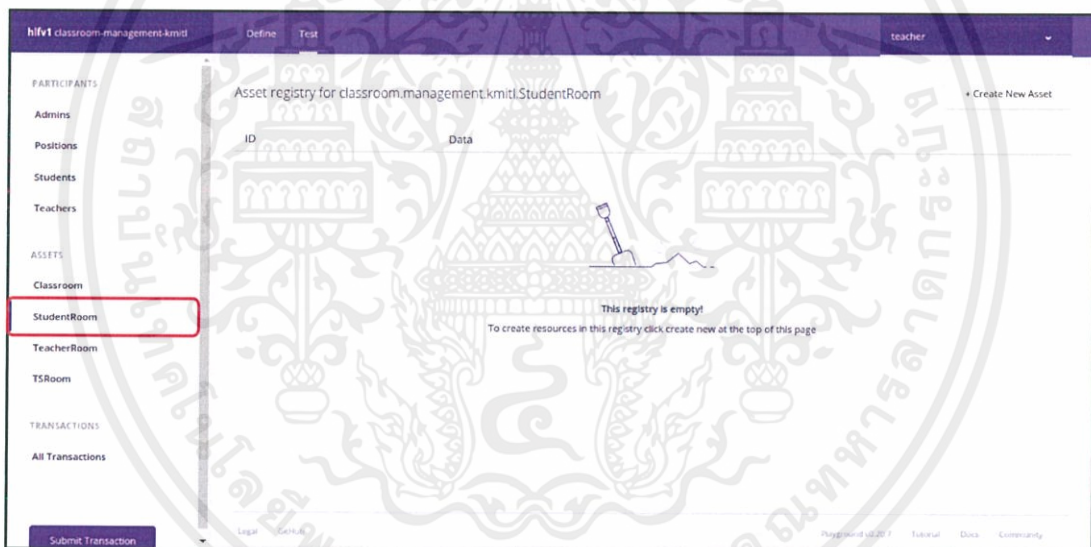


รูปที่ 4.83 หน้าแสดงผล Participants ของ Admins ผ่านการ์ดเครือข่ายของอาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.84 หน้าแสดงผล Participants ของ Students ผ่านการ์ดเครือข่ายของอาจารย์



รูปที่ 4.85 หน้าแสดงผล Assets ของ StudentRoom ผ่านการ์ดเครือข่ายของอาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Asset registry for classroom.management.kmitl.TeacherRoom

ID	Data
T101	{ <pre> "className": "classroom.management.kmitl.TeacherRoom", "roomId": "T101", "roomKey": false, "counts": 1, "matching": {}, "teacher": {}, "structure": {}, "timestamp": "\2019-03-24T16:58:08.238Z", "transactionId": "a654762b7fec644da6a84546962b9c5e96e75aa7e771b3def1b58b575d179d7", "transaction p": "b820d283001516865cfc6426d9549a40a64b71050dbce03c765989f1439a0r" </pre>

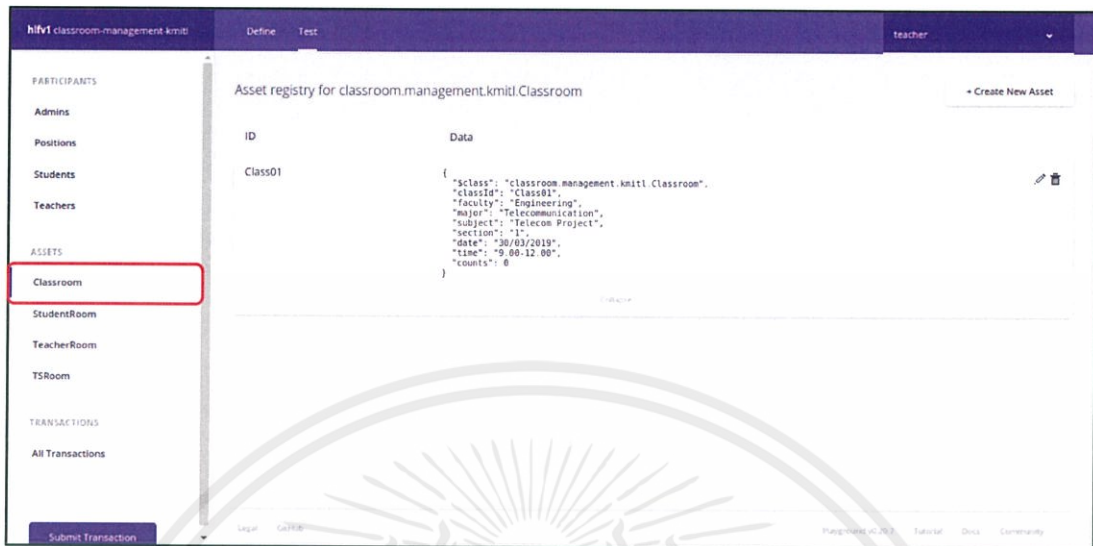
รูปที่ 4.86 หน้าแสดงผล Assets ของ TeacherRoom ผ่านการ์ดเครือข่ายของอาจารย์

Asset registry for classroom.management.kmitl.TSRoom

ID	Data
TS101	{ <pre> "className": "classroom.management.kmitl.TSRoom", "roomId": "TS101", "roomKey": false, "counts": 2, "teacherCounts": 1, "studentCounts": 1, "teacherMatching": {}, "studentMatching": {}, "teacher": {}, "student": {}, "teacherStructure": {}, "studentStructure": {}, "timestamp": "\2019-03-25T19:09:50.614Z", "transactionId": "aee58eac55a49f8346f9c90253e81927b09b5cac29c8ba5796d1d79f79e7028", "transaction p": "c0f9d8935c4e67708ec09098a0197e75451261a56665098423556067f2183e" </pre>

รูปที่ 4.87 หน้าแสดงผล Assets ของ TSRoom ผ่านการ์ดเครือข่ายของอาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.88 หน้าแสดงผล Assets ของ Classroom ผ่านการ์ดเครือข่ายของอาจารย์

4.2 ผลการทดลองในส่วนของ Webpage

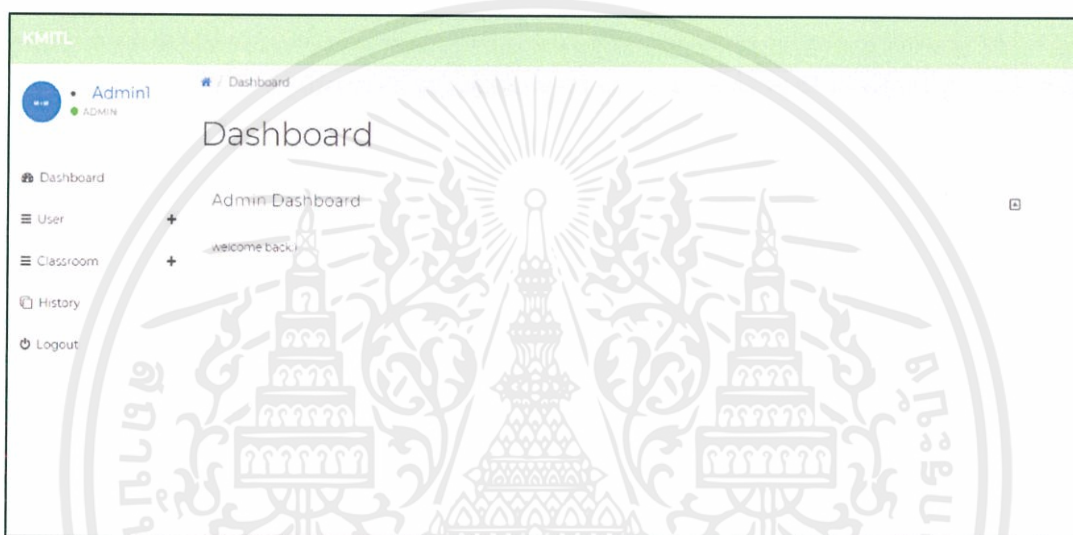
4.2.1 ผลการทดสอบในส่วนของการแสดงผล Admin Webpage

จากการทดสอบการแสดงผลของ Webpage ที่ทำการเชื่อมต่อกับ Blockchain พบว่าผู้จัดทำสามารถจัดเก็บผลการทดลองและใช้งานระบบได้ตามปกติ โดยได้ทำการจัดเก็บผลการทดลองตามที่ได้ทำการออกแบบไว้ ซึ่งเริ่มจากการ login แสดงดังรูปที่ 4.89 เนื่องจากในส่วนของการ register ของ Admin User นั้นจะกระทำผ่านทางส่วนของ Blockchain

รูปที่ 4.89 การ login เข้าสู่ระบบ

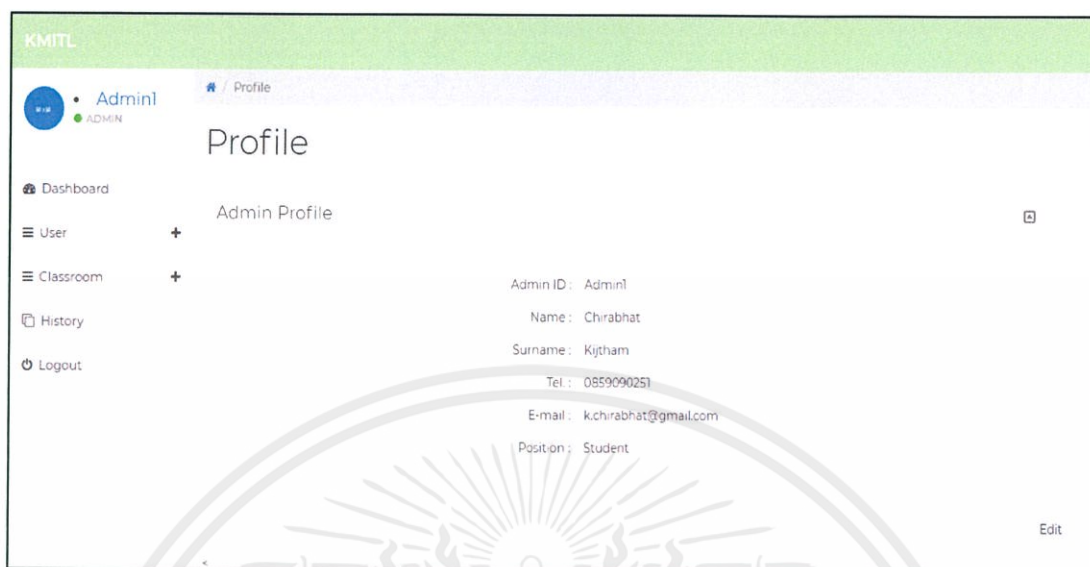
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหัวข้อ 3.1.2.1 ในส่วนของ Admin Dashboard ซึ่งจะประกอบไปด้วยส่วนของเมนูย่อย และจะเป็นหน้าหลักที่คอยรองรับผู้ใช้งาน โดยรูปที่ 4.90 จะเป็นรูปแสดงหน้าจอของ Admin Dashboard จะพบว่าทางด้านซ้ายจะมีแถบเมนู ซึ่งประกอบไปด้วย Admin Profile, User, Classroom และ History จากโฟลว์ชาร์ตในหัวข้อ 3.1.2.1 จะเห็นว่าผู้จัดทำได้ทำการออกแบบแยกย่อยไปตามแต่ละหน้า ดังนั้นทางคณะผู้จัดทำจะขอทำการอธิบายผลการเก็บผลการทดสอบพอสังเขป

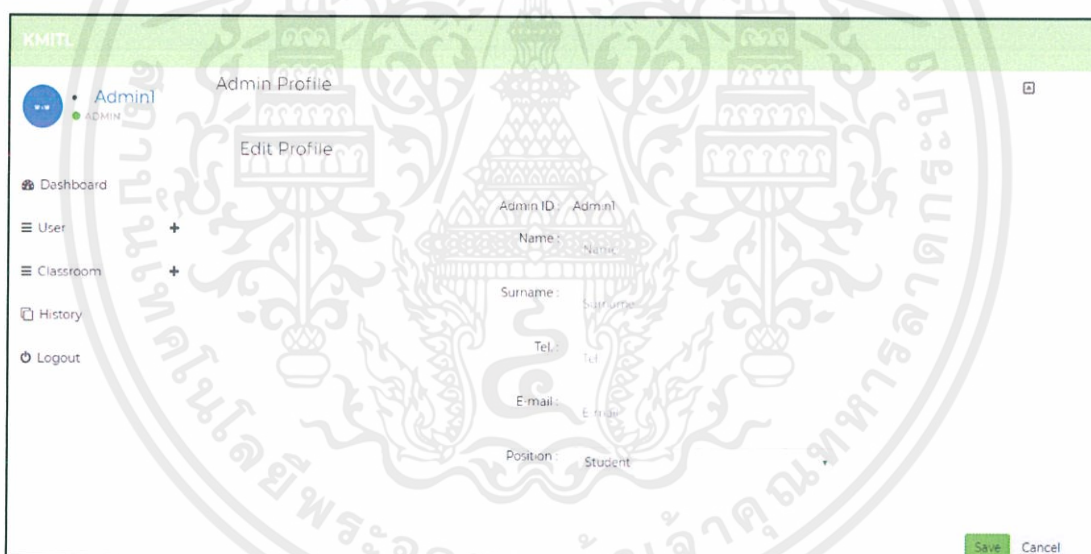


รูปที่ 4.90 การแสดงผลหน้า Admin Dashboard

1) Admin Profile เป็นส่วนการแสดงผลของข้อมูลของผู้ใช้งาน ซึ่งจะมีการบันทึกข้อมูลที่จำเป็นต่อการใช้งานของผู้ใช้งานดังรูปที่ 4.91 จะพบว่านอกจากข้อมูลที่จำเป็นของผู้ใช้งาน ซึ่งประกอบไปด้วย Admin ID, Name, Surname, Tel., E-mail, และ Position ในส่วนมุมขวาล่างจะมีปุ่ม Edit สำหรับการแก้ไขข้อมูลส่วนตัว ซึ่งผลการทดสอบจะอยู่ในรูปที่ 4.92 ผู้ใช้งานสามารถแก้ไขและจัดเก็บข้อมูลที่ได้ทำการแก้ไข หรือสามารถยกเลิกระหว่างการแก้ไขได้



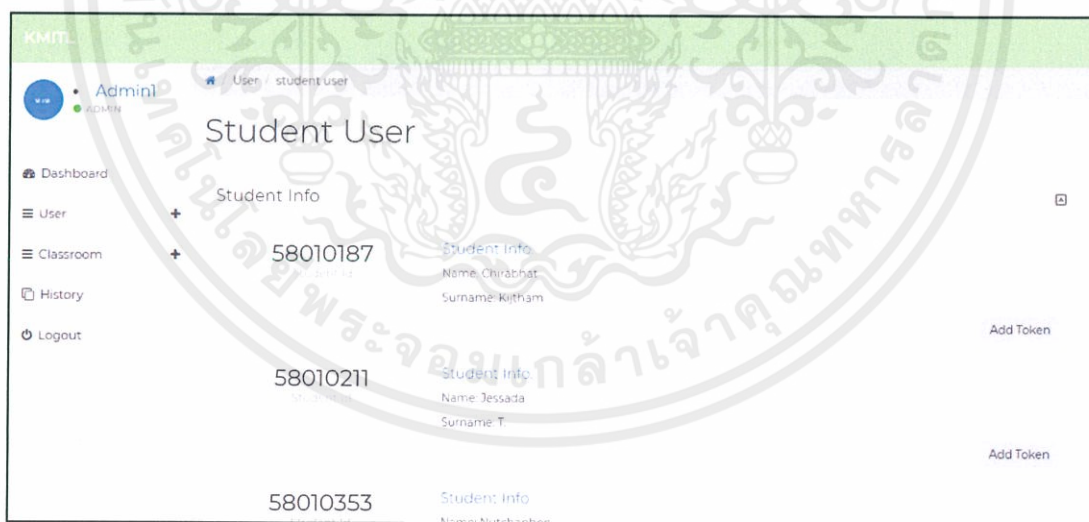
รูปที่ 4.91 การแสดงผลข้อมูลผู้ใช้งานในส่วนของ Admin Profile



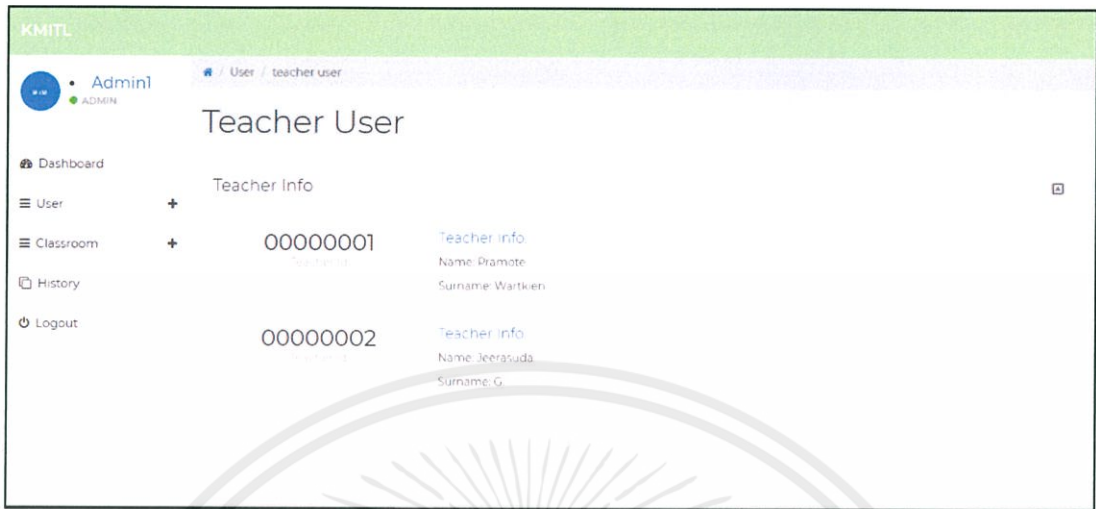
รูปที่ 4.92 การแก้ไขข้อมูลผู้ใช้งานของ Admin Profile

2) User คือ ส่วนของรายชื่อบัญชีผู้ใช้งานในระบบ ซึ่งจะถูกแบ่งออกเป็น 2 ประเภท ได้แก่ Student User และ Teacher User โดยจะเป็นไปตามรูปที่ 4.93 - 4.94 ตามลำดับ ในส่วนการใช้งานของ Student User ในรูปที่ 4.93 จะเห็นได้ว่าในหนึ่งรายชื่อจะมีการแสดง ชื่อ รหัสศั น์ศึกษา พร้อมกับปุ่มอีก 2 ปุ่มในการทำงาน ได้แก่ ปุ่ม Student Info ซึ่งจะเป็นส่วนของข้อมูล บัญชีผู้ใช้นั้น เมื่อทำการคลิกเข้าไปภายในจะเป็นดังรูปที่ 4.95 นอกจากผู้ใช้งานจะสามารถเห็นข้อมูล

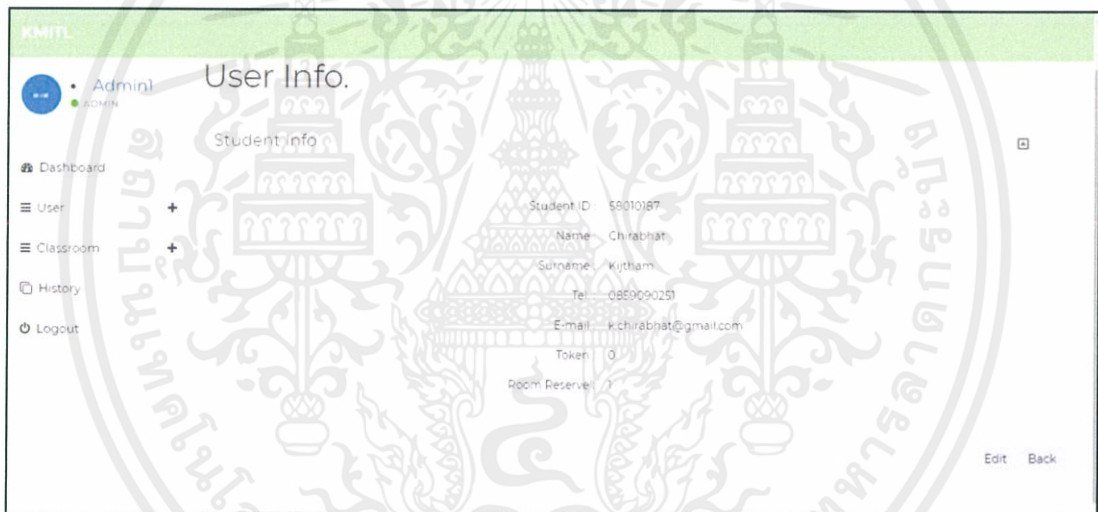
บัญชีผู้ใช้งานของ Student แล้วยังสามารถทำการแก้ไขข้อมูลในกรณีที่บัญชีผู้ใช้งานนั้นมีเหตุขัดข้องไม่สามารถแก้ไขข้อมูลเองได้อีกด้วย จากรูปที่ 4.96 จะเห็นได้อีกว่ามีการแสดงผลของห้องที่บัญชีผู้ใช้งานนี้ทำการจองไว้ ผู้ใช้งานสามารถเข้าไปดู และทำการเปิดห้องหรือยกเลิกการจองห้องเรียนของผู้ใช้งานนี้ได้ดังรูปที่ 4.97 ในกรณีฉุกเฉินต่างๆ โดยภายในกล่องข้อมูลของห้องเรียนจะแสดง รหัสของห้องเรียน รหัสของบัญชีผู้ใช้ที่จองไว้ วันเวลาในการจอง และที่ตั้งของห้องเรียน ซึ่งสีของกล่องจะมี 2 สี ได้แก่ สีฟ้า ซึ่งมีความหมายว่าห้องเรียนนี้ยังไม่ถึงเวลาในการใช้งาน และ สีส้ม ที่มีความหมายว่าห้องเรียนนี้กำลังถูกใช้งานในขณะนั้น หากเป็นสีฟ้าหรือยังไม่ถึงเวลาการใช้งาน ในกรณีที่ทำการยกเลิกการจองห้องเรียนในรูปที่ 4.98 จะเห็นว่าในกล่องส่วนด้านล่างขวาจะมีปุ่ม Unreserve สำหรับยกเลิกการจอง เมื่อทำการกดปุ่มนั้นจะพบว่าห้องเรียนได้ถูกยกเลิกการจองแล้ว เมื่อผู้ใช้งานต้องการจะเปิดห้องเรียนจะต้องกดปุ่ม Unlock เพื่อทำการปลดล็อคประตู ดังแสดงผลตามรูปที่ 4.99 ในส่วนของห้องเรียนที่ถึงเวลาในการใช้งานแล้วหรือกำลังถูกใช้งานจะพบว่า ระบบจะทำการปลดล็อคประตูเป็นเวลา 15 นาที ดังรูปที่ 4.100 และเมื่อหมดเวลาประตูจะทำการล็อคโดยอัตโนมัติ และผู้ใช้งานจะสามารถกด Unlock เพื่อเปิดประตูได้อีกครั้ง ดังรูปที่ 4.101 นอกจากนี้ในกรณีของบัญชีผู้ใช้งานของ Student User ยังสามารถทำการ Add Token เพื่อใช้สำหรับการจองห้องเรียน Teacher and Student Room ได้อีกด้วย ดังรูปที่ 4.102 – 4.104



รูปที่ 4.93 การแสดงผลในส่วนของรายชื่อบัญชีผู้ใช้งานของ Student User



รูปที่ 4.94 การแสดงผลในส่วนของรายชื่อบัญชีผู้ใช้งานของ Teacher User



รูปที่ 4.95 การแสดงผลข้อมูลบัญชีผู้ใช้งานของ Student User

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Edit Profile

Student ID: ID

Name: Name

Surname: Surname

Tel.: Tel

E-mail: E-mail

Room Reserve: This user has reserved 1 room.

Save Cancel

รูปที่ 4.96 การแก้ไขข้อมูลบัญชีผู้ใช้งานของ Student User

KMITL

Admin | 58010187 | Room

Student user room

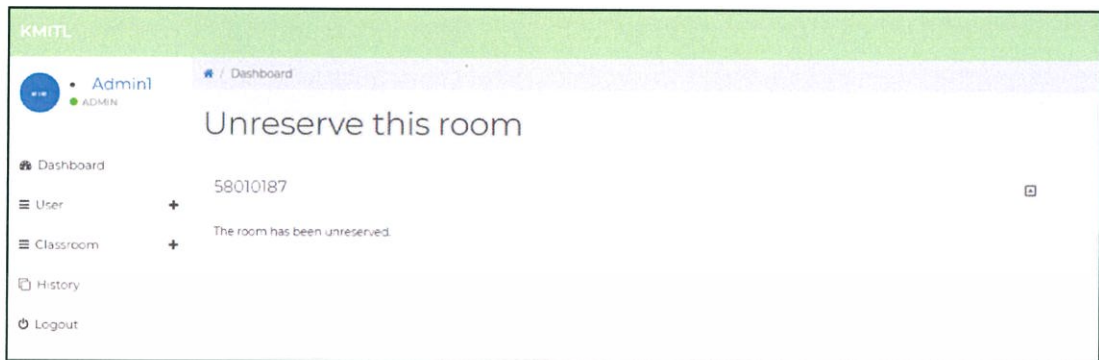
Student Recently Reserve

Student ID	Room	Date	Address	Action
58010187	HM-303	25-01-2019	อาคาร 7 ชั้น 3 คณะวิศวกรรมศาสตร์	Unreserve

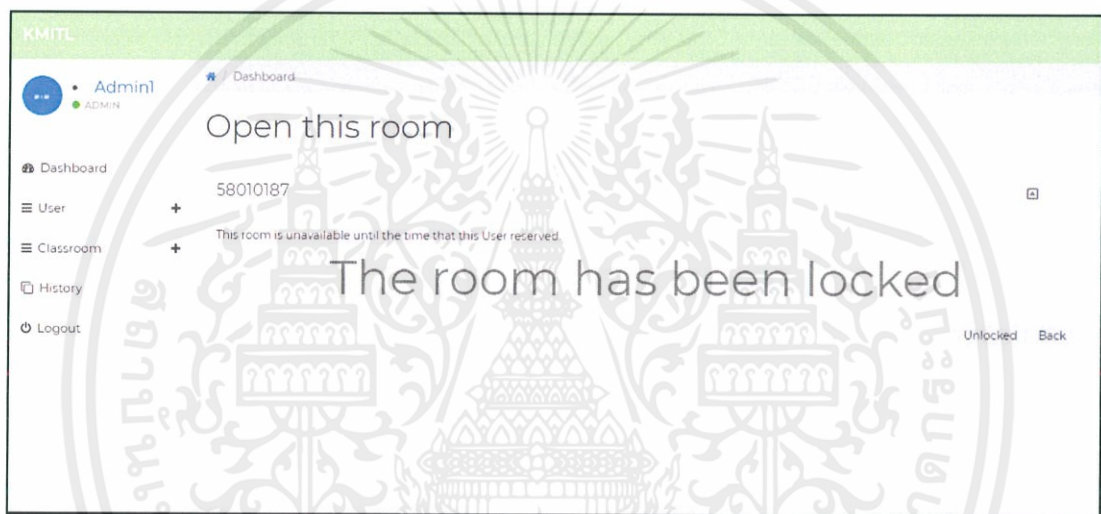
Reserved On using

รูปที่ 4.97 รายการห้องเรียนที่บัญชีผู้ใช้งานของ Student User ได้ทำการจองไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

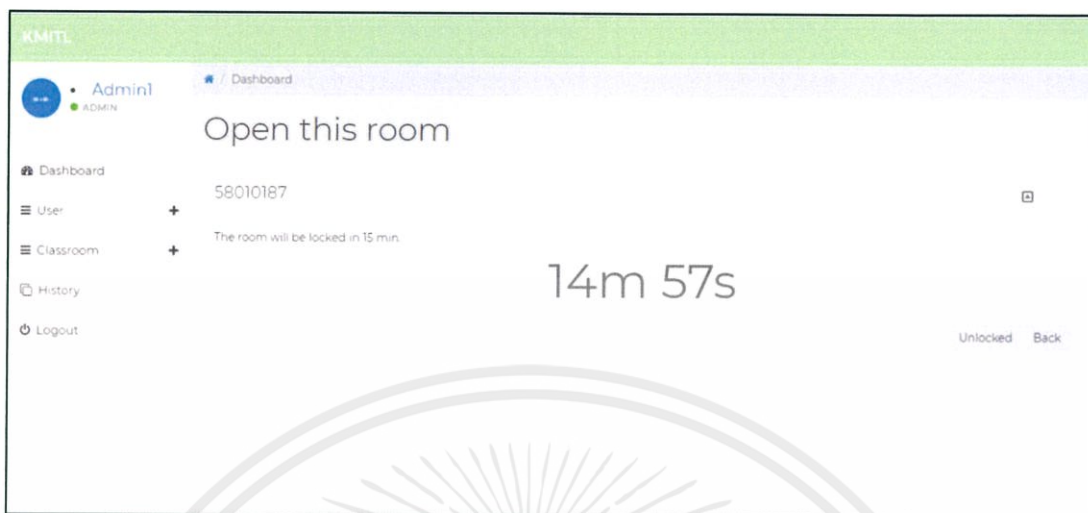


รูปที่ 4.98 การยกเลิกการจองห้องเรียนผ่าน Admin User

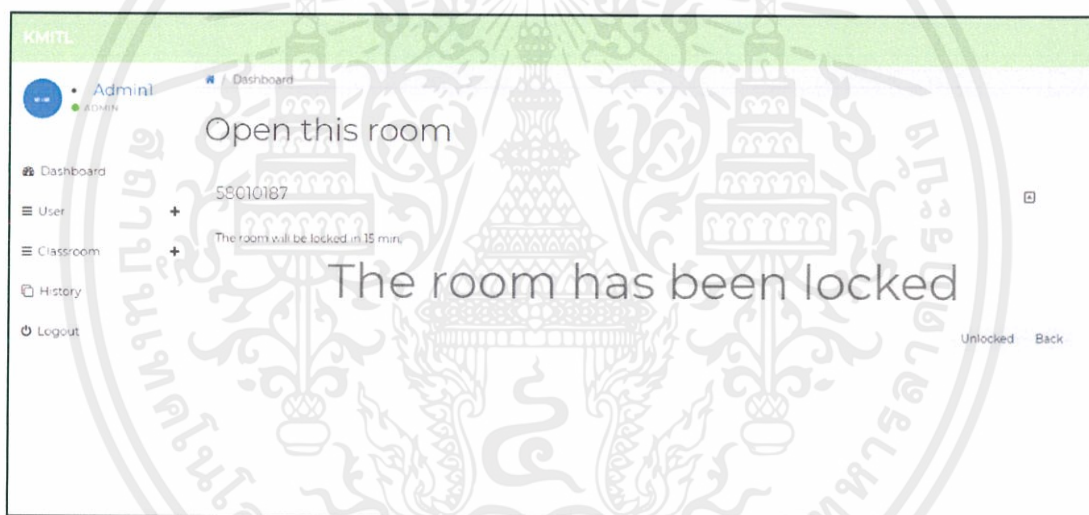


รูปที่ 4.99 การปลดล็อคห้องเรียนผ่าน Admin User

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.100 การนับถอยหลังของการปลดล็อคประตูในกรณีที่ห้องเรียนกำลังถูกใช้งาน



รูปที่ 4.101 การแสดงผลการล็อคห้องเรียนอีกครั้งหลังจากผ่านไป 15 นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Add Token

Add Token for 58010187

Student ID: 58010187

Name: Chirabhat

Surname: Kijtham

Token: 0

Add Token Back

รูปที่ 4.102 การ Add Token ให้กับ Student User ผ่าน Admin User

Add Token

Add Token for 58010187

Student ID: 58010187

Name: Chirabhat

Surname: Kijtham

Token: 20

Add Token Back

รูปที่ 4.103 การกำหนดจำนวน token ที่ต้องการเพิ่มให้กับ Student User

The screenshot shows a web interface titled "Add Token". Below the title, it says "Add Token for 58010187". The user information is displayed as follows:

Student ID:	58010187
Name:	Chirabhat
Surname:	Kijtham
Token:	20

At the bottom right, there are two buttons: "Add Token" and "Back".

รูปที่ 4.104 การแสดงผลหลังทำการ Add Token ให้กับ Student User

ในส่วนของ Teacher User จะมีความคล้ายคลึงกับ Student User แต่จะมีความต่างเล็กน้อยในส่วนของคุณสมบัติส่วนตัวและบัญชีผู้ใช้งานของ Teacher User จะมีส่วนที่บ่งบอกถึงจำนวน Class ที่มีอยู่ของบัญชีผู้ใช้งานนั้นด้วย ดังรูปที่ 4.105 - 4.106

The screenshot shows a web interface titled "Teacher Info". The user information is displayed as follows:

Teacher ID:	00000001
Position:	Asoc.Dr.
Name:	Pramote
Surname:	Wardkien
Tel.:	0816997616
E-mail:	pramote@telecom.kmitl.ac.th
Room Reserve:	1
Class in hand:	0

At the bottom right, there are two buttons: "Edit" and "Back".

รูปที่ 4.105 การแสดงผลข้อมูลบัญชีผู้ใช้งานของ Teacher User

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Teacher ID: ID

Position: -

Name: Name

Surname: Surname

Tel.: Tel

E-mail: E-mail

Room Reserve: This user has reserved 1 room.

Class in hand: This user has 0 class in hand.

Save Cancel

รูปที่ 4.106 การแก้ไขข้อมูลบัญชีผู้ใช้งานของ Teacher User

3) Classroom ในส่วนของ Classroom จะพบว่าได้ถูกแบ่งออกเป็น 3 ประเภท ได้แก่ Student Room, Teacher Room, Teacher and Student Room ดังรูปที่ 4.107 – 4.109 ตามลำดับ ซึ่งโดยหลักแล้วทั้ง 3 ประเภทจะมีหน้าที่เหมือนกันต่างกันตรงชนิดของห้องเรียนที่กำหนด เพื่อสิทธิ์การเข้าถึงของบัญชีผู้ใช้งานที่ต่างกัน โดย Admin User สามารถระงับการใช้งานห้องเรียนได้ชั่วคราวเมื่อทำการกด Close Room อีกทั้งสามารถ Add Room เพื่อทำการเพิ่มห้องเรียนที่เปิดให้ใช้งานได้ ดังรูปที่ 4.110 – 4.112

Student Room

Student Room Info

HM-303 Room ID	Reserve & Edit ที่ตั้ง: อาคาร7ชั้น ชั้น3 คณะวิศวกรรมศาสตร์ KMITL	Close Room
D-513 Room ID	Reserve & Edit ที่ตั้ง: อาคารพระเทพฯ ชั้น5 คณะวิศวกรรมศาสตร์ KMITL	Close Room
E12-304 Room ID	Reserve & Edit ที่ตั้ง: อาคาร12ชั้น ชั้น3 คณะวิศวกรรมศาสตร์ KMITL	Close Room

Add Room

รูปที่ 4.107 การแสดงผลห้องเรียนสำหรับ Student Room

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Teacher Room

Teacher Room Info

HM-304 <small>Room id</small>	Reserve & Edit ที่ตั้ง อาคาร7ชั้น ชั้น3 คณะวิศวกรรมศาสตร์ KMITL	Close Room
E12-305 <small>Room id</small>	Reserve & Edit ที่ตั้ง อาคารพระเทพฯ ชั้น5 คณะวิศวกรรมศาสตร์ KMITL	Close Room
T-101 <small>Room id</small>	Reserve & Edit ที่ตั้ง อาคารโทรคมนาคม ชั้น1 คณะวิศวกรรมศาสตร์ KMITL	Close Room
		Add Room

รูปที่ 4.108 การแสดงผลห้องเรียนสำหรับ Teacher Room

Teacher and Student Room

Teacher and Student Room Info

C-113 <small>Room id</small>	Reserve & Edit ที่ตั้ง อาคาร12ชั้น ชั้น7 คณะวิศวกรรมศาสตร์ KMITL	Close Room
E12-704 <small>Room id</small>	Reserve & Edit ที่ตั้ง อาคารพระเทพฯ ชั้น1 คณะวิศวกรรมศาสตร์ KMITL	Close Room
E12-604 <small>Room id</small>	Reserve & Edit ที่ตั้ง อาคาร12ชั้น ชั้น6 คณะวิศวกรรมศาสตร์ KMITL	Close Room
		Add Room

รูปที่ 4.109 การแสดงผลห้องเรียนสำหรับ Teacher and Student Room

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Student Room Info

HM-303
Room ID

Reserve & Edit
ที่ตั้ง: อาคาร7ชั้น ชั้น3 คณะวิศวกรรมศาสตร์ KMUTL

Reopen Room

D-513
Room ID

Reserve & Edit
ที่ตั้ง: อาคารพระเทพฯ ชั้น5 คณะวิศวกรรมศาสตร์ KMUTL

Close Room

รูปที่ 4.110 การปิดการใช้งานห้องเรียน

Student Room Info

HM-303
Room ID

Reserve & Edit
ที่ตั้ง: อาคาร7ชั้น ชั้น3 คณะวิศวกรรมศาสตร์ KMUTL

Close Room

D-513
Room ID

Reserve & Edit
ที่ตั้ง: อาคารพระเทพฯ ชั้น5 คณะวิศวกรรมศาสตร์ KMUTL

Close Room

รูปที่ 4.111 การเปิดการใช้งานห้องเรียน

Add Room

Add Room

Edit Room Data

Room ID: ID

Room Type: -

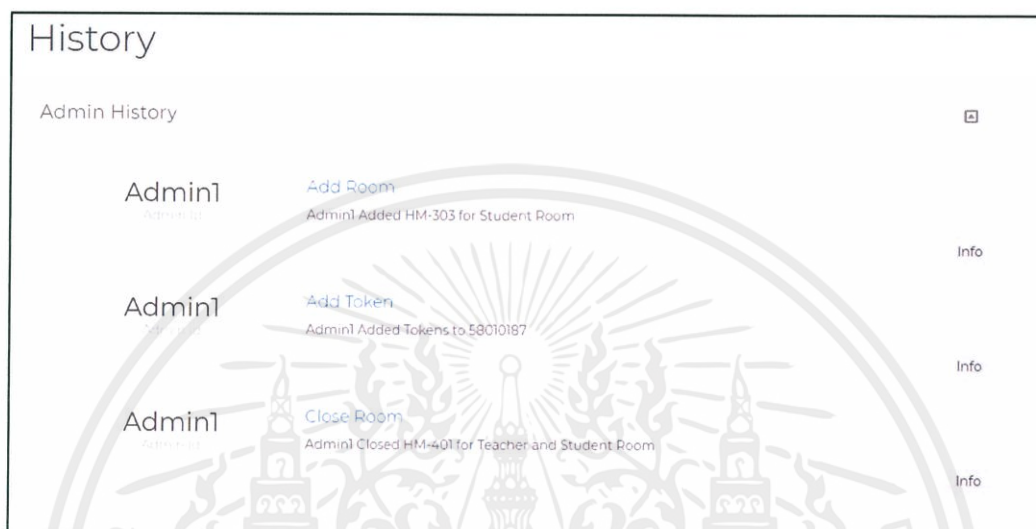
Address: Address

Confirm Back

รูปที่ 4.112 การเพิ่มห้องเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) History จะแสดงผลของการใช้งานต่างๆของผู้ใช้งาน ทั้งการเพิ่มเติม ปรับเปลี่ยน หรือยกเลิก ไม่ว่าจะเป็นส่วนหนึ่งของ User หรือ Classroom จะถูกบันทึกไว้และสามารถเรียกดูข้อมูลการทำงานได้ในภายหลัง ดังรูปที่ 4.113



รูปที่ 4.113 ประวัติการใช้งานของ Admin User

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

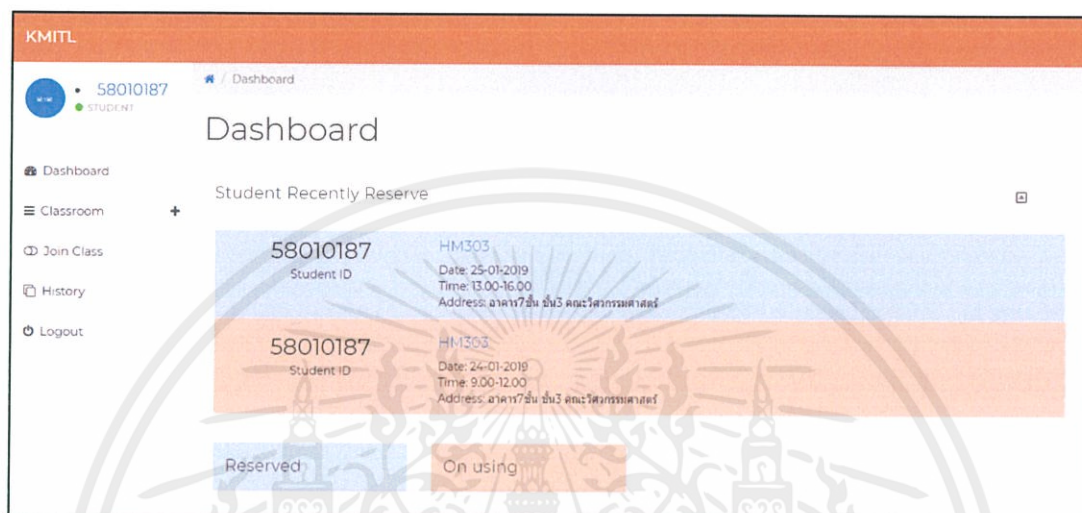
4.2.2 ผลการทดสอบในส่วนของการแสดงผล Student Webpage

ในส่วนผลการทดสอบของ Student Webpage นี้จะเป็นการจัดเก็บผลการทดลอง เหมือนกับหัวข้อ 4.2.1 จากการทดสอบการแสดงผลของ Student Webpage ที่ทำการเชื่อมต่อกับ Blockchain ซึ่งได้ทำการจัดเก็บผลการทดลองตามที่ได้ทำการการออกแบบไว้ โดยเริ่มจากการ register ดังรูปที่ 4.114 ซึ่งในส่วนของการ register จะพบว่าจำเป็นต้องมีการกรอกข้อมูลส่วนตัว เพื่อลงทะเบียนผู้ใช้งานของระบบ

รูปที่ 4.114 การลงทะเบียนผู้ใช้งาน Student User

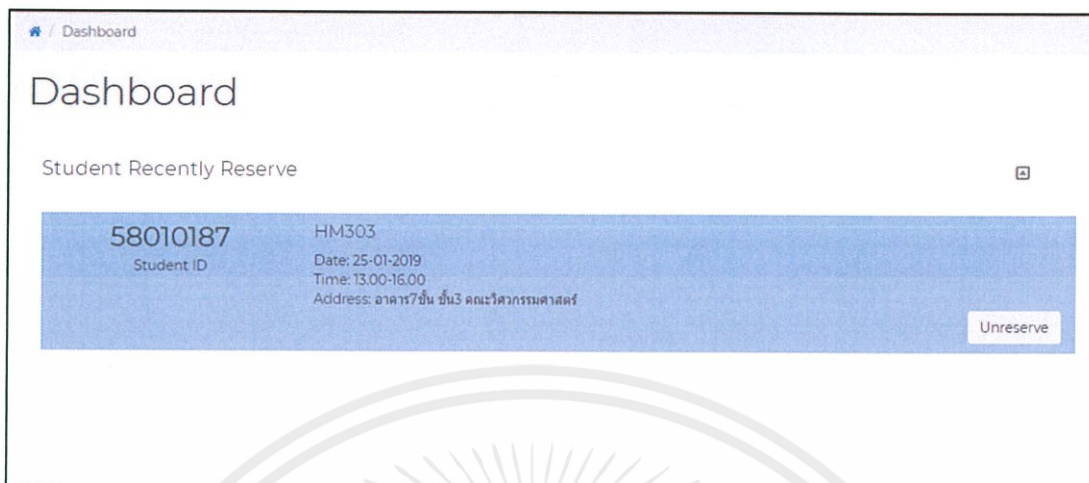
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหัวข้อ 3.1.2.2 ในส่วนของ Student Dashboard ซึ่งจะประกอบไปด้วยส่วนของเมนูต่างๆ ซึ่งจะเป็นหน้าหลักที่คอยรองรับผู้ใช้งาน โดยรูปที่ 4.115 จะเป็นรูปแสดงผลหน้าจอของ Student Dashboard



รูปที่ 4.115 การแสดงผลหน้า Student Dashboard

ในหน้า Dashboard ผู้ใช้งานสามารถดูการใช้งานล่าสุดของตนเองได้ โดยจะมีการใช้สีเพื่อบ่งบอกความหมายของการทำงานนั้น สีส้ม หมายถึง ผู้ใช้งานกำลังใช้งานห้องเรียนอยู่ สีฟ้า หมายถึง ห้องที่ได้ทำการจองไว้แล้วแต่ยังไม่ถึงเวลาใช้งาน อีกทั้งผู้ใช้งานยังสามารถเข้าไปดูรายละเอียดการใช้งานเพิ่มเติมได้ หากเป็นกรณีห้องที่ได้ทำการจองไว้ผู้ใช้งานสามารถสั่งยกเลิกการจองได้โดยไม่ต้องไปที่หน้าการจองห้องเรียน หากเป็นส่วนของห้องเรียนที่กำลังใช้งานจะสามารถทำการปลดล็อคประตูได้จากหน้านี้เช่นกัน ประตูจะทำการปลดล็อคเป็นเวลา 15 นาที แล้วจึงกลับมาทำการล็อคเช่นเดิม โดยผู้ใช้งานสามารถใช้คำสั่งปลดล็อคประตูได้จนกว่าจะหมดเวลาที่ได้ทำการจอง หากหมดเวลาที่ยังสามารถใช้งานได้ผู้ใช้งานสามารถใช้สวิตช์ฉุกเฉินที่ติดตั้งไว้ภายในห้องเพื่อทำการเปิดประตู ทั้งนี้ในส่วนของ Dashboard ได้ทำการแยกย่อยออกเป็น 4 ส่วน ดังรูปที่ 4.116 – 4.121



รูปที่ 4.116 ข้อมูลของห้องเรียนที่ได้ทำการจองไว้บน Student Dashboard

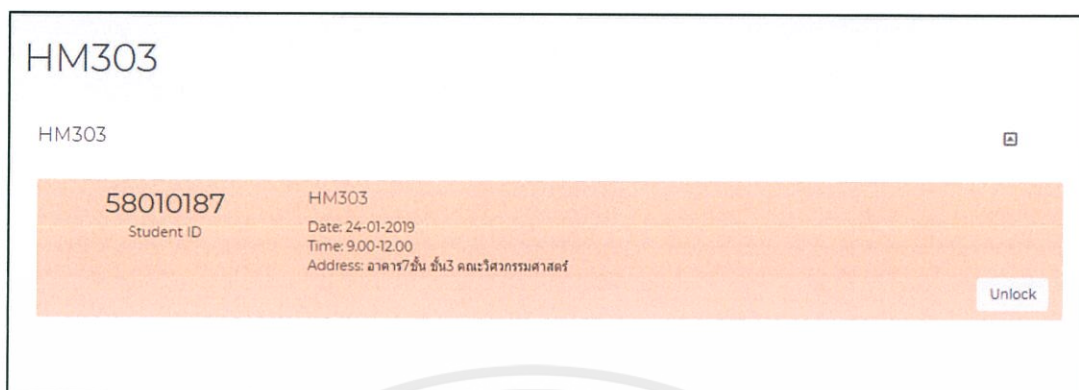


รูปที่ 4.117 การยกเลิกการจองห้องเรียน



รูปที่ 4.118 การแสดงผลหลังจากยกเลิกการจองห้องเรียนบน Student Dashboard

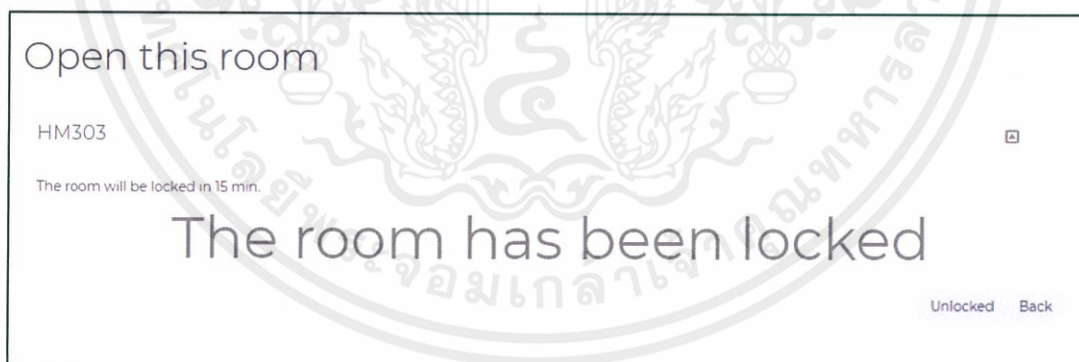
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.119 การแสดงผลห้องเรียนที่กำลังใช้งานบน Student Dashboard



รูปที่ 4.120 ระบบทำการจับเวลา 15 นาที เมื่อทำการปลดล็อกประตู



รูปที่ 4.121 ระบบทำการล็อกประตูหลังจากหมดเวลา 15 นาที

1) Student Profile ในส่วนนี้เป็นส่วนการแสดงผลของข้อมูลของผู้ใช้งาน ซึ่งจะมีการบันทึกข้อมูลที่จำเป็นต่อการใช้งานของผู้ใช้งานดังรูปที่ 4.122 นอกจากนี้ยังสามารถแก้ไขข้อมูลส่วนตัวได้อีกด้วยดังรูปที่ 4.123

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Profile

Student Profile ✖

Student ID: 58010187

Name: Chirabhat

Surname: Kijtham

Tel.: 0859090251

E-mail: k.chirabhat@gmail.com


Faculty: Engineering

Major: Telecommunication

Token: 20

[Edit](#)

รูปที่ 4.122 การแสดงผลข้อมูลผู้ใช้งานของ Student Profile



Student ID: 58010187

Name:

Surname:

Tel.:

E-mail:

Faculty:

Major:

Token: 20

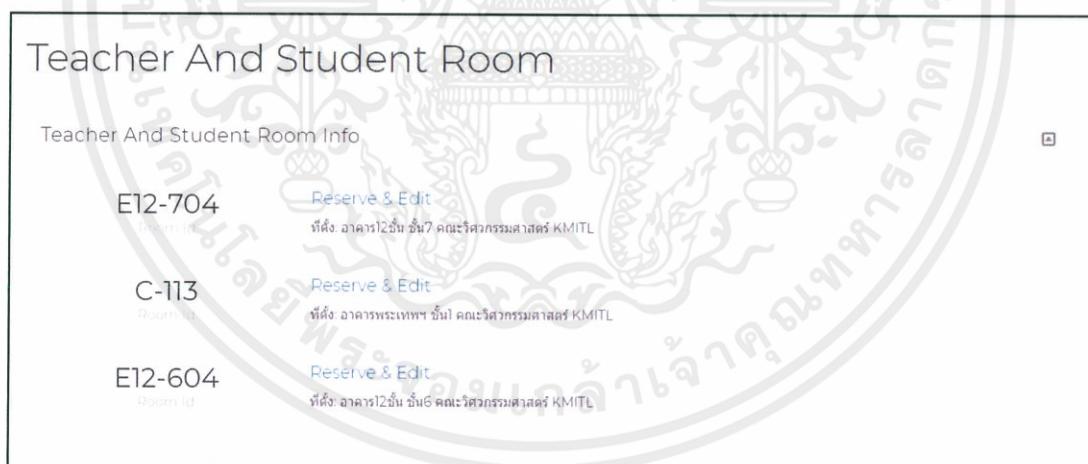
รูปที่ 4.123 การแก้ไขข้อมูลผู้ใช้งานของ Student Profile

2) Classroom ในส่วนนี้จะเป็นส่วนสำหรับการจองห้องเรียน โดยระบบจะแยกห้องเรียนออกเป็น 2 ประเภท ได้แก่ Student Room และ Teacher and Student Room ทั้ง 2 ประเภทจะมีรูปแบบที่เหมือนกัน แต่ในกรณีผู้ใช้งาน Student จะทำการจอง Teacher and Student Room ได้ จำเป็นต้องมี token สำหรับการจองห้องเรียนด้วย จึงจะสามารถทำการใช้งานได้ตามปกติ โดยในกล่องจะมีรายละเอียดของห้องเรียนเพื่อช่วยให้ง่ายต่อการใช้งานด้วย เมื่อผู้ใช้ได้ทำการเลือกห้องเรียนที่ต้องการได้แล้วให้ทำการเลือกวันที่ที่ต้องการทำการจอง จากนั้นให้ทำการเลือกช่วงเวลาที่ยาวอยู่ โดยดูจากสีของกล่อง สีเขียว มีความหมายว่า ห้องว่างสามารถทำการจองได้

สีแดง มีความหมายว่า ห้องนั้นได้ถูกผู้ใช้งานอื่นทำการจองไว้แล้วไม่สามารถเลือกได้ และสีฟ้า มีความหมายว่า ห้องได้ถูกทำการจองโดยผู้ใช้งานเองและสามารถยกเลิกการจองได้ จากนั้นเมื่อเลือกวันจองได้แล้วให้ผู้ใช้ทำการยืนยัน แล้วระบบจะทำการบันทึกไว้และเมื่อถึงเวลาที่ทำการจองจึงจะสามารถใช้งานห้องเรียนได้ ดังรูปที่ 4.124 – 4.131



รูปที่ 4.124 รายการห้องเรียนของ Student Room



รูปที่ 4.125 รายการห้องเรียนของ Teacher and Student Room

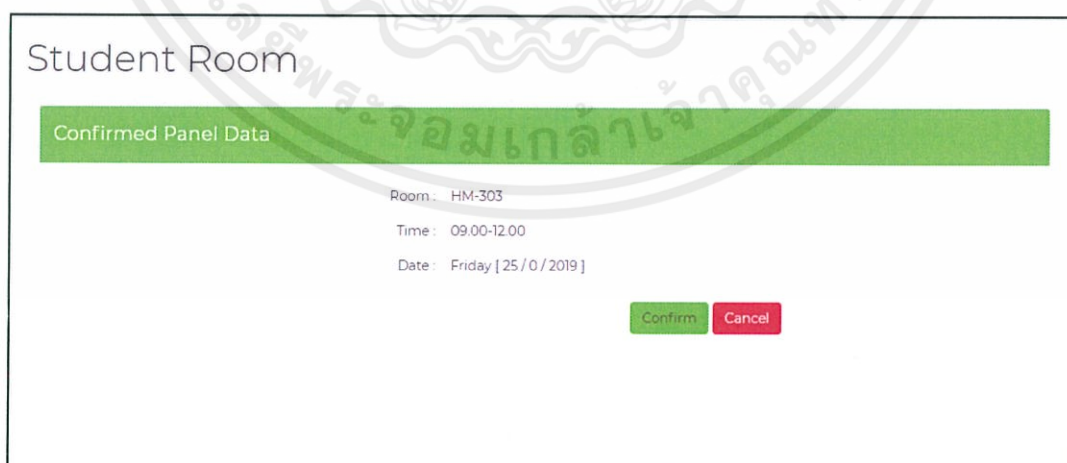
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.126 ปฏิทินสำหรับเลือกวันที่ต้องการจองห้องเรียน Student Room



รูปที่ 4.127 การเลือกช่วงเวลาที่ต้องการจองห้องเรียน Student Room

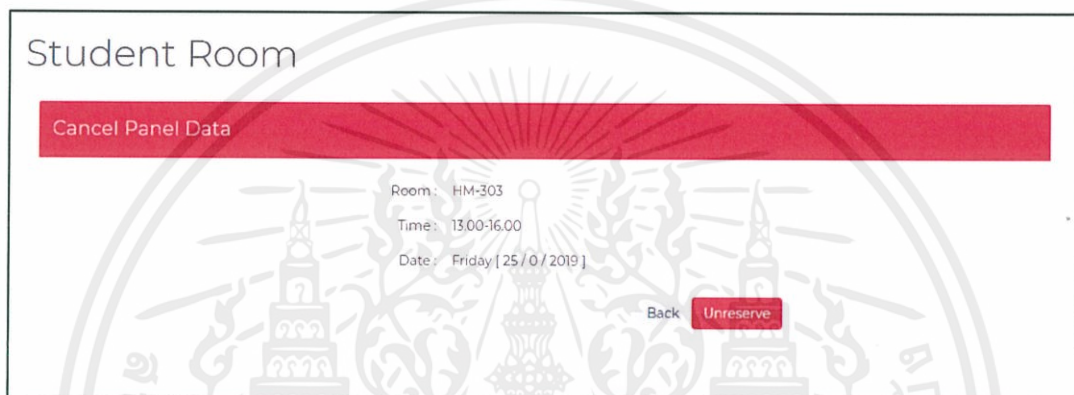


รูปที่ 4.128 การยืนยันการจองห้องเรียน Student Room

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

58010187 Student ID	HM303 Date: 25-01-2019 Time: 9:00-12:00 Address: อาคาร7ชั้น ชั้น3 คณะวิศวกรรมศาสตร์
58010187 Student ID	HM303 Date: 25-01-2019 Time: 13:00-16:00 Address: อาคาร7ชั้น ชั้น3 คณะวิศวกรรมศาสตร์

รูปที่ 4.129 การแสดงผลห้องเรียน Student Room ที่ได้จองไว้บนหน้า Dashboard



รูปที่ 4.130 การยกเลิกการจองห้องเรียน Student Room

Student Recently Reserve	
58010187 Student ID	HM303 Date: 25-01-2019 Time: 9:00-12:00 Address: อาคาร7ชั้น ชั้น3 คณะวิศวกรรมศาสตร์
58010187 Student ID	HM303 Date: 24-01-2019 Time: 9:00-12:00 Address: อาคาร7ชั้น ชั้น3 คณะวิศวกรรมศาสตร์

รูปที่ 4.131 การแสดงผลหลังจากยกเลิกการจองห้องเรียน Student Room

3) Join Class ส่วนต่อมาเป็นส่วนของการเข้าร่วมชั้นเรียนโดยนักเรียนจะได้รับ QR Code จากการสร้างชั้นเรียนของ Teacher User จากนั้นทำการสแกนผ่านทาง การ Join Class ระบบจะทำการบันทึกข้อมูลการเข้าร่วมชั้นเรียนไว้ ซึ่งจะมีรายละเอียดของชั้นเรียนนั้นอยู่ด้วย ดังรูปที่ 4.132 – 4.133

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

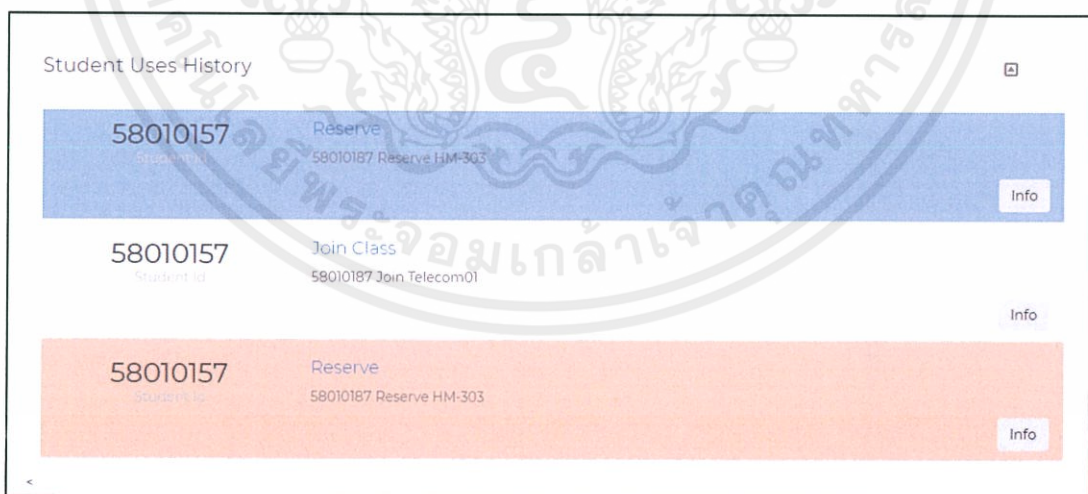


รูปที่ 4.132 การแสดงผลการเลือก QR Code ของชั้นเรียนที่ต้องการเข้าร่วม



รูปที่ 4.133 การแสดงผลการเข้าร่วมชั้นเรียน

4) History ส่วนสุดท้ายเป็นส่วนของประวัติการใช้งาน โดยจะทำการบันทึกข้อมูลการใช้งานทั้งหมดแยกออกเป็นหมวดหมู่สำหรับแต่ละประเภทการใช้งานดังรูปที่ 4.134



รูปที่ 4.134 ประวัติการใช้งานของ Student User

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ผลการทดสอบในส่วนของการแสดงผล Teacher Webpage

ในส่วนผลการทดสอบของ Teacher Webpage นี้จะเป็นการเก็บผลการทดลอง เหมือนกับ 2 ส่วนที่ได้กล่าวไป จากการทดสอบการแสดงผลของ Teacher Webpage ที่ทำการ เชื่อมต่อกับ Blockchain โดยได้ทำการจัดเก็บผลการทดลองตามที่ได้ทำการออกแบบไว้ เริ่มจากการ register ดังรูปที่ 4.135 ซึ่งในส่วนของการ register จำเป็นต้องมีการกรอกข้อมูลเพื่อทำการ ลงทะเบียนผู้ใช้งานเหมือนกับ Student Webpage

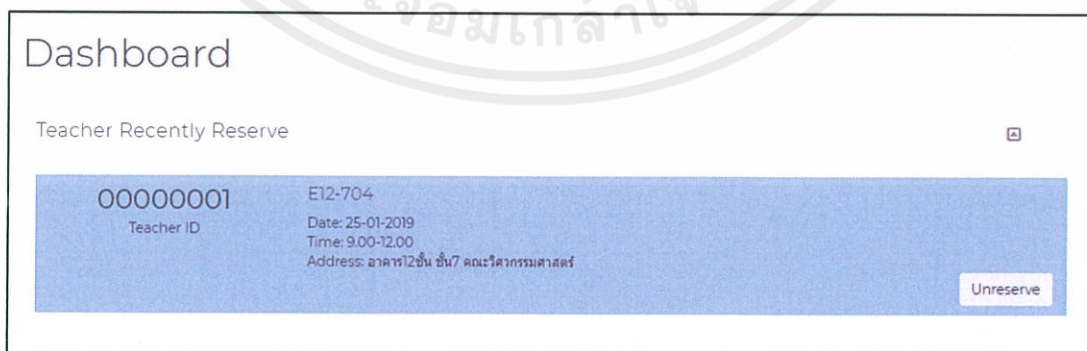
รูปที่ 4.135 การลงทะเบียนผู้ใช้งาน Teacher User

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหัวข้อ 3.1.2.2 ในส่วนของ Teacher Dashboard ซึ่งจะประกอบไปด้วยส่วนของเมนูย่อย ซึ่งจะเป็นหน้าหลักที่คอยรองรับผู้ใช้งาน โดยรูปที่ 4.136 จะเป็นรูปแสดงหน้าจอของ Teacher Dashboard ภายในหน้า Dashboard ผู้ใช้สามารถดูการใช้งานล่าสุดของตนเองได้ โดยจะมีการใช้สีเพื่อบ่งบอกความหมายของการทำงานนั้น สีส้ม หมายถึง ผู้ใช้กำลังใช้งานห้องเรียนอยู่ สีฟ้า หมายถึง ห้องที่ได้ทำการจองไว้แล้วแต่ยังไม่ถึงเวลาใช้งาน อีกทั้งผู้ใช้งานยังสามารถเข้าไปดูรายละเอียดการใช้งานเพิ่มเติมได้ หากเป็นกรณีห้องที่ได้ทำการจองไว้ผู้ใช้งานสามารถสั่งยกเลิกการจองได้โดยไม่ต้องไปที่หน้าการจองห้องเรียน หากเป็นส่วนของห้องเรียนที่กำลังใช้งานจะสามารถทำการปลดล็อคประตูได้จากหน้านี้ ประตูจะทำการปลดล็อคเป็นเวลา 15 นาที แล้วจะกลับมาทำการล็อคดังเดิม โดยผู้ใช้งานสามารถใช้คำสั่งปลดล็อคประตูได้จนกว่าจะหมดเวลาการจอง หากหมดเวลาการจองสามารถใช้สวิตช์ฉุกเฉินที่ติดตั้งไว้ภายในห้องในการเปิดประตู ทั้งนี้ในส่วนของ Dashboard ได้ทำการแยกย่อยออกเป็น 4 ส่วน ดังรูปที่ 4.137 – 4.142

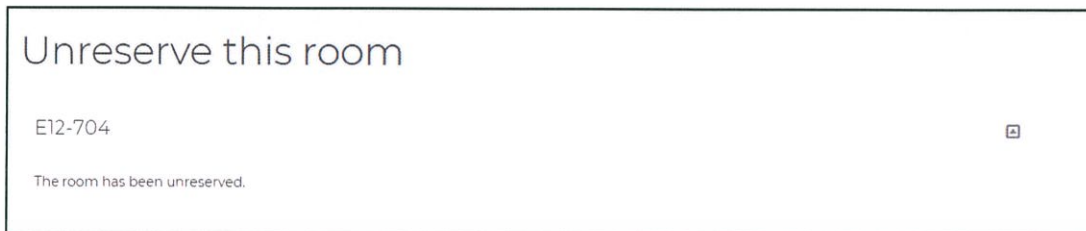


รูปที่ 4.136 การแสดงผลหน้า Teacher Dashboard

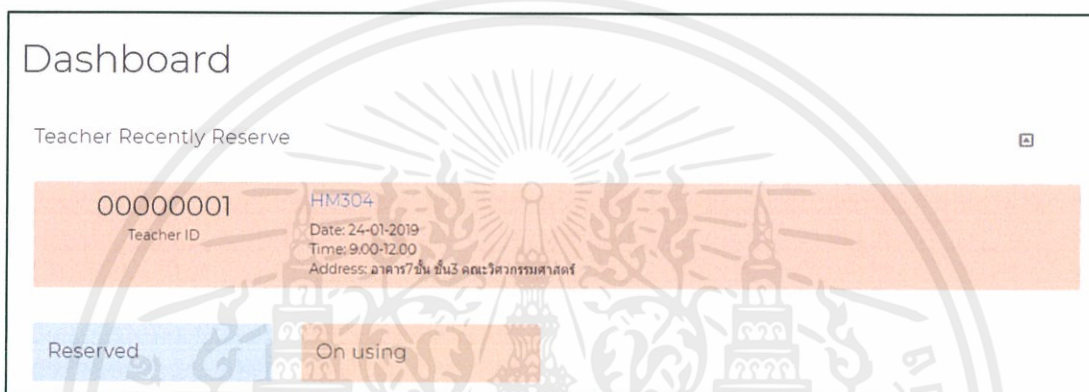


รูปที่ 4.137 ข้อมูลของห้องเรียนที่ได้ทำการจองไว้บน Teacher Dashboard

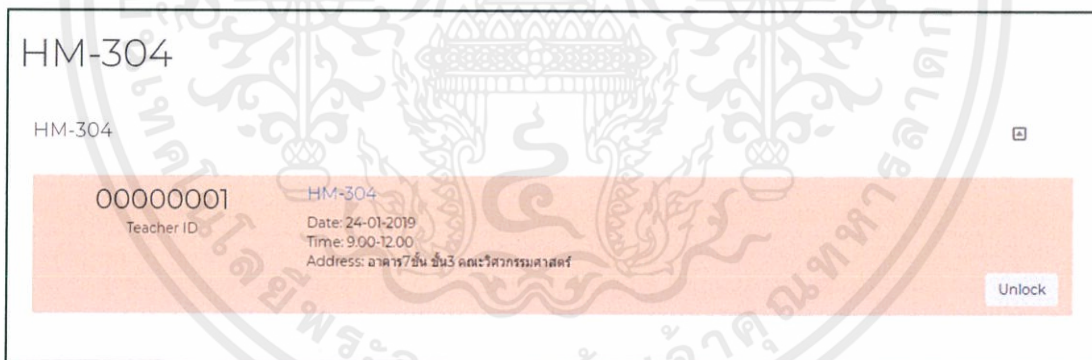
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



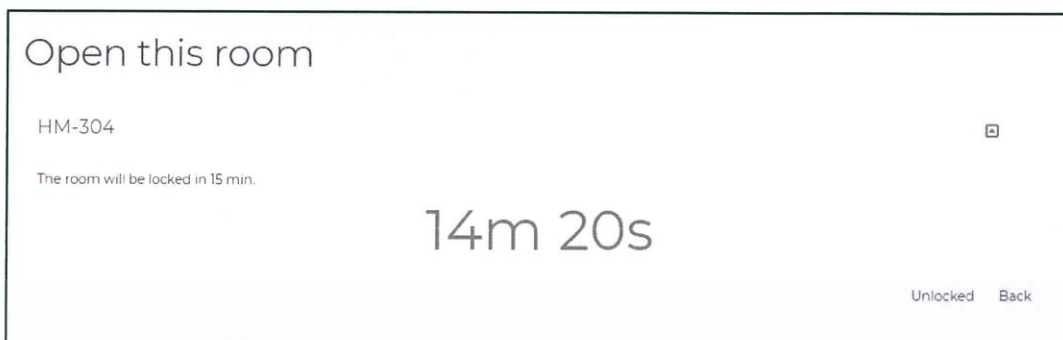
รูปที่ 4.138 การยกเลิกการจองห้องเรียน



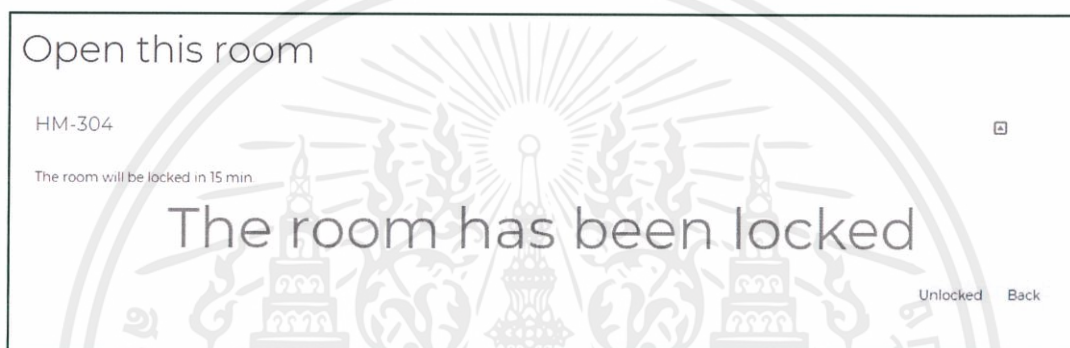
รูปที่ 4.139 การแสดงผลหลังจากยกเลิกการจองห้องเรียนบน Teacher Dashboard



รูปที่ 4.140 การแสดงผลห้องเรียนที่กำลังใช้งานบน Teacher Dashboard



รูปที่ 4.141 ระบบทำการจับเวลา 15 นาที เมื่อทำการปลดล็อคประตู



รูปที่ 4.142 ระบบทำการล็อคประตูหลังจากหมดเวลา 15 นาที

1) Teacher Profile ในส่วนนี้เป็นส่วนการแสดงผลของข้อมูลของผู้ใช้งาน ซึ่งจะมีการบันทึกข้อมูลที่จำเป็นต่อการใช้งานของผู้ใช้งานดังรูปที่ 4.143 นอกจากนี้ยังสามารถแก้ไขข้อมูลส่วนตัวได้อีกด้วยดังรูปที่ 4.144



รูปที่ 4.143 การแสดงผลข้อมูลผู้ใช้งานของ Teacher Profile

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Teacher ID: 00000001

Position:

Name:

Surname:

Tel.:

E-mail:

Faculty:

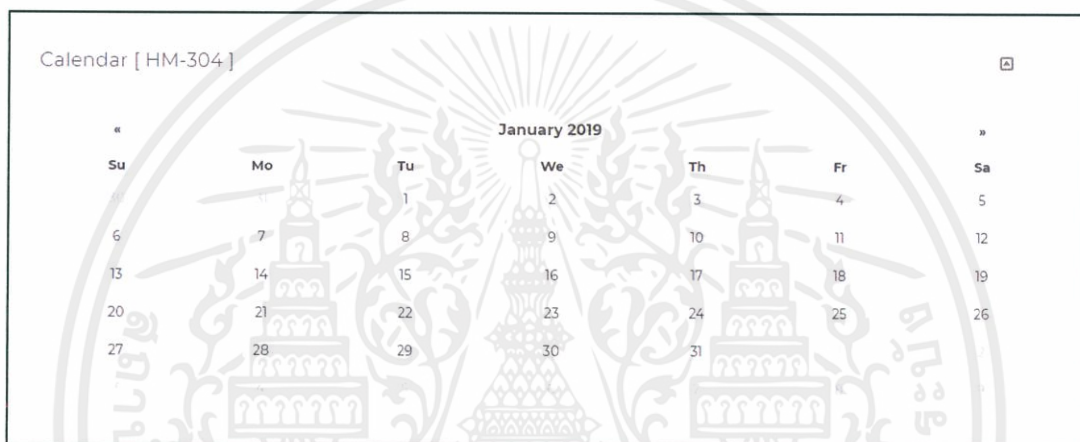
Major:

รูปที่ 4.144 การแก้ไขข้อมูลผู้ใช้งานของ Teacher Profile

2) Classroom ในส่วนนี้จะเป็นส่วนสำหรับการจองห้องเรียนโดยระบบจะแยกห้องเรียนออกเป็น 2 ประเภท ได้แก่ Teacher Room และ Teacher and Student Room ทั้ง 2 ประเภทจะมีรูปแบบการใช้งานที่เหมือนกัน แต่ในกรณีผู้ใช้งาน Teacher User จะทำการจอง Teacher and Student Room ได้ 2 วิธี คือ การจองแบบปกติ และการจองแบบสร้างชั้นเรียน ซึ่งจะต้องทำการสร้างชั้นเรียนขึ้นมาเพื่อตรวจสอบการเข้าชั้นเรียนจึงจะสามารถใช้งานได้ เมื่อผู้ใช้ได้ทำการเลือกห้องเรียนที่ต้องการแล้ว ให้ทำการเลือกวันที่ที่ต้องการทำการจอง จากนั้นให้ทำการเลือกช่วงเวลาที่ยังว่างอยู่ โดยดูจากสีของกล่อง สีเขียว มีความหมายว่า ห้องว่างสามารถจองได้ สีแดง มีความหมายว่า ห้องนั้นได้ถูกผู้ใช้งานอื่นทำการจองไว้แล้ว ไม่สามารถเลือกได้ และสีฟ้า มีความหมายว่า ห้องได้ถูกทำการจองโดยผู้ใช้งานเอง สามารถยกเลิกการจองได้ หลังจากเลือกวันเวลาที่ต้องการได้แล้วให้ผู้ใช้ทำการยืนยันแล้วระบบจะทำการบันทึกไว้ และเมื่อถึงวันเวลาที่ทำการจองจึงจะสามารถใช้งานห้องเรียนได้ ดังรูปที่ 4.145 – 4.149

HM-304 <small>Room Id</small>	Reserve & Edit ที่ตั้ง: อาคาร12ชั้น ชั้น7 คณะวิศวกรรมศาสตร์ KMUTL
E12-305 <small>Room Id</small>	Reserve & Edit ที่ตั้ง: อาคาร12ชั้น ชั้น1 คณะวิศวกรรมศาสตร์ KMUTL
T-101 <small>Room Id</small>	Reserve & Edit ที่ตั้ง: อาคารโทรคมนาคม ชั้น6 คณะวิศวกรรมศาสตร์ KMUTL

รูปที่ 4.145 รายการห้องเรียนของ Teacher Room



รูปที่ 4.146 ปฏิทินสำหรับเลือกวันที่ต้องการจองห้องเรียน Teacher Room

teacher Room

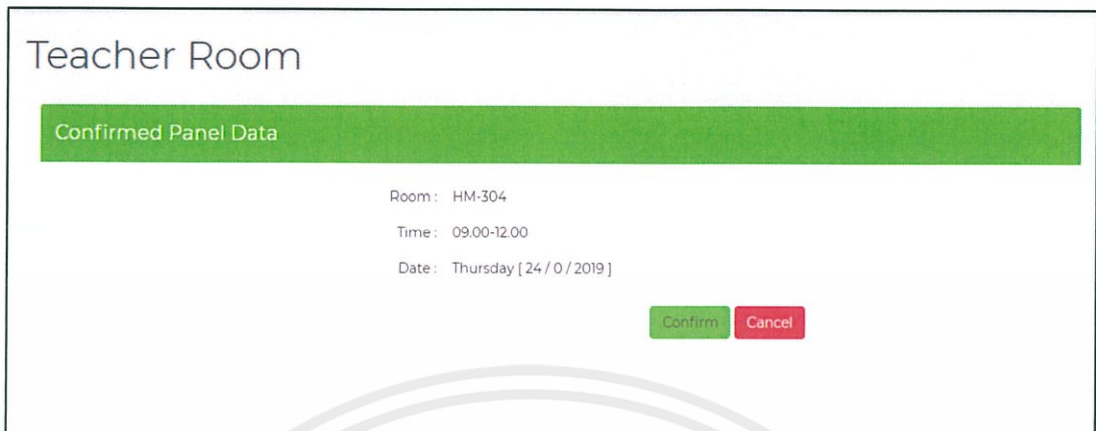
Choose Time to Reserve

9.00-12.00 Date:	Room Id: HM-304	Reserve
13.00-16.00 Date:	Room Id: HM-304	Cancel

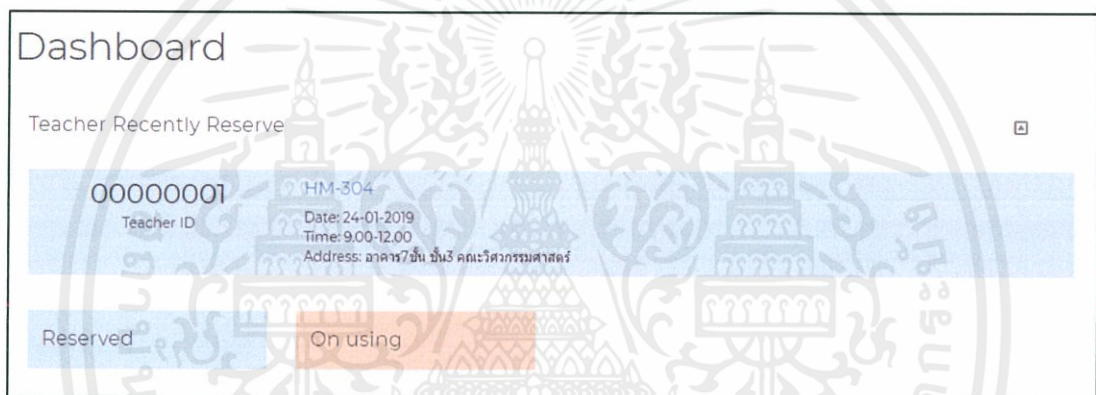
Available Unavailable Reserved

รูปที่ 4.147 การเลือกช่วงเวลาที่ต้องการจองห้องเรียน Teacher Room

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.148 การยืนยันการจองห้องเรียน Teacher Room

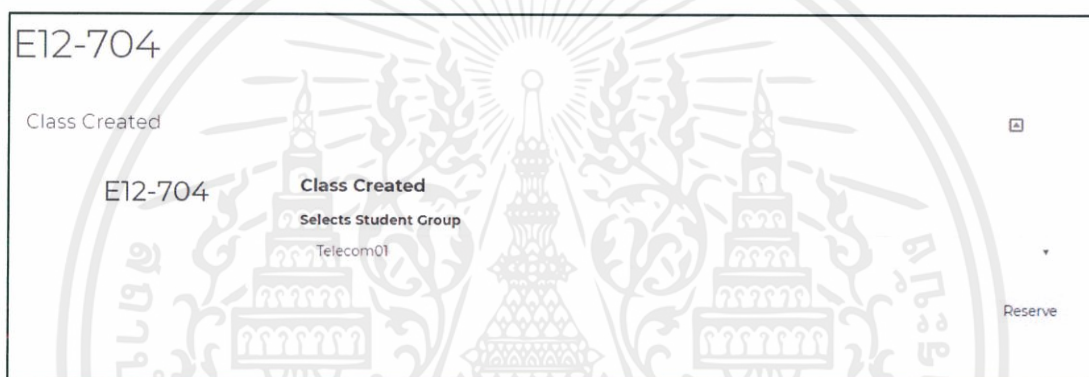


รูปที่ 4.149 การแสดงผลห้องเรียน Teacher Room ที่ได้จองไว้บนหน้า Dashboard

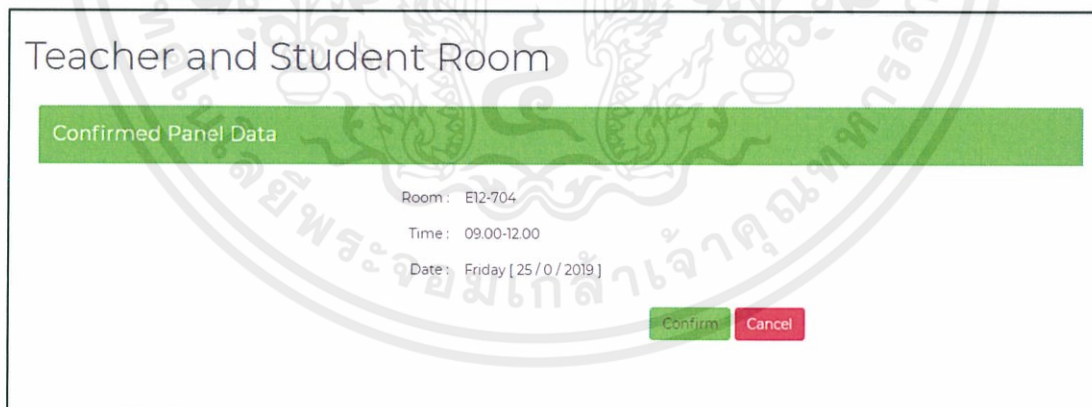
ในกรณีของ Teacher and Student Room ผู้ใช้สามารถเลือกวิธีการจองได้ หากจองแบบปกติสามารถจองได้ด้วยวิธีเดียวกันกับ Teacher Room แต่หากจะจองแบบสร้างชั้นเรียน ผู้ใช้จะต้องทำการเลือกรายชื่อชั้นเรียนที่สร้างไว้ใน Create Class ซึ่งเนื้อหาจะอยู่ในส่วนถัดไป จากนั้นจึงทำการจองปกติ โดย QR Code ที่ได้จากการสร้างชั้นเรียนนี้จะใช้ในการตรวจสอบการเข้าห้องเรียนในครั้งนั้นๆของ Student User ดังรูปที่ 4.150 – 4.152



รูปที่ 4.150 รายชื่อห้องเรียนของ Teacher and Student Room



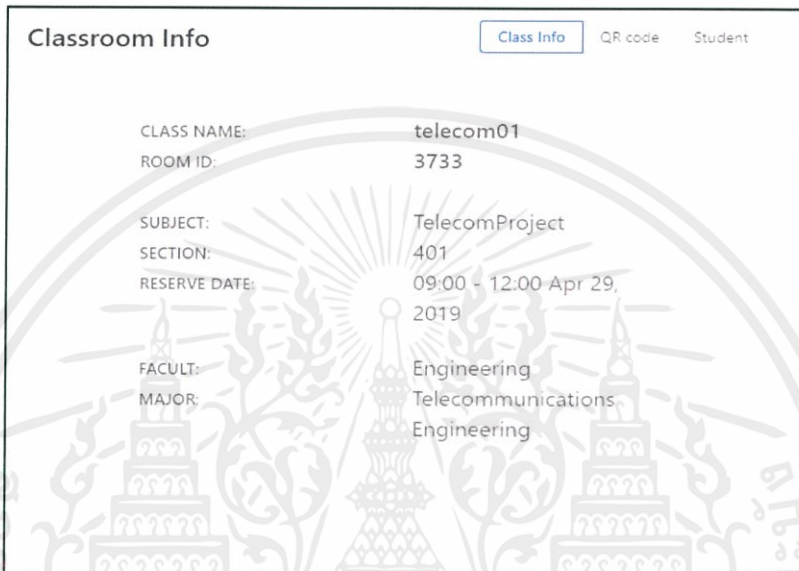
รูปที่ 4.151 การเลือกรายชื่อชั้นเรียนที่ต้องการ



รูปที่ 4.152 การยืนยันการจองห้องเรียนแบบสร้างชั้นเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Create Class ส่วนต่อมาเป็นส่วนของการสร้างชั้นเรียน จะเห็นได้จากการสร้างชั้นเรียนในการจองห้องเรียนของ Teacher and Student Room นั้นสามารถเลือกชั้นเรียนที่ได้ Create Class ไว้เพื่อใช้ในการจัดเก็บรายชื่อ Student User ผ่านการสแกน QR Code โดยจะประกอบด้วย ข้อมูลของชั้นเรียน, QR Code และรายชื่อ Student User ดังรูปที่ 4.153 – 4.156



The screenshot shows a 'Classroom Info' form with the following details:

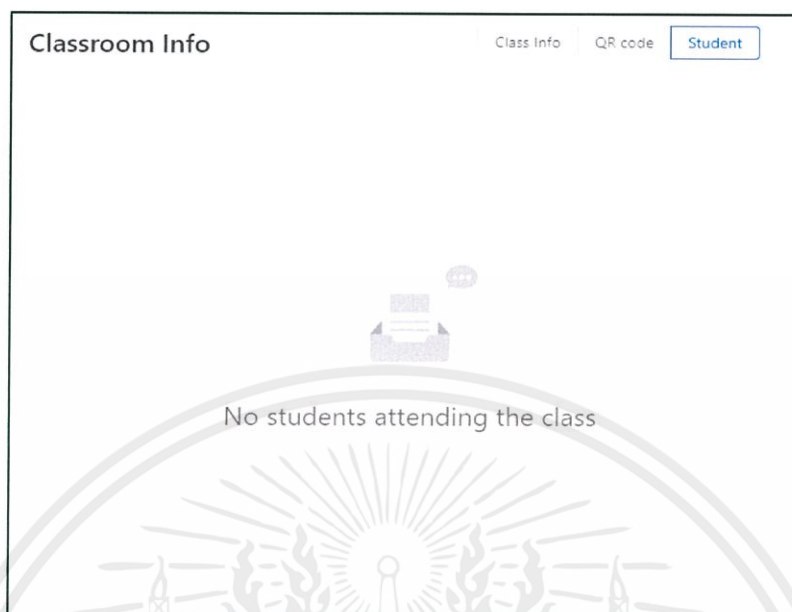
Field	Value
CLASS NAME:	telecom01
ROOM ID:	3733
SUBJECT:	TelecomProject
SECTION:	401
RESERVE DATE:	09:00 - 12:00 Apr 29, 2019
FACULT:	Engineering
MAJOR:	Telecommunications Engineering

รูปที่ 4.153 ข้อมูลของชั้นเรียน

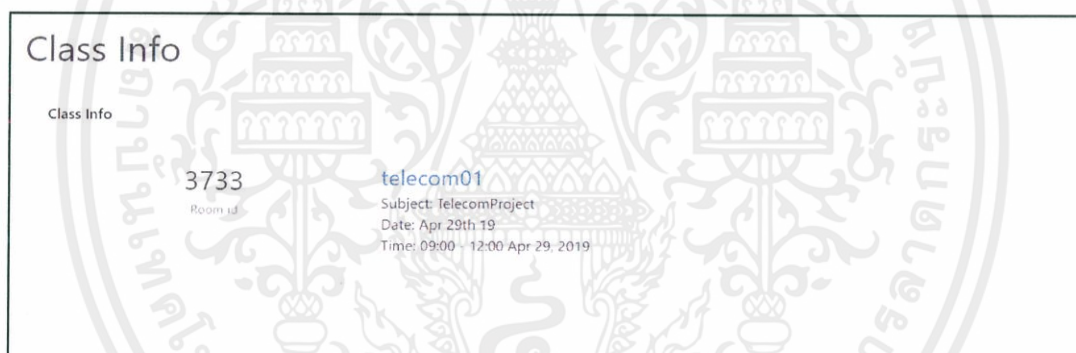


รูปที่ 4.154 QR Code ที่ใช้ในการสแกนเพื่อ login เข้าชั้นเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.155 หน้าแสดงรายชื่อ Student User ที่จัดเก็บผ่าน QR Code



รูปที่ 4.156 การแสดงผลรายชื่อชั้นเรียนที่ได้สร้างไว้

4) Add Token ในส่วนการ Add Token ของ Teacher User จะมีความแตกต่างกับของ Admin User เล็กน้อย โดยในส่วนของ Teacher User จะสามารถ Add Token ได้เฉพาะ Student User ที่อยู่ในชั้นเรียนของผู้ใช้เท่านั้น ดังรูปที่ 4.157 – 4.160

Add Token

Choose Class ☰

Choose Class **Teacher ID: 000000001**

Class Name

Telecom01 ▼

Choose

รูปที่ 4.157 การเลือกรายชื่อชั้นเรียนเพื่อ Add Token

Telecom01

Student in Telecom01 ☰

58010187 <small>Student ID</small>	Name: Chirabhat Surname: Kijtham	Add Token
58010211 <small>Student ID</small>	Name: Jessada Surname: T.	Add Token
58010353 <small>Student ID</small>	Name: Nutchaphon Surname: U.	Add Token

รูปที่ 4.158 การเลือกรายชื่อ Student User จากชั้นเรียน

Add Token

Add Token for 58010187 ☰

Student ID: 58010187

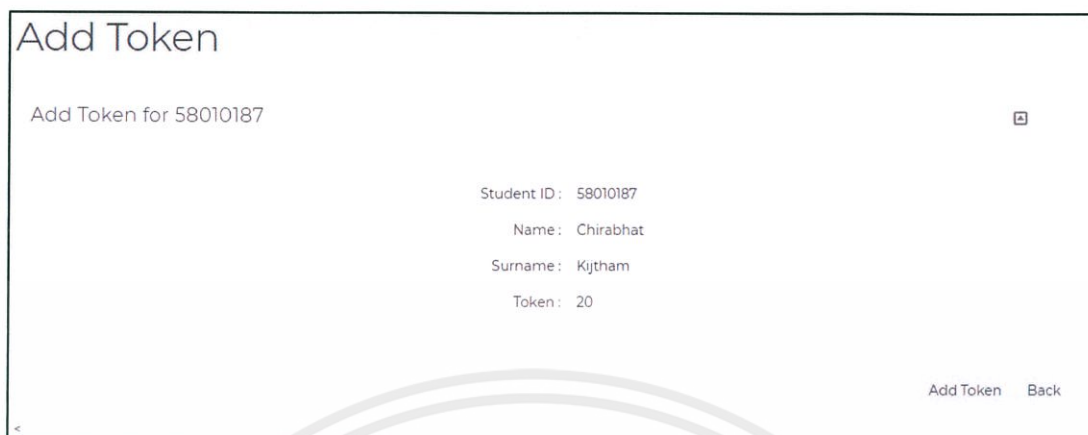
Name: Chirabhat

Surname: Kijtham

Token:

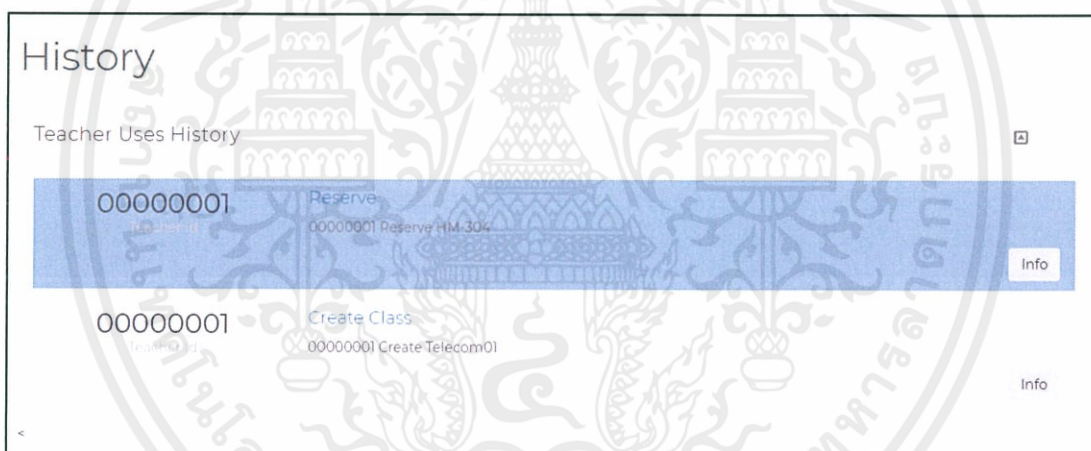
[Add Token](#) [Back](#)

รูปที่ 4.159 การ Add Token ให้กับ Student User



รูปที่ 4.160 การแสดงผลหลังจากทำการ Add Token

5) History ส่วนสุดท้ายเป็นส่วนของประวัติการใช้งาน โดยส่วนนี้จะทำการบันทึกข้อมูลการใช้งานทั้งหมดแยกออกเป็นหมวดหมู่สำหรับแต่ละประเภทการใช้งาน ดังรูปที่ 4.161



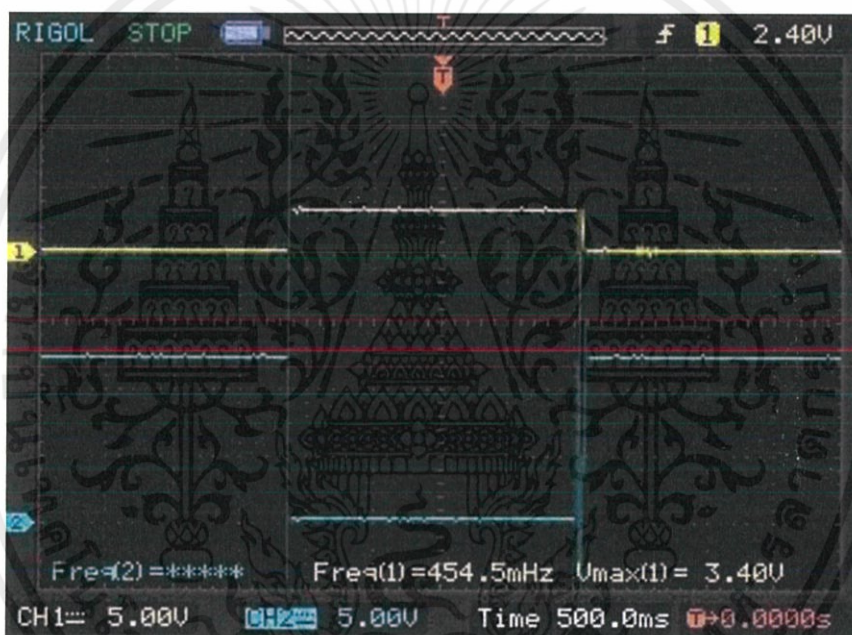
รูปที่ 4.161 ประวัติการใช้งานของ Teacher User

4.3 ผลการทดลองในส่วนควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า

4.3.1 ผลการทดสอบวงจรควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า

ในการทดลองนี้ได้กำหนดให้ Raspberry Pi ทำการจ่ายแรงดันไฟตรงที่ขา GPIO 2 สถานะสลับกัน คือ 0V และ 3.3V โดยกำหนดระยะเวลาในการจ่ายสัญญาณจาก Raspberry Pi ไว้ที่ 10 วินาที เพื่อความสะดวกในการบันทึกผล ซึ่งสัญญาณนี้ถูกป้อนเข้าที่ขาเบสของทรานซิสเตอร์แล้วทำการวัดสัญญาณเอาต์พุตของรีเลย์ ซึ่งจะเป็นสัญญาณไปควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า และแสดงผลการทดลองในรูปที่ 4.162 โดยช่อง 1 คือ สัญญาณที่ขาเบสของ

ทรานซิสเตอร์ซึ่งเป็นสัญญาณที่ได้รับมาจากขา GPIO ของ Raspberry Pi และช่อง 2 คือ สัญญาณเอาต์พุตของรีเลย์ที่ไปควบคุมกลอนแม่เหล็กไฟฟ้า โดยเริ่มแรกรีเลย์จะจ่ายกระแสไฟตรงแรงดัน 12V เพื่อให้กลอนแม่เหล็กไฟฟ้าทำงานตามปกติ และเมื่อทำการสั่งให้ Raspberry Pi จ่ายสัญญาณแรงดันไฟตรง 3.3V ไปยังขาเบสของทรานซิสเตอร์ ทำให้มีแรงดันไฟเลี้ยง 5V เข้าสู่รีเลย์เพื่อทำให้รีเลย์ทำงานและทำการสับสวิทซ์ตัดกระแสไฟที่จ่ายไปยังกลอนแม่เหล็กไฟฟ้า ทำให้สัญญาณเอาต์พุตของรีเลย์มีค่าเป็น 0V และทำให้กลอนแม่เหล็กไฟฟ้าหยุดการทำงาน และหลังจาก 10 วินาทีผ่านไป Raspberry Pi หยุดการจ่ายสัญญาณทำให้ไม่มีแรงดันไฟเลี้ยงไปยังรีเลย์ กระแสไฟจึงไม่ถูกตัดทำให้สัญญาณเอาต์พุตจากรีเลย์กลับมาเป็น 12V และกลอนแม่เหล็กไฟฟ้ากลับมาทำงาน

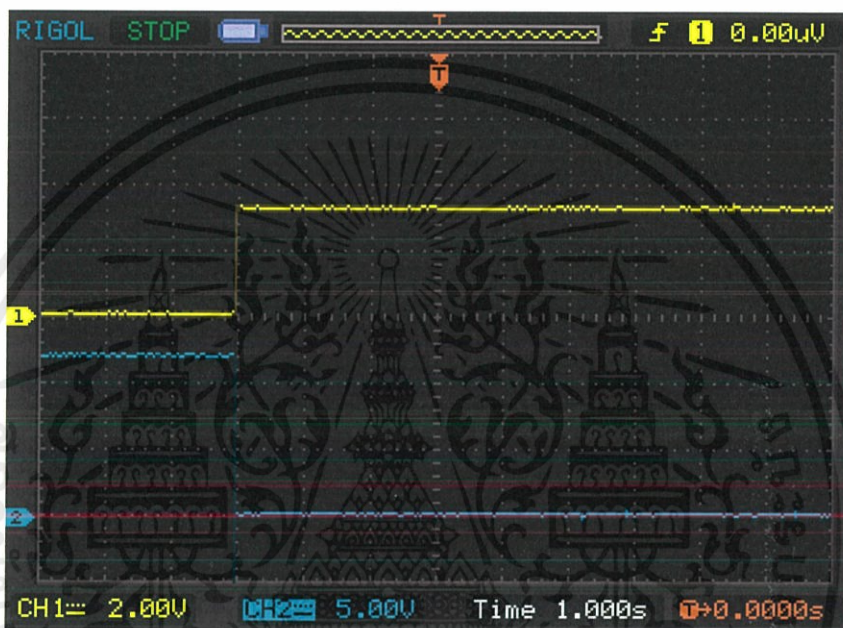


รูปที่ 4.162 ผลการทดลองการทำงานของวงจรควบคุมกลอนแม่เหล็กไฟฟ้า

4.3.2 ผลการทดสอบการควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้าผ่านไมโครคอนโทรลเลอร์

ในการทดลองนี้จะใช้ Raspberry Pi เป็นไมโครคอนโทรลเลอร์ในการรับค่าตัวแปร Room Key จาก API เพื่อใช้กำหนดเงื่อนไขในการจ่ายสัญญาณอินพุตเข้าสู่ขาเบสของทรานซิสเตอร์ และทำการวัดสัญญาณเอาต์พุตที่ออกจากขาของรีเลย์ไปควบคุมการทำงานของกลอนแม่เหล็กไฟฟ้า มีผลการทดลองดังแสดงในรูปที่ 4.163 โดยช่อง 1 คือ สัญญาณอินพุตจาก GPIO ของ Raspberry Pi เข้าสู่ขาเบสของทรานซิสเตอร์ และช่อง 2 คือ สัญญาณเอาต์พุตจากรีเลย์ที่ไปควบคุมการทำงาน

ของกลอนแม่เหล็ก หากตัวแปร Room Key มีค่าเป็น true Raspberry Pi จะทำการจ่ายแรงดันไฟตรง 3.3V เข้าสู่ขาเบสของทรานซิสเตอร์เพื่อให้รีเลย์ทำงานและทำการตัดกระแสไฟ 12V ที่จ่ายให้กับกลอนแม่เหล็กไฟฟ้าเป็นระยะเวลา 15 นาที ทำให้สัญญาณเอาต์พุตจากรีเลย์มีค่าแรงดัน 0V กลอนแม่เหล็กไฟฟ้าหยุดการทำงาน จากนั้นจึงหยุดการจ่ายสัญญาณและทำการส่ง Transaction ผ่าน API เพื่อทำการเปลี่ยนค่าตัวแปร Room Key ให้มีค่าเป็น false

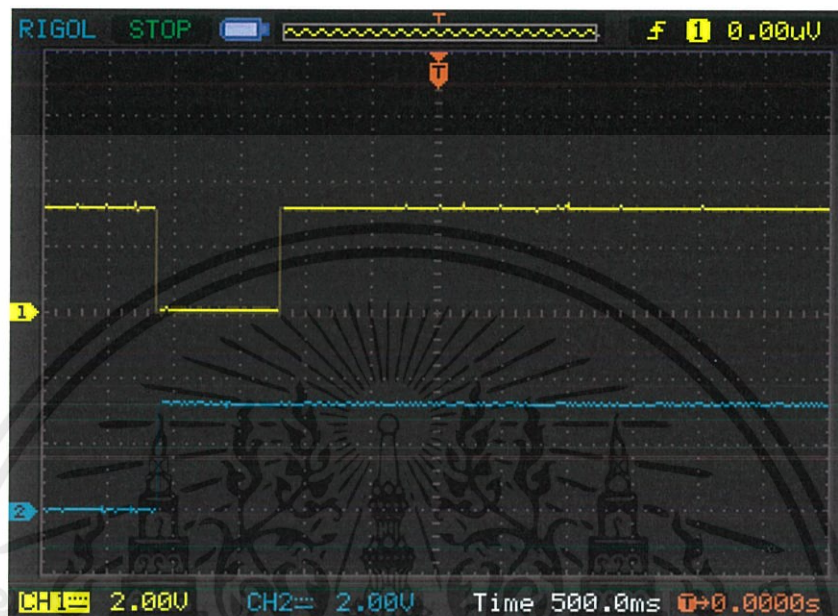


รูปที่ 4.163 ผลการทดลองการทำงานวงจรควบคุมกลอนแม่เหล็กไฟฟ้าผ่านไมโครคอนโทรลเลอร์

4.3.3 ผลการทดสอบการทำงานของสวิทช์ในการควบคุมการทำงานของไมโครคอนโทรลเลอร์

กำหนดให้สวิทช์เป็นสัญญาณอินพุตที่ต่อกับขา GPIO ของ Raspberry Pi โดยมี 2 สถานะ คือ 3.3V (HIGH) และ 0V (LOW) ซึ่งเป็นสัญญาณที่ใช้ควบคุมการจ่ายสัญญาณแรงดันไฟตรง 3.3V ของ Raspberry Pi ได้อีกทางหนึ่ง โดยมีผลการทดลองดังรูปที่ 4.164 ช่อง 1 คือ สัญญาณอินพุตจากสวิทช์เข้าสู่ขา GPIO ของ Raspberry Pi และช่อง 2 คือ สัญญาณเอาต์พุตจากขา GPIO เข้าสู่วงจรที่ขาเบสของทรานซิสเตอร์ โดยสวิทช์นั้นจะทำการต่อกับขาอินพุตของ GPIO และขากราวด์ของ Raspberry Pi ซึ่งจะทำงานแบบ Active LOW คือ สัญญาณอินพุตจะคงสถานะมีค่าเป็น 3.3V (HIGH) ตลอดเวลา เมื่อกดสวิทช์จะทำให้ขาอินพุตของ GPIO นั้นเชื่อมกับกราวด์ทำให้สัญญาณอินพุตมีค่าเป็น 0V (LOW) และสั่งให้ Raspberry Pi จ่ายสัญญาณเอาต์พุตแรงดันไฟตรง

3.3V ไปยังทรานซิสเตอร์ทำให้กลอนแม่เหล็กไฟฟ้าหยุดทำงานเป็นระยะเวลา 10 วินาที จากนั้นจึงหยุดจ่ายสัญญาณทำให้กลอนแม่เหล็กไฟฟ้ากลับมาทำงาน



รูปที่ 4.164 ผลการทดลองการทำงานของปุ่มสวิตซ์ในการควบคุมไมโครคอนโทรลเลอร์

4.4 ผลการทดลองในส่วนการตรวจสอบการเข้าชั้นเรียนผ่านการสแกน QR Code

ทำการทดลองโดยใช้บัญชีผู้ใช้ของอาจารย์ในการ Create Class ขึ้นมาดังรูปที่ 4.165 โดยทำการกรอกข้อมูลที่จำเป็นทั้งหมด เมื่อทำการ Create Class จะสามารถเรียกดูข้อมูลของชั้นเรียนที่สร้างขึ้น ซึ่งจะประกอบด้วย รายละเอียดของชั้นเรียน, QR Code และรายชื่อนักศึกษาที่ทำการ Join Class เข้ามาผ่านการสแกน QR Code ดังรูปที่ 4.166 – 4.169 ตามลำดับ

Create Class Next Cancel

1 **Create** 2 Reserve 3 Confirm
 Create class Reserve room Finish process

* Class name: * Subject:

* Faculty: * Major:
 ----Please select----

* Section:

รูปที่ 4.165 หน้าสำหรับกรอกข้อมูลในการ Create Class

[/ Class Info](#)

Class Info

Class Info

E12-806 Project01
 Room id Subject: Project
Date: May 30th 19
Time: 09.00 - 12.00 May 30, 2019

รูปที่ 4.166 หน้าแสดงผลชั้นเรียนที่ได้ทำการสร้างไว้

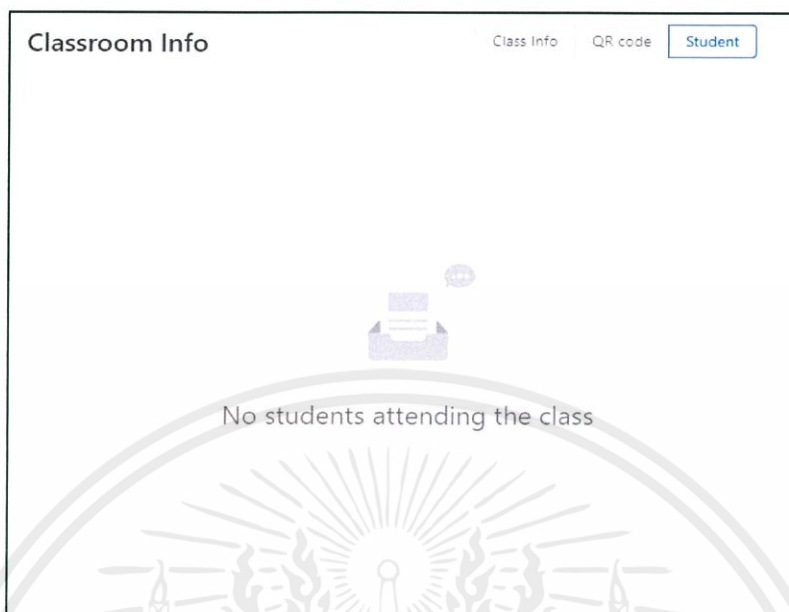
Classroom Info		Class Info	QR code	Student
CLASS NAME:	Project01			
ROOM ID:	E12-806			
SUBJECT:	Project			
SECTION:	01			
RESERVE DATE:	09:00 - 12:00 May 30, 2019			
FACULT:	Engineering			
MAJOR:	Telecommunications Engineering			

รูปที่ 4.167 รายละเอียดของชั้นเรียน

Classroom Info		Class Info	QR code	Student
				

รูปที่ 4.168 QR Code ของชั้นเรียนที่ใช้ในการจัดเก็บรายชื่อนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

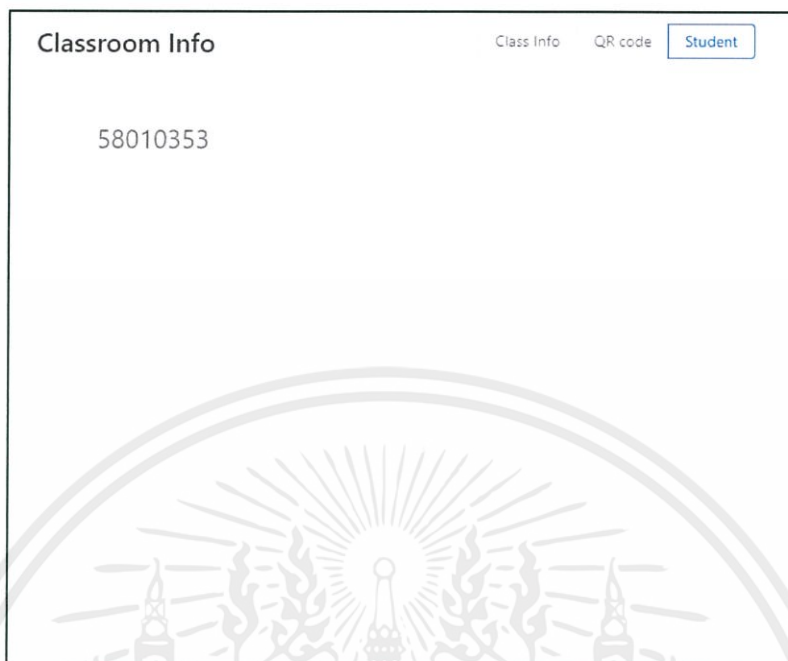


รูปที่ 4.169 หน้าสำหรับแสดงรายชื่อนักศึกษาที่ทำการ Join Class

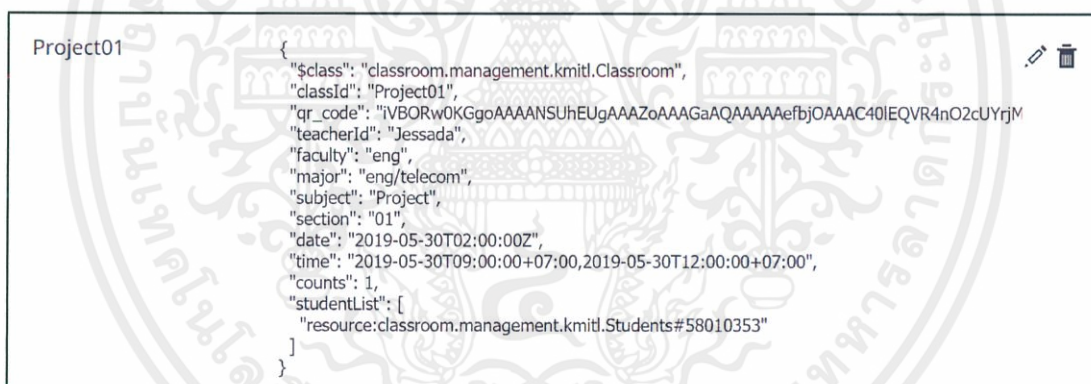
จากนั้นทดลองทำการสแกน QR Code ในรูปที่ 4.168 และทำการ login เข้าสู่ระบบ โดยใช้บัญชีผู้ใช้ของนักศึกษา เพื่อทำการจัดเก็บ ID ของนักศึกษาเข้ามายังรายชื่อนักศึกษาของ Classroom เพื่อเป็นการยืนยันการเข้าชั้นเรียน ดังรูปที่ 4.170 – 4.171 และบันทึกผลการจัดเก็บข้อมูลของระบบผ่าน Composer Playground ดังรูปที่ 4.172

 A screenshot of a login form. At the top left, there are two links: "Log in" (underlined) and "Sign up". Below these are two input fields: the first contains the email address "58010353@kmitl.ac.th" and the second contains a masked password "****". Under the password field, the text "Student" is displayed with a checkmark icon to its right. Below that is a "Remember me" checkbox. At the bottom, there is a blue "Log in" button followed by the text "Or Forgot password". The background features a large, faint watermark of a university seal.

รูปที่ 4.170 หน้าสำหรับทำการ login เข้าสู่ระบบเพื่อ Join Class



รูปที่ 4.171 หน้าแสดงรายชื่อนักศึกษาที่ Join Class เข้ามาแล้ว



รูปที่ 4.172 หน้าแสดงผลการจัดเก็บข้อมูลชั้นเรียนของ Blockchain

บทที่ 5

สรุปผลและข้อเสนอแนะ

เนื่องจากทางคณะผู้จัดทำได้ทำการศึกษาและนำเทคโนโลยี Blockchain มาปรับใช้กับโครงการชิ้นนี้ ทำให้ทางคณะผู้จัดทำเล็งเห็นถึงความเป็นไปได้ที่น่าสนใจในเทคโนโลยี Blockchain เนื่องจาก Blockchain เป็นเทคโนโลยีใหม่ที่มีการพัฒนาอย่างต่อเนื่อง อีกทั้งยังเป็นที่ได้รับความสนใจจากหลากหลายสาขาอาชีพ ไม่ว่าจะเป็นเชิง ธุรกิจ อุตสาหกรรม หรือเทคโนโลยีพื้นฐาน ทางคณะผู้จัดทำจึงมีความเห็นว่าเทคโนโลยี Blockchain สามารถเป็นส่วนหนึ่งและส่วนสำคัญของการทำงานในอนาคต โดยผลลัพธ์ที่ได้จากการทำการทดลองนำระบบ Blockchain มาประยุกต์ใช้ร่วมกับระบบจัดการการลงทะเบียน สามารถสรุปได้ดังนี้

5.1 สรุปผล

จากวัตถุประสงค์และผลของการทดลองในครั้งนี้มีความสอดคล้องและเป็นไปตามเป้าหมาย โดยสามารถสรุปผลการดำเนินงานได้ดังนี้

- 1) สามารถนำแนวทางการทำงานของระบบ Hyperledger Fabric Blockchain มาประยุกต์ใช้ในการเก็บข้อมูลการทำงานของระบบจัดการการลงทะเบียนได้อย่างมีประสิทธิภาพ
- 2) สามารถใช้งานระบบการจัดการการลงทะเบียนได้อย่างมีประสิทธิภาพตามที่คาดหวังไว้ โดยสามารถจัดการการลงทะเบียนอย่างเป็นระบบและมีความสอดคล้องกับการใช้งานจริง รวมถึงการตรวจสอบการเข้าเรียนและการเรียกใช้ข้อมูลที่มีความสะดวกต่อการใช้งาน
- 3) สามารถใช้เว็บแอปพลิเคชันในการใช้งานระบบการจัดการลงทะเบียนผ่านเว็บไซต์ ร่วมกับการทำงานของไมโครคอนโทรลเลอร์ในการควบคุมการใช้งานห้องเรียน
- 4) สามารถใช้งาน QR Code ในการตรวจสอบการเข้าเรียนและจัดเก็บรายชื่อของนักศึกษา

5.2 ข้อเสนอแนะ

จากการศึกษาและทดลองใช้งานพบว่าสามารถนำทฤษฎี Blockchain ไปประยุกต์ใช้งานกับระบบต่างๆในด้านของการเก็บข้อมูลและจัดการระบบพื้นฐานของหน่วยงานหรือองค์กรได้เป็นอย่างดี มีความปลอดภัยและความน่าเชื่อถือของข้อมูล มีการจัดเก็บข้อมูลอย่างเป็นระบบ ป้องกันการปลอมแปลงและแก้ไขข้อมูล เนื่องจาก Blockchain เป็นระบบที่มีการเก็บข้อมูล

แบบถาวร ไม่สามารถลบหรือแก้ไขข้อมูลในอดีตได้ และมีความสามารถในการเก็บข้อมูลได้ในปริมาณมาก และยังสามารถออกแบบ Smart Contract เพื่อกำหนดเงื่อนไขและการทำงานให้สอดคล้องกับความต้องการในการใช้งานของระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Pairoj Ruamviboonsuk. “มาทำความรู้จักเทคโนโลยี Blockchain.”
<https://www.iknowplus.co.th/single-post/2018/03/12/มาทำความรู้จักเทคโนโลยี-Blockchain>
- [2] it24hrs. “Blockchain คืออะไร.”
<https://www.it24hrs.com/2018/block-chain-what-is-blockchain>
- [3] Matana Wiboonyasake. “ลักษณะการทำงานของ Blockchain.”
<https://www.aware.co.th/ลักษณะการทำงานของ-blockchain>
- [4] Pongsatorn Tonglairoum. “Distributed Ledger คืออะไร.”
<https://medium.com/@pongsatt/distributed-ledger-คืออะไร>
- [5] Digital Venture. “What is Smart Contract.”
<http://www.dv.co.th/blog-th/smart-contract-blockchain>
- [6] สำนักงานคณะกรรมการธุรกรรมทางอิเล็กทรอนิกส์. “Smart Contract คืออะไร.”
<http://www.etcommission.go.th/article-Techno-Smartcontract.html>
- [7] Nuuneoicom. “Blockchain Consensus Protocol.”
https://nuuneoi.com/blog/blog.php?read_id=933
- [8] Methus Kaewsai kao. “Hyperledger: Linux แห่งโลก Blockchain.”
<https://medium.com/cochain/hyperledger-the-linux-of-blockchain>
- [9] Hyperledger Composer. “Welcome to Hyperledger Composer.”
<https://hyperledger.github.io/composer/v0.19/introduction/introduction.html>
- [10] Rachata Tongpagdee. “Docker คืออะไร ใช้งานอย่างไร.”
<https://medium.com/@rachatatongpagdee/docker-คืออะไร-ใช้งานอย่างไร>
- [11] Yodsawat Santhi. “ทำความรู้จักกับ Laravel Framework กัน.”
<https://medium.com/@yodsawatse/ทำความรู้จักกับ-laravel-framework-กัน>
- [12] PSP TECH Co., Ltd. “รีเลย์ (Relay) คืออะไร.”
<http://www.pspstech.co.th/รีเลย์relayคืออะไร-15696.page>