การประยุกต์บล็อกเชนสำหรับระบบเช็คชื่อเข้าเรียน

APPLICATION OF BLOCKCHAIN FOR CLASS ATTENDANCE SYSTEMS

จิติวัฒนา รอบรู้
ณัฐภัทร ทิพย์โพธิ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

การประยุกต์บล็อกเชนสำหรับระบบเช็คชื่อเข้าเรียน

APPLICATION OF BLOCKCHAIN FOR CLASS ATTENDANCE SYSTEMS

จิติวัฒนา รอบรู้
ณัฐภัทร ทิพย์โพธิ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

ปริญญานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การประยุกต์บล็อกเชนสำหรับระบบเช็คชื่อเข้าเรียน

APPLICATION OF BLOCKCHAIN FOR CLASS ATTENDANCE SYSTEMS

ผู้จัดทำ

1. นายจิติวัฒนา รอบรู้ รหัสนักศึกษา 58010176

2. นายณัฐภัทร ทิพย์โพธิ์ รหัสนักศึกษา 58010418

_____ อาจารย์ที่ปรึกษา

(ผศ. ดร.ศักดิ์ชัย ทิพย์จักษุรัตน์)

การประยุกต์บล็อกเชนสำหรับระบบเช็คชื่อเข้าเรียน

นายจิติวัฒนา รอบรู้ 58010176
นายณัฐภัทร ทิพย์โพธิ์ 58010418
ผศ. ดร.ศักดิ์ชัย ทิพย์จักษุรัตน์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2561

บทคัดย่อ

การเช็คชื่อในปัจจุบันส่วนใหญ่จะใช้วิธีการขานชื่อนักศึกษาหรือให้นักศึกษาลงลายมือชื่อในใบเซ็นชื่อเข้าเรียน ซึ่งอาจทำให้ไม่สะดวกและใช้เวลานานในการเช็คชื่อเข้าเรียน โครงงานนี้มีวัตถุประสงค์เพื่อพัฒนาระบบโดยการประยุกต์ใช้บล็อกเชนสำหรับระบบเช็คชื่อเข้าเรียนบนระบบปฏิบัติการแอนดรอยด์ ระบบที่นำเสนอเรียกว่า "การประยุกต์บล็อกเชนสำหรับระบบเช็คชื่อเข้าเรียน" หรือ "เอบีคาส" หลักการออกแบบของระบบเอบีคาสแบ่งออกเป็น 3 ส่วน คือ บล็อกเชน, เซิร์ฟเวอร์ และโมไบล์แอปพลิเคชัน ในส่วนของบล็อกเชนเราเลือกแพลตฟอร์มอีเธอร์เลียมในการรันสมาร์ตคอนแทร็กต์สำหรับการประมวลผลข้อมูล สำหรับการเช็คชื่อเข้าเรียนจะทำงานผ่านโทรศัพท์เคลื่อนที่ของอาจารย์ผู้สอนกับของนักศึกษาโดยการสื่อสารกันด้วยเทคโนโลยีบลูทูธ โดยขั้นตอนแรกจะทำการแพริ่งกันหรือจับคู่กันก่อน โดยขั้นตอนนี้จะทำเพียงครั้งเดียวเท่านั้น ส่วนการเช็คชื่อเข้าเรียนระบบเอบีคาสจะทำการเช็คโดยอัตโนมัติทุก ๆ 30 นาที ด้วยการใช้ระบบเอบีคาสอาจจะทำให้การเช็คชื่อเข้าเรียนสะดวกและรวดเร็วขึ้น นอกจากนี้ระบบเอบีคาสยังสามารถสรุปผลข้อมูลในเชิงสถิติสำหรับอาจารย์ผู้สอนในรูปแบบของไฟล์เอ็กซ์เซลด้วย

Application of Blockchain for Class Attendance Systems

Mr. Jitiwattana	Robru	58010176
Mr. Nattapat	Tippo	58010418
Asst.Prof.Dr. Sakchai	Thipchaksurat	Advisor

Academic Year 2018

ABSTRACT

Nowadays, class attendance checking systems mostly use student name calling method or signing method which may not be convenient and time consuming. The propose of this project is to develop the blockchain application for class attendance systems on Android operating system. The proposed system is called "Application of Blockchain for Class Attendance Systems (ABCAS)". Concept design of a system composed three parts: Blockchain, Server, and Mobile application. In Blockchain part, we select the Ethereum platform which can run smart contract for processing the data. For class attendance checking, the smartphone of lecturer and those of students are automatically pairing via Bluetooth technology. The class attendance process will be automatically checking every 30 minutes periodically. By using our proposed system, the class attendance checking might be more convenient. Furthermore, the system can summarize the statistics data for the lecturer in the form of Excel file.

กิตติกรรมประกาศ

โครงงานการประยุกต์บล็อกเชนสำหรับระบบเช็คชื่อเข้าเรียนสำเร็จลุล่วงไปได้ด้วยดีด้วยความกรุณาและความช่วยเหลือจากอาจารย์ศักดิ์ชัย ทิพย์จักษุรัตน์ ซึ่งเป็นอาจารย์ที่ปรึกษาของโครงงานนี้ ซึ่งได้ให้คำปรึกษาและช่วยเหลือการวางแผนการดำเนินงานตลอดจนให้คำแนะนำแนวทางปฏิบัติ รวมทั้งเสียสละเวลาเพื่อให้คำปรึกษาอย่างสม่ำเสมอ และแก้ไขข้อบกพร่องของเนื้อหาและสำนวนของภาษาด้วยความเอาใจใส่

ขอขอบคุณอาจารย์ทุกท่านที่ได้ให้ความรู้ ให้คำแนะนำ และช่วยเหลือในการทำโครงงานนี้จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณบิดามารดาและครอบครัวที่ให้กำลังใจและสนับสนุนในทุก ๆ ด้าน และยังให้คำแนะนำในการทำงาน เพื่อส่งเสริมให้โครงงานนี้เป็นไปตามเป้าหมายที่กำหนดไว้

ขอขอบคุณรุ่นพี่ รุ่นน้อง และเพื่อน ๆ ตลอดจนผู้เกี่ยวข้องทุกท่านที่ไม่ได้กล่าวนามไว้ ณ ที่นี้ ที่ได้ให้ความช่วยเหลือและแสดงความคิดเห็นต่อโครงงานในด้านต่าง ๆ

สุดท้ายนี้ด้วยคุณค่าและประโยชน์อันพึงมีจากโครงงานนี้ ขอมอบแด่ผู้มีพระคุณทุกท่านที่กล่าวมาข้างต้น และขอกราบขอบพระคุณทุกท่านมา ณ ที่นี้

จิติวัฒนา รอบรู้

ณัฐภัทร ทิพย์โพธิ์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตาราง หน้า

สารบัญรูป

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

การเช็คชื่อเข้าเรียนในปัจจุบันส่วนใหญ่จะใช้วิธีการขานชื่อนักศึกษาหรือให้นักศึกษาลงลายมือชื่อในใบเซ็นชื่อเข้าเรียน ซึ่งอาจทำให้ไม่สะดวกและใช้เวลานานในการเช็คชื่อส่งผลให้เวลาในการเรียนการสอนลดน้อยลง เมื่อนักศึกษาเข้ามาเช็คชื่อแล้วก็อาจออกจากห้องในระหว่างเรียนหรือไม่เข้าเรียนจนกระทั่งถึงเวลาเช็คชื่อ หากให้นักศึกษาลงลายมือชื่อเองก็อาจมีนักศึกษาที่เช็คชื่อแทนเพื่อน และหากนักศึกษาต้องการทราบข้อมูลการเข้าเรียนของตนเองก็ทำได้ไม่สะดวกเนื่องจากต้องติดต่อกับอาจารย์เพื่อขอข้อมูล และอาจารย์ยังต้องนำใบเซ็นชื่อเข้าห้องเรียนด้วยทุกครั้งและต้องสรุปผลการเข้าเรียนด้วยตนเองเมื่อสิ้นภาคการศึกษา หากนักศึกษามีจำนวนมากอาจทำให้การสรุปผลขาดความแม่นยำส่งผลให้เกิดความผิดพลาดขึ้นได้ นอกจากนี้ยังมีเรื่องของการชำรุดหรือสูญหายของเอกสารหรืออุปกรณ์ที่ใช้ในการเก็บข้อมูลการเช็คชื่อเข้าเรียนอีกด้วย

จากปัญหาเหล่านี้เองจึงทำให้เราคิดพัฒนาระบบอัตโนมัติสำหรับการเช็คชื่อเข้าเรียนโดยการใช้เทคโนโลยีเข้ามาช่วย โครงงานนี้จึงเลือกใช้สมาร์ตโฟน (Smartphone) มาเป็นอุปกรณ์ที่ใช้สำหรับระบุแทนตัวตน เนื่องจากสมาร์ตโฟนเป็นอุปกรณ์ที่ปัจจุบันทั้งอาจารย์และนักศึกษาพกพาติดตัวอยู่เสมอ ทำให้ไม่จำเป็นต้องเสียค่าใช้จ่ายหรือพกพาอุปกรณ์อื่น ๆ เฉพาะเพิ่มเติม และเลือกใช้เทคโนโลยีบลูทูธ (Bluetooth) ซึ่งเป็นเทคโนโลยีที่สนับสนุนการติดต่อสื่อสารข้อมูลแบบไร้สายระยะสั้นระหว่างอุปกรณ์ มาเป็นตัวช่วยในการเช็คชื่อเข้าเรียนของนักศึกษา โดยหากสมาร์ตโฟนของอาจารย์และนักศึกษาเชื่อมต่อกันสำเร็จก็สามารถบอกได้ว่านักศึกษามาเข้าเรียน โดยมีการพัฒนาโมไบล์แอปพลิเคชัน (Mobile Application) ขึ้นมาเพื่อให้ผู้ใช้ใช้งานระบบได้ง่าย สะดวก สามารถเช็คชื่อและดูข้อมูลสรุปผลการเข้าเรียนของนักศึกษาได้เลยผ่านทางแอปพลิเคชันและสามารถสรุปผลข้อมูลในเชิงสถิติสำหรับอาจารย์ผู้สอนในรูปแบบของไฟล์เอ็กซ์เซล (Excel) ได้อีกด้วย ในส่วนการประมวลผลหลังบ้านและฐานข้อมูล เราได้นำบล็อกเชน (Blockchain) มาใช้โดยเลือกใช้แพลตฟอร์มอีเธอร์เลียม (Ethereum) ในการรันสมาร์ตคอนแทร็กต์ (Smart Contract) สำหรับการประมวลผลข้อมูล โดยบล็อกเชนคือเทคโนโลยีการจัดเก็บข้อมูลแบบกระจายศูนย์ โดยเป็นรูปแบบการจัดเก็บข้อมูลที่รับประกันว่า ข้อมูลที่ถูกบันทึกไปก่อนหน้าไม่สามารถที่จะเปลี่ยนแปลงหรือแก้ไขได้ และทุกคนจะเห็นข้อมูลชุดเดียวกันทั้งหมด ซึ่งได้มีการใช้วิทยาการรหัสลับ (Cryptography) และการประมวลผลแบบกระจาย (Distributed Computing) มาเพื่อสร้างกลไก

ความน่าเชื่อถือ จากคุณสมบัติของเทคโนโลยีบล็อกเชนนี้เองทำให้ข้อมูลมีความถูกต้องเที่ยงตรง (Data Integrity) การเข้าถึงข้อมูลมีความโปร่งใส (Data Transparency) และระบบสามารถทำงานได้อย่างต่อเนื่อง (Availability) โดยระบบที่นำเสนอเรียกว่า "การประยุกต์บล็อกเชนสำหรับระบบเช็คชื่อเข้าเรียน" หรือ "เอบีคาส"

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อพัฒนาระบบเช็คชื่อเข้าเรียนที่สามารถช่วยลดเวลาและอำนวยความสะดวกในการเช็คชื่อเข้าเรียน และช่วยสรุปผลข้อมูลการเข้าเรียนได้
2) เพื่อลดปัญหาการปลอมแปลงการเช็คชื่อเข้าเรียนหรือการเช็คชื่อแทนกัน
3) เพื่อเพิ่มความปลอดภัยให้กับระบบเช็คชื่อเข้าเรียน ทำให้ระบบมีความน่าเชื่อถือ ช่วยป้องกันการแก้ไขหรือปลอมแปลงข้อมูลที่ได้เพิ่มเข้าไปในระบบแล้ว
4) เพื่อเพิ่มเสถียรภาพของระบบเช็คชื่อเข้าเรียน ช่วยลดการปฏิเสธการให้บริการ (Denial of Service) และป้องกันข้อมูลสูญหายด้วยการเก็บข้อมูลแบบกระจายศูนย์
5) เพื่อศึกษาการพัฒนาโมไบล์แอปพลิเคชันและหลักการทำงานของเทคโนโลยีบล็อกเชน

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้รับความรู้และความเข้าใจในเทคโนโลยีบล็อกเชน การพัฒนาสมาร์ตคอนแทร็กต์และการพัฒนาโมไบล์แอปพลิเคชัน
2) ลดเวลาในการเช็คชื่อและการตรวจสอบข้อมูลการเข้าเรียน
3) นำความรู้และทฤษฎีต่าง ๆ ที่ใช้ทำระบบเช็คชื่อเข้าเรียนไปประยุกต์ใช้กับงานที่เกี่ยวข้องกับการตรวจสอบการเข้าออกอื่น ๆ ได้

1.4 ขอบเขตของโครงงาน

1) ระบบเช็คชื่อเข้าเรียนออกแบบมาเพื่อใช้สำหรับสถาบันการศึกษา
2) เทคโนโลยีบล็อกเชนที่นำมาใช้ คือ แพลตฟอร์มอีเธอร์เลียม
3) การพัฒนาสมาร์ตคอนแทร็กต์บนแพลตฟอร์มอีเธอร์เลียมใช้ภาษาโซลิดิตี้ (Solidity) เวอร์ชัน 0.5.3
4) โมไบล์แอปพลิเคชันถูกพัฒนาโดยรองรับระบบปฏิบัติการแอนดรอยด์ (Android) 6 หรือ API ระดับ 23 ขึ้นไป และใช้บลูทูธคลาสสิกในการเชื่อมต่อ

5) โมไบล์แอปพลิเคชันสามารถใช้สร้างวิชาเรียนและเข้าร่วมวิชาเรียนที่สร้างได้
6) โมไบล์แอปพลิเคชันสามารถใช้เช็คชื่อเข้าเรียนได้ โดยบอกได้ว่านักศึกษามาเข้าเรียนในช่วงเวลาใดบ้าง
7) โมไบล์แอปพลิเคชันสามารถสรุปผลข้อมูลการเข้าเรียนออกมาในรูปแบบไฟล์เอ็กซ์เซลได้

1.5 แผนการดำเนินงาน

ตาราง 1.1 แผนการดำเนินงาน

หัวข้อกิจกรรม	เดือน								
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1. ศึกษาข้อมูลและงานวิจัยที่เกี่ยวข้องกับระบบเช็คชื่อเข้าเรียน									
2.วิเคราะห์และออกแบบระบบเช็คชื่อเข้าเรียน									
3.ศึกษาข้อมูลของเทคโนโลยีหรือแพลตฟอร์มที่เกี่ยวข้องเพื่อพัฒนาระบบเช็คชื่อเข้าเรียน									
4.ดำเนินการพัฒนาระบบเช็คชื่อเข้าเรียน									
5.ทดสอบและปรับปรุงแก้ไขข้อผิดพลาดของระบบ									
6.วิเคราะห์ผลระบบเช็คชื่อเข้าเรียน									
7.จัดทำเอกสารโครงงาน									

บทที่ 2
งานวิจัยและทฤษฎีที่เกี่ยวข้อง

ในบทนี้กล่าวถึงงานวิจัยและทฤษฎีต่าง ๆ ที่เกี่ยวข้อง ซึ่งใช้ในการทำโครงงานการประยุกต์บล็อกเชนสำหรับระบบเช็คชื่อเข้าเรียน ซึ่งงานวิจัยที่ค้นพบส่วนมากจะแตกต่างกันในเรื่องเทคโนโลยีที่ใช้ในการเช็คชื่อเข้าเรียนของนักศึกษา ในส่วนของทฤษฎีที่เกี่ยวข้องนั้น ได้แบ่งออกเป็น 4 ส่วนหลัก ๆ คือ บลูทูธ (Bluetooth) โมไบล์แอปพลิเคชัน (Mobile Application) แอปพลิเคชันเซิร์ฟเวอร์ (Application Server) และบล็อกเชน (Blockchain)

2.1 งานวิจัยที่เกี่ยวข้อง

2.1.1 ระบบบันทึกการเข้าชั้นเรียนผ่านบลูทูธ

งานวิจัยนี้เป็นการพัฒนาระบบบันทึกการเข้าชั้นเรียนอัตโนมัติผ่านบลูทูธที่มีชื่อเรียกว่าบลูการ์ด เพื่อแก้ไขปัญหาการตรวจสอบชื่อที่ใช้เวลานานเนื่องจากจำนวนนักศึกษาที่เพิ่มมากขึ้นและการเซ็นชื่อที่นักศึกษาสามารถเซ็นชื่อแทนเพื่อนได้ โดยนำเทคโนโลยีบลูทูธที่มีอยู่ในโทรศัพท์เคลื่อนที่และคอมพิวเตอร์มาประยุกต์ใช้ โดยสามารถตรวจสอบได้ว่านักศึกษานั่งเรียนอยู่จริงตลอดทั้งคาบเรียน โดยเป็นการเชื่อมต่อสัญญาณบลูทูธจากเครื่องคอมพิวเตอร์ของอาจารย์ที่ติดตั้งระบบกับอุปกรณ์บลูทูธของนักศึกษา โดยนักศึกษาสามารถติดตามการตรวจสอบรายชื่อได้ผ่านทางเว็บไซต์ และอาจารย์สามารถเข้ามาตรวจสอบและจัดการข้อมูลการเข้าชั้นเรียนของนักศึกษาได้ผ่านทางเว็บไซต์ (วริญทร เจนชัย, จิติมนต์ อั่งสกุล และธรา อั่งสกุล, 2555)

2.1.2 การพัฒนาระบบบันทึกเวลาเรียนด้วยการตรวจจับและรู้จำใบหน้า

งานวิจัยนี้เป็นการพัฒนาระบบบันทึกเวลาเรียนด้วยการตรวจจับใบหน้าโดยทำขึ้นเพื่อแก้ปัญหาการบันทึกเวลาเรียนที่ส่วนใหญ่มักจะใช้การเรียกชื่อเพื่อระบุตัวตนของนักศึกษา รวมทั้งความผิดพลาดที่อาจเกิดขึ้นจากการขาดความแม่นยำ เป็นการพัฒนาระบบในลักษณะเว็บแอปพลิเคชันโดยประยุกต์ใช้เทคโนโลยี WebRTC ในการสนับสนุนการสื่อสารระหว่างเว็บเบราว์เซอร์ในลักษณะเรียลไทม์ โดยจุดเด่นคือเป็นการสร้างการเชื่อมต่อระหว่างเว็บเบราว์เซอร์กับอุปกรณ์นำเข้าโดยตรง สามารถนำไปใช้กับห้องเรียนใดก็ได้เพียงแค่มีกล้องเว็บแคมกับคอมพิวเตอร์เท่านั้น และระบบมีลักษณะเป็นเว็บแอปพลิเคชันสามารถใช้งานได้ง่ายโดยไม่จำเป็นต้องลงโปรแกรมเสริม (พิชญา จตุรวัฒน์, ภาสินี พงศ์มานะวุฒิ และมานพ พันธ์โคกกรวด, 2560)

2.1.3 ระบบตรวจสอบการเข้าชั้นเรียนด้วย QR Code ในรายวิชาศึกษาทั่วไป

งานวิจัยนี้นำเสนอกระบวนการสร้างระบบตรวจสอบการเข้าชั้นเรียนและประเมินผลการเรียนด้วยระบบคิวอาร์โค้ดเพื่อแก้ไขความล่าช้าในการตรวจสอบการเข้าชั้นเรียนของนักศึกษาจำนวนมากและสามารถประเมินผลการเรียนรู้ได้อย่างรวดเร็วและมีประสิทธิภาพ โดยเป็นการสร้างแบบสอบถามออนไลน์จากกูเกิลฟอร์มเพื่อให้นักศึกษาเช็คชื่อเข้าชั้นเรียนโดยการตอบแบบสอบถามออนไลน์ โดยนำยูอาร์แอลของแบบสอบถามออนไลน์มาทำให้สั้นลงและเปลี่ยนยูอาร์แอลนั้นไปเป็นภาพคิวอาร์โค้ดเพื่อให้นักศึกษาสแกนไปยังแบบสอบถามออนไลน์ผ่านแอปพลิเคชันไลน์ในโทรศัพท์มือถือ (ประทีป พืชทองหลาง, ญาตาวีมินทร์ พืชทองหลาง และอาภากร ปัญโญ, 2561)

2.1.4 ระบบบันทึกเวลาอัตโนมัติด้วยลายนิ้วมือแบบไร้สาย

งานวิจัยนี้เป็นระบบที่พัฒนาขึ้นโดยมีวัตถุประสงค์เพื่อเพิ่มความสะดวกสบายและความแม่นยำของการบันทึกการเข้าชั้นเรียนด้วยการสแกนลายนิ้วมือแบบไร้สาย โดยจะประกอบไปด้วยชุดสแกนลายนิ้วมือซึ่งเชื่อมต่อกับไมโครคอนโทรลเลอร์และชุดสื่อสารรับส่งสัญญาณดิจิทัลไร้สายซึ่งมีคอมพิวเตอร์เป็นตัวกลางในการรับส่งสัญญาณเพื่อเปรียบเทียบกับระบบฐานข้อมูลของผู้เข้าเรียนและบันทึกเวลาเข้าเรียน พร้อมทั้งสรุปผลและวิเคราะห์จำนวนการเข้าเรียนของนักศึกษาออกมา (วิชาญ เพชรมณี, ขจรศักดิ์ พงศ์ธนา, 2552)

2.1.5 การประยุกต์บล็อกเชนสำหรับระบบเช็คชื่อเข้าเรียน (ABCAS)

โครงงานนี้เป็นระบบที่ผู้จัดทำนำเสนอ ซึ่งเป็นการนำเทคโนโลยีบล็อกเชนมาประยุกต์ใช้เพื่อนำเอาคุณสมบัติของบล็อกเชนมาใช้ คือ เป็นการเก็บข้อมูลแบบกระจายศูนย์ ข้อมูลที่เพิ่มไปแล้วจะแก้ไขได้ยากและสามารถตรวจสอบที่มาที่ไปได้ โดยใช้เทคโนโลยีบลูทูธสื่อสารกันระหว่างสมาร์ตโฟนของอาจารย์และนักศึกษาเพื่อตรวจสอบการเข้าเรียน และมีโมไบล์แอปพลิเคชันเพื่อเป็นตัวกลางช่วยอำนวยความสะดวกทั้งในเรื่องของการเช็คชื่อ การตรวจสอบและสรุปผลข้อมูลการเข้าเรียนของนักศึกษา โดยจะมีการเปรียบเทียบความสามารถในการทำงานดังตาราง 2.1 ด้านล่าง โดยข้อมูลที่นำมาเปรียบเทียบนั้นได้จากการวิเคราะห์งานวิจัยที่พบ

ตาราง 2.1 แสดงการเปรียบเทียบคุณลักษณะพื้นฐานของระบบต่าง ๆ

ความสามารถของระบบ	งานวิจัย (เทคโนโลยีที่ใช้)				โครงงานนี้
	บลูทูธ	การรู้จำใบหน้า	QR Code	ลายนิ้วมือ	
เช็คชื่อผ่านโมไบล์แอปพลิเคชัน	-	-	✓	-	✓
ตรวจสอบข้อมูลผ่านโมไบล์แอปพลิเคชัน	-	-	✓	-	✓
ทราบช่วงเวลาขณะอยู่ในห้องเรียน	✓	✓	✓	✓	✓
ทราบช่วงเวลาขณะไม่อยู่ในห้องเรียน	✓	✓	-	-	✓
แสดงผลสถิติสรุปการเข้าเรียน	✓	✓	✓	✓	✓
นักศึกษาสามารถตรวจสอบการเข้าเรียนของตนเองได้ผ่านสมาร์ตโฟน	✓	-	✓	-	✓
ส่งออกข้อมูลรายงาน	✓	-	✓	✓	✓
จัดเก็บข้อมูลในฐานข้อมูลแบบกระจายศูนย์ผ่านเครือข่าย	-	-	-	-	✓
ข้อมูลที่เพิ่มแล้วเปลี่ยนแปลงได้ยาก	-	-	-	-	✓
ตรวจสอบที่มาที่ไปของข้อมูลได้ทั้งหมด	-	-	-	-	✓

2.2 ทฤษฎีที่เกี่ยวข้อง

2.2.1 บลูทูธ (Bluetooth)

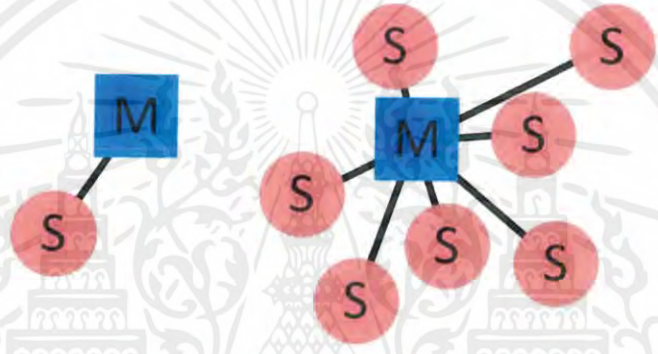
บลูทูธ คือ เทคโนโลยีที่ใช้สัญญาณวิทยุสำหรับการติดต่อสื่อสารระยะสั้นระหว่างอุปกรณ์อิเล็กทรอนิกส์ ทำให้อุปกรณ์สามารถสร้างการเชื่อมต่อไร้สายเพื่อแลกเปลี่ยนข้อมูลกันได้ เช่น โทรศัพท์มือถือ เมาส์ คีย์บอร์ด หูฟัง คอมพิวเตอร์ เป็นต้น โดยมีราคาถูกและใช้พลังงานน้อย

รูป 2.1 สัญลักษณ์ของบลูทูธ

บลูทูธทำงานที่สัญญาณวิทยุความถี่สูง 2.4 GHz โดยอยู่ระหว่างความถี่ 2400-2483.5 MHz ซึ่งช่วงความถี่ที่ใช้งานอาจแตกต่างกันในบางประเทศ โดยจะใช้วิธีการส่งข้อมูลที่เรียกว่าการส่งข้อมูลผ่านคลื่นวิทยุด้วยการแผ่สเปกตรัม (Frequency Hopping Spread Spectrum: FHSS) ซึ่งจะแบ่งช่องสัญญาณความถี่ออกเป็น 79 ช่องสัญญาณ แต่ละช่องจะมีแบนวิดธ์ 1 MHz จะใช้ย่านความถี่ช่วง 2 MHz จากล่างสุด และช่วง 3.5 MHz จากบนสุดเป็นช่วงความถี่ป้องกัน โดยจะใช้ช่องสัญญาณที่แบ่งนี้สลับไปมาถึง 1,600 ครั้งต่อ 1 วินาที เพื่อใช้ส่งข้อมูลไปมาระหว่าอุปกรณ์โดยไม่จำเป็นต้องเรียงตามหมายเลขช่อง และยังสามารถเลือกเปลี่ยนความถี่ที่ใช้ได้เองโดยอัตโนมัติ เพื่อป้องกันการดักฟังหรือลักลอบขโมยข้อมูล

รูป 2.2 การแบ่งช่องสัญญาณของบลูทูธ

เครือข่ายบลูทูธหรือที่รู้จักกันว่า พิโคเน็ต (piconet) ใช้โมเดลมาสเตอร์/สลาฟ (master/slave) คือมีอุปกรณ์ตัวหนึ่งทำหน้าที่เป็นมาสเตอร์หรือตัวแม่ข่าย และอุปกรณ์ตัวอื่น ๆ ในเครือข่ายเดียวกันทำหน้าที่เป็นสลาฟหรือตัวลูกข่าย โดยมาสเตอร์จะทำหน้าที่ควบคุมการทำงานและประสานงานให้กับสลาฟในเครือข่ายเดียวกันเพื่อควบคุมว่าอุปกรณ์ไหนจะส่งข้อมูลเมื่อไร โดยมาสเตอร์หนึ่งตัวสามารถเชื่อมต่อกับสลาฟได้ถึง 7 ตัว แต่สลาฟใด ๆ ในพิโคเน็ตหนึ่งสามารถมีมาสเตอร์ได้เพียง 1 ตัวเท่านั้น นอกจากนี้ยังมีเครือข่ายบลูทูธอีกแบบที่เรียกว่า สแคทเทอร์เน็ต (scatternet) คือ การนำพิโคเน็ตตั้งแต่ 2 เครือข่ายขึ้นไปมารวมกัน โดยมีอุปกรณ์ตัวหนึ่งที่ทำหน้าเป็นมาสเตอร์ในพิโคเน็ตหนึ่งแต่เป็นสลาฟในพิโคเน็ตอื่น



รูป 2.3 เครือข่ายพิโคเน็ตของบลูทูธ

กำลังส่งและระยะทำการของอุปกรณ์บลูทูธจะขึ้นอยู่กับคลาสของอุปกรณ์นั้น ๆ โดยจะแบ่งได้เป็น 3 คลาส ดังตาราง 2.2

ตาราง 2.2 คลาสของอุปกรณ์บลูทูธ

Class Number	Max Output Power (dBm)	Max Output Power (mW)	Max Range
Class 1	20 dBm	100 mW	100 m
Class 2	4 dBm	2.5 mW	10 m
Class 3	0 dBm	1 mW	10 cm

2.2.2 โมไบล์แอปพลิเคชัน (Mobile Application)

2.2.2.1 จาวาสคริปต์ (JavaScript)

จาวาสคริปต์เป็นภาษาสคริปต์เชิงวัตถุครอสแพลตฟอร์ม (Cross platform) สำหรับฝั่งไคลเอ็นต์ (Client-side) จะช่วยให้เว็บเพจมีการโต้ตอบกับผู้ใช้งาน เช่น แอนิเมชันที่มีความซับซ้อน ปุ่มที่สามารถคลิกได้ เมนูแบบพอปอัป (popup) เป็นต้น ซึ่งจะเป็นการอัปเดตส่วนประกอบต่าง ๆ ของฟอร์มเอชทีเอ็มแอล (HTML form) โดยตอบสนองต่อการกระทำของผู้ใช้งาน สำหรับฝั่งเซิร์ฟเวอร์ (Server-side) จะช่วยให้แอปพลิเคชันสามารถเชื่อมต่อกับฐานข้อมูลหรือแม้แต่จัดการไฟล์ต่าง ๆ บนเซิร์ฟเวอร์ได้ เป็นต้น

รูป 2.4 สัญลักษณ์ของจาวาสคริปต์

2.2.2.2 โหนดเจเอส (Node.js)

โหนดเจเอสเป็นจาวาสคริปต์รันไทม์เอ็นไวรอนเมนท์ (Runtime environment) ของฝั่งเซิร์ฟเวอร์ เป็นโอเพนซอร์ซ (Opensource) และยังรองรับข้ามแพลตฟอร์ม ช่วยในการรันแอปพลิเคชันที่เขียนด้วยภาษาจาวาสคริปต์ โดยถูกสร้างขึ้นมาจากวีแปดจาวาสคริปต์เอนจิน (V8 JavaScript engine) ของกูเกิลโครม (Google Chrome) มีลักษณะแบบอิงเหตุการณ์ (event-driven) และเป็นอะซิงโครนัส (Asynchronous) คือ การที่ไม่จำเป็นต้องรอทำงานตามลำดับ

โหนดเจเอสสามารถใช้สร้างแอปพลิเคชันได้อย่างหลากหลาย เช่น คอมมานไลน์แอปพลิเคชัน (Command-line Application) เว็บแอปพลิเคชัน (Web Application) แอปพลิเคชันแชทแบบเรียลไทม์ (Real-time Chat Application) หรือแม้แต่เรสเอพีไอเซิร์ฟเวอร์ (REST API Server) เป็นต้น แต่ส่วนใหญ่แล้วจะใช้ทำเว็บเซิร์ฟเวอร์

รูป 2.5 สัญลักษณ์ของโหนดเจเอส

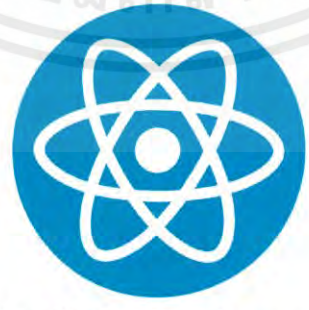
2.2.2.3 เอ็นพีเอ็ม (npm)

เอ็นพีเอ็มหรือโหนดแพ็กเกจแมเนเจอร์ (Node Package Manager) เป็นเครื่องมือที่ใช้สำหรับจัดการแพ็กเกจต่าง ๆ สำหรับโหนดเจเอส ซึ่งจะถูกติดตั้งมาพร้อมกับโหนดเจเอส โดยเอ็นพีเอ็มจะประกอบด้วย 2 ส่วน คือ ส่วนที่เป็นฐานข้อมูลออนไลน์ (online database) เรียกว่า เอ็นพีเอ็มรีจิสทรี (npm registry) ซึ่งใช้เก็บแพ็กเกจต่าง ๆ ของโหนดเจเอส และส่วนที่เป็นเครื่องมือสำหรับจัดการและเข้าถึงเอ็นพีเอ็มรีจิสทรีนั้น ช่วยในการติดตั้ง การจัดการเวอร์ชันและการจัดการดีเพนเดนซี (dependency) โดยใช้เพียงคำสั่งเดียวก็สามารถติดตั้งแพ็กเกจที่ต้องการได้เลย และสามารถเลือกได้ว่าจะติดตั้งเฉพาะโปรเจคหรือติดตั้งแบบโกลบอลอีกด้วย

รูป 2.6 สัญลักษณ์ของเอ็นพีเอ็ม

2.2.2.4 รีแอคเนทีฟ (React Native)

รีแอคเนทีฟเป็นจาวาสคริปต์เฟรมเวิร์ก (JavaScript Framework) ตัวหนึ่งซึ่งถูกพัฒนาโดยเฟซบุ๊ค (Facebook) ใช้ในการสร้างโมไบล์แอปพลิเคชันแบบครอสแพลตฟอร์มช่วยให้เขียนโค้ดเพียงครั้งเดียวก็สามารถสร้างแอปพลิเคชันที่ทำงานได้ทั้งบนระบบปฏิบัติการแอนดรอยด์ (Android) และไอโอเอส (iOS) โดยใช้ภาษาจาวาสคริปต์เป็นหลักในการพัฒนาและสามารถเข้าถึงส่วนที่เป็นเนทีฟของแพลตฟอร์มไอโอเอสและแอนดรอยด์ได้โดยตรง ซึ่งข้อดีของรีแอคเนทีฟคือช่วยให้สามารถออกแบบส่วนติดต่อผู้ใช้งานได้ง่าย มีโมดูลมากมายที่รองรับการใช้งานได้หลากหลายรูปแบบและยังมีประสิทธิภาพเทียบเท่าได้กับการเขียนแบบเนทีฟ



รูป 2.7 สัญลักษณ์ของรีแอคเนทีฟ

2.2.2.5 โคลเจอร์สคริปต์ (ClojureScript)

โคลเจอร์สคริปต์ คือ โคลเจอร์เวอร์ชันที่ถูกคอมไพล์มาให้อยู่ในภาษาจาวาสคริปต์ คือมีความสามารถของโคลเจอร์แต่สามารถเข้าถึงได้ด้วยภาษาจาวาสคริปต์ มีประสิทธิภาพมากในการพัฒนาเว็บแอปพลิเคชัน โดยโคลเจอร์คือภาษาสำหรับการเขียนโปรแกรมเชิงฟังก์ชัน (Functional programming language) สำหรับจาวาเวอร์ชวลแมชชีน (Java Virtual Machine) และเนื่องจากการคอมไพล์โคลเจอร์ให้เป็นจาวาสคริปต์ จึงทำให้โคลเจอร์สคริปต์สามารถรันได้บนทุกเบราว์เซอร์ บนอุปกรณ์โมไบล์ และบนโหนดเจเอส ด้วยลักษณะของภาษาที่คล้ายกับภาษา LISP จึงทำให้โค้ดสั้น กระชับ และด้วยคุณสมบัติของภาษาเชิงฟังก์ชันจึงช่วยลดบั๊กด้วย

รูป 2.8 สัญลักษณ์ของโคลเจอร์สคริปต์

2.2.2.6 เว็บทรี (web3.js)

เว็บทรี คือ ชุดไลบรารีที่ใช้สำหรับการโต้ตอบกับอีเธอร์เลียม (Ethereum) โหนดทั้งแบบโลคอลโหนด (Local node) และรีโมตโหนด (Remote node) ด้วยการเชื่อมต่อผ่านโพรโทคอลเอชทีทีพี (HTTP) หรือไอพีซี (IPC) ช่วยให้สามารถดูข้อมูลต่าง ๆ ส่งทรานแซคชั่นหรือติดต่อกับสมาร์ตคอนแทร็กต์ (Smart Contract) และอื่น ๆ อีกมากมาย โดยเว็บทรีดอทเจเอสจะเป็นเว็บทรีที่เป็นจาวาสคริปต์ จึงทำให้สามารถใช้ได้บนทุกเว็บเบราว์เซอร์ ส่วนใหญ่จะใช้กับฝั่งเซิร์ฟเวอร์ซึ่งอยู่ในโหนดเจเอสแอปพลิเคชัน

รูป 2.9 สัญลักษณ์ของเว็บทรี

2.2.2.7 รีนาเทล (Re-Natal)

รีนาเทลเป็นเครื่องมือที่ช่วยลดขั้นตอนต่าง ๆ สำหรับการสร้างรีแอคเนทีฟแอปพลิเคชันที่เขียนด้วยโคลเจอร์สคริปต์ โดยช่วยตั้งค่าต่าง ๆ ที่จำเป็น การติดตั้งโมดูลที่แอปพลิเคชันต้องการ รวมถึงช่วยให้แอปพลิเคชันที่พัฒนาทำงานร่วมกันกับโมดูลที่ต้องการได้โดยอัตโนมัติ

2.2.2.8 เลนอินเจน (Leiningen)

เลนอินเจนเป็นเครื่องมือที่ช่วยในการติดตั้งและจัดการแพ็กเกจต่าง ๆ รวมถึงการตั้งค่าบางอย่างที่จำเป็นสำหรับการพัฒนาแอปพลิเคชันที่เขียนด้วยภาษาโคลเจอร์ โดยทำให้เป็นระบบอัตโนมัติเพื่อให้ผู้พัฒนามุ่งเน้นไปที่การเขียนโค้ด

รูป 2.10 สัญลักษณ์ของเลนอิเจน

2.2.3 แอปพลิเคชันเซิร์ฟเวอร์ (Application Server)

2.2.3.1 เอนจิ้นเอ็ก (Nginx)

เอนจิ้นเอ็กเป็นโอเพนซอร์ซซอฟต์แวร์ใช้สำหรับการให้บริการเว็บเซิร์ฟเวอร์ ทำรีเวิร์สพร็อกซี (Reverse Proxy) แคชชิ่ง (Caching) ทำโหลดบาลานซ์ (Load Balance) การสตรีมสื่อมีเดีย (Media Streaming) และอื่น ๆ อีกมากมาย

รูป 2.11 สัญลักษณ์ของเอนจิ้นเอ็ก

เอ็นจิ้นเอ็กมีความสามารถในการปรับขยายโครงสร้างสูง (Highly Scalable) มีลักษณะเป็นแบบโมดูลลาร์ (Modular System) คือ ระบบจะประกอบด้วยหน่วยแยกต่าง ๆ มีการทำงานแบบอิงเหตุการณ์และอะซิงโครนัส มีโครงสร้างเป็นแบบซิงเกิลเธรด (Single-Thread) สามารถปรับตัวได้ดีแม้ทำงานบนฮาร์ดแวร์เซิร์ฟเวอร์ทั่วไปหรือการประมวลแบบมัลติโพรเซสเซอร์ข้ามระบบ ใช้ทรัพยากรเครื่องน้อยและมีประสิทธิภาพ สามารถจัดการกับการเชื่อมต่อจำนวนมากพร้อม ๆ กันได้ ส่วนใหญ่จึงจะใช้เอนจิ้นเอ็กทำเป็นรีเวิร์สพร็อกซีหรือโหลดบาลานซ์เซอร์

2.2.3.2 เอ็กเพรส (Express.js)

เอ็กเพรสเป็นเว็บแอปพลิเคชันเฟรมเวิร์กสำหรับทำงานบนแพลตฟอร์มของโหนดเจเอส เป็นโอเพนซอร์ซซอฟต์แวร์ถูกออกแบบมาเพื่อใช้สร้างเว็บแอปพลิเคชันและเอพีไอ มีความยืดหยุ่นสูงและมีโมดูลต่าง ๆ มากมายที่สามารถติดตั้งและนำมาใช้งานได้ทันที และยังมีความสามารถต่าง ๆ ที่จะช่วยให้ทำเว็บแอปพลิเคชันได้สะดวกมากยิ่งขึ้น เช่น การจัดการหาเส้นทาง (routing) การจัดการข้อมูลระหว่างทาง (middleware) การจัดการการร้องขอ (request) และการตอบกลับ (response) เป็นต้น

express

รูป 2.12 สัญลักษณ์ของเอ็กเพรส

2.2.4 บล็อกเชน (Blockchain)

2.2.4.1 บล็อกเชน

บล็อกเชน คือ เทคโนโลยีการจัดเก็บข้อมูลแบบใช้ฐานข้อมูลร่วมกัน (Shared Database) หรือที่รู้จักกันในชื่อ การจัดเก็บข้อมูลแบบกระจายศูนย์ (Distributed Ledger Technology , DLT) เป็นรูปแบบการบันทึกข้อมูลที่รับประกันความปลอดภัยว่าข้อมูลที่ถูกบันทึกไปก่อนหน้านั้น ไม่สามารถที่จะเปลี่ยนแปลงหรือแก้ไขได้ ซึ่งผู้ใช้งานทุกคนจะได้เห็นข้อมูลชุดเดียวกันทั้งหมด โดยจะใช้หลักการของวิทยาการรหัสลับ (Cryptography) และความสามารถของการประมวลผลแบบกระจาย (Distributed Computing) เพื่อสร้างกลไกความน่าเชื่อถือ

ข้อมูลในระบบบล็อกเชนจะมีการเชื่อมโยงกันทั้งระบบ ซึ่งทุกโหนดในเครือข่ายจะต้องมีข้อมูลที่ตรงกัน โดยข้อมูลทั้งหมดจะถูกจัดเก็บอยู่ภายใต้โครงสร้างของ

เทคโนโลยีบล็อกเชนและถูกสำเนากระจายไปยังโหนดทุกโหนดในเครือข่าย ดังนั้นเทคโนโลยีบล็อกเชนจึงไม่จำเป็นต้องมีตัวกลางเข้ามาเกี่ยวข้องเพื่อคอยทำหน้าที่ในการจัดเก็บรายการทรานแซคชั่น (Transaction) เมื่อมีรายการทรานแซคชั่นใหม่เกิดขึ้น จะมีการประกาศบอกทุกโหนดในเครือข่ายให้รับรู้ โดยรายการทรานแซคชั่นดังกล่าวจะต้องผ่านการตรวจสอบด้วยกระบวนการยืนยันข้อมูลร่วมกัน (Consensus) จากทั้งเครือข่ายก่อน จึงจะสามารถบันทึกข้อมูลเข้าสู่ระบบบล็อกเชนได้ และถ้ามีคนพยายามสร้างรายการทรานแซคชั่นปลอมขึ้นมา ข้อมูลนั้นก็จะขัดแย้งกันกับข้อมูลในเครื่องของโหนดอื่น ๆ ในเครือข่าย เนื่องจากทุกโหนดจะต้องมีข้อมูลเหมือนกันทั้งหมด ดังนั้นระบบจะไม่อนุญาตให้สร้างรายการดังกล่าว จะมีแต่รายการที่ทุกโหนดในเครือข่ายยอมรับเท่านั้นที่จะสามารถบันทึกเข้าสู่ระบบบล็อกเชนได้ และข้อมูลที่ถูกบันทึกเข้าสู่ระบบบล็อกเชนไปแล้วจะไม่สามารถเปลี่ยนแปลงหรือแก้ไขย้อนหลังได้ จึงทำให้เทคโนโลยีบล็อกเชนเป็นเทคโนโลยีที่มีการจัดเก็บข้อมูลที่มีความน่าเชื่อถือสูง

รูป 2.13 ระบบแบบ Centralised, Decentralised และ Distributed

บล็อกเชนมีคุณสมบัติที่สำคัญ 3 ประการ คือ

1) ความถูกต้องเที่ยงตรงของข้อมูล (Data Integrity)

เนื่องจากการเชื่อมโยงบล็อกเข้าด้วยกันด้วยการใช้ฟังก์ชันแฮชและทำการกระจายข้อมูลให้กับโหนดทุกโหนดในเครือข่ายเก็บไว้ จึงทำให้ข้อมูลที่ถูกบันทึกลงในบล็อกเชนแล้วไม่สามารถแก้ไข หรือเปลี่ยนแปลงข้อมูลได้ (Immutability) ดังนั้นหากมีความพยายามในการแก้ไขหรือเปลี่ยนแปลงข้อมูลที่ถูกบันทึกลงไปแล้ว จะสามารถทราบได้ทันทีเนื่องจากข้อมูลของโหนดดังกล่าวต่างไปจากโหนดอื่น ๆ ในระบบ จึงไม่สามารถสร้างการยืนยันข้อมูลร่วมกันได้และจะถูกแยกออกจากเชน (Chain) หลักไปในที่สุด

2) ความโปร่งใสในการเข้าถึงข้อมูล (Data Transparency)

เนื่องจากทุกโหนดในระบบจะเก็บข้อมูลที่เป็นชุดเดียวกันทั้งหมดและไม่มีตัวกลางเข้ามาเกี่ยวข้องในการคอยเก็บข้อมูล ทำให้การเข้าถึงข้อมูลใด ๆ ทำได้เองโดยไม่จำเป็นต้องไปร้องขอจากตัวกลาง

3) ความสามารถในการทำงานได้อย่างต่อเนื่องของระบบ (Availability)

เนื่องจากการที่ทุกโหนดในระบบเก็บข้อมูลชุดเดียวกันทั้งหมด ดังนั้นหากมีโหนดใดไม่สามารถให้บริการได้ในขณะนั้น โหนดอื่น ๆ ในระบบจึงสามารถทำงานแทนโหนดนั้นได้ และเมื่อโหนดกลับมาให้บริการได้อีกครั้งก็จะอัปเดตข้อมูลของตัวเองได้โดยอัตโนมัติ

หลักการทำงานพื้นฐานที่สำคัญของเทคโนโลยีบล็อกเชน อย่างน้อยจะประกอบไปด้วย 4 ขั้นตอนหลัก ๆ คือ

1) การสร้างบล็อก

เป็นการนำรายการคำสั่งขอทำทรานแซคชั่นมารวบรวมบรรจุลงเป็นบล็อก

2) การการจายบล็อก

เป็นการนำบล็อกที่ได้จากขั้นตอนแรก ส่งไปยังทุก ๆ โหนดในเครือข่าย เพื่อบอกโหนดทุกโหนดในระบบว่ามีข้อมูลใหม่เกิดขึ้น

3) การตรวจสอบบล็อก

โดยเมื่อโหนดอื่น ๆ ในเครือข่ายได้รับบล็อกใหม่ จะมีการตรวจสอบความถูกต้องของบล็อกนั้นตามเงื่อนไขของกระบวนการยืนยันข้อมูลร่วมกันของระบบบล็อกเชนนั้น ๆ

4) การนำบล็อกเข้าสู่ระบบ

หลังจากทำการยืนยันและตรวจสอบความถูกต้องของบล็อกใหม่ที่ได้รับแล้ว ก็จะนำบล็อกใหม่นั้นไปเรียงต่อกับบล็อกก่อนหน้า เพื่ออัปเดตฐานข้อมูล

2.2.4.2 โหนด (Node)

โหนด คือ อุปกรณ์ในเครือข่ายบล็อกเชน โดยอาจเป็นเครื่องคอมพิวเตอร์ โทรศัพท์มือถือ หรืออื่น ๆ สามารถจำแนกได้เป็น 2 ประเภท คือ โหนดที่ทำหน้าที่ในการจัดเก็บสำเนาข้อมูล ประกอบด้วย ฟูลโหนด (Full Node) และไลท์โหนด (Light Node) และอีกประเภทคือโหนดที่ทำหน้าที่ตรวจสอบความถูกต้องของข้อมูล

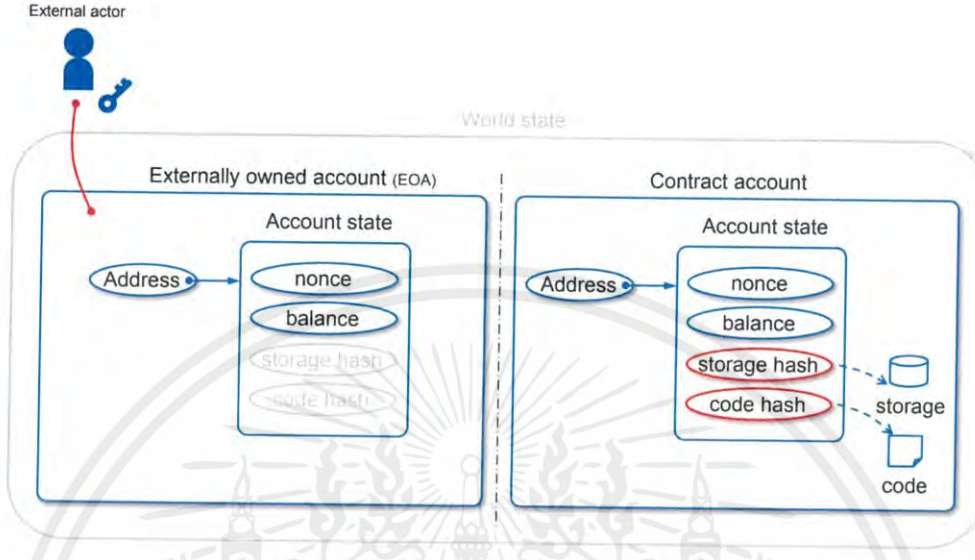
2.2.4.3 สมุดบัญชี (Ledger)

สมุดบัญชีหรือก็คือประวัติรายการการทำทรานแซคชั่น เป็นรายการบันทึกการทำกิจกรรมต่าง ๆ โดยทุกโหนดจะเก็บสมุดบัญชีที่เหมือนกันไว้ ซึ่งข้อมูลในสมุดบัญชีนี้จะเพิ่มขึ้นเรื่อย ๆ ตามเวลา

2.2.4.4 บัญชี (Account)

ในอีเธอร์เลียมจะมีบัญชีอยู่ 2 ประเภท คือ Externally owned account (EOA) เป็นบัญชีที่ถูกควบคุมด้วยไพรเวทคีย์ หมายความว่าถ้าเรามีไพรเวทคีย์ของบัญชีนั้นเราก็สามารถใช้บัญชีนั้นเพื่อทำทรานแซคชั่นได้ แต่ไม่มีโค้ด และอีกประเภทคือ Contract account เป็นบัญชีที่มี

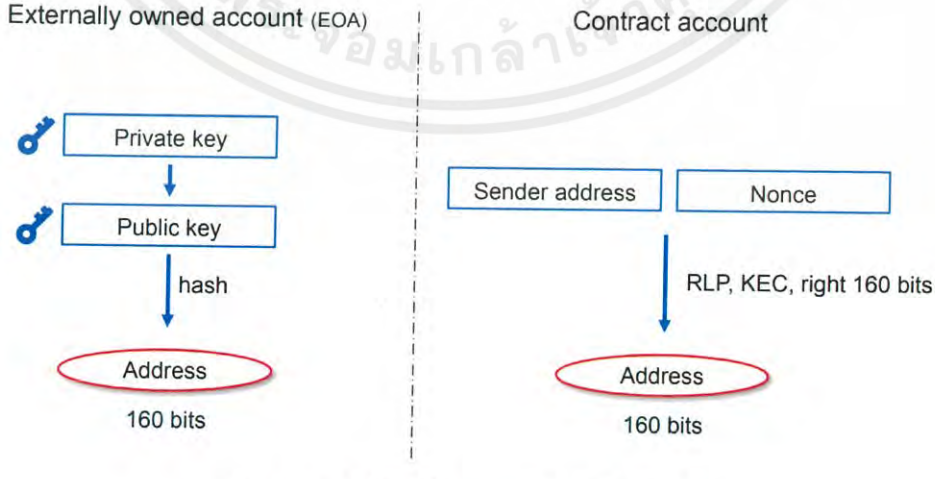
โค้ดของตัวเอง และถูกควบคุมด้วยโค้ดนั้น แต่บัญชีทั้ง 2 ประเภทจะมีสิ่งที่เหมือนกันคือ บาลานซ์ (balance) และนอนซ์ (nonce)



รูป 2.14 ประเภทของบัญชีในอีเธอร์เลียม

2.2.4.5 แอดเดรส (Address)

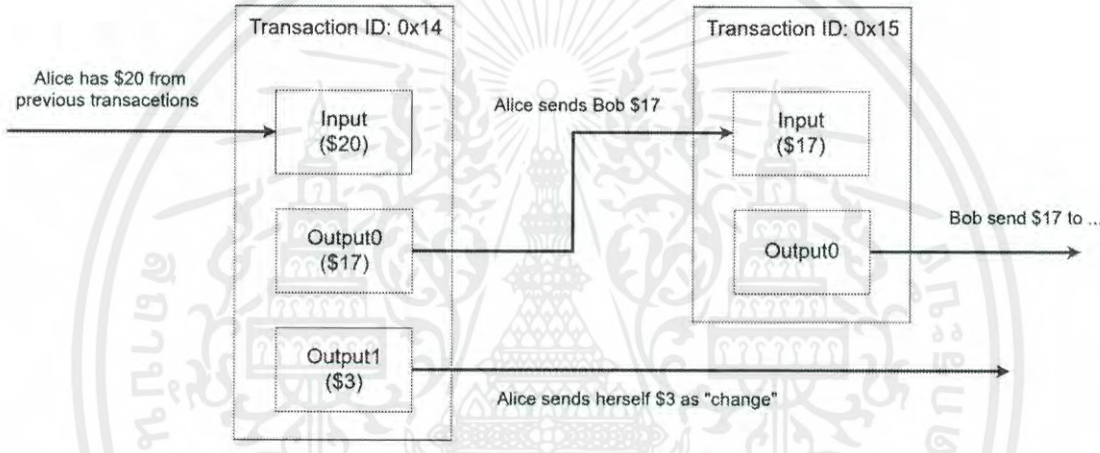
ในอีเธอร์เลียม แอดเดรสใช้สำหรับระบุแทนบัญชี มีขนาด 160 บิต โดยต้องสร้างไพรเวทคีย์มาก่อนจากนั้นสร้างพับบลิคคีย์จากไพรเวทคีย์แล้วนำไปเข้าฟังก์ชันแฮช โดยแอดเดรสจะมีขนาดสั้นกว่าพับบลิคคีย์และไม่เป็นความลับ แต่สำหรับคอนแทร็กต์แอดเดรสนั้นจะมาจากแอดเดรสของผู้สร้างคอนแทร็กต์กับค่านอนซ์ของบัญชีผู้สร้าง



รูป 2.15 การได้มาซึ่งแอดเดรสในอีเธอร์เลียม

2.2.4.6 ทรานแซคชั่น (Transaction)

ทรานแซคชั่นแสดงถึงการมีปฏิสัมพันธ์กันของทั้งสองฝ่ายหรือเป็นการบันทึกกิจกรรมที่เกิดขึ้น ตัวอย่างเช่น สำหรับเงินตราเข้ารหัส ทรานแซคชั่นจะแสดงถึงการโอนเงินตราเข้ารหัส สำหรับสมาร์ตคอนแทร็กต์ ทรานแซคชั่นจะแสดงถึงการส่งข้อมูล การประมวลผลข้อมูลหรือการเก็บข้อมูลผลลัพธ์บางอย่างบนบล็อกเชน ซึ่งบล็อกแต่ละบล็อกจะมีหรือไม่มีทรานแซคชั่นก็ได้ โดยปกติทรานแซคชั่นหนึ่งอย่างน้อยจะประกอบด้วย Inputs และ Outputs โดย Inputs คือรายการของสินทรัพย์ดิจิทัลที่จะถูกโอน ซึ่งผู้ส่งต้องพิสูจน์ได้ว่าตนเองสามารถเข้าถึงสินทรัพย์ดิจิทัลที่อ้างโดยทำการลงลายมือชื่อดิจิทัล และ Outputs คือ บัญชีผู้รับและจำนวนที่จะได้รับ



รูป 2.16 ตัวอย่างทรานแซคชั่นโอนเงิน

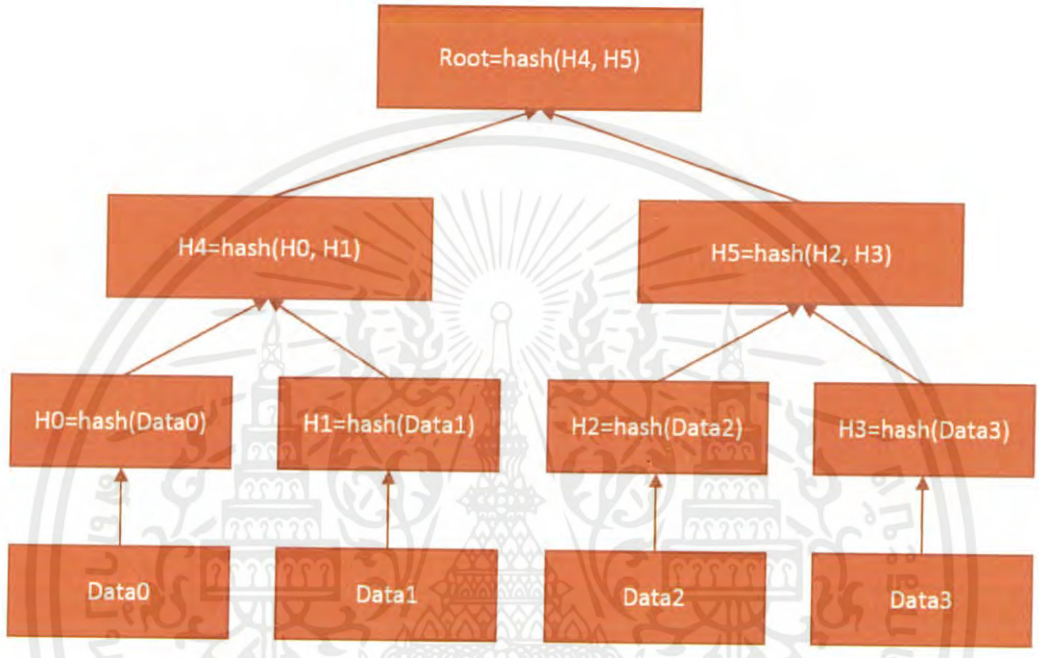
2.2.4.7 บล็อก (Block)

บล็อก คือ ชุดบรรจุข้อมูล แบ่งออกเป็น 2 ส่วน คือ ส่วนหัวของบล็อก (Block Header) ใช้เพื่อบอกให้ทราบว่าภายในบรรจุข้อมูลอะไร และส่วนข้อมูลของบล็อก (Block Data) เป็นส่วนที่ใช้บรรจุข้อมูลต่าง ๆ เช่น รายการทรานแซคชั่น โดยส่วนใหญ่โครงสร้างของส่วนหัวจะมี 6 ส่วน คือ

1) หมายเลขบล็อก (Block Number)
   คือ ตัวเลขที่แสดงถึงลำดับของบล็อกนั้น ๆ
2) แฮชของบล็อกก่อนหน้า (Previous Hash)
   คือ ค่าแฮชของส่วนหัวของบล็อกก่อนหน้า

3) แฮชของส่วนข้อมูล (Hash)

คือ แฮชของรายการทรานแซคชั่นทั้งหมดของส่วนข้อมูลของบล็อก ซึ่งเป็นวิธีการแฮชข้อมูลชุดใหญ่ โดยใช้รูปแบบ Hash Tree เช่น Merkle Root ของระบบบิทคอย (Bitcoin) เป็นต้น ซึ่งอาจแตกต่างกันไปแล้วแต่ระบบ



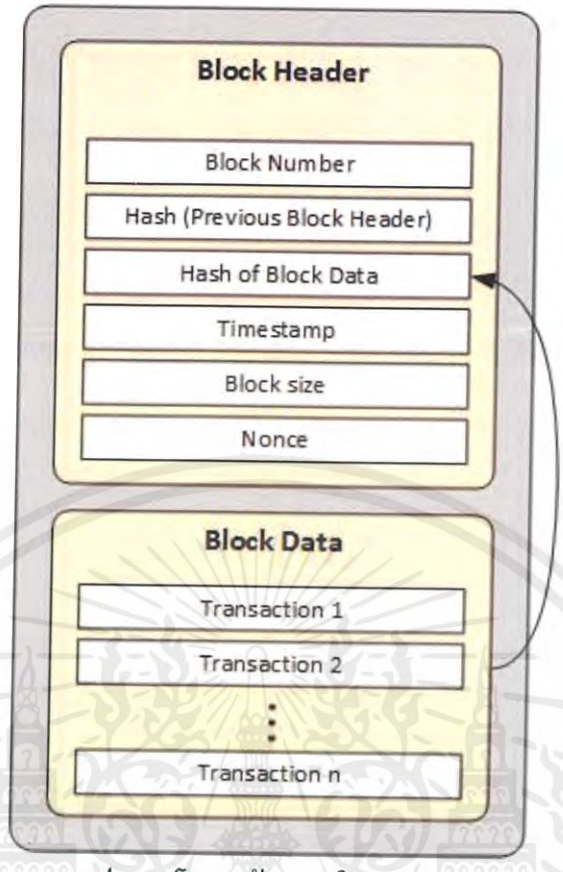
รูป 2.17 หลักการแฮชของ Merkle Root

4) เวลาที่สร้างบล็อก (Timestamp)

5) ขนาดของบล็อก (Block size)

6) ค่านอนซ์ (Nonce)

สำหรับระบบบล็อกเชนที่ใช้การขุด (Mining) จะเป็นค่าที่ใช้ในการแก้โจทย์ปัญหา สำหรับระบบอื่น ๆ อาจใช้ทำหน้าที่อื่นหรือไม่ใช้ก็ได้



รูป 2.18 โครงสร้างภายในของบล็อก

2.2.4.8 ฟังก์ชันแฮช (Cryptographic Hash Functions)

ฟังก์ชันแฮชเป็นกระบวนการในการนำข้อมูลมาผ่านกระบวนการทางคณิตศาสตร์ จะได้ผลลัพธ์เป็นข้อมูลที่ไม่มีความเกี่ยวข้องกับข้อมูลนำเข้าเลย เรียกว่า ข้อความย่อย โดยข้อมูลนำเข้าสามารถมีขนาดเกือบจะเท่าไรก็ได้ ซึ่งเมื่อนำข้อมูลนำเข้าเดิมมาเข้าฟังกชันแฮชแล้วจะได้ผลลัพธ์เหมือนเดิมเสมอ ซึ่งทำให้พิสูจน์ได้ว่าข้อมูลนั้นไม่มีการเปลี่ยนแปลง แม้จะเปลี่ยนข้อมูลนำเข้าเพียงบิตเดียวก็ส่งผลให้ผลลัพธ์ที่ได้เปลี่ยนแปลงไปอย่างสิ้นเชิงจากข้อความย่อยเดิม

คุณสมบัติของฟังก์ชันแฮชที่สำคัญเกี่ยวกับความปลอดภัย คือ

1) เป็นฟังก์ชันแบบทางเดียว นั่นคือไม่สามารถคำนวณหาข้อมูลนำเข้าจากข้อมูลผลลัพธ์หรือข้อความย่อยได้
2) แม้จะมีข้อมูลนำเข้าชุดหนึ่ง ก็ไม่สามารถหาข้อมูลนำเข้าอีกชุดหนึ่งซึ่งเมื่อเข้าฟังก์ชันแฮชแล้วจะได้ผลลัพธ์หรือข้อความย่อยเหมือนกัน

3) ไม่มีข้อมูลนำเข้า 2 ชุดใด ๆ ที่เมื่อเข้าฟังก์ชันแฮชแล้วให้ผลลัพธ์หรือข้อความย่อยเหมือนกัน

2.2.4.9 การเข้ารหัสลับแบบกุญแจอสมมาตร (Asymmetric-Key Cryptography)

การเข้ารหัสลับแบบกุญแจอสมมาตรจะใช้กุญแจเป็นคู่ คือ มีพับบลิคคีย์ (Public Key) และไพรเวทคีย์ (Private Key) ซึ่งมีความสัมพันธ์กันในทางคณิตศาสตร์ พับบลิคคีย์ทำขึ้นมาเพื่อให้สามารถแจกจ่ายได้โดยไม่กระทบถึงความปลอดภัย แต่ไพรเวทคีย์ต้องเก็บไว้เป็นความลับ แม้ว่ากุญแจทั้งสองจะมีความสัมพันธ์กันแต่ก็ไม่สามารถหาไพรเวทคีย์จากข้อมูลเพียงแค่พับบลิคคีย์ได้ โดยเราสามารถเข้ารหัสลับได้โดยใช้ไพรเวทคีย์และจะสามารถถอดรหัสลับได้ด้วยพับบลิคคีย์ ในทำนองเดียวกันหากเราเข้ารหัสลับด้วยพับบลิคคีย์ จะต้องถอดรหัสลับด้วยไพรเวทคีย์

การเข้ารหัสลับแบบกุญแจอสมมาตรสร้างความสัมพันธ์ที่เชื่อถือได้ระหว่างผู้ใช้ที่ไม่รู้จักกัน โดยมีกลไกที่สามารถตรวจสอบความสมบูรณ์ของข้อมูลและการพิสูจน์ตัวตนของการทำทรานแซคชั่น ในขณะเดียวกันทรานแซคชั่นนั้นยังคงเป็นสาธารณะ โดยกลไกที่ว่านี้สามารถทำได้โดยการลงลายมือชื่อดิจิทัล (Digital Signature) กับทรานแซคชั่นนั้น นั่นหมายความว่าเราใช้ไพรเวทคีย์เพื่อเข้ารหัสลับทรานแซคชั่น ใครก็ตามที่มีพับบลิคคีย์ของเราสามารถถอดรหัสลับได้เนื่องจากพับบลิคคีย์เข้าถึงได้โดยอิสระ การเข้ารหัสลับทรานแซคชั่นด้วยไพรเวทคีย์พิสูจน์ได้ว่าผู้ลงลายมือชื่อดิจิทัลของทรานแซคชั่นนั้นสามารถเข้าถึงไพรเวทคีย์ได้ ในอีกทางหนึ่ง หากเราเข้ารหัสลับข้อมูลด้วยพับบลิคคีย์ของบุคคลนั้น เฉพาะผู้ที่สามารถเข้าถึงไพรเวทคีย์ที่เป็นคู่ของพับบลิคคีย์นั้น ถึงจะสามารถถอดรหัสลับข้อมูลนั้นได้

2.2.4.10 การลงลายมือชื่อดิจิทัล (Digital Signature)

การลงลายมือชื่อดิจิทัลเป็นกระบวนการที่จะใช้สำหรับการตรวจสอบความแท้จริงและความสมบูรณ์ของข้อความ ซอฟต์แวร์หรือเอกสารทางดิจิทัล เป็นการใช้วิทยาการเข้ารหัสลับและฟังก์ชันแฮชมาช่วยพิสูจน์ว่าข้อความที่ได้รับเป็นข้อความต้นฉบับจริง ไม่ถูกปลอมแปลง รวมถึงความเป็นเจ้าของ

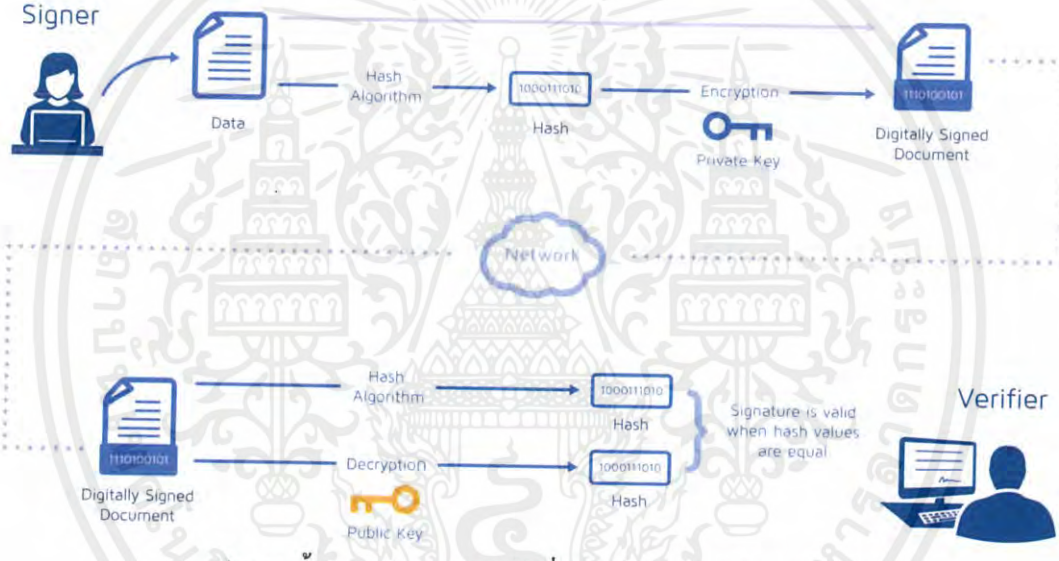
การลงลายมือชื่อดิจิทัลมีขั้นตอนดังนี้

1) ใช้อัลกอริทึมในการสร้างคีย์ เช่น RSA สำหรับสร้างคู่กุญแจ เรียกว่า พับบลิคคีย์ (Public key) และไพรเวทคีย์ (Private key) โดยให้แจกจ่ายพับบลิคคีย์ให้กับทุกคน แต่ให้เก็บไพรเวทคีย์ไว้กับตัวเองเท่านั้น
2) ใช้ฟังก์ชันแฮช เช่น sha256 โดยนำข้อมูลที่ต้องการส่ง มาเข้าฟังก์ชันแฮช เรียกค่าแฮชที่ได้ว่า ข้อความย่อย (Message digest)
3) ผู้ส่งทำการเข้ารหัสข้อความย่อยที่ได้ด้วยไพรเวทคีย์ของตน จะได้ลายมือชื่อดิจิทัล

4) นำลายมือชื่อดิจิทัลที่ได้แนบไปกับข้อมูลที่ต้องการจะส่ง เพื่อส่งเอกสารที่มีการลงลายมือชื่อดิจิทัลแล้ว

สำหรับการตรวจสอบลายมือชื่อดิจิทัลมีขั้นตอนดังนี้

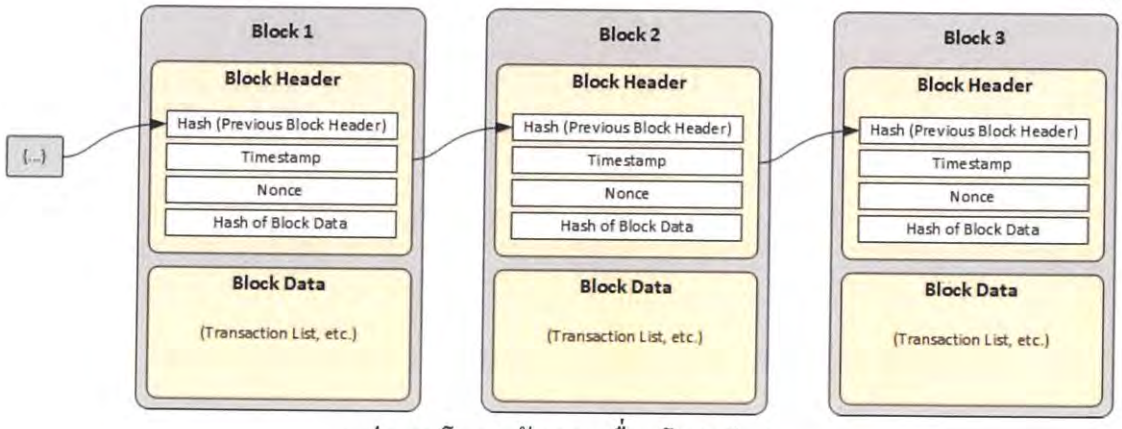
1) ให้ผู้รับถอดรหัสลายมือชื่อดิจิทัลของผู้ส่งที่แนบมาด้วยพับบลิคคีย์ของผู้ส่ง ซึ่งจะได้เป็นข้อความย่อย
2) นำข้อมูลที่ผู้ส่งส่งมาให้ ไปเข้าฟังก์ชันแฮช จากนั้นจะได้ข้อความย่อยของ
3) ถ้าข้อความย่อยจากขั้นตอนที่ 1 และขั้นตอนที่ 2 ตรงกัน ก็สามารถรับรองได้ว่าข้อมูลที่ได้รับ ส่งมาจากผู้ส่งจริงและข้อมูลไม่มีการเปลี่ยนแปลงระหว่างทาง



รูป 2.19 ขั้นตอนการลงลายมือชื่อดิจิทัลและการตรวจสอบ

2.2.4.11 การเชื่อมโยงบล็อก (Chaining blocks)

เนื่องจากแต่ละบล็อกจะมีการเก็บค่าแฮชของส่วนหัวของบล็อกก่อนหน้าไว้ ซึ่งเปรียบเสมือนเป็นการเชื่อมโยงบล็อกปัจจุบันกับบล็อกก่อนหน้า จึงก่อเกิดเป็นสายของบล็อก เรียกว่า บล็อกเชน ถ้าบล็อกก่อนหน้ามีการเปลี่ยนแปลง ค่าแฮชของส่วนหัวของบล็อกนั้นก็จะเปลี่ยนแปลงไปด้วย ส่งผลให้บล็อกถัดไปทั้งหมดจะมีค่าแฮชเปลี่ยนแปลงไปด้วย นั่นก็เพราะว่ามีการเก็บค่าแฮชของบล็อกก่อนหน้าไว้ ทำให้สามารถตรวจสอบได้ง่ายและปฏิเสธบล็อกที่ไม่ถูกต้อง โดยจะเรียกบล็อกแรกสุดว่า Genesis Block



รูป 2.20 โครงสร้างการเชื่อมโยงบล็อก

2.2.4.12 กระบวนการยืนยันข้อมูลร่วมกัน (Consensus)

กระบวนการยืนยันข้อมูลร่วมกัน เป็นกลไกในการควบคุมความถูกต้องของข้อมูลในทุกโหนดผ่านอัลกอริทึมต่าง ๆ เพื่อให้ข้อมูลของทั้งระบบมีความถูกต้องเที่ยงตรงและเป็นข้อมูลชุดเดียวกัน รวมทั้งมีการจัดเก็บและลำดับที่สอดคล้องกันด้วย โดยกระบวนการยืนยันข้อมูลร่วมกันนั้นมีอยู่ด้วยกันหลายวิธี จะเลือกวิธีใดนั้นขึ้นอยู่กับความเหมาะสม ตัวอย่างที่พบบ่อยมี 3 ตัวอย่าง ดังนี้

2.2.4.12.1 การพิสูจน์ด้วยพลังประมวลผล (Proof of Work)

การพิสูจน์ด้วยพลังประมวลผล คือ กระบวนการยืนยันข้อมูลร่วมกันโดยการใช้การแก้ปัญหาทางคณิตศาสตร์ซึ่งมีความซับซ้อนและต้องใช้เวลาในการแก้ปัญหานั้น ๆ เพื่อยืนยันความน่าเชื่อถือของข้อมูลที่จะถูกบันทึกลงในเครือข่าย โดยจะเรียกโหนดในระบบว่านักขุด (Miner) โดยโหนดที่เจอคำตอบก่อนก็จะได้สิทธิ์ในการเขียนทรานแซคชั่นบนบล็อกถัดไปและได้รับค่าธรรมเนียมในการดำเนินงานเป็นรางวัล

2.2.4.12.2 การพิสูจน์ด้วยสินทรัพย์ (Proof of Stake)

การพิสูจน์ด้วยสินทรัพย์ คือ กระบวนการยืนยันข้อมูลร่วมกันโดยใช้หลักการวางสินทรัพย์เพื่อให้ได้รับสิทธิ์เป็นผู้ตรวจสอบ ผู้ที่ทำการวางสินทรัพย์จำนวนมากก็จะมีโอกาสสูงที่จะได้รับสิทธิ์ในการเขียนข้อมูลทรานแซคชั่นบนบล็อกถัดไป โดยผู้ทำการเขียนข้อมูลบนบล็อกถัดไปก็จะได้รับค่าธรรมเนียมการดำเนินงานเป็นรางวัลตอบแทน

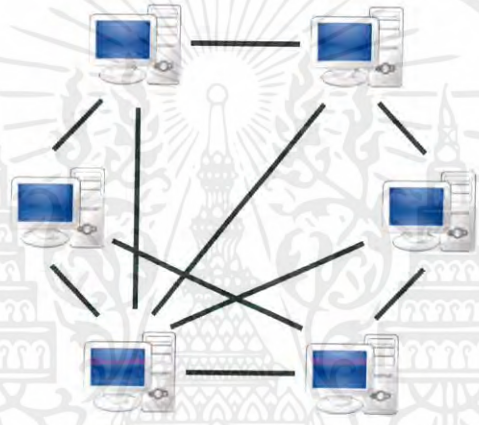
2.2.4.12.3 การพิสูจน์ด้วยผู้มีอำนาจ (Proof of Authority)

การพิสูจน์ด้วยผู้มีอำนาจ คือ กระบวนการยืนยันข้อมูลร่วมกันโดยใช้การทำข้อตกลงร่วมกันในการกำหนดสิทธิ์ให้กับผู้ใช้งานที่เชื่อถือได้ เพื่อทำหน้าที่เป็นผู้

ตรวจสอบ (Validator) ซึ่งจะทำหน้าที่ในการเขียนคำขอการทำทรานแซคชั่นลงบนบล็อก โดยจะมีการหมุนเวียนสิทธิ์กับผู้ตรวจสอบทุกคนเพื่อกระจายความรับผิดชอบ

2.2.4.13 เครือข่ายแบบเพียร์ทูเพียร์ (Peer-to-Peer Network)

เครือข่ายแบบเพียร์ทูเพียร์เป็นเครือข่ายของระบบคอมพิวเตอร์ซึ่งเชื่อมต่อเข้าด้วยกันผ่านอินเทอร์เน็ต สามารถส่งผ่านไฟล์กันได้โดยตรงโดยไม่จำเป็นต้องมีตัวกลาง โดยคอมพิวเตอร์แต่ละเครื่องที่มาเชื่อมต่อทำหน้าที่เป็นทั้งเซิร์ฟเวอร์และไคล์เอนต์ ซึ่งมีข้อจำกัดแค่ว่าคอมพิวเตอร์ที่จะเชื่อมต่อเครือข่ายเพียร์ทูเพียร์ต้องเชื่อมต่ออินเทอร์เน็ตได้และมีซอฟต์แวร์เพียร์ทูเพียร์



รูป 2.21 เครือข่ายเพียร์ทูเพียร์

2.2.4.14 เงินตราเข้ารหัส (Cryptocurrency)

เงินตราเข้ารหัสเป็นสกุลเงินซึ่งมีมูลค่าเหมือนกับธนบัตรในสกุลเงินของประเทศต่าง ๆ แต่จะอยู่ในรูปแบบดิจิทัล สามารถใช้เป็นสื่อกลางในการแลกเปลี่ยนแบบดิจิทัล ยกตัวอย่างเช่น บิทคอย (Bitcoin)

รูป 2.22 สัญลักษณ์ของบิทคอย

2.2.4.15 อีเธอร์เลียม (Ethereum)

อีเธอร์เลียมเป็นแพลตฟอร์มที่สามารถสร้างแอปพลิเคชันแบบกระจายศูนย์ซึ่งรันอยู่บนเทคโนโลยีบล็อกเชนได้ โดยอีเธอร์เลียมเป็นโปรเจคแบบโอเพนซอร์ซถูกพัฒนาโดยผู้คนจากทั่วโลก ต่างจากบิทคอยตรงที่อีเธอร์เลียมถูกออกแบบมายืดหยุ่นกว่าและสามารถปรับตัวได้ โดยอีเธอร์เลียมมีความสามารถที่เพิ่มขึ้นมาเรียกว่า สมาร์ตคอนแทร็กต์ ซึ่งอนุญาตให้ผู้ใช้สามารถเขียนโปรแกรมลงไปได้ ทำให้เกิดรูปแบบการใช้งานที่หลากหลายแตกต่างจากระบบบิตคอยที่เน้นเพียงการทำธุรกรรมทางการเงินเพียงอย่างเดียว โดยสกุลเงินในระบบอีเธอร์เลียมจะเรียกว่า อีเธอร์ (Ether)

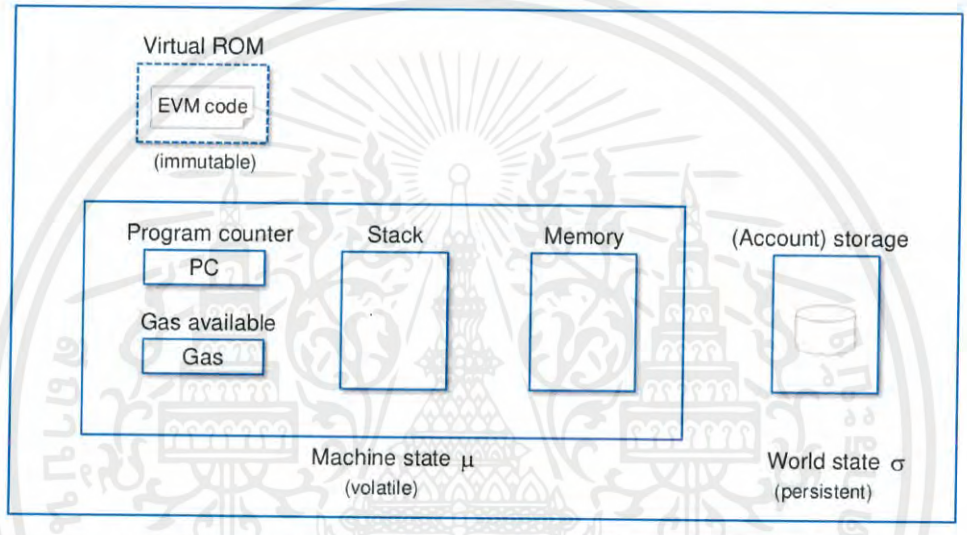
รูป 2.23 สัญลักษณ์ของอีเธอร์เลียม

2.2.4.16 สมาร์ตคอนแทร็กต์ (Smart Contract)

สมาร์ตคอนแทร็กต์ คือ โปรแกรมคอมพิวเตอร์ที่สามารถดำเนินการตามข้อตกลงโดยอัตโนมัติทันทีที่เกิดเหตุการณ์ตามเงื่อนไขที่ได้ตั้งไว้ ทำงานอยู่บนเทคโนโลยีบล็อกเชน โดยเป็นความสามารถเฉพาะตัวของอีเธอร์เลียม ทำให้ผู้ใช้สามารถเขียนโปรแกรมลงไปบนบล็อกเชนได้ โดยสมาร์ตคอนแทร็กต์จะประกอบด้วยส่วนโค้ดและข้อมูลหรืออาจจะเรียกว่าฟังก์ชันและสเตท โดยจะถูกเก็บไว้บนเครือข่ายบล็อกเชนในรูปแบบ EVM bytecode สมาร์ตคอนแทร็กต์จะถูกรันด้วยโหนดที่อยู่ภายในเครือข่ายบล็อกเชนและต้องได้ผลลัพธ์เดียวกันจึงจะนำไปบันทึกลงในบล็อกเชน ผู้ใช้เครือข่ายอีเธอร์เลียมสามารถสร้างทรานแซคชั่นเพื่อโต้ตอบกับสมาร์ตคอนแทร็กต์ได้ ซึ่งสมาร์ตคอนแทร็กต์จะถูกรันหากเป็นไปตามเงื่อนไขที่ได้ถูกกำหนดไว้ โดยสมาร์ตคอนแทร็กต์เปรียบเสมือนเป็นตัวช่วยบังคับใช้สัญญาตามที่ได้โปรแกรมไว้โดยอัตโนมัติ มีความปลอดภัยและความเป็นมาตรฐาน ช่วยลดขั้นตอนการทำงานและข้อผิดพลาดของมนุษย์ที่อาจเกิดขึ้น

2.2.4.17 อีเธอร์เลียมเวอร์ชวลแมชชีน (Ethereum Virtual Machine)

อีเธอร์เลียมเวอร์ชวลแมชชีน คือ รันไทม์เอ็นไวรอนเมนท์สำหรับรันสมาร์ตคอนแทร็กต์ โดยเป็นสภาพแวดล้อมที่แยกออกอย่างสิ้นเชิง หมายความว่าโค้ดที่รันอยู่ในอีวีเอ็มไม่สามารถเข้าถึงเครือข่าย ระบบแฟ้ม (File system) หรือโพรเซสอื่น ๆ (Process) ของเครื่องได้ และแม้แต่สมาร์ตคอนแทร็กต์ด้วยกันเองก็ยังมีการจำกัดการเข้าถึงสมาร์ตคอนแทร็กต์อื่น โดยอีวีเอ็มเป็นเวอร์ชวลแมชชีนแบบทัวริงคอมพลีท (Turing Complete) นั่นคือ มีความสามารถในการคำนวณสิ่งต่าง ๆ ได้ สมมุติว่าให้ทรัพยากรที่เพียงพอ



รูป 2.24 แบบจำลองสถาปัตยกรรมอีเธอร์เลียมเวอร์ชวลแมชชีน

2.2.4.18 โซลิดิตี้ (Solidity)

โซลิดิตี้ คือ ภาษาที่ใช้สำหรับการเขียนสมาร์ตคอนแทร็กต์เพื่อไปรันบนอีเธอร์เลียมเวอร์ชวลแมชชีน (Ethereum Virtual Machine, EVM) โดยมีไวยากรณ์คล้ายกับภาษาจาวาสคริปต์

รูป 2.25 สัญลักษณ์ของโซลิดิตี้

2.2.4.19 เก็ตท์ (Geth)

เก็ตท์ เป็นเครื่องมือแบบคอมมานไลน์สำหรับใช้ในการรันอีเธอร์เลียมโหนด ถูกสร้างขึ้นด้วยภาษาโก (Go) ช่วยให้สามารถขุด โอนเงิน สร้างคอนแทร็กต์หรือส่งทรานแซคชั่น รวมถึงการสำรวจรายการบล็อกทั้งหมดย้อนหลัง เป็นต้น

รูป 2.26 หน้าตาของเครื่องมือเก็ตท์

2.2.4.20 ทรัฟเฟิล (Truffle)

ทรัฟเฟิล เป็นเฟรมเวิร์ก (Framework) ที่ใช้สำหรับพัฒนาสมาร์ตคอนแทร็กต์ โดยช่วยสร้างสภาพแวดล้อมที่เหมาะสมในการพัฒนาสมาร์ตคอนแทร็กต์ให้ทำได้ง่ายขึ้น ใช้งานง่ายและมีความสามารถมากมาย เช่น การรวบรวมสมาร์ตคอนแทร็กต์ที่สร้างขึ้นและ deploy ขึ้นไปบนระบบอีเธอร์เลียม การทดสอบสมาร์ตคอนแทร็กต์ การดีบั๊กและโต้ตอบกับสมาร์ตคอนแทร็กต์ผ่านคอนโซล เป็นต้น

รูป 2.27 สัญลักษณ์ของทรัฟเฟิล

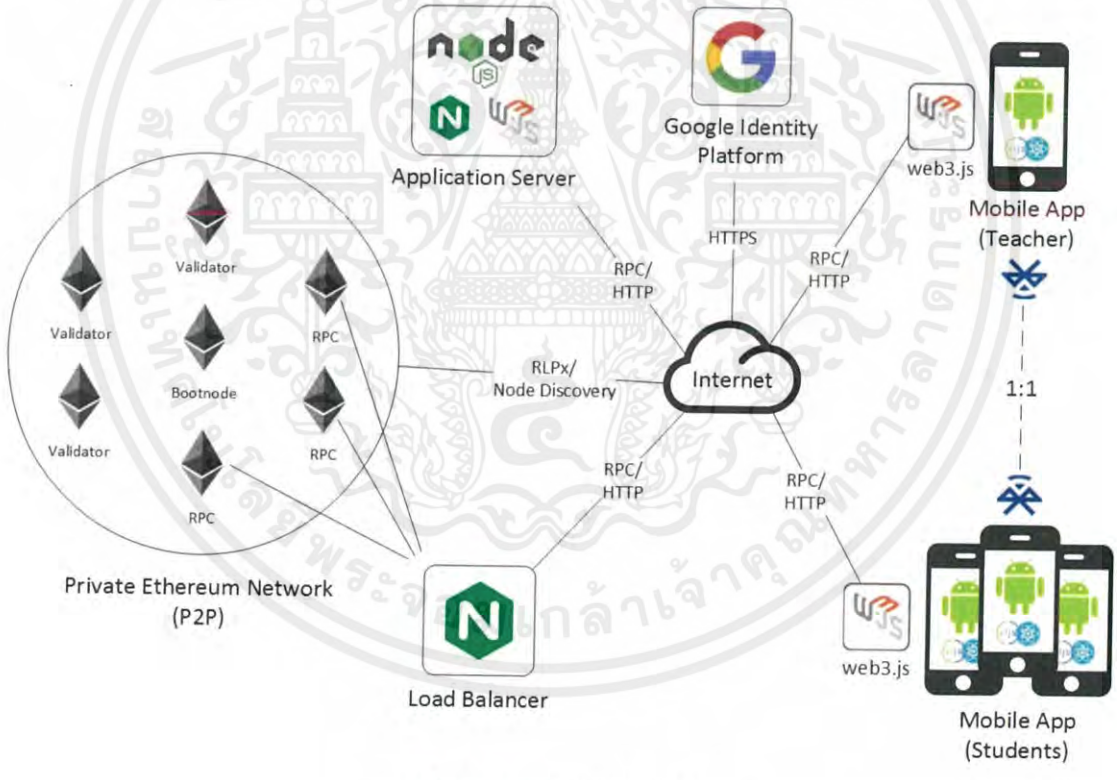
2.2.4.21 พัพเพ็ต (puppeth)

พัพเพ็ตเป็นเครื่องมือที่ช่วยในการจัดการเครือข่ายอีเธอร์เลียมให้ง่ายขึ้น เช่น การสร้างไฟล์ genesis เพื่อเริ่มต้นระบบบล็อกเชน ทำให้ไม่จำเป็นต้องเริ่มเขียนใหม่หมดเองตั้งแต่ต้น รวมทั้งช่วยในการ deploy ระบบสำเร็จรูป เป็นต้น

บทที่ 3
การออกแบบระบบ

ในบทนี้จะกล่าวถึงการออกแบบระบบโดยจะมี 7 ส่วนด้วยกัน คือ ภาพรวมของระบบ ความต้องการของระบบ แผนภาพยูสเคส (Use Case Diagram) แผนภาพซีเควนซ์ (Sequence Diagram) แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล (Entity Relationship Diagram) ส่วนติดต่อผู้ใช้งาน (User Interface) และสมาร์ตคอนแทร็กต์ (Smart Contract)

3.1 ภาพรวมของระบบ



รูป 3.1 ภาพรวมของระบบ

สำหรับภาพรวมของระบบนั้นจะแบ่งออกได้เป็น 3 ส่วนหลัก ๆ คือ โมไบล์แอปพลิเคชัน (Mobile Application) แอปพลิเคชันเซิร์ฟเวอร์ (Application Server) และบล็อกเชน (Blockchain)

3.1.1 โมไบล์แอปพลิเคชัน

ทำหน้าที่เป็นตัวกลางในการสื่อสารระหว่างผู้ใช้งานกับเครือข่ายบล็อกเชน ช่วยให้ผู้ใช้ใช้งานได้ง่าย สะดวก และใช้เช็คชื่อเข้าเรียนรวมทั้งการสรุปผลข้อมูลการเข้าเรียนของนักศึกษา โดยโมไบล์แอปพลิเคชันจะสรุปข้อมูลการเช็คชื่อเข้าเรียนเป็นรอบ ๆ โดยนำผลลัพธ์ที่ได้มาสร้างทรานแซคชั่น (Transaction) ส่งไปยังเครือข่ายอีเธอร์เลียมส่วนตัวด้วยไลบรารีเว็บทรี (web3.js) ผ่านโพรโทคอล JSON-RPC โดยจะติดต่อผ่านโหลดบาลานซ์เซอร์ (Load Balancer) ไปยังโหนดที่เปิดใช้งาน RPC

3.1.2 แอปพลิเคชันเซิร์ฟเวอร์

ทำหน้าที่ช่วยเหลือในการคัดกรองและตรวจสอบผู้ที่สมัครเข้าใช้บริการหรือเข้าสู่ระบบเพื่อให้เฉพาะผู้ที่กำหนดไว้เท่านั้นจึงจะสามารถสมัครเข้าใช้บริการหรือเข้าสู่ระบบได้ โดยเมื่อผู้ใช้เข้าสู่ระบบด้วยกูเกิล (Google) บนโมไบล์แอปพลิเคชัน จะได้รับข้อมูลบัญชีและโทเค็น (Token) จากนั้นจะส่งไปให้แอปพลิเคชันเซิร์ฟเวอร์เพื่อทำการติดต่อกับเซิร์ฟเวอร์ของกูเกิล เพื่อตรวจสอบข้อมูลบัญชีและโทเค็นที่ได้รับอีกครั้งว่าถูกต้องจริงและเป็นไปตามเงื่อนไขที่กำหนด นอกจากนี้ยังช่วยส่งออกรายงานสรุปผลการเข้าเรียนเป็นไฟล์เอ็กซ์เซล (Excel) ด้วย

3.1.3 บล็อกเชน

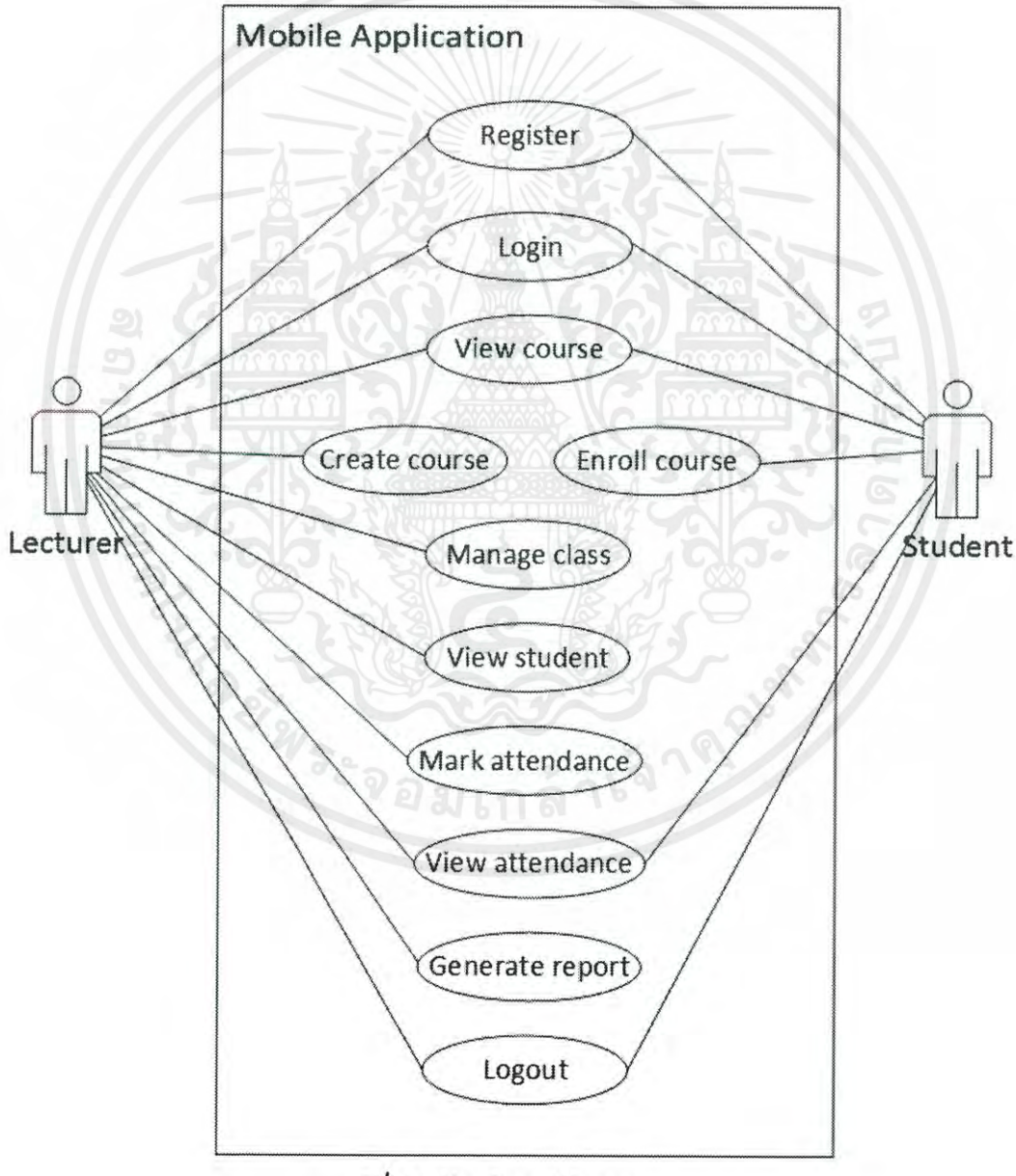
ในส่วนของบล็อกเชนนั้นระบบนี้ได้ใช้อีเธอร์เลียม (Ethereum) สำหรับรันสมาร์ตคอนแทร็กต์ โดยนำมาใช้เป็นส่วนที่ช่วยประมวลผลของระบบ โดยเครือข่ายบล็อกเชนจะเป็นเครือข่ายส่วนตัว ใช้กระบวนการยืนยันข้อมูลร่วมกันโดยใช้การพิสูจน์ด้วยผู้มีอำนาจ (Proof of Authority) โดยตั้งให้มีการตรวจสอบและเขียนบล็อกใหม่ทันทีเฉพาะเมื่อมีรายการทรานแซคชั่นในเครือข่ายเท่านั้น โดยจะมีโหนดหลัก ๆ 3 แบบคือโหนดของผู้ตรวจสอบ (Validator) จะมีหน้าที่ในการนำคำขอทำรายการทรานแซคชั่นมาเขียนลงบล็อกและเพิ่มลงในระบบ โหนดอาร์พีซี (RPC) ซึ่งจะทำหน้าที่เป็นตัวกลางที่จะให้โมไบล์แอปพลิเคชันติดต่อเครือข่ายบล็อกเชนด้วยไลบรารีเว็บทรีเพื่อให้สามารถทำทรานแซคชั่นได้ และบูตโหนด (Bootnode) สำหรับให้เพียร์ในเครือข่ายรู้จักกัน

3.2 ความต้องการของระบบ

1) อาจารย์สามารถสร้างวิชาเรียนได้
2) อาจารย์สามารถเช็คชื่อเข้าเรียนให้กับนักศึกษาได้
3) อาจารย์สามารถตรวจสอบรายชื่อและช่วงเวลาในการเข้าเรียนของนักศึกษาได้
4) นักศึกษาสามารถลงเรียนและตรวจสอบการเข้าเรียนของตนเองได้

5) มีโมไบล์แอปพลิเคชันสำหรับจัดการและแสดงผลข้อมูลเพื่อให้ทั้งอาจารย์และนักศึกษามีความสะดวกในการใช้งาน
6) มีแอปพลิเคชันเซิร์ฟเวอร์ช่วยในการตรวจสอบและคัดกรองผู้ใช้งาน ช่วยเริ่มต้นบัญชีผู้ใช้และช่วยสร้างไฟล์รายงานในรูปแบบเอ็กซ์เซล
7) มีอีเธอร์เลียมเป็นส่วนช่วยในการประมวลผลและเก็บข้อมูล

3.3 แผนภาพยูสเคส (Use Case Diagram)



รูป 3.2 Use Case Diagram

รายละเอียด Use Case จะถูกแสดงรายละเอียดที่ตาราง 3.1 ถึงตาราง 3.11 ด้านล่างต่อไปนี้ โดยแสดงรายละเอียดตามรูปแบบของ UML Use Case Description

ตาราง 3.1 Register Use Case

Use Case Name	Register
Actors	Lecturer, Student
Use Case Purpose	เพื่อสมัครสมาชิกสำหรับใช้งานระบบ
Pre-conditions	ยังไม่ได้สมัครสมาชิก
Post-conditions	-
Main Course	1.ผู้ใช้งานกรอก Email และ Password ของ Gmail ของ @kmitl.ac.th 2.กดปุ่มสมัครสมาชิก
Exceptions	1.Email หรือ Password ไม่ถูกต้อง 2.ไม่สามารถติดต่อกับ Google Servers ได้ 3.ระบบเกิดข้อผิดพลาด

ตาราง 3.2 Login Use Case

Use Case Name	Login
Actors	Lecturer, Student
Use Case Purpose	เข้าสู่ระบบเพื่อใช้งาน
Pre-conditions	ยังไม่ได้เข้าสู่ระบบ
Post-conditions	เข้าสู่ระบบสำเร็จ
Main Course	1.ผู้ใช้งานกรอก Private key ของบัญชีตนเอง 2.ผู้ใช้งานกรอก Email และ Password ของ Gmail ของ @kmitl.ac.th 3.กดปุ่มเข้าสู่ระบบ
Exceptions	1.Email หรือ Password ไม่ถูกต้อง 2.ไม่สามารถติดต่อกับ Google Servers ได้ 3.ระบบเกิดข้อผิดพลาด

ตาราง 3.3 View course Use Case

Use Case Name	View course
Actors	Lecturer, Student
Use Case Purpose	เพื่อแสดงรายการของวิชาที่เปิดสอน/เข้าร่วม
Pre-conditions	เข้าสู่ระบบสำเร็จ
Post-conditions	-
Main Course	ที่หน้าแรกหลังจากเข้าสู่ระบบ จะแสดงรายชื่อวิชาทั้งหมดจากวิชาที่เปิดสอน/เข้าร่วม หรือหากอยู่หน้าอื่น ๆ สามารถเข้าถึงได้โดยคลิกที่ปุ่มเมนูด้านบนซ้าย
Exceptions	ระบบเกิดข้อผิดพลาด

ตาราง 3.4 Create course Use Case

Use Case Name	Create course
Actors	Lecturer
Use Case Purpose	เพื่อสร้างวิชาเรียน
Pre-conditions	เข้าสู่ระบบสำเร็จ
Post-conditions	วิชาเรียนที่สร้างจะถูกเพิ่มเข้าไปในระบบ
Main Course	1.ในหน้าแรก ให้กดปุ่มสร้างวิชาเรียน 2.กรอกข้อมูลของวิชาเรียน 3.กดยืนยัน
Exceptions	1.ข้อมูลที่ระบุไม่ถูกต้อง 2.ระบบเกิดข้อผิดพลาด

ตาราง 3.5 Enroll course Use Case

Use Case Name	Enroll course
Actors	Student
Use Case Purpose	เพื่อเข้าร่วมวิชาเรียน
Pre-conditions	เข้าสู่ระบบสำเร็จ
Post-conditions	นักศึกษาจะถูกเพิ่มลงไปในวิชานั้น
Main Course	1.กดปุ่มลงเรียน 2.กรอกข้อมูลของวิชาเรียน 3.กดยืนยัน
Exceptions	1.ข้อมูลที่ระบุไม่ถูกต้อง 2.ระบบเกิดข้อผิดพลาด

ตาราง 3.6 Manage class Use Case

Use Case Name	Manage class
Actors	Lecturer
Use Case Purpose	เพื่อเพิ่มตารางเวลาเรียนพิเศษ
Pre-conditions	1.เข้าสู่ระบบสำเร็จ 2.เลือกวิชาเรียนที่ต้องการ
Post-conditions	วิชาเรียนที่เลือกจะมีเวลาเรียนเพิ่มขึ้นมา
Main Course	1.กดที่แท็บตารางเรียนและเลือกวัน 2.กดปุ่มเพิ่มเวลาเรียน 3.กรอกเวลาและกดยืนยัน
Exceptions	ระบบเกิดข้อผิดพลาด

ตาราง 3.7 View student Use Case

Use Case Name	View student
Actors	Lecturer
Use Case Purpose	เพื่อแสดงรายชื่อของนักศึกษาทั้งหมดที่อยู่ในวิชาที่เปิดสอน
Pre-conditions	1.เข้าสู่ระบบสำเร็จ 2.เลือกวิชาเรียนที่ต้องการ
Post-conditions	-
Main Course	กดที่แท็บสมาชิก เพื่อแสดงผู้ที่เกี่ยวข้องกับวิชานั้น
Exceptions	ระบบเกิดข้อผิดพลาด

ตาราง 3.8 Mark attendance Use Case

Use Case Name	Mark attendance
Actors	Lecturer
Use Case Purpose	เพื่อเช็คชื่อเข้าเรียนนักศึกษา
Pre-conditions	1.เข้าสู่ระบบสำเร็จ 2.เลือกวิชาเรียนที่ต้องการ 3.มีนักศึกษาลงเรียนในวิชา 4.เวลาที่เริ่มเช็คชื่อต้องตรงกับเวลาเรียน
Post-conditions	วิชาเรียนนั้นจะถูกเพิ่มการเข้าเรียนของนักศึกษา
Main Course	1.กดแท็บเช็คชื่อ 2.กดปุ่มเริ่มการเช็คชื่อ
Exceptions	ระบบเกิดข้อผิดพลาด

ตาราง 3.9 View attendance Use Case

Use Case Name	View attendance
Actors	Lecturer, Student
Use Case Purpose	เพื่อแสดงข้อมูลการเข้าเรียนของนักศึกษาทั้งหมดในวิชาที่เปิดสอน/ที่เข้าร่วม
Pre-conditions	1.เข้าสู่ระบบสำเร็จ 2.เลือกวิชาเรียนที่ต้องการ
Post-conditions	-
Main Course	หลังจากเลือกวิชา จะแสดงรายการการเข้าเรียนของวิชาที่เปิดสอน/เข้าร่วมให้โดยอัตโนมัติ แต่หากอยู่หน้าอื่นให้กดที่แท็บไทม์ไลน์
Exceptions	ระบบเกิดข้อผิดพลาด

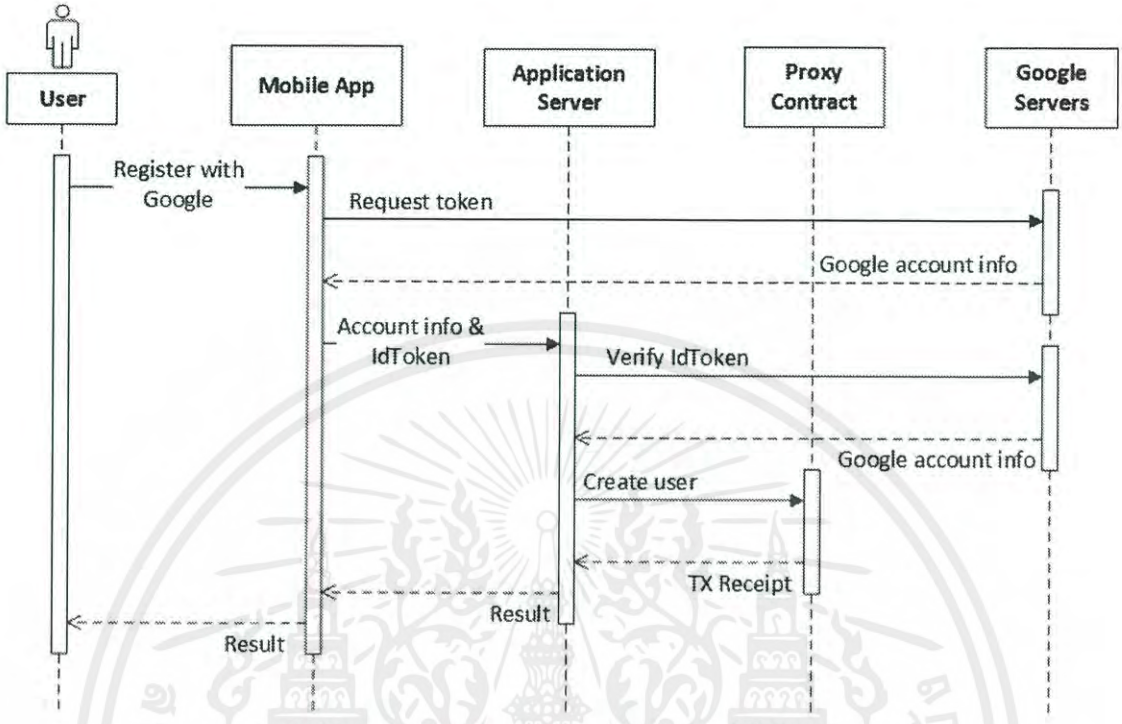
ตาราง 3.10 Generate report Use Case

Use Case Name	Generate report
Actors	Lecturer
Use Case Purpose	เพื่อส่งออกข้อมูลการเข้าเรียนของวิชาที่เปิดสอน
Pre-conditions	1.เข้าสู่ระบบสำเร็จ 2.เลือกวิชาเรียนที่ต้องการ
Post-conditions	-
Main Course	1.กดที่แท็บไทม์ไลน์ 2.กดที่ปุ่มส่งออกรายงาน
Exceptions	ระบบเกิดข้อผิดพลาด

ตาราง 3.11 Logout Use Case

Use Case Name	Logout
Actors	Lecturer, Student
Use Case Purpose	ออกจากระบบ
Pre-conditions	เข้าสู่ระบบสำเร็จ
Post-conditions	ออกจากระบบสำเร็จ
Main Course	กดปุ่มออกจากระบบ
Exceptions	ระบบเกิดข้อผิดพลาด

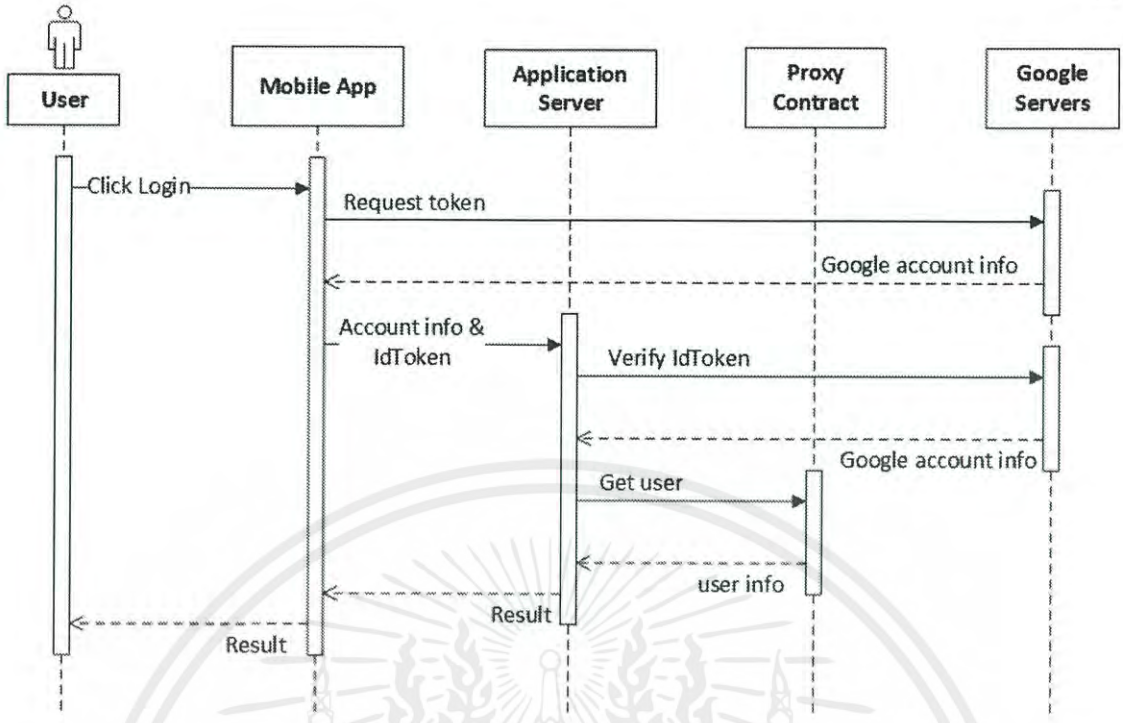
3.4 แผนภาพซีเควนซ์ (Sequence Diagram)



รูป 3.3 Sequence Diagram การสมัครเข้าใช้งาน

3.4.1 การสมัครใช้งานแอปพลิเคชัน

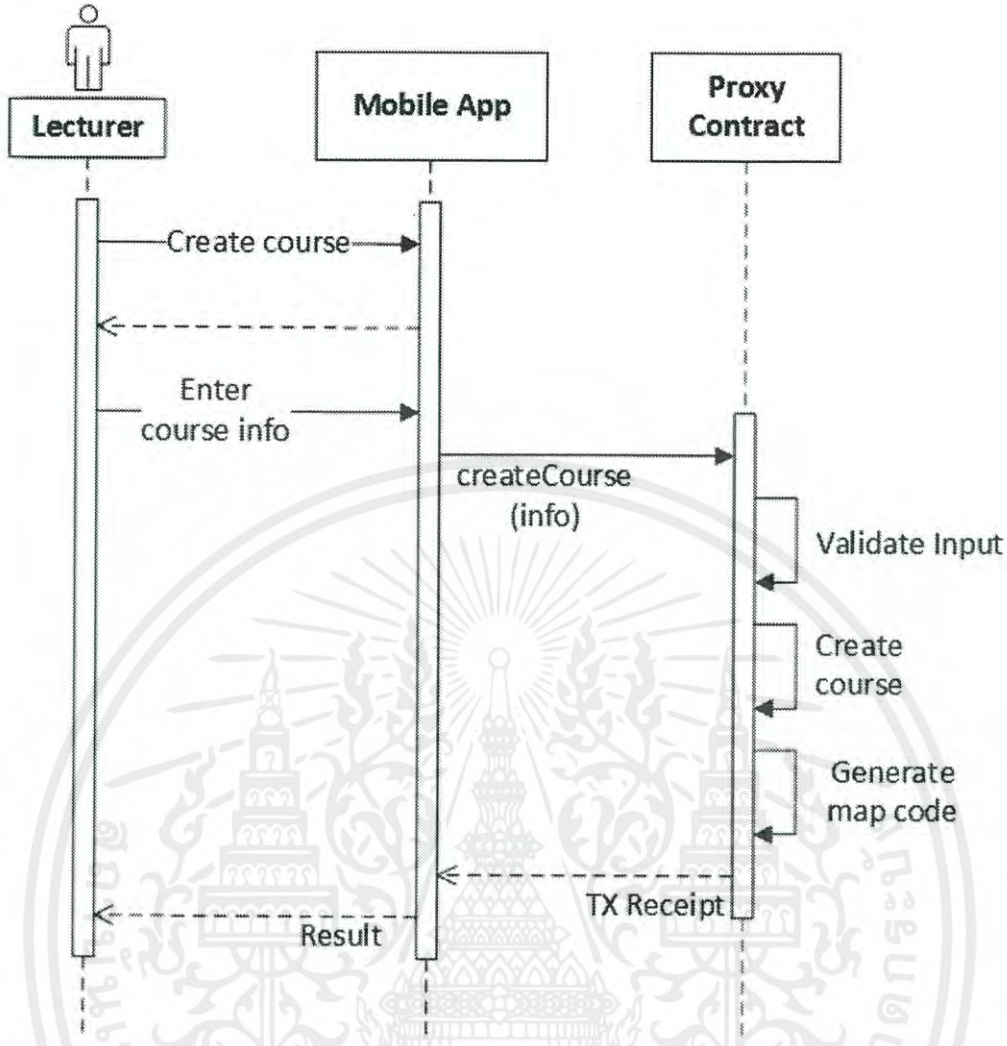
ผู้ใช้งานจะสมัครใช้งานโดยการใช้บัญชีอีเมลของกูเกิล ซึ่งจะเป็นการใช้บริการกูเกิลโอออท (Google OAuth) เพื่อให้ได้รับกูเกิลแอ็กเซสโทเค็น (Google access token) โดยจะนำไปใช้ดึงข้อมูลของผู้ใช้มาตรวจสอบว่าเป็นอีเมลของสถาบันหรือไม่ หากใช่จะส่งโทเค็นไปที่แอปพลิเคชันเซิร์ฟเวอร์ เพื่อแอปพลิเคชันเซิร์ฟเวอร์จะได้ตรวจสอบโทเค็นอีกครั้งด้วยตนเอง หากเป็นอีเมลสถาบันจริงจะสร้างบัญชีให้โดยการส่งข้อมูลที่ดึงได้จากโทเค็นไปยังเครือข่ายอีเธอร์เลียมด้วยเว็บทรีโดยสร้างทรานแซคชั่นติดต่อไปยังคอนแทร็กต์พร็อกซีและเรียกใช้ฟังก์ชันการสร้างบัญชี หลังจากสร้างบัญชีสำเร็จก็จะโอนอีเธอร์ให้กับบัญชีนั้นเพื่อให้สามารถใช้งานเครือข่ายอีเธอร์เลียมได้อย่างมีประสิทธิภาพ



รูป 3.4 Sequence Diagram การเข้าสู่ระบบ

3.4.2 การเข้าสู่ระบบเพื่อใช้งานแอปพลิเคชัน

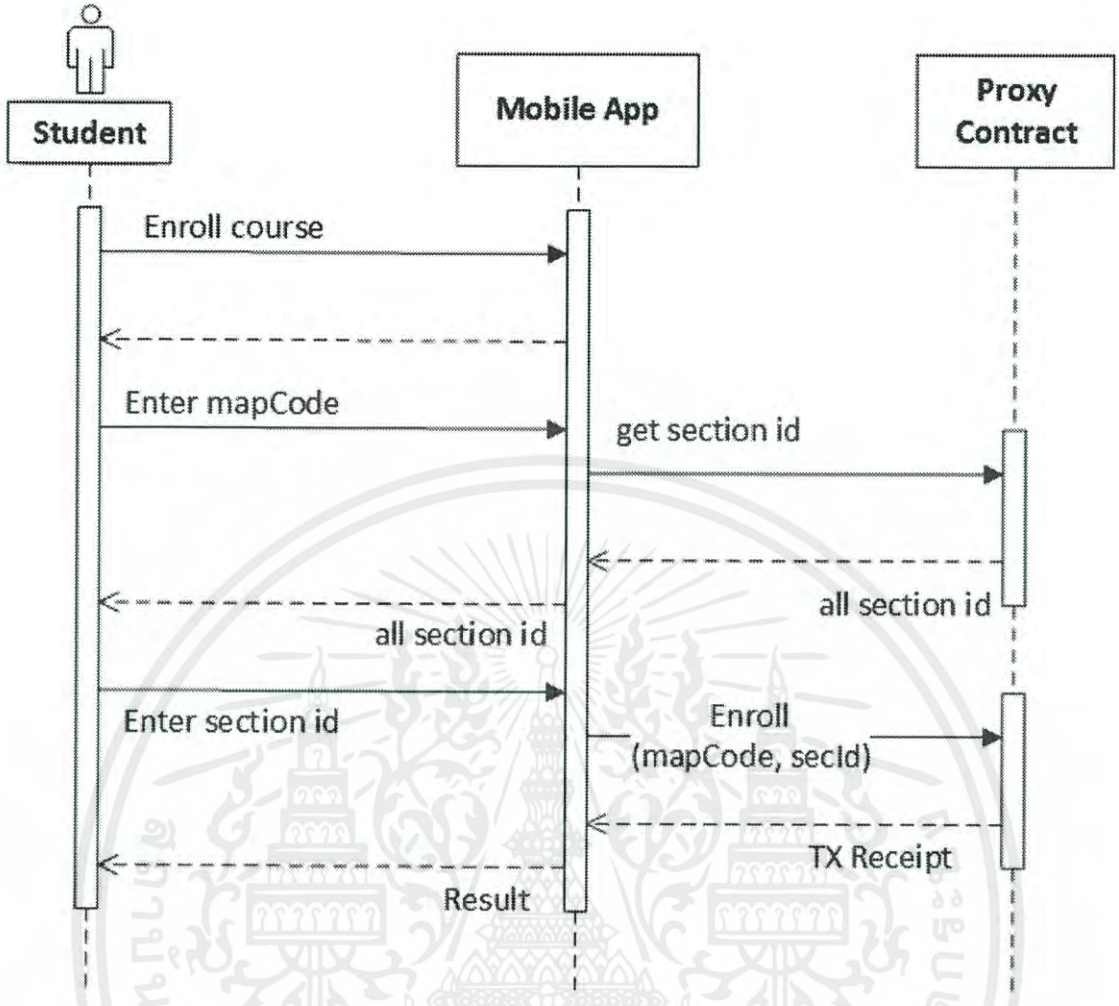
จะคล้ายกับตอนสมัครใช้งาน คือผู้ใช้เพียงแค่เข้าสู่ระบบด้วยอีเมลที่เคยใช้สมัครสมาชิกไปแล้ว โดยหากแอปพลิเคชันเซิร์ฟเวอร์ตรวจสอบโทเค็นแล้วพบว่าถูกต้อง จะทำการดึงข้อมูลจากคอนแทร็กต์พร็อกซีที่ตรงกับอีเมลนั้น และนำมาแสดงที่หน้าหลักหลังจากเข้าสู่ระบบสำเร็จ เช่น ชื่อวิชา เป็นต้น



รูป 3.5 Sequence Diagram การสร้างวิชาเรียนของอาจารย์

3.4.3 การสร้างวิชาเรียนของอาจารย์

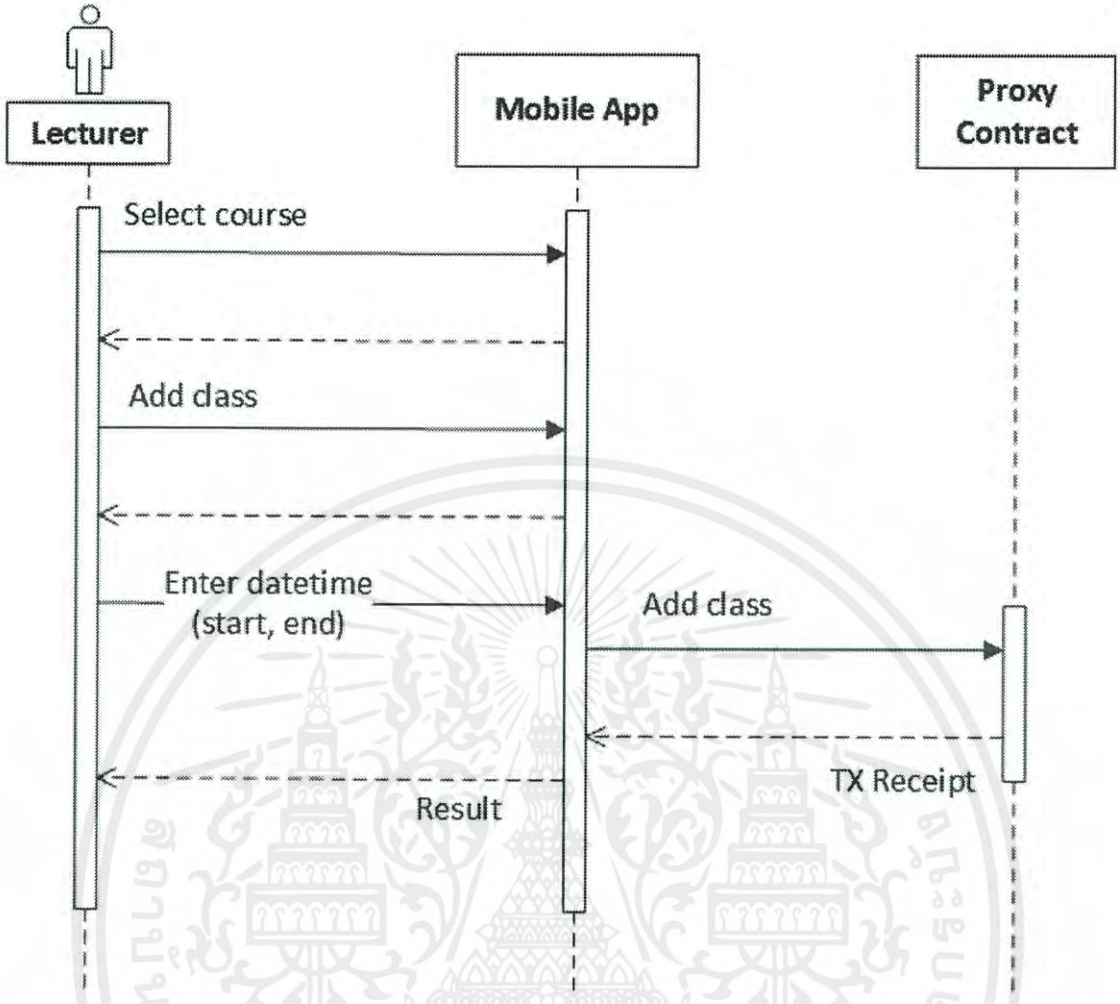
หลังจากเข้าสู่ระบบสำเร็จโมไบล์แอปพลิเคชันจะพามาที่หน้าแรก ซึ่งเมื่ออาจารย์กดปุ่มสร้างวิชาเรียน โมไบล์แอปพลิเคชันจะให้อาจารย์กรอกข้อมูลของวิชาเรียนที่จะสร้าง เมื่อกรอกสำเร็จจะทำการนำข้อมูลที่ได้มาสร้างเป็นทรานแซคชั่นและส่งไปยังคอนแทร็กต์พร็อกซีเพื่อเรียกใช้งานฟังก์ชันการสร้างวิชาเรียน เมื่อคอนแทร็กต์พร็อกซีถูกเรียกใช้งานแล้วก็จะเริ่มทำการตรวจสอบข้อมูลของวิชาเรียนที่ได้ว่าถูกต้องตามรูปแบบที่กำหนดไว้หรือไม่ หากถูกต้องก็จะนำข้อมูลนั้นมาสร้างวิชาเรียนและเพิ่มเข้าสู่ระบบ แล้วจะทำการสร้าง mapCode ซึ่งเป็นตัวอักษรภาษาอังกฤษผสมกับตัวเลข ใช้สำหรับการอ้างอิงถึงวิชาเรียนเมื่อนักศึกษาต้องการลงเรียน



รูป 3.6 Sequence Diagram การลงเรียนของนักศึกษา

3.4.4 การลงเรียนของนักศึกษา

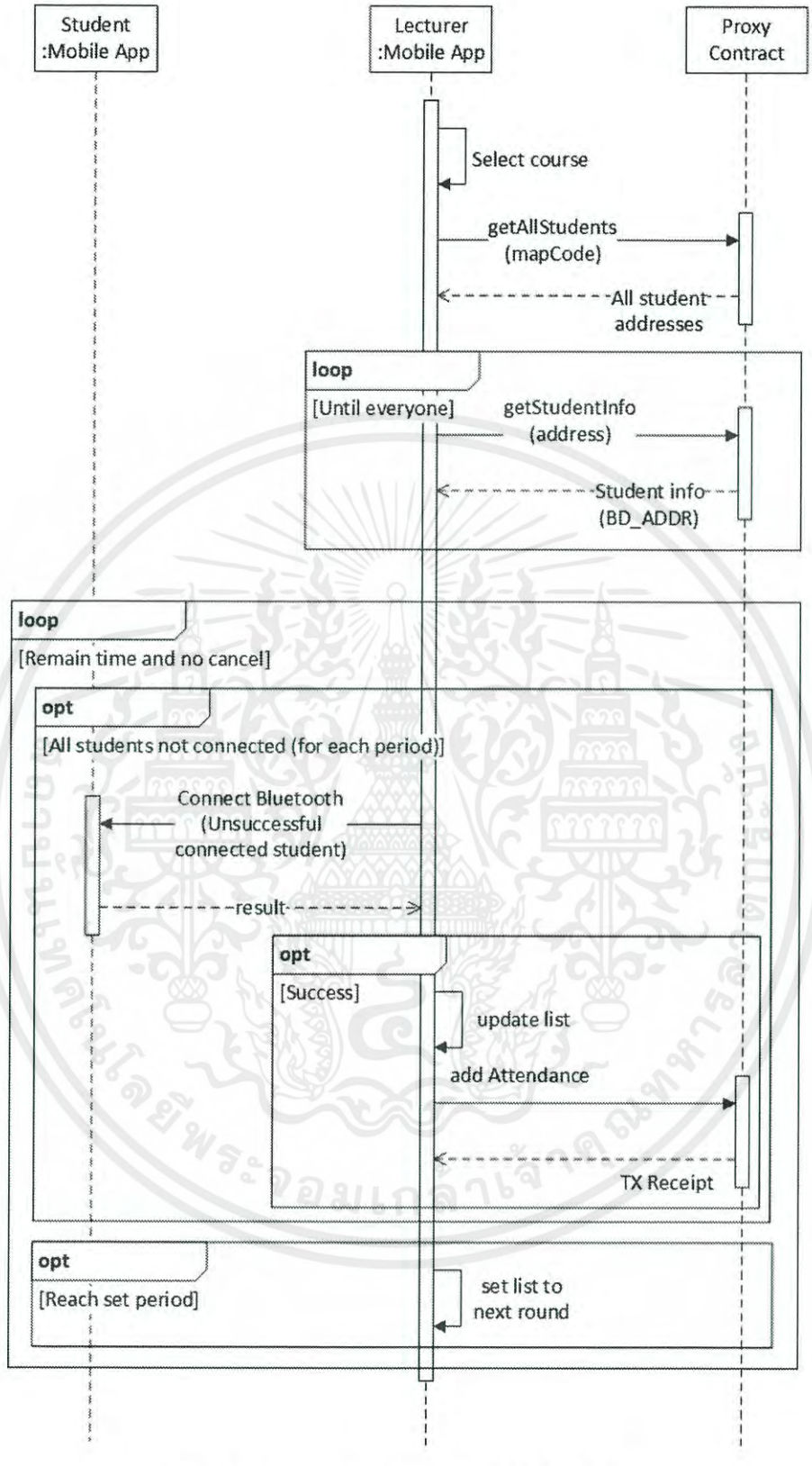
หลังจากทำการเข้าสู่ระบบสำเร็จแล้ว หากนักศึกษากดปุ่มลงเรียน โมไบล์แอปพลิเคชันจะให้นักศึกษากรอก mapCode ซึ่งใช้สำหรับอ้างอิงถึงวิชาเรียนที่นักศึกษาต้องการจะลงเรียน เมื่อกรอกและกดเข้าร่วมแล้ว โมไบล์แอปพลิเคชันจะทำการเรียกใช้งานคอนแทร็กต์พร็อกซีเพื่อดึงข้อมูลกลุ่มเรียนทั้งหมดของวิชานั้น โดยนำข้อมูลที่ได้รับมาแสดงเพื่อให้นักศึกษาเลือกว่าจะลงเรียนกลุ่มเรียนไหน เมื่อนักศึกษาเลือกแล้วโมไบล์แอปพลิเคชันก็จะทำการสร้างทรานแซคชั่นเพื่อเรียกใช้งานคอนแทร็กต์พร็อกซีฟังก์ชันการเพิ่มนักศึกษาเข้าวิชาเรียน โดยหลังจากลงเรียนเสร็จแล้ว โมไบล์แอปพลิเคชันก็จะอัพเดทข้อมูลวิชาเรียนของนักศึกษาและนำมาแสดงผลให้ใหม่ในหน้าแรก



รูป 3.7 Sequence Diagram การเพิ่มเวลาเรียนของอาจารย์

3.4.5 การเพิ่มเวลาเรียนของอาจารย์

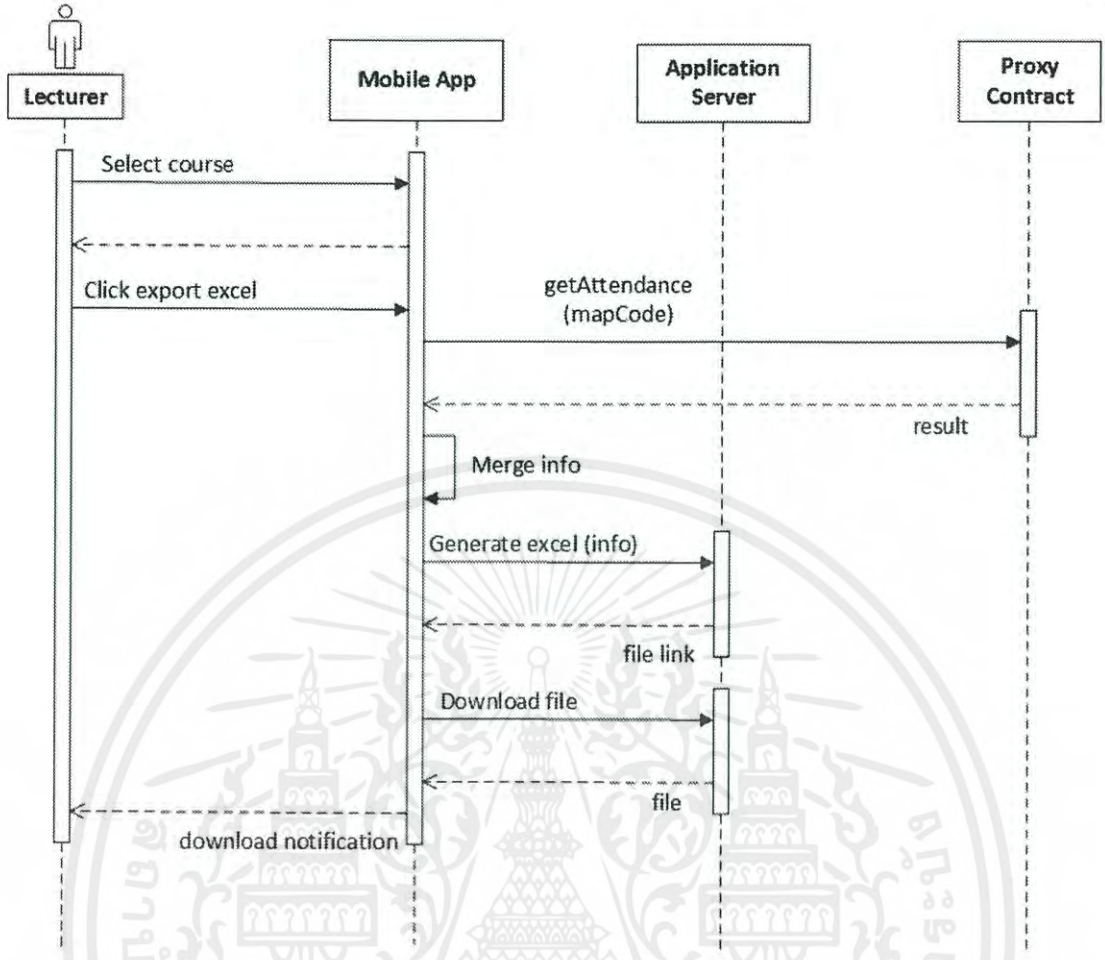
หลังจากที่เข้าสู่ระบบสำเร็จแล้ว เมื่ออาจารย์ต้องการจะเพิ่มเวลาเรียน จะต้องทำการเลือกวิชาที่ต้องการเพิ่มเวลาเรียนใหม่ก่อน จากนั้นเมื่อกดที่แท็บตารางเวลาเรียนแล้วเลือกวันที่ต้องการจะเพิ่มเวลาเรียน โมไบล์แอปพลิเคชันจะให้อาจารย์เลือกเวลาเริ่มต้นและสิ้นสุดของเวลาเรียนใหม่ หลังจากกดยืนยันแล้ว โมไบล์แอปพลิเคชันก็จะสร้างทรานแซคชั่นและส่งไปยังคอนแทร็กต์พร็อกซีเพื่อเรียกใช้ฟังก์ชันการเพิ่มเวลาเรียน หลังจากเพิ่มเวลาเรียนใหม่เสร็จแล้วโมไบล์แอปพลิเคชันก็จะทำการอัปเดตข้อมูลในหน้าตารางเรียนให้ใหม่



รูป 3.8 Sequence Diagram การเช็คชื่อเข้าเรียน

3.4.6 การเช็คชื่อเข้าเรียน

หลังจากเข้าสู่ระบบสำเร็จ เมื่ออาจารย์ต้องการเช็คชื่อเข้าเรียนนักศึกษา อาจารย์จะต้องเลือกวิชาเรียนที่ต้องการเช็คชื่อก่อน จากนั้นเมื่อกดปุ่มเริ่มเช็คชื่อเข้าเรียน โมไบล์แอปพลิเคชันจะทำการติดต่อคอนแทร็กต์พร็อกซีเพื่อดึงข้อมูลของนักศึกษาที่ได้ลงเรียนในวิชานั้น โดยจะได้ออกมาเป็นแอดเดรสบนอีเธอร์เลียม จากนั้นนำแต่ละแอดเดรสที่ได้ไปดึงข้อมูลของนักศึกษาโดยเฉพาะหมายเลขบลูทูธของอุปกรณ์สมาร์ตโฟนที่ได้ทำการบันทึกไว้ โดยเมื่อได้รับข้อมูลครบหมดทุกคนที่ลงเรียนในวิชานั้นแล้ว จะเริ่มทำการเช็คชื่อเข้าเรียน โดยจะวนทำตลอดจนกว่าจะหมดเวลาเรียนของคลาสนั้น ๆ หรืออาจารย์กดยกเลิกการเช็คชื่อเอง โดยในแต่ละรอบนั้นโมไบล์แอปพลิเคชันของอาจารย์จะนำข้อมูลหมายเลขบลูทูธที่ได้มาทำการเชื่อมต่อกับสมาร์ตโฟนของนักศึกษาผ่านบลูทูธทีละอุปกรณ์ หากเชื่อมต่อสำเร็จโมไบล์แอปพลิเคชันจะทำการบันทึกไว้ในรายการเพื่อใช้สำหรับดูสถานะการเชื่อมต่อ โดยหากเชื่อมต่อสำเร็จไปแล้วจะไม่ทำการเชื่อมต่อซ้ำ จากนั้นจะสร้างทรานแซคชั่นส่งไปยังคอนแทร็กต์พร็อกซีเพื่อบันทึกข้อมูลการเข้าเรียนของนักศึกษา เมื่อเชื่อมต่อกับนักศึกษาครบทุกคนแล้วในรอบนั้น ๆ ก็จะหยุดการเชื่อมต่อบลูทูธและรอจนกว่าจะถึงเวลาสิ้นสุดของรอบนั้น ๆ เมื่อสิ้นสุดรอบปัจจุบันโมไบล์แอปพลิเคชันจะนำรายการสถานะการเชื่อมต่อที่ได้บันทึกไว้มาตั้งค่าใหม่สำหรับรอบถัดไป และเริ่มทำการเชื่อมต่อบลูทูธกับสมาร์ตโฟนของนักศึกษาในรอบใหม่ต่อไป

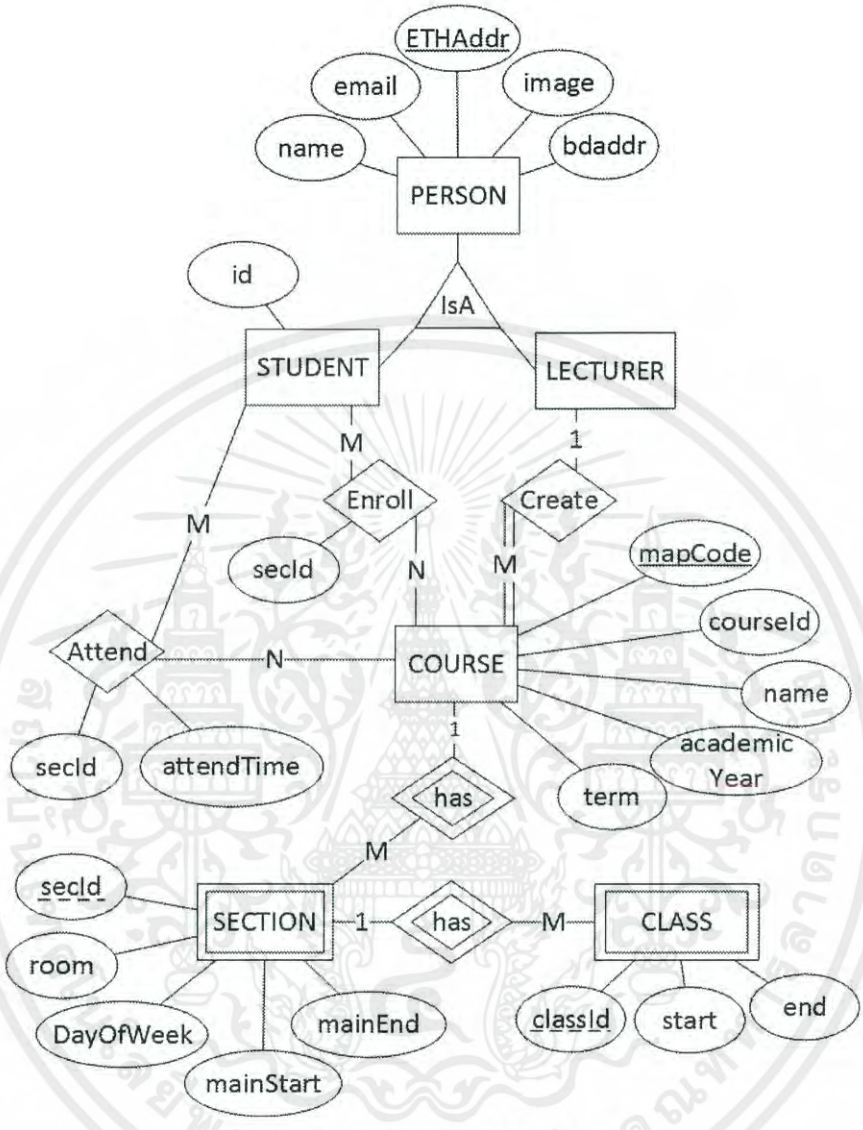


รูป 3.9 Sequence Diagram การส่งออกรายงานของอาจารย์

3.4.7 การส่งออกรายงานของอาจารย์

หลังจากเข้าสู่ระบบสำเร็จแล้ว หากต้องการส่งออกรายงาน อาจารย์จะต้องเลือกวิชาเรียนที่ต้องการส่งออกรายงานในหน้าแรกก่อน จากนั้นกดที่ปุ่มส่งออก หลังจากนั้นโมไบล์แอปพลิเคชันจะทำการดึงข้อมูลการเข้าเรียนของนักศึกษาทุกคนที่ลงเรียนในวิชานั้นจากคอนแทร็กต์พร็อกซี และจะนำข้อมูลที่ได้ทั้งหมดมาทำให้อยู่ในรูปแบบตามที่กำหนดไว้ จากนั้นจะทำการส่งข้อมูลไปที่แอปพลิเคชันเซิร์ฟเวอร์เพื่อที่จะนำข้อมูลที่ได้มาเขียนเป็นไฟล์รายงานในรูปแบบเอ็กซ์เซล และเมื่อเขียนไฟล์เสร็จจะส่งลิงก์ของไฟล์เอ็กซ์เซลที่สร้างได้กลับไปยังโมไบล์แอปพลิเคชัน โมไบล์แอปพลิเคชันก็จะดาวน์โหลดไฟล์ตามลิงก์ที่ได้รับและจะขึ้นแจ้งว่าได้ดาวน์โหลดไฟล์สำเร็จ โดยไฟล์จะถูกเก็บอยู่บนโมไบล์แอปพลิเคชัน

3.5 แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล (Entity Relationship Diagram)



รูป 3.10 Entity Relationship Diagram

โดยเอนทิตีและรีเลชันต่าง ๆ มีรายละเอียดดังนี้

3.5.1 Person

เป็นเอนทิตีของบุคคลทั่วไป โดยเป็นซูเปอร์ไทป์ (Supertype) ของเอนทิตีที่เกี่ยวข้องกับผู้ใช้ทั้งหมด คือ อาจารย์ (Lecturer) และนักศึกษา (Student) ซึ่งมีแอตทริบิวต์ดังต่อไปนี้

1) ETHAddr คือแอดเดรสของบัญชีบนอีเธอร์เลียม ใช้เป็น Primary Key สำหรับระบุตัวของแต่ละบุคคล

2) name คือชื่อของแต่ละบุคคล

3) email คืออีเมลของแต่ละบุคคล

4) img คือลิงก์รูปภาพของแต่ละบุคคล

5) bdaddr คือหมายเลขอุปกรณ์บลูทูธของแต่ละบุคคล

3.5.2 Lecturer

เป็นเอนทิตีของอาจารย์ โดยจะได้รับแอตทริบิวต์ของเอนทิตี Person ด้วย

3.5.3 Student

เป็นเอนทิตีของนักศึกษา จะได้รับแอตทริบิวต์ของเอนทิตี Person ด้วย และจะมีแอตทริบิวต์ที่เพิ่มขึ้นมาคือ id ซึ่งเป็นรหัสนักศึกษา

3.5.4 Course

เป็นเอนทิตีของวิชาเรียน โดยมีแอตทริบิวต์ดังต่อไปนี้

1) mapCode เป็นตัวอักษรผสมตัวเลขสุ่มขึ้นเพื่อใช้ในการระบุถึงวิชาเรียน

2) courseId เป็นรหัสวิชาเรียน

3) name เป็นชื่อวิชาเรียน

4) academicYear เป็นปีการศึกษาของวิชาเรียน

5) term เป็นภาคการเรียนของวิชาเรียนในปีการศึกษานั้น ๆ

3.5.5 Section

เป็นเอนทิตีของกลุ่มเรียน เป็น weak เอนทิตี ซึ่งขึ้นอยู่กับเอนทิตี Course โดยมีแอตทริบิวต์ดังต่อไปนี้

1) secId เป็นรหัสของกลุ่มเรียนแต่ละกลุ่ม

2) room เป็นสถานที่หรือห้องที่ใช้ในการเรียนของแต่ละกลุ่มเรียน

3) DayOfWeek เป็นวันที่มีเรียนคลาสเรียนหลัก

4) mainStart เป็นเวลาที่เริ่มเรียนคลาสเรียนหลัก

5) mainEnd เป็นเวลาสิ้นสุดคลาสเรียนหลัก

3.5.6 Class

เป็นเอนทิตีของคลาสเรียน เป็น weak เอนทิตี ซึ่งขึ้นอยู่กับเอนทิตี Section โดยมีแอตทริบิวต์ดังต่อไปนี้

1) classId เป็นตัวระบุสำหรับคลาสแต่ละคลาส

2) start เป็นวันเวลาเริ่มเรียนของคลาส อยู่ในรูปแบบของ unix timestamp

3) end เป็นวันเวลาสิ้นสุดการเรียนของคลาส อยู่ในรูปแบบของ unix timestamp

3.5.7 Enroll

เป็นรีเลชันระหว่างนักศึกษาและวิชาเรียน หมายความว่า นักศึกษาแต่ละคนสามารถลงเรียนได้หลายวิชาเรียน และแต่ละวิชาเรียนก็สามารถถูกลงเรียนได้โดยนักศึกษาหลายคน โดยมีแอตทริบิวต์คือ secId เพื่อใช้สำหรับอ้างถึงกลุ่มเรียนที่ได้ลงเรียนในวิชานั้น ๆ

3.5.8 Attend

เป็นรีเลชันของการเข้าเรียนระหว่างนักศึกษาและวิชาเรียน หมายความว่า นักศึกษาแต่ละคนสามารถเข้าเรียนได้หลายวิชาวิชาเรียน และทุกวิชาเรียนสามารถถูกเข้าเรียนจากนักศึกษาได้หลายคน โดยมีแอตทริบิวต์ดังต่อไปนี้

1) secId เป็นตัวระบุกลุ่มเรียนที่ได้เข้าเรียน

2) attendTime เป็น timestamp สำหรับเก็บเวลาที่เข้าเรียน

3.5.9 Create

เป็นรีเลชันการสร้างวิชาเรียนระหว่างอาจารย์และวิชาเรียน หมายความว่า อาจารย์แต่ละคนสามารถสร้างวิชาเรียนได้หลายวิชาเรียน แต่ทุก ๆ วิชาเรียนสามารถถูกสร้างจากอาจารย์ได้เพียงคนเดียว

3.5.10 Has

เป็นรีเลชันที่หมายถึงการมีอยู่ สำหรับระหว่างวิชาเรียนและกลุ่มเรียนนั้นหมายความได้ว่า ในแต่ละวิชาเรียนจะมีกลุ่มเรียนได้หลายกลุ่ม แต่ในแต่ละกลุ่มเรียนจะอยู่ในวิชาเรียนใดวิชาเรียนหนึ่งเท่านั้น ส่วนสำหรับระหว่างกลุ่มเรียนและคลาสก็มีความหมายในทำนองเดียวกัน คือ กลุ่มเรียนหนึ่งมีได้หลายคลาส แต่ทุก ๆ คลาสจะต้องเป็นของกลุ่มเรียนใดกลุ่มเรียนหนึ่ง

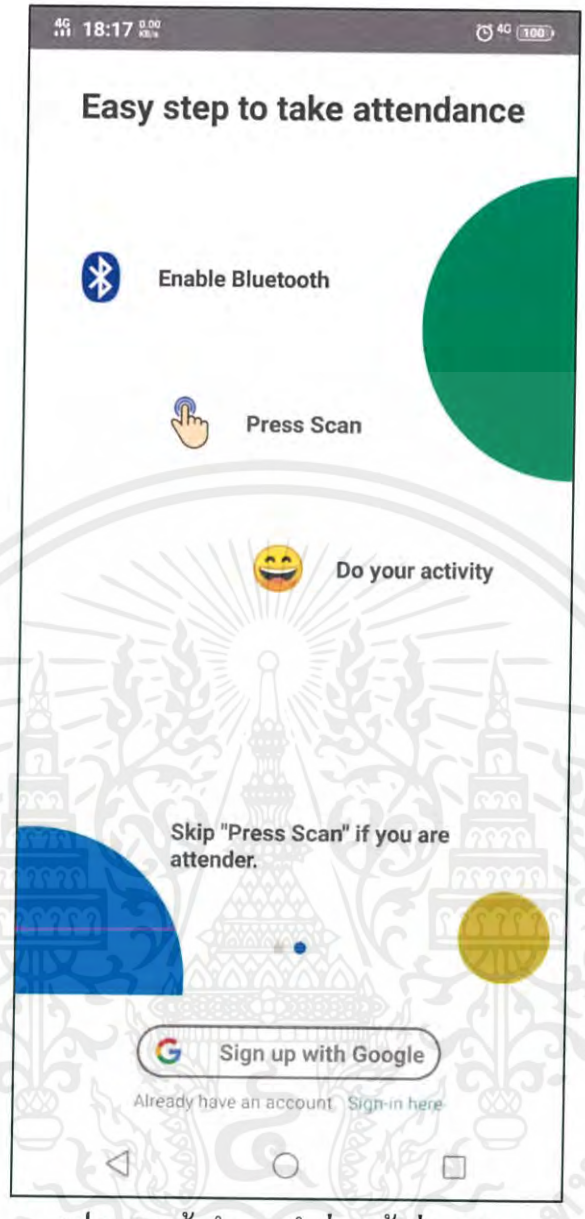
3.6 ส่วนติดต่อผู้ใช้งาน

สำหรับส่วนติดต่อผู้ใช้งาน จะแบ่งได้เป็น 3 ส่วนคือ ส่วนที่เหมือนกันของอาจารย์และนักศึกษา ส่วนอาจารย์และส่วนของนักศึกษา โดยมีการออกแบบดังนี้

3.6.1 ส่วนที่เหมือนกันของอาจารย์และนักศึกษา

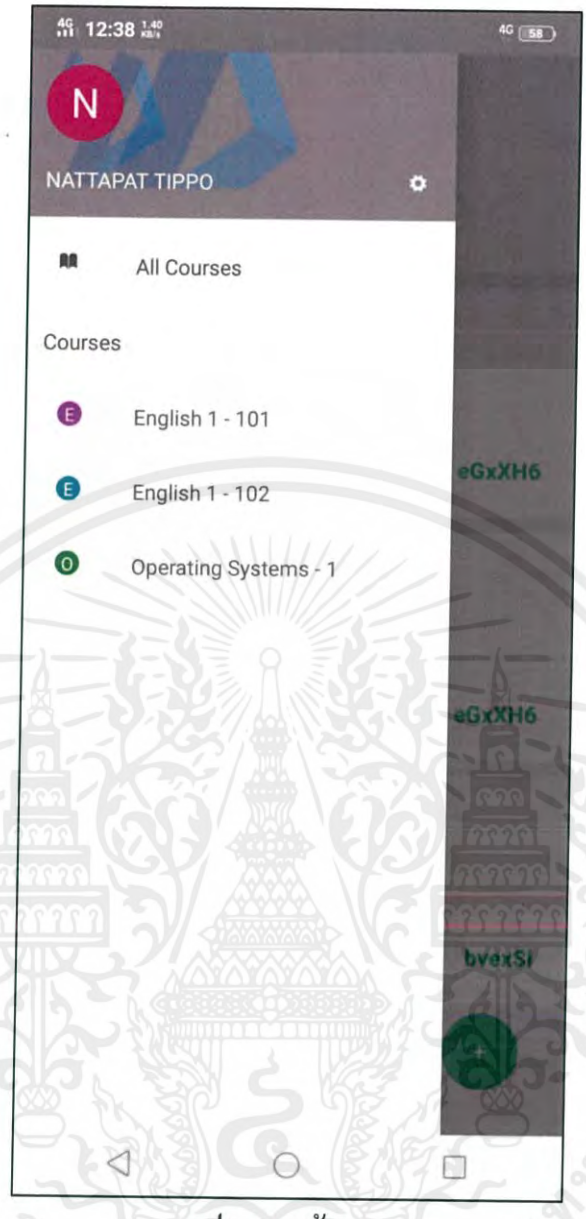
รูป 3.11 หน้าคำแนะนำก่อนเข้าสู่ระบบ 1

เป็นหน้าสำหรับบอกถึงประโยชน์ที่จะได้รับจากการใช้แอปพลิเคชันนี้ มีรูปแบบเป็นสไลด์เลื่อนซ้ายขวา โดยหน้านี้เป็นสไลด์แรก ซึ่งบอกถึงการช่วยประหยัดเวลาในการเช็คชื่อเข้าเรียน ซึ่งไม่จำเป็นต้องขานชื่อนักศึกษาในการเช็คชื่อให้เสียเวลา



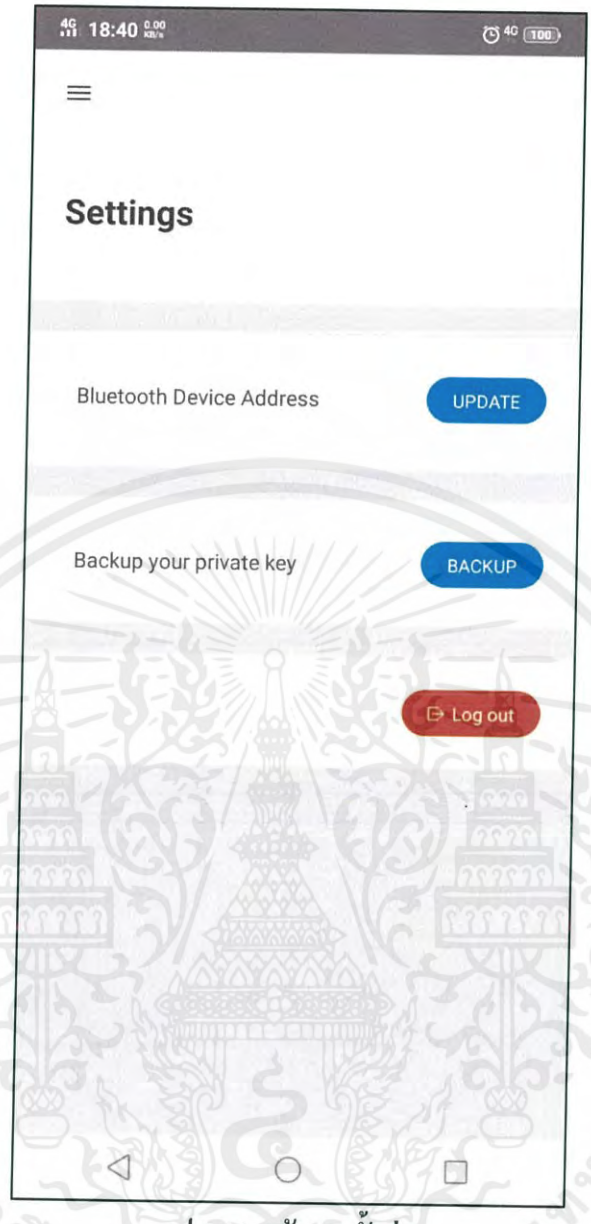
รูป 3.12 หน้าคำแนะนำก่อนเข้าสู่ระบบ 2

เป็นหน้าสำหรับบอกถึงประโยชน์ที่จะได้รับจากการใช้แอปพลิเคชันนี้ มีรูปแบบเป็นสไลด์เลื่อนซ้ายขวา โดยหน้านี้เป็นสไลด์ที่สอง ซึ่งบอกถึงการความง่ายในการเช็คชื่อเข้าเรียน เพียงแค่เปิดบลูทูธไว้แล้วกดเริ่มการเช็คชื่อ หลังจากนั้นก็ทำกิจกรรมของคุณต่อไป



รูป 3.13 หน้าเมนู

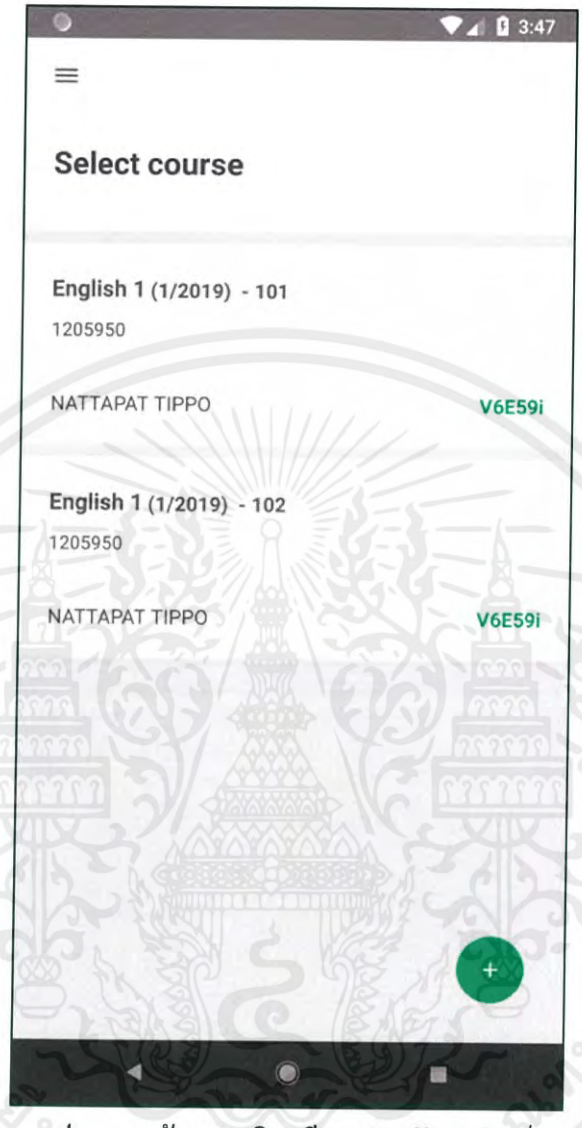
หน้าปุ่มเมนูสามารถเข้าได้ด้วยการคลิกที่ปุ่มด้านซ้ายบน โดยสามารถใช้สลับวิชาได้ หาก เป็นนักศึกษาจะแสดงวิชาที่ได้ลงเรียนไว้ แต่หากเป็นอาจารย์จะแสดงวิชาที่ได้เปิดสอนไว้



รูป 3.14 หน้าการตั้งค่า

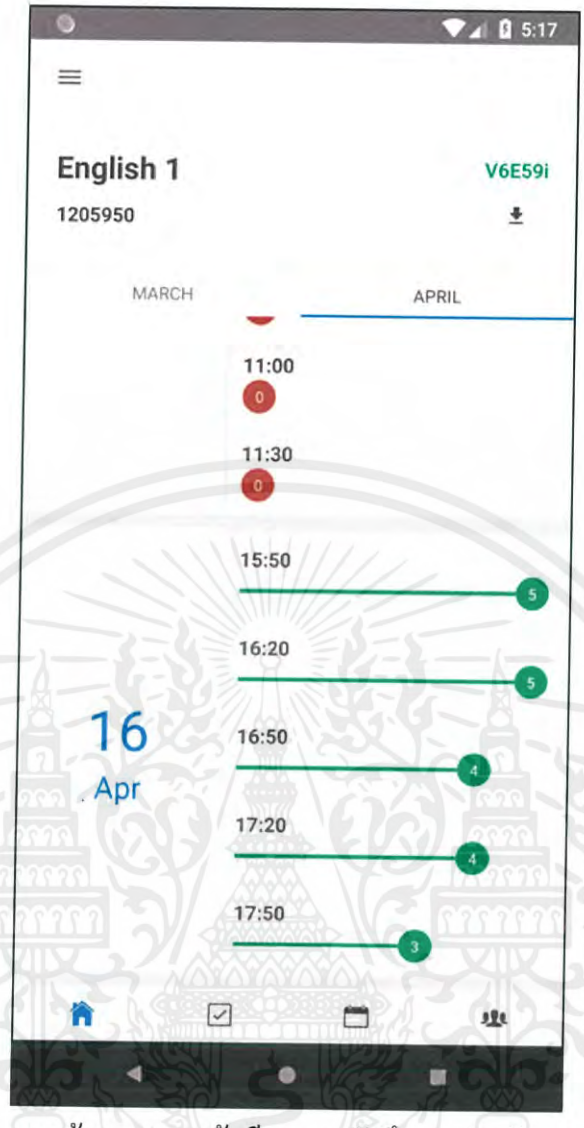
เป็นหน้าสำหรับใช้ตั้งค่า ซึ่งจะมีการอัปเดตหมายเลขบลูทูธ การ Export Private key เพื่อใช้เข้าสู่ระบบในครั้งหน้า หรือการออกจากระบบ

3.6.2 ส่วนของอาจารย์



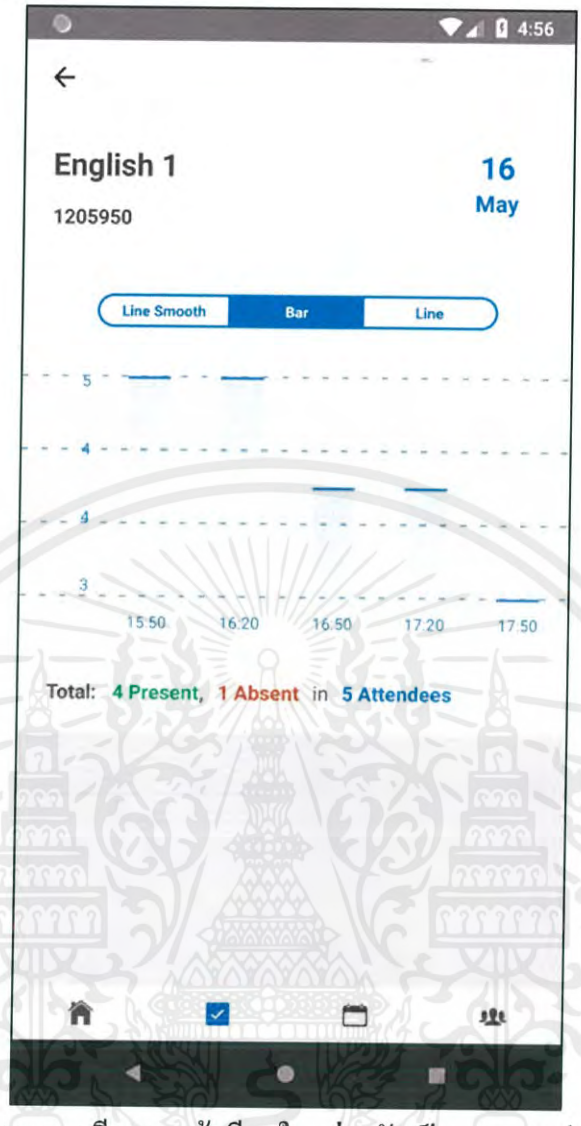
รูป 3.15 หน้าแสดงวิชาเรียน สำหรับอาจารย์

เมื่อเข้าสู่ระบบได้สำเร็จ แอปพลิเคชันจะพามาหน้านี้โดยอัตโนมัติซึ่งเป็นหน้าแรก โดยจะแสดงรายการวิชาเรียนทั้งหมดที่ได้สร้างไว้ และหากต้องการสร้างวิชาเรียนใหม่ ก็สามารถกดที่ปุ่มบวกด้านล่างได้



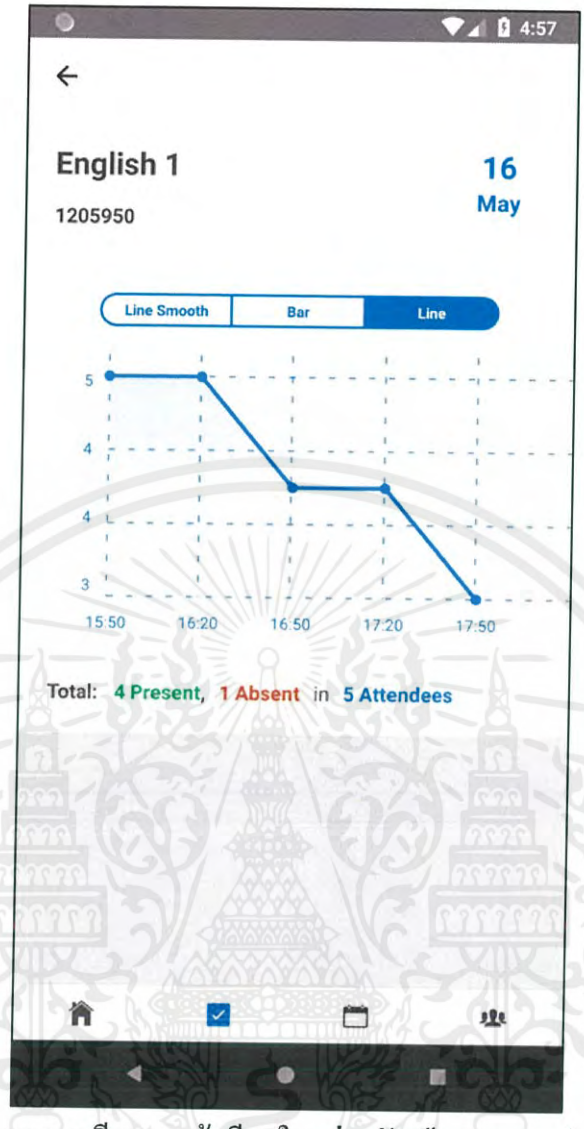
รูป 3.16 หน้าแสดงการเข้าเรียนของนักศึกษา สำหรับอาจารย์

หลังจากที่อาจารย์ได้ทำการเข้าสู่ระบบสำเร็จแล้ว และเลือกวิชาที่แสดงไว้ในหน้าแรก เมื่อเข้ามาในวิชาที่เลือก จะแสดงสรุปรายการการเข้าเรียนของนักศึกษาในทั้งเดือน โดยจะทำการสรุปเป็นแต่ละช่วงเวลาของคลาสนั้น ๆ และในแต่ละช่วงเวลาจะมีแถบสีเพื่อแบ่งให้ดูได้ง่าย

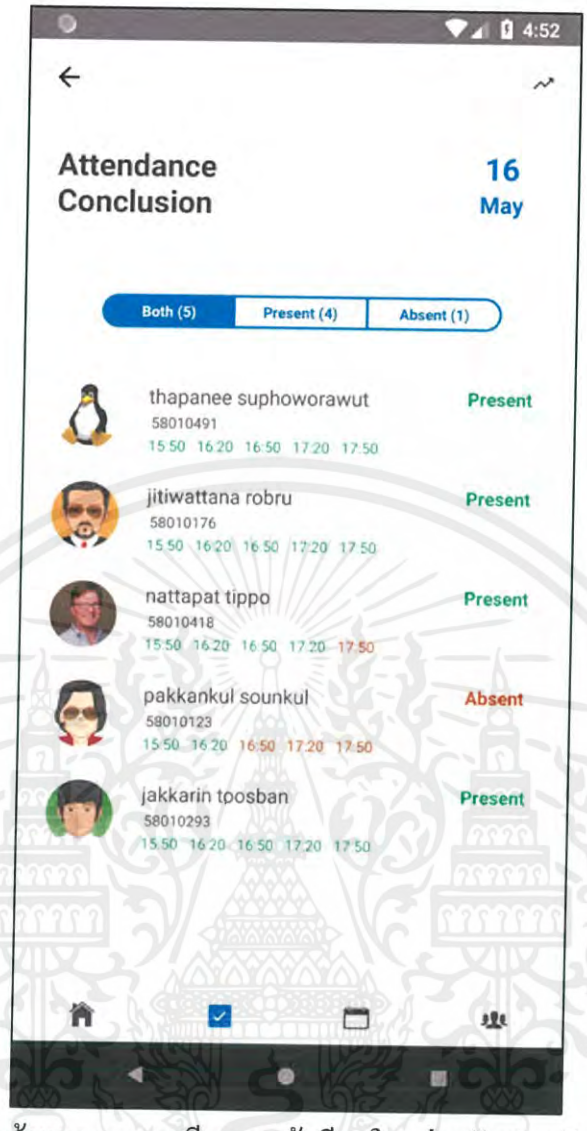


รูป 3.17 หน้าแสดงรายละเอียดการเข้าเรียนในแต่ละวันเป็นแบบกราฟ สำหรับอาจารย์ 1

หลังจากที่เลือกวิชาเรียนที่ต้องการแล้วและกดที่วันนั้น ๆ เพื่อแสดงรายละเอียดเพิ่มเติม ที่มุมบนจะมีปุ่มสำหรับการพามาหน้านี้ ซึ่งเป็นหน้าแสดงการสรุปการเข้าเรียนเป็นกราฟ โดยจะสามารถเลือกได้ถึง 3 แบบ โดยมีกราฟเส้นแบบธรรมดา กราฟเส้นแบบลื่นไหลและกราฟแท่ง ผู้ใช้สามารถเลือกได้ว่าต้องการจะดูแบบไหนตามต้องการ

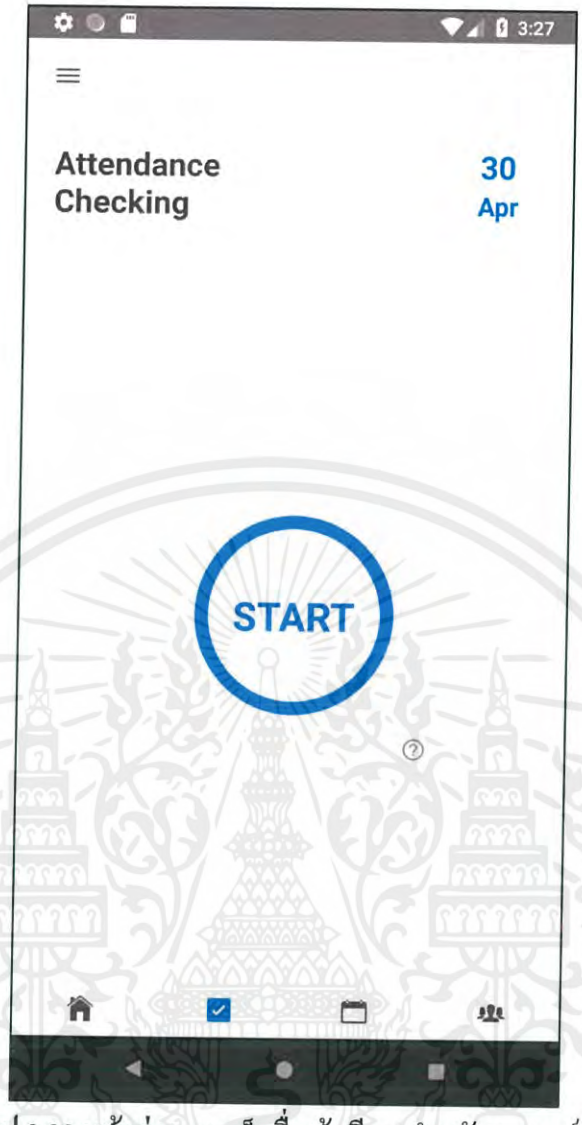


รูป 3.18 หน้าแสดงรายละเอียดการเข้าเรียนในแต่ละวันเป็นแบบกราฟ สำหรับอาจารย์ 2



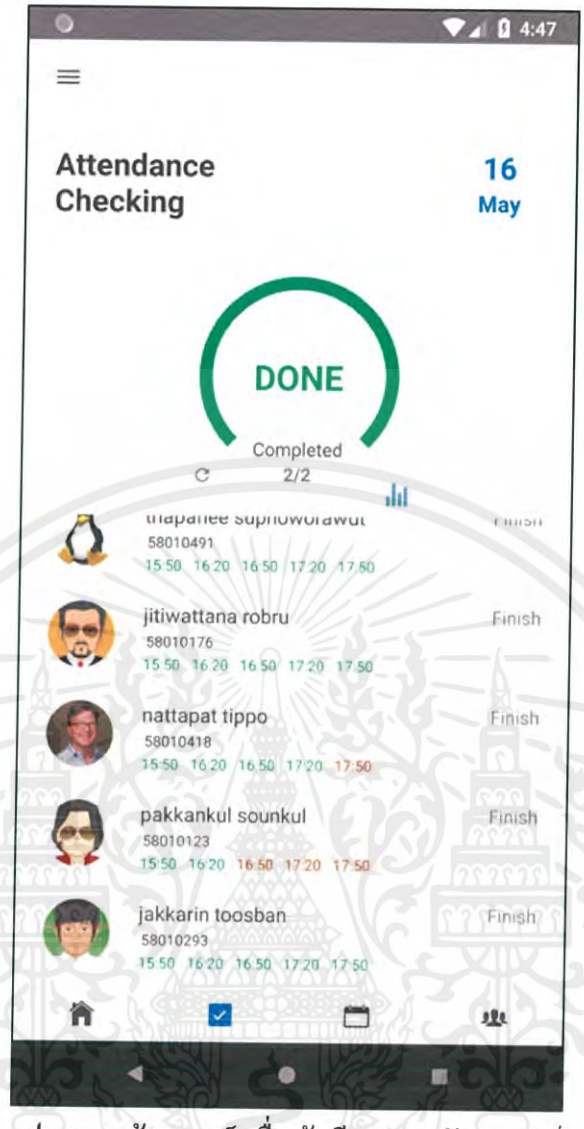
รูป 3.19 หน้าแสดงรายละเอียดการเข้าเรียนในแต่ละวัน สำหรับอาจารย์

เมื่อเข้าหน้าแสดงสรุปการเข้าเรียน หากกดที่วันนั้น ๆ จะพามาหน้าแสดงรายละเอียดการเข้าเรียนที่มากขึ้น โดยจะแสดงถึงชื่อและรหัสนักศึกษาด้วย และมีแถบสำหรับการกรองว่าจะแสดงเฉพาะนักศึกษาที่มาเข้าเรียนหรือเฉพาะนักศึกษาที่ไม่มาเข้าเรียนหรือแสดงทั้งสอง



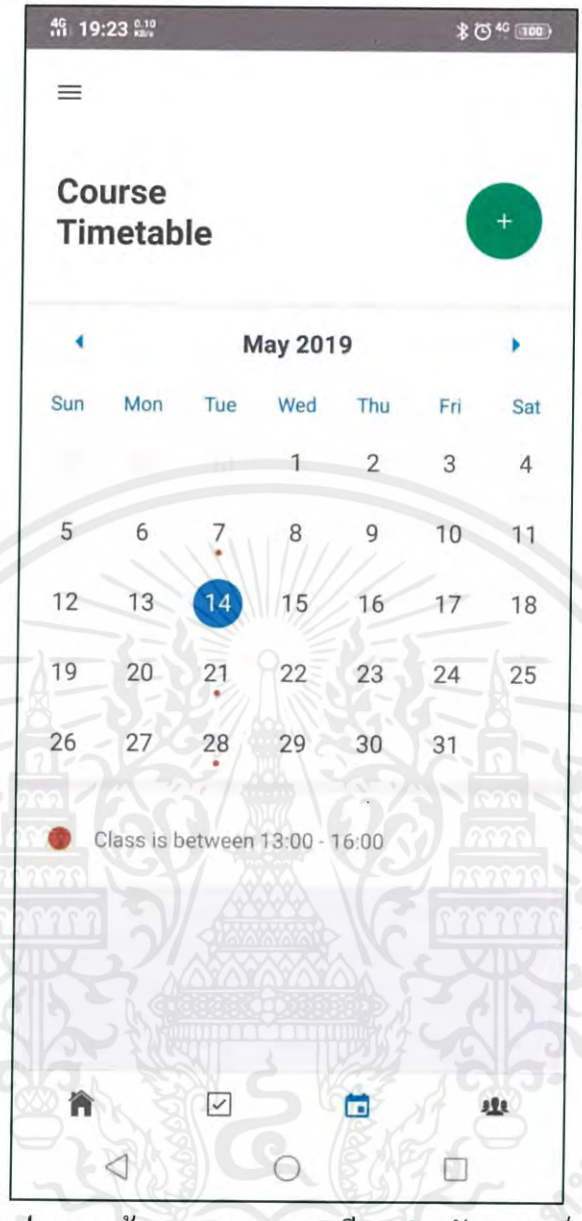
รูป 3.20 หน้าก่อนการเช็คชื่อเข้าเรียน สำหรับอาจารย์

หลังจากเลือกวิชาเรียนแล้ว เมื่อกดที่ปุ่มเช็คชื่อด้านล่างจะพามาหน้านี้ โดยจะเป็นหน้าสำหรับการเริ่มการเช็คชื่อเข้าเรียน โดยที่มุมขวาล่างจะสามารถกดได้เพื่อดูรายละเอียดหรือข้อมูลอื่นเพิ่มเติม และที่ปุ่มเช็คชื่อจะแสดงเวลาเริ่มคลาสเรียนนั้น ๆ ด้วย



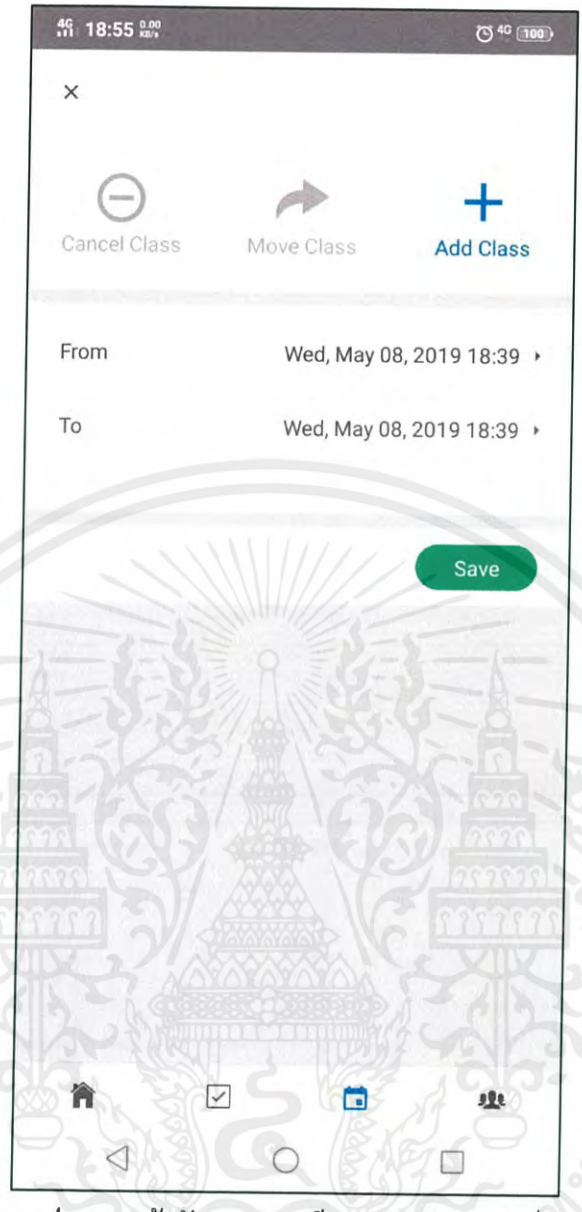
รูป 3.21 หน้าขณะเช็คชื่อเข้าเรียน สำหรับอาจารย์

หลังจากกดปุ่มเริ่มการเช็คชื่อ จะพามาหน้านี้ ซึ่งเป็นหน้าขณะเช็คชื่อเข้าเรียน จะแสดงผลโดยบอกเป็นจำนวนเปอร์เซ็นต์ และสถานะการเช็คชื่อในแต่ละรอบของนักศึกษาแต่ละคน



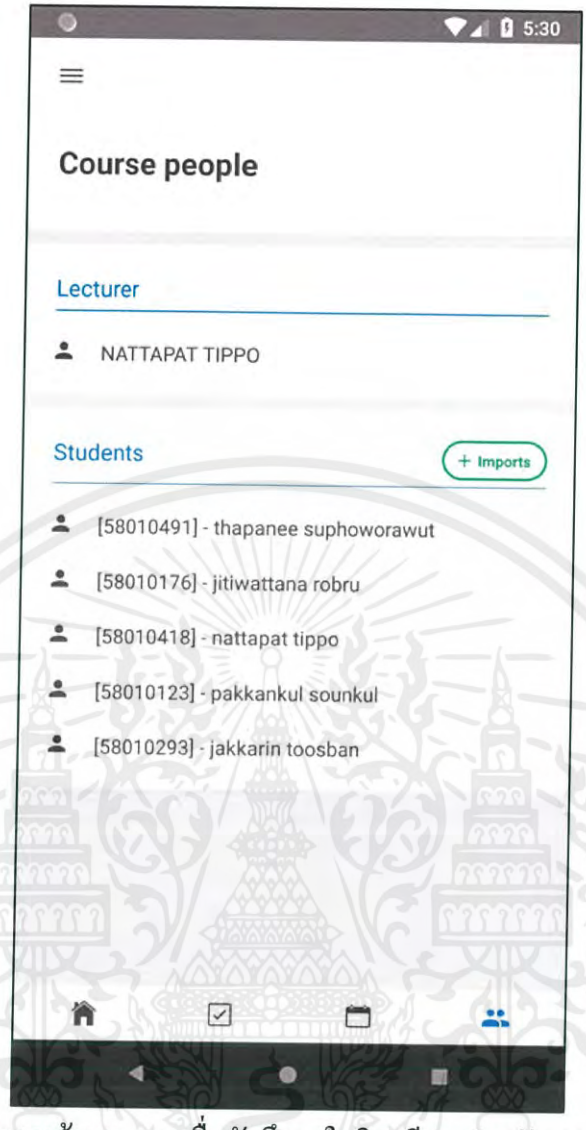
รูป 3.22 หน้าแสดงตารางเวลาเรียน สำหรับอาจารย์

หน้าแสดงตารางเวลาเรียน จะบอกว่าวันไหนมีเรียนบ้าง โดยปุ่มบวกขวาบน สามารถใช้เพื่อเพิ่มเวลาเรียนได้ สำหรับจุดสีแดงหมายถึงว่าในวันนั้นมีการเรียนการสอน



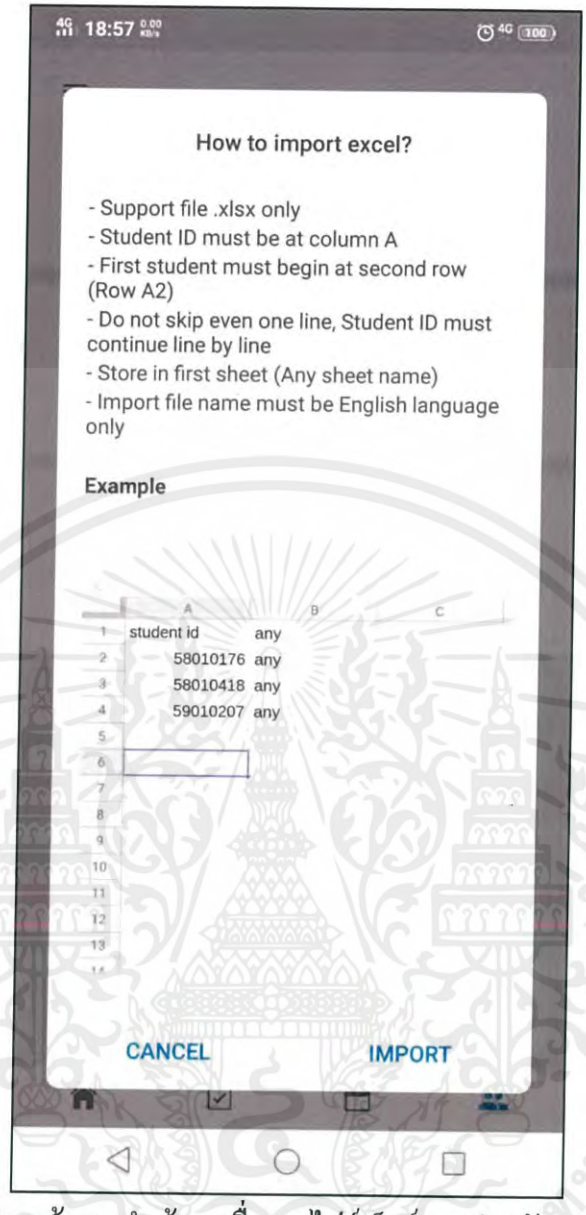
รูป 3.23 หน้าจัดการเวลาเรียน สำหรับอาจารย์

หน้านี้ใช้สำหรับจัดการเวลาเรียนของวิชานั้น ๆ โดยสามารถเพิ่ม ลบ หรือย้ายเวลาเรียนไปยังวันเวลาอื่น ๆ ได้



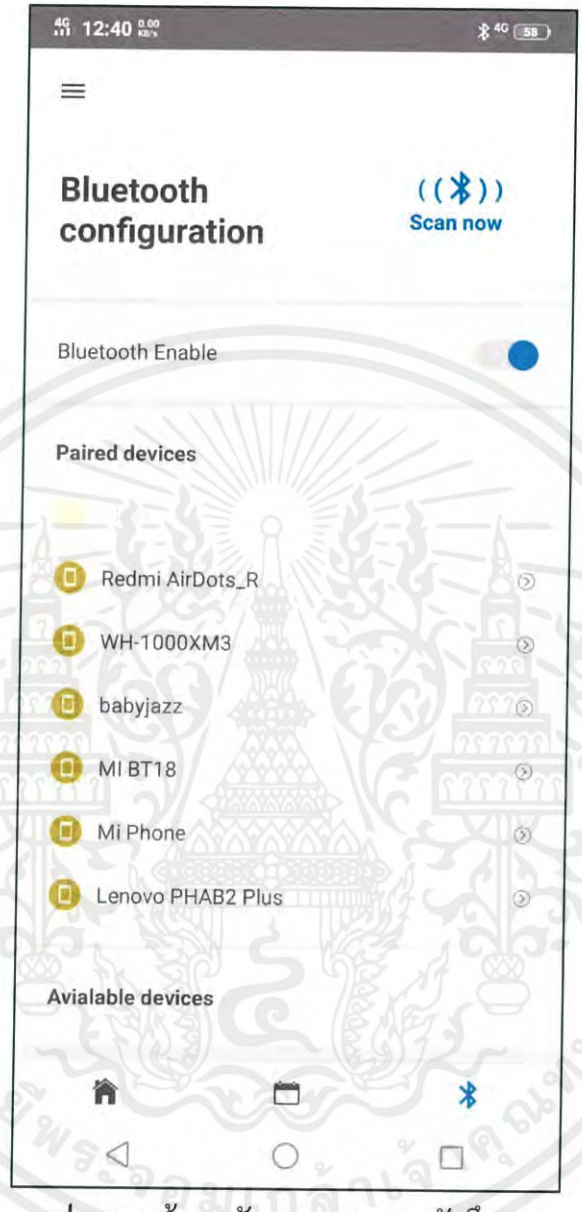
รูป 3.24 หน้าแสดงรายชื่อนักศึกษาในวิชาเรียน สำหรับอาจารย์

หน้านี้ที่ด้านบนจะแสดงชื่ออาจารย์ผู้สอน ส่วนด้านล่างจะแสดงรายชื่อของนักศึกษาที่ได้มีการลงเรียนในวิชาแล้ว



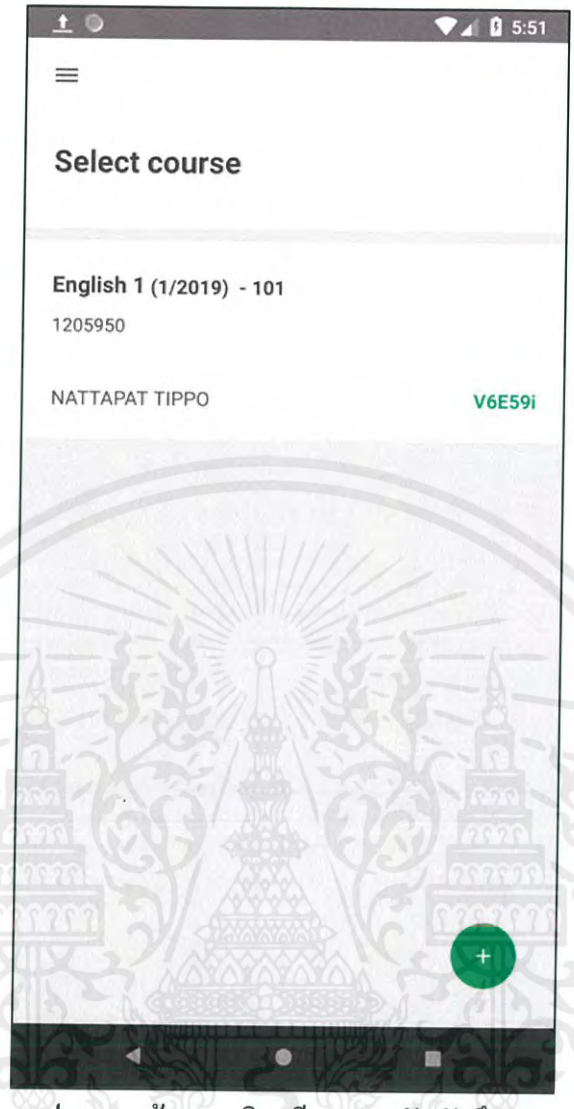
รูป 3.25 หน้าการนำเข้ารายชื่อจากไฟล์เอ็กซ์เซล สำหรับอาจารย์

3.6.3 ส่วนของนักศึกษา



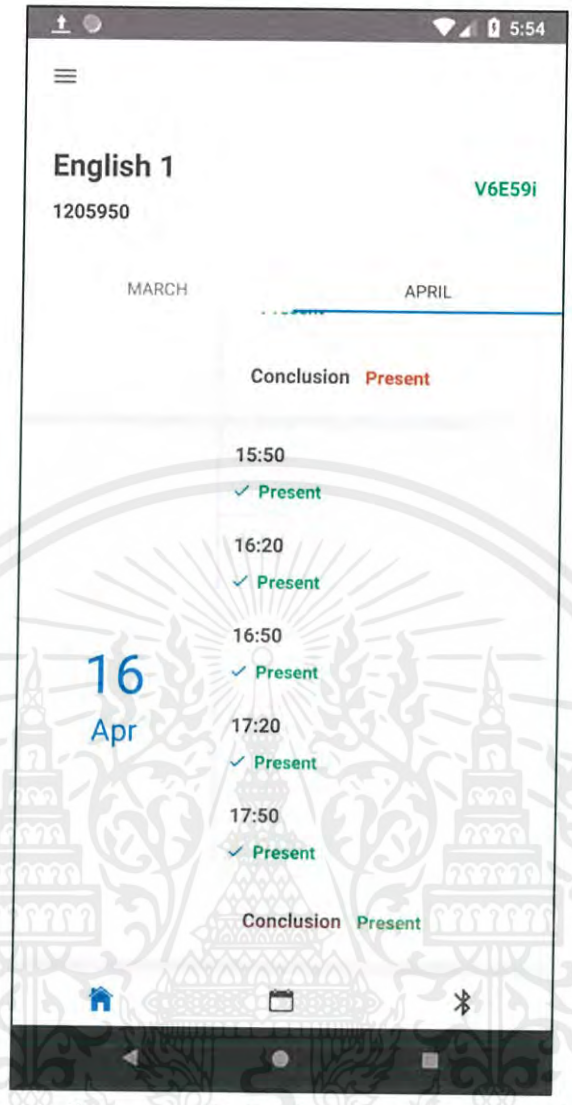
รูป 3.26 หน้าการจัดการบลูทูธของนักศึกษา

หน้าจัดการบลูทูธ ใช้สำหรับการจับคู่หรือตรวจสอบการจับคู่ของนักศึกษาและอาจารย์ โดยสามารถเข้าหน้านี้ได้ด้วยการกดปุ่มสัญลักษณ์บลูทูธที่มุมขวาล่าง



รูป 3.27 หน้าแสดงวิชาเรียน สำหรับนักศึกษา

เมื่อเข้าสู่ระบบได้สำเร็จ แอปพลิเคชันจะพามาหน้านี้โดยอัตโนมัติซึ่งเป็นหน้าแรก โดยจะแสดงรายการวิชาเรียนทั้งหมดที่ได้ลงเรียน และหากต้องการลงวิชาเรียนเพิ่มก็สามารถกดที่ปุ่มบวกได้



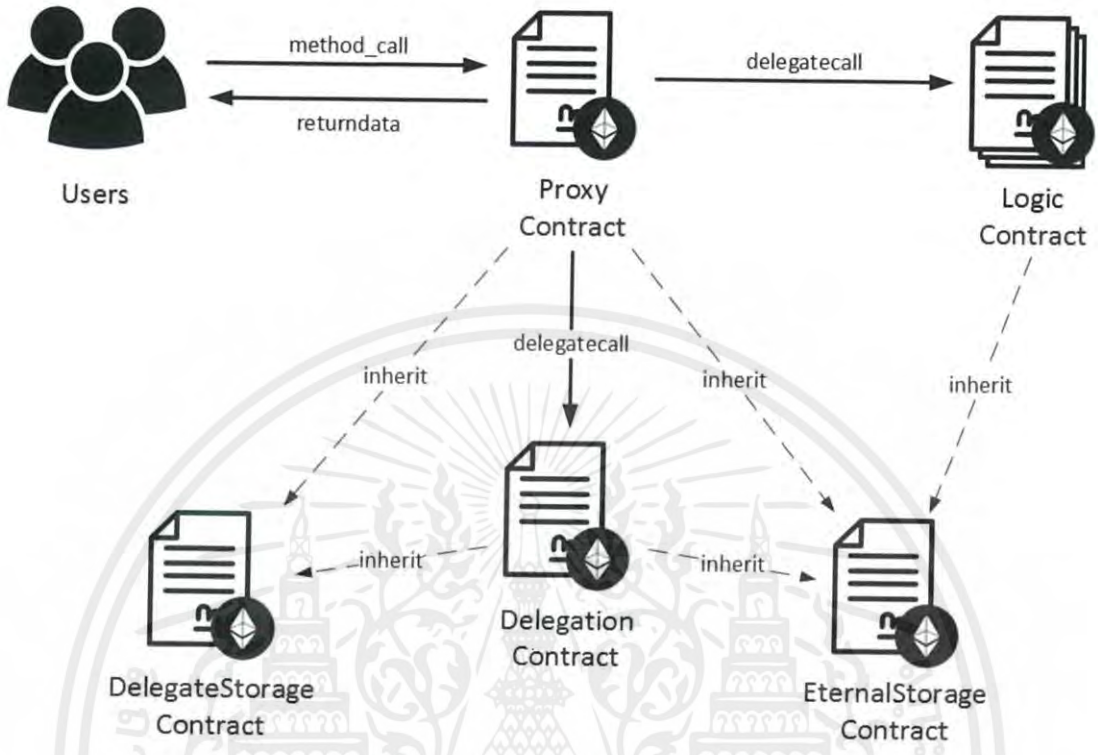
รูป 3.28 หน้าแสดงการเข้าเรียนในแต่ละวัน สำหรับนักศึกษา

หลังจากที่นักศึกษาได้ทำการเข้าสู่ระบบสำเร็จแล้ว และเลือกวิชาที่แสดงไว้ในหน้าแรก เมื่อเข้ามาในวิชาที่เลือก จะแสดงสรุปรายการการเข้าเรียน ในแต่ละวันของตนเองว่า ในแต่ละช่วงเวลาของวิชานั้น ๆ ได้เข้าเรียนหรือไม่ โดยสีเขียวแสดงถึงมา และสีแดงแสดงถึงการไม่มาเข้าเรียน

รูป 3.29 หน้าแสดงเวลาเรียน สำหรับนักศึกษา

หน้าแสดงตารางเวลาเรียน จะบอกว่าวันไหนมีเรียนบ้าง สำหรับจุดสีแดงหมายถึงว่าในวันนั้นมีการเรียนการสอน

3.7 Smart Contract



รูป 3.30 ภาพรวมสมาร์ตคอนแทร็กต์

ในขณะพัฒนาสมาร์ตคอนแทร็กต์นั้นค่อนข้างมีข้อจำกัดหลายอย่าง โดยเฉพาะเรื่องของคอนแทร็กต์ไซซ์ จึงได้พัฒนาสมาร์ตคอนแทร็กต์โดยใช้โครงสร้างที่เรียกว่า "พร็อกซี" ซึ่งจะมีการทำงานในลักษณะคล้ายกับพร็อกซี คือ จะมีคอนแทร็กต์หลักที่ใช้สำหรับการติดต่อและคอนแทร็กต์นี้จะไปเรียกใช้งานคอนแทร็กต์ที่รับผิดชอบของงานส่วนนั้น ๆ ไป จากรูป 3.30 จะแบ่งสมาร์ตคอนแทร็กต์หลัก ๆ ออกได้ 5 ส่วน ดังนี้

3.7.1 EternalStorage Contract

คอนแทร็กต์นี้จะทำหน้าที่เปรียบเสมือนการเป็นโครงสร้างข้อมูลให้กับคอนแทร็กต์ที่สืบทอด โดยถึงแม้จะเป็นคอนแทร็กต์คนละตัวกันแต่จะมีโครงสร้างข้อมูลเดียวกัน กล่าวคือที่อยู่ของข้อมูลอยู่ตำแหน่งเดียวกัน ทำให้ตัวคอนแทร็กต์ส่วนที่เป็นลอจิกต่าง ๆ สามารถเรียกดูหรือเปลี่ยนแปลงข้อมูลเดียวกันจากการทำดีลิเกทคอลของคอนแทร็กต์พร็อกซีได้

3.7.2 DelegateStorage Contract

คอนแทร็กต์นี้จะทำหน้าที่สำหรับการเก็บข้อมูลเพื่อใช้สำหรับการทำดีลิเกทคอลของคอนแทร็กต์พร็อกซี โดยจะเก็บแอดเดรสรวมทั้งฟังก์ชันซิกเนเจอร์ของแต่ละฟังก์ชันไว้

3.7.3 Delegation Contract

คอนแทร็กต์นี้จะทำหน้าที่สำหรับการเพิ่ม ลบ และเปลี่ยนแปลงข้อมูลที่จะใช้สำหรับการทำดีลิเกทคอลของคอนแทร็กต์พร็อกซี โดยมีการสืบทอดจากคอนแทร็กต์ดีลิเกทสตอเรจเพื่อต้องการใช้โครงสร้างข้อมูลเดียวกัน

3.7.4 Proxy Contract

คอนแทร็กต์นี้เป็นตัวสำคัญ โดยจะเป็นคอนแทร็กต์ที่โมไบล์แอปพลิเคชันหรือแอปพลิเคชันเซิร์ฟเวอร์ปฏิสัมพันธ์ด้วย ซึ่งเมื่อมีการเรียกใช้งาน คอนแทร็กต์พร็อกซีจะทำการดีลิเกทคอลไปยังคอนแทร็กต์ส่วนที่เป็นลอจิกต่าง ๆ โดยตรวจสอบจากฟังก์ชันซิกเนเจอร์และดึงแอดเดรสจากข้อมูลที่เก็บไว้ตามโครงสร้างของคอนแทร็กต์ดีลิเกทสตอเรจ

3.7.5 Logic Contract

คอนแทร็กต์เหล่านี้เป็นส่วนลอจิกของระบบ ซึ่งแบ่งได้หลายคอนแทร็กต์ขึ้นอยู่กับงานที่เกี่ยวข้อง เช่น การจัดการผู้ใช้ ทั้งการเพิ่มสมาชิก การลบสมาชิก การแก้ไขข้อมูลต่าง ๆ การจัดการวิชาเรียน ทั้งการเปิดวิชาเรียน การปิดวิชา การเพิ่มกลุ่มเรียน การลงทะเบียนเรียนของนักศึกษา รวมถึงการจัดการการเข้าเรียนของนักศึกษา ทั้งการบันทึกเวลาที่นักศึกษามาเข้าเรียน การลบหรือล้างข้อมูล และเวลาการเริ่มเช็คชื่อเข้าเรียนของอาจารย์ เป็นต้น

บทที่ 4

การทดลองและผลการทดลอง

บทนี้จะกล่าวถึงการทดลองการทำงานส่วนต่าง ๆ ของระบบ รวมทั้งการทำงานร่วมกัน ทั้งส่วนของโมไบล์แอปพลิเคชัน (Mobile Application) แอปพลิเคชันเซิร์ฟเวอร์ (Application Server) และบล็อกเชน (Blockchain) โดยมีการทดลองดังนี้

1) การสร้างระบบบล็อกเชนส่วนตัว
2) การ deploy สมาร์ตคอนแทร็กต์บนระบบบล็อกเชนส่วนตัว
3) การติดต่อระบบบล็อกเชนส่วนตัวด้วยเว็บทรี
4) การเชื่อมต่อบลูทูธด้วยโมไบล์แอปพลิเคชัน
5) การทำงานเบื้องหลังของโมไบล์แอปพลิเคชัน
6) การสมัครสมาชิกและการเข้าสู่ระบบเพื่อใช้งาน
7) การสร้างและเข้าร่วมวิชาเรียน
8) การเช็คชื่อเข้าเรียน
9) การส่งออกรายงาน

4.1 การสร้างระบบบล็อกเชนส่วนตัว

4.1.1 วัตถุประสงค์

เพื่อสร้างระบบบล็อกเชนส่วนตัวและเชื่อมต่อโหนดทุกโหนดเข้าด้วยกัน

4.1.2 วิธีการทดลอง

1) สร้างไฟล์ genesis.json เพื่อใช้สำหรับการเริ่ม genesis block ด้วย puppeth

```
{
  "config": {
    "chainId": 12311920,
    "homesteadBlock": 1,
    "eip150Block": 2,
    "eip150Hash": "0x0000000000000000000000000000000000000000000000000000000000000000",
    "eip155Block": 3,
    "eip158Block": 3,
    "byzantiumBlock": 4,
    "constantinopleBlock": 5,
    "clique": {
      "period": 0,
      "epoch": 30000
    }
  },
  "nonce": "0x0",
  "timestamp": "0x5c8589c7",
  "extraData":
  "0x0000000000000000000000000000000000000000000000000000000000000000327342d5c0de6db5320353a593e0956
  e958056272000d0a82ef59697acd04bcebc3029238a24fb1899bc7c145c3d4cffef286bcaea55ca383b8fd7000000000000
  0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
  0000000000000000",
  "gasLimit": "0x1312D00",
  "difficulty": "0x1",
  "mixHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "coinbase": "0x0000000000000000000000000000000000000000",
  "alloc": {
    "0327342d5c0de6db5320353a593e0956e9580562": {
      "balance": "0x200000000000000000000000000000000000000000000000000000000000000"
    },
    "2db39bd5977ad4c634094a8068eb9177260ed0d3": {
      "balance": "0x200000000000000000000000000000000000000000000000000000000000000"
    },
    "72000d0a82ef59697acd04bcebc3029238a24fb1": {
      "balance": "0x200000000000000000000000000000000000000000000000000000000000000"
    },
    "899bc7c145c3d4cffef286bcaea55ca383b8fd70": {
      "balance": "0x200000000000000000000000000000000000000000000000000000000000000"
    },
    "e88e5fbd140a5e874c63b292d5f9152ecec6461b": {
      "balance": "0x200000000000000000000000000000000000000000000000000000000000000"
    }
  },
  "number": "0x0",
  "gasUsed": "0x0",
  "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000"
}
```


รูป 4.1 ไฟล์ genesis.json

2) เริ่มต้นบล็อกเชนจากไฟล์ genesis.json ด้วย geth กับทุกโหนด

```
~ [~/Downloads] geth init genesis.json
INFO [04-28|23:20:12.802] Maximum peer count                       ETH=25 LES=0 total=25
INFO [04-28|23:20:12.803] Allocated cache and file handles         database=/home/zenerdeor/.ethereum/geth/chaindata
INFO [04-28|23:20:12.818] Writing custom genesis block
INFO [04-28|23:20:12.818] Persisted trie from memory database      nodes=7 size=1.15kB time=72.594µs gcnodes=0 gcsize
INFO [04-28|23:20:12.819] Successfully wrote genesis state         database=chaindata
INFO [04-28|23:20:12.819] Allocated cache and file handles         database=/home/zenerdeor/.ethereum/geth/lightchain
INFO [04-28|23:20:12.849] Writing custom genesis block
INFO [04-28|23:20:12.850] Persisted trie from memory database      nodes=7 size=1.15kB time=64.74µs  gcnodes=0 gcsize
INFO [04-28|23:20:12.850] Successfully wrote genesis state         database=lightchaindata
```

รูป 4.2 การเริ่มต้นระบบบล็อกเชน

3) รัน bootnode สำหรับให้โหนดแต่ละโหนดสามารถเชื่อมต่อกันได้ ด้วยคำสั่ง "bootnode -nodekey boot.key -addr :30301"

4) รันโหนดสำหรับ Validator ด้วย geth

```
geth \
    --networkid 12311920 \
    --identity sealer_000 \
    --syncmode full \
    --port 30303 \
    --maxpeers 50 \
    --ethstats sealer_000:secret@ethstats:3000 \
    --unlock 899bc7c145c3d4cffef286bcaea55ca383b8fd70 \
    --password /signer.pass \
    --etherbase 899bc7c145c3d4cffef286bcaea55ca383b8fd70 \
    --minerthreads 1 \
    --mine  \
    --targetgaslimit 20000000 \
    --nodiscover \
    --gcmode archive \
    --gasprice 1000000000
```

รูป 4.3 คำสั่งสำหรับรัน Validator node

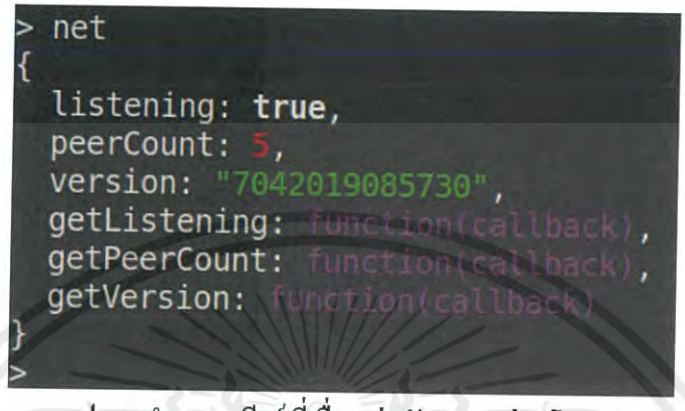
5) รันโหนดสำหรับ RPC ด้วย geth

```
geth \
    --networkid 12311920 \
    --identity rpc_000 \
    --syncmode full \
    --port 30303 \
    --rpc \
    --rpcapi personal,net,eth,web3 \
    --rpcaddr 0.0.0.0 \
    --rpcport 8545 \
    --rpccorsdomain '*' \
    --maxpeers 50 \
    --ethstats rpc_000:secret@ethstats:3000 \
    --rpcvhosts='*' \
    --nodiscover \
    --gcmode archive \
    console
```

รูป 4.4 คำสั่งสำหรับรัน RPC Node

4.1.3 ผลการทดลอง

จากผลลัพธ์ของการใช้คำสั่ง net กับโหนดแต่ละโหนดเพื่อแสดงจำนวนเพียร์ทั้งหมดที่เชื่อมต่ออยู่ จึงสามารถสรุปได้ว่าโหนดทุกโหนดสามารถทำงานและเชื่อมต่อกันได้



รูป 4.5 จำนวนเพียร์ที่เชื่อมต่อกันของแต่ละโหนด

4.2 การ deploy สมาร์ตคอนแทร็กต์บนระบบบล็อกเชนส่วนตัว

4.2.1 วัตถุประสงค์

เพื่อ deploy สมาร์ตคอนแทร็กต์ที่พัฒนาขึ้นไปบนระบบบล็อกเชนส่วนตัวและสามารถเรียกใช้งานได้

4.2.2 วิธีการทดลอง

1) เขียนโค้ด Smart Contract สำหรับระบบเช็คชื่อเข้าเรียน

2) คอมไพล์โค้ดที่พัฒนาบน truffle ด้วยคำสั่ง truffle compile

```
Compiling ./contracts/Migrations.sol...
Compiling ./contracts/Proxy/AdvanceProxy.sol...
Compiling ./contracts/Proxy/DelegateStorage.sol...
Compiling ./contracts/Proxy/Delegation.sol...
Compiling ./contracts/Proxy/OwnerStorage.sol...
Compiling ./contracts/Proxy/QueryDelegates.sol...
Compiling ./contracts/Schedule/ScheduleAdder.sol...
Compiling ./contracts/Schedule/ScheduleDeleter.sol...
Compiling ./contracts/Schedule/ScheduleSetter.sol...
Compiling ./contracts/User/UserAdder.sol...
Compiling ./contracts/User/UserGetter.sol...
Compiling ./contracts/User/UserSetter.sol...
Compiling ./contracts/User/UserSetter2.sol...
Compiling ./contracts/User/UserUtils.sol...
Compiling ./contracts/Validate/Regex.sol...
Compiling ./contracts/Validate/TestRegex.sol...
Writing artifacts to ./build/contracts
```

รูป 4.6 ผลลัพธ์การคอมไพล์สมาร์ตคอนแทร็กต์

3) ทำการตั้งค่า truffle ให้เชื่อมต่อไป RPC โหนด

4) Deploy Smart Contract ด้วย truffle

```
Summary
=======
> Total deployments:   32
> Final cost:          100000001.58638406 ETH
```

รูป 4.7 ผลลัพธ์ของการ deploy สมาร์ตคอนแทร็กต์

5) เรียกใช้งาน Smart Contract

4.2.3 ผลการทดลอง

สามารถ deploy smart contract ขึ้นได้สำเร็จและสามารถเรียกใช้งาน smart contract ได้

```
truffle(poa)> let con = await AdvanceProxy.deployed()
undefined
truffle(poa)> con.getAllMapCodes.call()
[ '0x6d5559766b4a000000000000000000000000000000000000000000000000000000',
  '0x36665a4451310000000000000000000000000000000000000000000000000000',
  '0x4b697779515200000000000000000000000000000000000000000000000000' ]
truffle(poa)>
```

รูป 4.8 ผลการทดลองเรียกใช้สมาร์ตคอนแทร็กต์

4.3 การติดต่อระบบบล็อกเชนส่วนตัวด้วยเว็บทรี

4.3.1 วัตถุประสงค์

เพื่อให้สามารถติดต่อกับเครือข่ายบล็อกเชนผ่าน RPC โหนดด้วยไลบรารี web3.js ได้

4.3.2 วิธีการทดลอง

1) ที่แอปพลิเคชันเซิร์ฟเวอร์ สร้างโปรเจคด้วยคำสั่ง “npm init ABCAS”

2) ติดตั้งไลบรารีเว็บทรีและเอ็กเพรสลงที่แอปพลิเคชันเซิร์ฟเวอร์

3) เขียนโค้ดการเชื่อมต่อระบบบล็อกเชนจากแอปพลิเคชันเซิร์ฟเวอร์

```
const web3 = new Web3('http://161.246.34.37:8080')
export const coinbase = '0x831Cca0bCE7a18c62DBE7DdE7cdcE4395af874AD'
export const coinbasePWD = 'root'
export const contractUser = new web3.eth.Contract(userABI, '0xe1F8f0427F22aD93f27aE3A378740D844ce339B0')
```

รูป 4.9 โค้ดการเชื่อมต่อบล็อกเชนของแอปพลิเคชันเซิร์ฟเวอร์

4) ที่โมไบล์แอปพลิเคชัน สร้างโปรเจคด้วยแพ็กเกจรีนาเทล (re-natal)

5) ติดตั้งไลบรารีเว็บทรีที่โมไบล์แอปพลิเคชัน

6) เขียนโค้ดส่วนการเชื่อมต่อระบบบล็อกเชนจากโมไบล์แอปพลิเคชัน

```
(def web-url "http://rpc.z3n.pw")
(defonce web3 (new c/Web3 web-url))
(def contract-user (atom (new web3.eth.Contract
                              c/abi-classfactory
                              "0x5Bd6018962Dc69e4B76b34A9b64a2E58aD9dFE71")))
```

รูป 4.10 โค้ดการเชื่อมต่อบล็อกเชนของโมไบล์แอปพลิเคชัน

7) รันโค้ดเพื่อทดสอบการเชื่อมต่อบล็อกเชน

4.3.3 ผลการทดลอง

ทั้งแอปพลิเคชันเซิร์ฟเวอร์และโมไบล์แอปพลิเคชันสามารถติดต่อกับเครือข่ายบล็อกเชนด้วย web3.js ได้

```
web3.eth.net.isListening().then(() => {
  console.log('connected')
}).catch(() => {
  console.log('disconnected')
})
```

รูป 4.11 โค้ดทดสอบการเชื่อมต่อบล็อกเชนของแอปพลิเคชันเซิร์ฟเวอร์

```
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
ABCast listening on port 8000!
connected
```

รูป 4.12 ผลการทดลองเชื่อมต่อบล็อกเชนจากแอปพลิเคชันเซิร์ฟเวอร์

```
(-> web3 .-eth .-net
      (.isListening)
      (.then #(prn "Connected"))
      (.catch #(prn "Disconnected")))
#object[Promise [object Promise]]
android:abcast.utils.web3=> "Connected"
```

รูป 4.13 ผลการทดลองเชื่อมต่อบล็อกเชนจากโมไบล์แอปพลิเคชัน

4.4 การเชื่อมต่อบลูทูธด้วยโมไบล์แอปพลิเคชัน

4.4.1 วัตถุประสงค์

เพื่อทดลองเชื่อมต่อบลูทูธระหว่างสมาร์ตโฟนด้วยโมไบล์แอปพลิเคชัน

4.4.2 วิธีการทดลอง

1) ที่โปรเจคโมไบล์แอปพลิเคชัน ติดตั้งรีแอคเนทีฟคลาสสิกบลูทูธ (react-native-classic-bluetooth)

2) เขียนโค้ดส่วนการตั้งค่าเริ่มต้นคุณสมบัติของบลูทูธ

```
(def bluetooth-config #js {:uuid "0000110a-0000-1000-8000-00805f9b34fb"
                           :deviceName "babyjazz"
                           :bufferSize 1024
                           :characterDelimiter "\n"})

(defn init-bluetooth-config []
   (-> (.init c/bluetooth-classic bluetooth-config)
      (.then (fn [config]
               (prn "Bluetooth successfully init configuration")
               (prn config)))
      (.catch (fn [err]
                (prn "Bluetooth failed init configuration")
                (prn err)))))
```

รูป 4.14 โค้ดการตั้งค่าเริ่มต้นคุณสมบัติของบลูทูธ

3) เขียนโค้ดฟังก์ชันค้นหาและเชื่อมต่อบลูทูธ

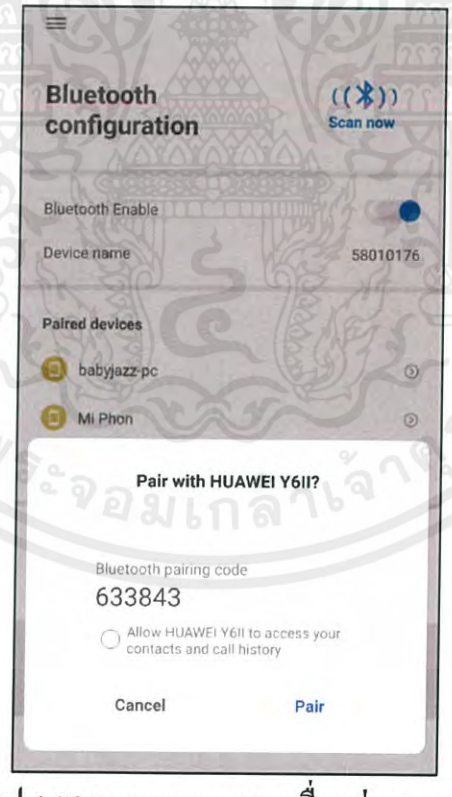
```
(defn bluetooth-connect [device]
    (-> (.connect c/bluetooth-classic device)
        (.then (fn [] (js/alert "Connected!")
                      (prn "Connected"))))

(defn scan-bluetooth []
    (-> (.startScan c/bluetooth-classic)
        (.then (fn [devices]
                   (prn "Scaning is done\n")
                   (prn "Devices found: ")
                   (prn devices)))
        (.catch (fn [err]
                   (prn "Scaning is error")
                   (prn err)))))
```

รูป 4.15 โค้ดการค้นหาและเชื่อมต่อบลูทูธ

4.4.3 ผลการทดลอง

สามารถเชื่อมต่อบลูทูธคลาสสิกระหว่างสองอุปกรณ์ได้สำเร็จ



รูป 4.16 ผลการทดลองการเชื่อมต่อบลูทูธ

4.5 การทำงานเบื้องหลังของโมไบล์แอปพลิเคชัน

4.5.1 วัตถุประสงค์

เพื่อให้โมไบล์แอปพลิเคชันยังคงสามารถทำงานได้อย่างต่อเนื่องแม้มีการซ่อนแอปพลิเคชันหรือปิดหน้าจอสมาร์ตโฟนไปแล้ว

4.5.2 วิธีการทดลอง

1) เขียนโค้ดส่วนการทำงานเบื้องหลัง โดยให้แสดงตัวเลขทุก 1 วินาที

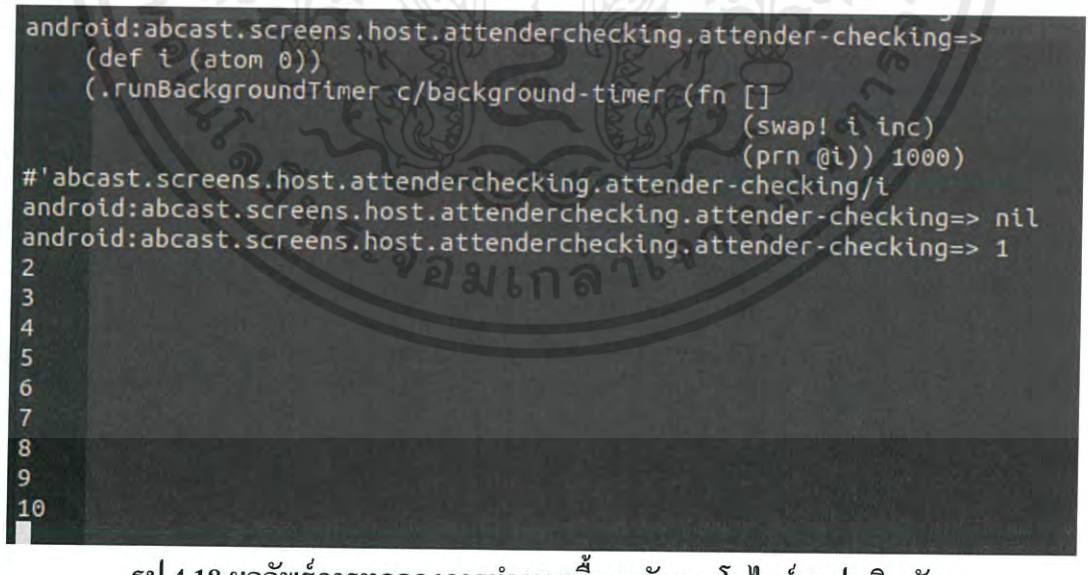
```
(def i (atom 0))
(.runBackgroundTimer c/background-timer (fn []
                                          (swap! i inc)
                                          (prn @i)) 1000)
```

รูป 4.17 โค้ดการทำงานเบื้องหลังของโมไบล์แอปพลิเคชัน

2) ทดสอบส่วนการทำงานเบื้องหลัง

4.5.3 ผลการทดลอง

โมไบล์แอปพลิเคชันยังคงสามารถทำงานได้อย่างต่อเนื่อง แม้มีการซ่อนแอปพลิเคชัน ปิดหน้าจอสมาร์ตโฟน หรือใช้งานแอปพลิเคชันอื่น ๆ อยู่



รูป 4.18 ผลลัพธ์การทดลองการทำงานเบื้องหลังของโมไบล์แอปพลิเคชัน

4.6 การสมัครสมาชิกและการเข้าสู่ระบบเพื่อใช้งาน

4.6.1 วัตถุประสงค์

เพื่อทดลองสมัครสมาชิกและเข้าสู่ระบบเพื่อให้สามารถใช้งานส่วนต่าง ๆ ของระบบต่อไปได้

4.6.2 วิธีการทดลอง

1) เข้าโมไบล์แอปพลิเคชัน

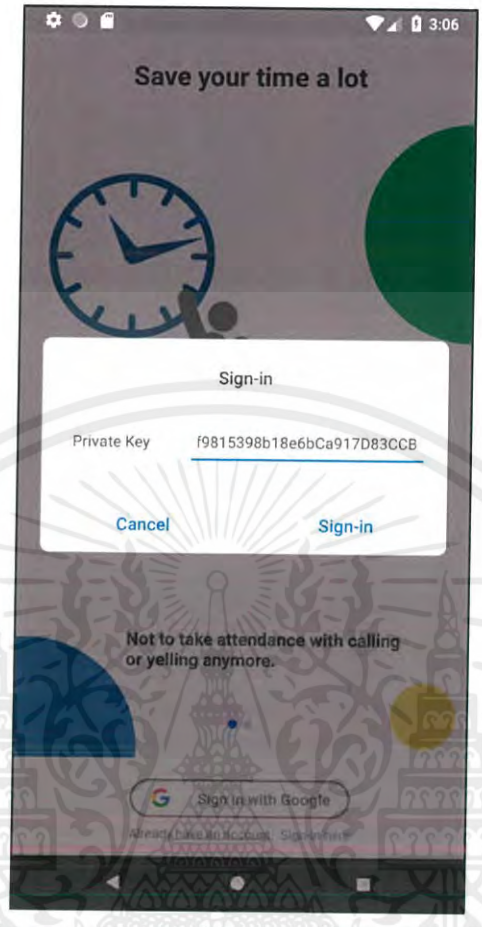
2) กด Sign in with Google

3) กรอกอีเมล์และรหัสผ่านของ Google ด้วยอีเมล์สถาบัน (@kmitl.ac.th) เพื่อสมัครสมาชิก

รูป 4.19 ผลลัพธ์หลังการสมัครสมาชิก

4) หลังเข้าสู่ระบบให้ Export Private key เพื่อใช้สำหรับการเข้าสู่ระบบครั้งต่อไป

5) ออกจากระบบและเข้าสู่ระบบใหม่อีกครั้งด้วย Private key ที่ได้จากขั้นตอนที่ 4



รูป 4.20 การทดลองเข้าสู่ระบบด้วย Private key

4.6.3 ผลการทดลอง

สามารถทำการสมัครสมาชิกและเข้าสู่ระบบได้สำเร็จ โดยหลังจากสมัครสมาชิกใหม่จะยังไม่มีวิชาใดอยู่ ๆ และหากทดลองเข้าสู่ระบบใหม่อีกครั้งด้วย Private key ก็จะให้ผลลัพธ์เหมือนเดิม คือหน้าแรกยังไม่มีวิชาอะไร

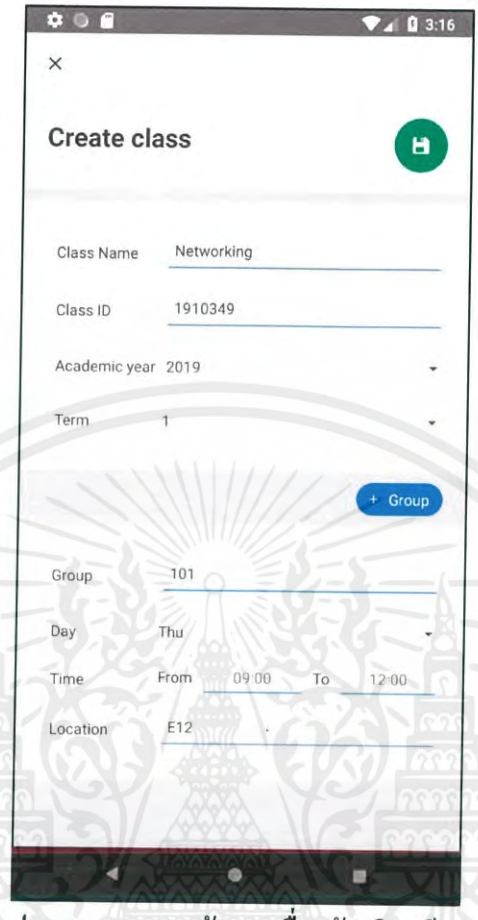
4.7 การสร้างและเข้าร่วมวิชาเรียน

4.7.1 วัตถุประสงค์

เพื่อทดลองสร้างวิชาเรียนและสามารถเข้าร่วมวิชาเรียนที่สร้างได้

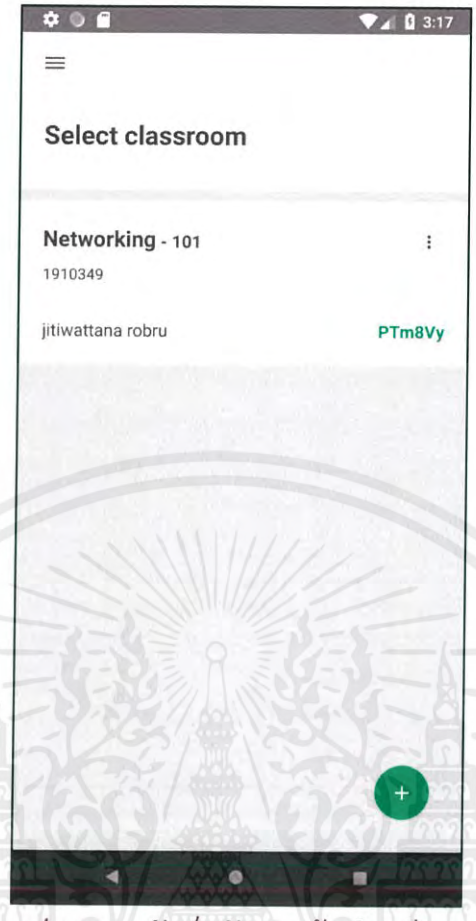
4.7.2 วิธีการทดลอง

1) ที่สมาร์ตโฟนของอาจารย์ เข้าโมไบล์แอปพลิเคชันและทำการเข้าสู่ระบบ
2) ทำการสร้างวิชาเรียนโดยกดที่ปุ่มสร้างวิชาเรียนและกรอกข้อมูลวิชาเรียน



รูป 4.21 การกรอกข้อมูลเพื่อสร้างวิชาเรียน

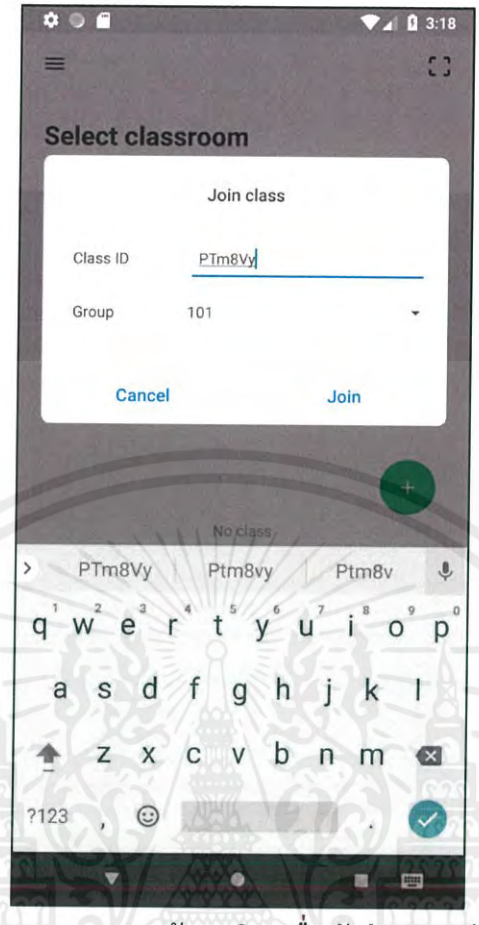
3) กดปุ่มยืนยันเพื่อสร้างวิชาเรียน



รูป 4.22 ผลลัพธ์หลังการสร้างวิชาเรียน

4) ที่สมาร์ตโฟนของนักศึกษา เข้าโมไบล์แอปพลิเคชันและทำการเข้าสู่ระบบ

5) กดปุ่มเข้าร่วมวิชาเรียนและกรอกรหัสที่ได้จากอาจารย์สำหรับการเข้าร่วมวิชาเรียน



รูป 4.23 การกรอกข้อมูลวิชาเพื่อเข้าร่วมวิชาเรียน

6) กดปุ่มยืนยันการเข้าร่วมวิชาเรียน

รูป 4.24 ผลลัพธ์หลังเข้าร่วมวิชาเรียน

4.7.3 ผลการทดลอง

สามารถสร้างวิชาเรียนและเข้าร่วมวิชาเรียนได้สำเร็จ

4.8 การเช็คชื่อเข้าเรียน

4.8.1 วัตถุประสงค์

เพื่อทดลองเช็คชื่อเข้าเรียนด้วยโมไบล์แอปพลิเคชัน

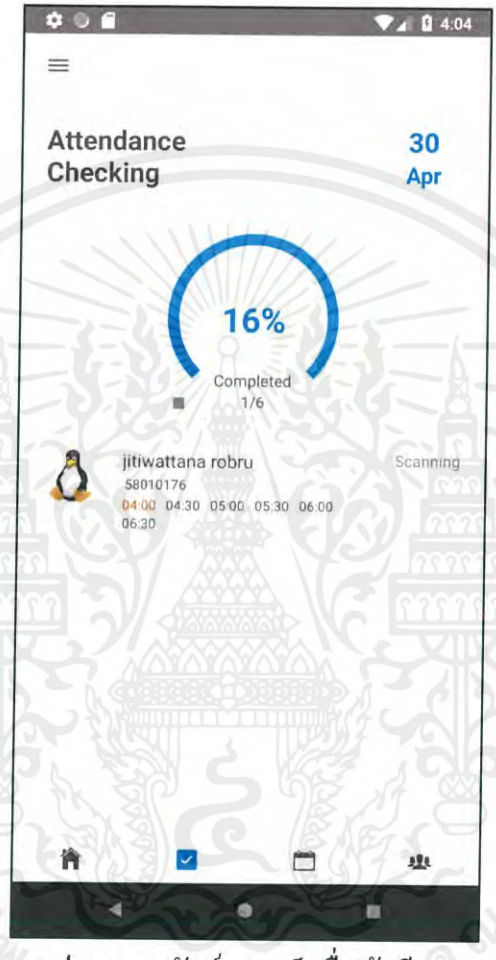
4.8.2 วิธีการทดลอง

1) ที่สมาร์ตโฟนของอาจารย์และนักศึกษา เข้าโมไบล์แอปพลิเคชันและทำการเข้าสู่ระบบ

2) ให้สมาร์ตโฟนของทั้งอาจารย์และนักศึกษาทำการแพริ่งกัน โดยผ่านเมนูของโมไบล์แอปพลิเคชันหรือผ่านเมนูของสมาร์ตโฟนเอง หากยังไม่ได้แพริ่ง

3) ที่สมาร์ตโฟนของอาจารย์กดที่วิชาเรียนที่ต้องการเช็คชื่อนักศึกษา

4) เมื่อถึงเวลาเรียนที่ตั้งไว้ ให้กดปุ่มเริ่มการเช็คชื่อเพื่อเริ่มการเช็คชื่อเข้าเรียน

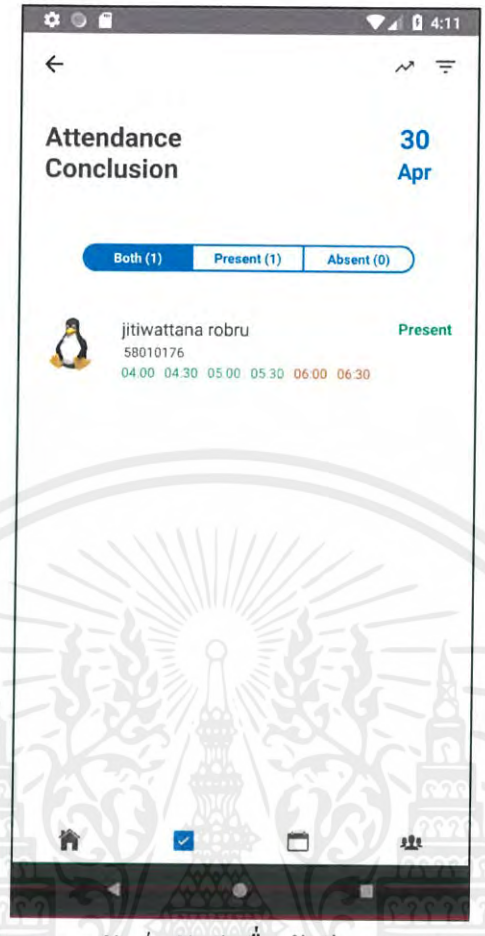


รูป 4.25 ผลลัพธ์ขณะเช็คชื่อเข้าเรียน

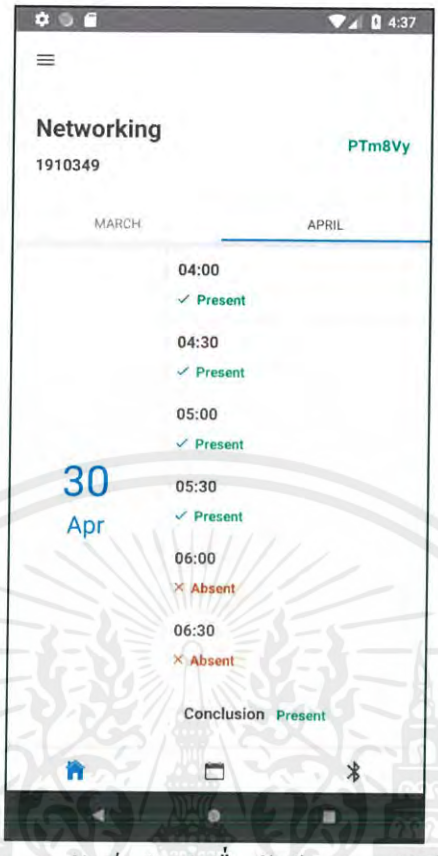
5) เมื่อหมดเวลาเรียน ดูผลสรุปการเข้าเรียนที่โมไบล์แอปพลิเคชัน

4.8.3 ผลการทดลอง

สามารถทำการเช็คชื่อเข้าเรียนและสรุปผลได้สำเร็จ



รูป 4.26 ผลลัพธ์หลังเช็คชื่อเข้าเรียนของอาจารย์



รูป 4.27 ผลลัพธ์หลังเช็คชื่อเข้าเรียนของนักศึกษา

4.9 การส่งออกรายงาน

4.9.1 วัตถุประสงค์

เพื่อทดลองส่งออกรายงานสรุปการเข้าเรียนในรูปแบบไฟล์เอ็กซ์เซล

4.9.2 วิธีการทดลอง

1) ที่สมาร์ตโฟนของอาจารย์ เข้าโมไบล์แอปพลิเคชันและทำการเข้าสู่ระบบ
2) กดที่วิชาเรียนที่ต้องการ
3) กดปุ่มดาวน์โหลดเพื่อดาวน์โหลดไฟล์เอ็กซ์เซลสรุปการเข้าเรียน

4.9.3 ผลการทดลอง

สามารถส่งออกเอกสารสรุปการเข้าเรียนในรูปแบบไฟล์เอ็กซ์เซลได้สำเร็จ

	A	B	C	D	E	F	G
1	studentId	2019-04-01[09:00]	2019-04-08[09:00]	2019-04-15[09:00]	2019-04-22[09:00]	2019-04-29[09:00]	2019-04-30[04:00]
2	58010176	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
3							

รูป 4.28 ตัวอย่างไฟล์รายงานที่ส่งออก

บทที่ 5
สรุปผลการดำเนินงาน

ในแต่ละหัวข้อจะแบ่งออกเป็น 3 ส่วน คือ ส่วนของโมไบล์แอปพลิเคชัน (Mobile Application) ส่วนของแอปพลิเคชันเซิร์ฟเวอร์ (Application Server) และส่วนของบล็อกเชน (Blockchain)

5.1 ผลสรุปของโครงงาน

โครงงานนี้เราได้นำเสนอการพัฒนาการประยุกต์บล็อกเชนสำหรับระบบเช็คชื่อเข้าเรียน เรียกว่า "เอบีคาส" แนวคิดหลักของเอบีคาสคือการใช้การติดต่อสื่อสารระหว่างสมาร์ตโฟนของอาจารย์และนักศึกษาเพื่อใช้สำหรับการเช็คชื่อเข้าเรียนโดยเชื่อมต่อผ่านเทคโนโลยีบลูทูธ ซึ่งไม่จำเป็นต้องใช้อุปกรณ์เฉพาะอื่น ๆ เพิ่มเติม ซึ่งระบบที่นำเสนอนี้สามารถเช็คชื่อเข้าเรียนนักศึกษาได้ซึ่งอาจช่วยลดระยะเวลาและอำนวยความสะดวกในการเช็คชื่อเข้าเรียน สำหรับในแต่ละส่วนจะมีรายละเอียดสรุปดังนี้

5.1.1 ส่วนของโมไบล์แอปพลิเคชัน

ทำการออกแบบส่วนติดต่อผู้ใช้งานโดยแตกต่างกันตามบทบาทว่าเป็นอาจารย์หรือนักศึกษา จากนั้นพัฒนาโมไบล์แอปพลิเคชัน โดยรองรับระบบปฏิบัติการแอนดรอยด์ (Android) และติดตั้งโมดูลต่าง ๆ ที่ใช้ เช่น เว็บทรี (web3.js) สำหรับการติดต่อกับระบบบล็อกเชน จากนั้นทดสอบทำงานทุกส่วนของระบบ เช่น การวนเชื่อมต่อบลูทูธ (Bluetooth) กับสมาร์ตโฟน (Smartphone) หลาย ๆ เครื่อง การสร้างวิชาเรียน การลงเรียน การเช็คชื่อเข้าเรียน การส่งออกรายงาน เป็นต้น

5.1.2 ส่วนของแอปพลิเคชันเซิร์ฟเวอร์

ทำการติดตั้งระบบปฏิบัติการอูบุนตู (ubuntu) เพื่อนำมาใช้งาน จากนั้นติดตั้งโหนดเจเอส (node.js) และทำการติดตั้งโมดูลต่าง ๆ จากนั้นพัฒนาในส่วนของการตรวจสอบเพื่อคัดกรองเฉพาะผู้ที่ตรงตามเงื่อนไขและการส่งออกรายงาน จากนั้นทดสอบการทำงาน

5.1.3 ส่วนของบล็อกเชน

ทำการติดตั้งระบบปฏิบัติการอูบุนตูและติดตั้งเก็ตท์ (geth) เพื่อนำมาใช้รันอีเธอร์เลียม (Ethereum) โหนด โดยจะใช้พัพเพ็ต (puppeth) เพื่อมาสร้างไฟล์ genesis.json เพื่อใช้สำหรับการตั้งค่าเริ่มต้นระบบบล็อกเชน และรันระบบไว้สำหรับทดสอบ จากนั้นติดตั้งทรัฟเฟิล (truffle) เพื่อใช้ติดต่อกับระบบบล็อกเชนที่รันไว้ เพื่อพัฒนาสมาร์ตคอนแทร็กต์ (Smart Contract) จากนั้นทดสอบ

การทำงานของสมาร์ตคอนแทร็กต์ในส่วนต่าง ๆ ตามที่ได้ออกแบบไว้ด้วยทรัฟเฟิล ทำการเพิ่มโหนดเป็นหลาย ๆ โหนดเพื่อทดลองการเชื่อมต่อและการทำงานร่วมกันหว่างโหนดว่าเป็นไปตามที่กำหนดไว้

5.2 ปัญหา อุปสรรค

5.2.1 ส่วนของโมไบล์แอปพลิเคชัน

การพัฒนาโมไบล์แอปพลิเคชันนั้นจะมีอุปสรรคในเรื่องของการทดสอบการเชื่อมต่อบลูทูธ เนื่องจากว่าซอฟต์แวร์ที่ใช้จำลองสภาพแวดล้อมในการพัฒนาโมไบล์แอปพลิเคชันไม่รองรับการทำงานในส่วนนี้ ทำให้ต้องทดสอบเชื่อมต่อกับสมาร์ตโฟนเครื่องจริง ซึ่งจำนวนสมาร์ตโฟนที่มี ไม่เพียงพอที่จะทดลองวนเชื่อมต่อกับหลาย ๆ เครื่องได้ ทำให้ทดลองได้เพียงการเชื่อมต่อระหว่าง 1-2 เครื่องเท่านั้น

การติดตั้งโมดูลเสริมต่าง ๆ บางตัวจะไปชนกับคอร์โมดูลของระบบ ซึ่งอาจมีผลทำให้ทั้งโมดูลเก่าและที่ติดตั้งเพิ่มเข้าไปใหม่ไม่สามารถใช้ได้ ทำให้ต้องเลี่ยงใช้โมดูลตัวนั้นหรือหาโมดูลตัวอื่น หรือทำการติดตั้งเองโดยตรง

ในเรื่องของการตรวจสอบข้อความข้อผิดพลาดจากระบบบล็อกเชนนั้นจะไม่ค่อยสะดวก เนื่องจากตัวไลบรารีเว็บทรีเอง ยังไม่สามารถอ่านข้อความข้อผิดพลาดได้ ทำให้ตัวโมไบล์แอปพลิเคชันต้องมีการตรวจสอบเองหากต้องการรู้ข้อผิดพลาดนั้น ๆ

5.2.2 ส่วนของแอปพลิเคชันเซิร์ฟเวอร์

สำหรับส่วนของแอปพลิเคชันเซิร์ฟเวอร์ยังไม่พบปัญหาหรืออุปสรรคต่อการทำโครงงานนี้

5.2.3 ส่วนของบล็อกเชน

ส่วนของบล็อกเชนนั้นค่อนข้างเป็นส่วนที่ต้องใช้เวลามากทั้งการศึกษา การทำความเข้าใจและการนำมาใช้ เช่นเรื่องของการพัฒนาสมาร์ตคอนแทร็กต์ที่ไม่เหมือนการพัฒนาซอฟต์แวร์ทั่วไป เนื่องจากเป็นแอปพลิเคชันแบบกระจายศูนย์และยังเกิดขึ้นมาไม่นาน ทำให้มีข้อจำกัดหลายอย่างและแหล่งอ้างอิงที่น้อย ข้อจำกัดหลัก ๆ เช่น ขนาดของคอนแทร็กต์ที่มีการจำกัดไว้ ทำให้การพัฒนาต้องใช้กลวิธีหลายอย่างเพื่อจะข้ามข้อจำกัดนั้น เช่น การใช้โมเดลอีเธอร์นอลสตอเรจ (EternalStorage) ร่วมกันกับโมเดลพร็อกซี ซึ่งเป็นการพัฒนาสมาร์ตคอนแทร็กต์แยกเป็นส่วน ๆ และเรียกใช้งานผ่านคอนแทร็กต์พร็อกซี ซึ่งคอนแทร็กต์พร็อกซีก็จะไปเรียกใช้งานคอนแทร็กต์ส่วนต่าง ๆ ต่ออีกทีหนึ่ง

การที่อีเธอร์เลียมเวอร์ชวลแมชชีนนั้นไม่สามารถเข้าถึงอินเทอร์เน็ตได้ ซึ่งหากจะทำอะไรที่ต้องมีการเชื่อมต่ออินเทอร์เน็ตหรืออะไรก็ตามที่ไม่ได้อยู่ในอีวีเอ็ม จะต้องเอาการทำงานส่วนนั้นไปทำที่ส่วนอื่น ๆ แทน

5.3 แนวทางการพัฒนาต่อ

5.3.1 ส่วนของโมไบล์แอปพลิเคชัน

ทำการพัฒนาให้สามารถรองรับการทำงานบนระบบปฏิบัติการไอโอเอสได้ เพื่อเพิ่มตัวเลือกให้กับผู้ใช้งาน

5.3.2 ส่วนของแอปพลิเคชันเซิร์ฟเวอร์

ทำระบบจัดการเบื้องหลัง เช่น การติดตามสถานะปัจจุบันของระบบ การเพิ่มผู้ตรวจสอบหรือการเพิ่มโหนด โดยจัดทำเป็นเว็บแอปพลิเคชันให้มีส่วนติดต่อผู้ใช้ เพื่อให้ผู้ดูแลสามารถดูแลและจัดการระบบได้สะดวก

5.3.3 ส่วนของบล็อกเชน

ทำการติดตั้ง RPC โหนดที่เป็น HTTPS เพื่อเข้ารหัสการส่งข้อมูล ช่วยเพิ่มความปลอดภัยให้กับระบบ

5.3.4 ส่วนอื่น ๆ

สำหรับการพัฒนาในส่วนอื่น ๆ เช่น การพัฒนาเว็บแอปพลิเคชันขึ้นมา เพื่อให้อาจารย์และนักศึกษาสามารถดูข้อมูล จัดการวิชาเรียน รวมทั้งลงเรียนได้เลยบนเว็บ เพื่อเพิ่มทางเลือกให้กับผู้ใช้งาน ให้สามารถใช้งานบนคอมพิวเตอร์โน้ตบุ๊กหรือเดสก์ท็อปได้ หรือทำการพัฒนาเดสก์ท็อปแอปพลิเคชันเพื่อใช้ในคอมพิวเตอร์ที่เชื่อมต่อกับอุปกรณ์บลูทูธอยู่ ซึ่งจะสามารถนำมาใช้แทนสมาร์ตโฟนได้หากไม่สะดวก

บรรณานุกรม

ประทีป พืชทองหลาง, ญาตาวีมินทร์ พืชทองหลาง และอาภากร ปัญโญ. 2561. "ระบบตรวจสอบการเข้าชั้นเรียนและประเมินผลการเรียนรู้ด้วย QR Code ในรายวิชาศึกษาทั่วไป." **วารสารพุทธศาสตร์ศึกษา.** 9(1). : 1-16.

พิชญา จตุรวัฒน์, ภาสินี พงศ์มานะวุฒิ และมานพ พันธ์โคกกรวด. 2560. "การพัฒนาระบบบันทึกเวลาเรียนด้วยการตรวจจับและรู้จำใบหน้า." **วารสารเทคโนโลยีสารสนเทศลาดกระบัง.** 5(1). : 1-11.

วรินทร เจนชัย, จิติมนต์ อั่งสกุล และธรา อั่งสกุล. 2555. "ระบบบันทึกการเข้าชั้นเรียนผ่านบลูทูธ." **วารสารเทคโนโลยีสุรนารี.** 6(1). : 37-56.

วิชาญ เพชรมณี และขจรศักดิ์ พงศ์ธนา. 2552. "ระบบบันทึกเวลาอัตโนมัติด้วยลายนิ้วมือแบบไร้สาย." **วารสาร ICT เพื่อพัฒนาการเรียนรู้.** 1(1). : 1-6.

สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน) (สพร.). 2562. **BLOCKCHAIN for GOVERNMENT SERVICES การใช้เทคโนโลยีบล็อกเชนสำหรับภาครัฐ เวอร์ชัน 1.1.** [Online]. Available : https://www.dga.or.th/upload/download/file_ff487bacfb3198a615ca75112b8d156c.pdf

Nakamoto, S. 2008. **Bitcoin: A peer-to-peer electronic cash system.** [Online]. Available : https://bitcoin.org/bitcoin.pdf.

Yaga, D. Mell, P. Roby, N. and Scarfone, K. 2018. **Blockchain Technology Overview.** [Online]. Available : https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf.

Nuuneoi. 2016. **Blockchain for Geek.** [Online]. Available : https://nuuneoi.com/blog/blog.php?read_id=901

Buterin, V. 2014. **A next-generation smart contract and decentralized application platform.** [Online]. Available : https://github.com/ethereum/wiki/wiki/White-Paper

Ethereum community. 2017. **Ethereum Homestead Documentation.** [Online]. Available : http://ethdocs.org/en/latest/index.html.

Ethereum. 2018. **Solidity Documentation.** [Online]. Available : https://solidity.readthedocs.io/en/v0.5.3/.

OpenZeppelin. 2018. **openzeppelin-solidity.** [Online]. Available : https://github.com/OpenZeppelin/openzeppelin-solidity.

Nadolinski, E. and Spagnuolo, F. 2018. **Proxy Patterns.** [Online]. Available : https://blog.zeppelinos.org/proxy-patterns/.

Mudge, N. 2018. **ERC1538: Transparent Contract Standard.** [Online]. Available : https://github.com/ethereum/EIPs/issues/1538.

Takenobu, T. 2018. **Ethereum EVM illustrated.** [Online]. Available : https://github.com/takenobu-hs/ethereum-evm-illustrated.

Péter Szilágyi. 2017. **Clique proof-of-authority consensus protocol.** [Online]. Available : https://github.com/ethereum/EIPs/issues/225.

Salanfe. 2018. **Setup your own private Proof-of-Authority Ethereum network with Geth.** [Online]. Available : https://hackernoon.com/setup-your-own-private-proof-of-authority-ethereum-network-with-geth-9a0a3750cda8.

Vogelsteller, F. Kotewicz, M. Wilcke, J. and Oancea, M. 2018. **web3.js Documentation**. [Online]. Available : https://web3js.readthedocs.io/en/1.0/.

Douglas Nassif Roma Junior. 2017. **Easy Bluetooth Classic**. [Online].

Available : https://github.com/douglasjunior/react-native-easybluetooth-classic.

Padgette, J. et. al. 2018. **Guide to Bluetooth Security**. [Online].

Available : https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-121r2.pdf.

Artur Girenko. 2018. **Re-Natal**. [Online].

Available : https://github.com/drapanjanas/re-natal.