โครงการ

การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ	การออกแบบบล็อกเชนเก็บประวัติการศึกษา	
	Education Transcript Blockchain Design	
ชื่อนิสิต	นายทัชนนท์ กุมมาลือ	583 36300 23
	นางสาวณัฏฐมณฑน์ กุลรัตนรักษ์	583 36191 23
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
	สาขาวิชา วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2561	

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ที่ส่งผ่านทางคณะที่สังกัดในจุฬาลงกรณ์มหาวิทยาลัย (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของโครงงานทางวิชาการที่ส่งผ่านทางคณะที่สังกัด
The abstract and full text of senior projects in Chulalongkorn University Intellectual Repository(CUIR)
are the senior project authors' files submitted through the faculty.

การออกแบบบล็อกเชนเก็บประวัติการศึกษา

นายทัชนนท์ กุมมาลือ
นางสาวณัฏฐมณฑน์ กุลรัตนรักษ์

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2561
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Education Transcript Blockchain Design

Mr.Tatchanon Kummalue
Ms. Nattamon Kulrattanarak

A Project Submitted in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Science Program in Computer Science
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic Year 2018

Copyright of Chulalongkorn University

หัวข้อโครงงาน	การออกแบบบล็อกเชนเก็บประวัติการศึกษา
โดย	นายทัชนนท์ กุมมาลือ
	นางสาวณัฏฐมณฑน์ กุลรัตนรักษ์
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาโครงงานหลัก	อาจารย์ ดร.อาธร เหลืองสดใส

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับโครงงานฉบับนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตรปริญญาบัณฑิต ในรายวิชา 2301499 โครงงานวิทยาศาสตร์ (Senior Project)

.....
(ศาสตราจารย์ ดร.กฤษณะ เนียมมณี) หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะกรรมการสอบโครงงาน

.....
(อาจารย์ ดร.อาธร เหลืองสดใส) อาจารย์ที่ปรึกษาโครงงานหลัก

.....
(ผู้ช่วยศาสตราจารย์ ดร.กรุง สินอภิรมย์สราญ) กรรมการ

.....
(ผู้ช่วยศาสตราจารย์ ดร.ธีรพงษ์ พงษ์พัฒนเจริญ) กรรมการ

ทัชนนท์ กุมมาลือ, ณัฏฐมณฑน์ กุลรัตนรักษ์: การออกแบบบล็อกเชนเก็บประวัติการศึกษา, (Education Transcript Blockchain Design) อ.ที่ปรึกษาโครงงาน: อาจารย์ ดร.อาธร เหลืองสดใส, 61 หน้า.

ในปัจจุบันมีการใช้อินเทอร์เน็ตในชีวิตประจำวันอยู่ตลอดเวลา ทุกคนสามารถส่งข้อความหาคนที่อยู่อีกฝั่งหนึ่งของโลกได้ แต่ข้อมูลในโลกอินเตอร์เน็ตนั้นไม่ปลอดภัย ข้อมูลอาจโดนเจาะแก้ไขดัดแปลงหรือโดนแอบดูได้ ผู้พัฒนาเห็นว่าต้องมีการพัฒนาระบบรักษาความปลอดภัยของข้อมูลก่อนที่จะทำการส่งข้อมูลสำคัญออกไป ดังนั้นผู้พัฒนาจึงใช้เทคโนโลยีบล็อกเชนเข้ามาใช้ที่มีความสามารถเก็บข้อมูลได้หลากหลายและทำให้ข้อมูลนั้นมีความปลอดภัยโดยวิทยาการเข้ารหัสลับเพื่อให้สามารถส่งใบแสดงผลการศึกษาผ่านทางออนไลน์ได้อย่างมั่นใจและน่าเชื่อถือ วัตถุประสงค์ของโครงงานนี้เพื่อนำเสนอการออกแบบบล็อกเชนสำหรับการเก็บประวัติการศึกษาของนิสิตแต่ละคน และให้นิสิตสามารถขอใบแสดงผลการศึกษาผ่านเว็บไซต์ซึ่งจะเพิ่มความสะดวกสบายมากขึ้น

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต ทัชนนท์ กุมมาลือ

ลายมือชื่อนิสิต ณัฏฐมณฑน์ กุลรัตนรักษ์

สาขาวิชา วิทยาการคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาโครงงานหลัก

ปีการศึกษา 2561

5833630023, 5833619123: MAJOR COMPUTER SCIENCE

KEYWORDS : WEB APPLICATION/ BLOCKCHAIN

TATCHANON KUMMALUE, NATTAMON KULRATTANARAK: Education Transcript Blockchain Design. ADVISOR : ARTHORN LUANGSODSAI, Ph.D., 61 pp

Nowadays, we use internet in daily life. We can send a text to someone who lives in the other side of the world. But the data in internet is unsecure. They may be hacked, changed or eavesdropped. We need to improve security of data before sending important documents. Therefore, we use Blockchain Technology that can store various information and make them secure on the Internet with cryptography method to submit Education Transcript documents online confidentialy and reliably. The objectives of this work are to propose the blockchain design for education records of individuals and issue education documents for students via website which will make it more convenient to request an education transcript.

Department :Mathematics and Computer Science Student's Signature ..... Student's Signature .....

Field of Study : Computer Science Advisor's Signature .....

Academic Year : 2018

กิตติกรรมประกาศ

โครงงานระบบประวัติการศึกษาของนิสิตบนบล็อกเชนสามารถสำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์ ช่วยเหลือและสนับสนุนจาก คณาจารย์และบุคลากรต่าง ๆ ตลอดระยะเวลาการดำเนินโครงงาน

ขอขอบพระคุณ อาจารย์ ดร.อาธร เหลืองสดใส อาจารย์ที่ปรึกษาโครงงานฉบับนี้ ที่ให้คำปรึกษา แนะนำ และช่วยเหลือผู้พัฒนาตลอดการพัฒนาโครงงาน

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.กรุง สินอภิรมย์สราญ และผู้ช่วยศาสตราจารย์ ดร.ธีรพงษ์ พงษ์พัฒนเจริญ ผู้เป็นกรรมการคุมสอบ ที่ช่วยแนะนำแนวทางต่าง ๆ ที่เป็นประโยชน์ต่อโครงงานนี้

ขอขอบพระคุณ บิดามารดา และครอบครัวของทีมผู้พัฒนา ที่ให้การสนับสนุน ช่วยเหลือ และ ให้กาลังใจผู้พัฒนา จนกระทั่งโครงงานนี้สำเร็จได้ด้วยดี

ขอขอบคุณพี่ ๆ เพื่อน ๆ ทุกคนที่คอยให้ความช่วยเหลือในด้านต่าง ๆ จนทำให้การดำเนินโครงงานประสบความสำเร็จ

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... จ
สารบัญ ..... ช
สารบัญตาราง ..... ฌ
สารบัญภาพ ..... ญ
บทที่ 1 บทนำ 1
1.1 หลักการและเหตุผล 1
1.2 วัตถุประสงค์ 2
1.3 ขอบเขตของโครงงาน 2
1.4 วิธีการดำเนินงาน 2
1.5 ประโยชน์ที่ได้รับ 3
1.6 โครงสร้างของรายงาน 4
บทที่ 2 ความรู้พื้นฐานที่เกี่ยวข้อง 5
2.1 JavaScript 5
2.2 React 7
2.3 Blockchain 8
2.4 Hyperledger Fabric 12
2.5 Hyperledger Composer 16
บทที่ 3 การวิเคราะห์และออกแบบระบบ 19
3.1 การเก็บรวบรวมข้อมูล 19
3.2 การเก็บวิเคราะห์ระบบ 19

หน้า

3.3 การกำหนดความสามารถของระบบ.....19
3.4 การออกแบบระบบ.....20
3.4.1 แผนภาพยูสเคส.....20
3.4.2 แผนภาพแสดงกิจกรรม.....22
3.5 การออกแบบเครือข่ายบล็อกเชน.....24
3.6 การออกแบบส่วนตัวประสานผู้ใช้.....25

บทที่ 4 เทคนิคการเขียนโปรแกรมและการทดสอบระบบ.....28
4.1 เทคนิคการเขียนโปรแกรม.....28
4.2 ขอบเขตการทดสอบ (Scope of Testing).....35
4.3 การทดสอบโดยผู้พัฒนา.....35
4.4 การทดสอบโดยผู้ใช้งาน.....36
บทที่ 5 ข้อสรุปและข้อเสนอแนะ.....37
5.1 ข้อสรุป.....37
5.2 ปัญหาและอุปสรรค.....37
5.3 วิธีการแก้ปัญหา.....37
5.4 ข้อเสนอแนะ.....37
รายการอ้างอิง.....38
ภาคผนวก ก แบบเสนอหัวข้อโครงงาน รายวิชา 2301399 Project Proposal ปีการศึกษา 2561..41
ภาคผนวก ข คู่มือการใช้งาน 46
ประวัติผู้เขียน 49

สารบัญตาราง

หน้า

ตารางที่ 3.1 แสดงรายละเอียดฟังก์ชัน “เข้าสู่ระบบ”.....20
ตารางที่ 3.2 แสดงรายละเอียดฟังก์ชัน “ดูประวัติการศึกษา”..... 21
ตารางที่ 3.3 แสดงรายละเอียดฟังก์ชัน “ดูรายละเอียดของวิชา”.....21
ตารางที่ 3.4 แสดงรายละเอียดฟังก์ชัน “ขอประวัติการศึกษา”.....22
ตารางที่ 4.1 ตารางแสดงกรณีทดสอบและผลการทดสอบ.....36
ตารางที่ 4.2 ผลการทดสอบของผู้ใช้งาน.....37

สารบัญภาพ

หน้า

ภาพที่ 2.1 แสดงที่อยู่ของบัญชีแบบ Public Ledger....9
ภาพที่ 2.2 แสดงการเลือกผู้ยืนยันบล็อก....10
ภาพที่ 2.3 แสดงตัวอย่างการทำงานของ Practical Byzantine Fault Tolerance Consensus....11
ภาพที่ 2.4 แสดง Model ของ Hyperledger Fabric....13
ภาพที่ 2.5 แสดงการส่งคำสั่งเข้าสู่เครือข่ายบล็อกเชน....14
ภาพที่ 2.6 แสดงการตอบกลับของบล็อกเชนไปสู่แอปพลิเคชัน....14
ภาพที่ 2.7 แสดงการตรวจสอบของ SDK....15
ภาพที่ 2.8 แสดงการส่งคำสั่งไปยัง Ordering Service....15
ภาพที่ 2.9 แสดงการทำงานของ Ordering Service....16
ภาพที่ 2.10 แสดงการเสร็จสิ้นของการทำงาน....16
ภาพที่ 2.11 แสดงหน้าที่ของแต่ละไฟล์ใน Hyperledger Composer....17
ภาพที่ 2.12 แสดง Architecture ของ Hyperledger Composer....18
ภาพที่ 3.1 แผนภาพยูสเคสของระบบเว็บแอปพลิเคชัน....20
ภาพที่ 3.2 แสดงแผนภาพแสดงกิจกรรมการ “เข้าสู่ระบบ”....22
ภาพที่ 3.3 แสดงแผนภาพแสดงกิจกรรมการ “ดูประวัติการศึกษา”....23
ภาพที่ 3.4 แสดงแผนภาพแสดงกิจกรรมการ “ดูรายละเอียดของวิชา”....23
ภาพที่ 3.5 แสดงแผนภาพแสดงกิจกรรมการ “ขอประวัติการศึกษา”....24
ภาพที่ 3.6 แบบจำลองเครือข่ายบล็อกเชน....24
ภาพที่ 3.7 หน้าจอล็อคอินเข้าสู่ระบบ....25
ภาพที่ 3.8 หน้าเว็บไซต์หลัก....26
ภาพที่ 3.9 หน้าแสดงรายละเอียดวิชา....26
ภาพที่ 3.10 หน้าแสดงผลการศึกษา....27
ภาพที่ 3.11 หน้าแสดงไฟล์ตัวอย่างเอกสารสำคัญทางการศึกษา....27
ภาพที่ 4.1 การออกแบบการเก็บข้อมูลนักเรียน....28
ภาพที่ 4.2 การออกแบบการเก็บข้อมูลหัวข้อวิชา....29
ภาพที่ 4.3 การออกแบบการเก็บข้อมูลรายวิชา....29
ภาพที่ 4.4 การออกแบบการเก็บข้อมูลชั้นเรียน....30

ภาพที่ 4.5 การออกแบบการเก็บข้อมูลใบแสดงผลการศึกษา.....31
ภาพที่ 4.6 การออกแบบรายการบัญชีในระบบ (Function) 32
ภาพที่ 4.7 การออกแบบฟังก์ชันการบันทึกเกรด.....33
ภาพที่ 4.8 การออกแบบฟังก์ชันสร้างใบแสดงผลการศึกษาเมื่อสร้างนักเรียน.....33
ภาพที่ 4.9 การออกแบบฟังก์ชันการแก้ไขข้อมูลชื่อนักเรียน.....34
ภาพที่ 4.10 ตัวอย่างการเขียนไฟล์ Access Control.....34
ภาพที่ 4.11 ตัวอย่างการ Query นักเรียนจากนักเรียนที่กำหนด.....3

บทที่ 1
บทนำ

ในบทนี้จะกล่าวถึงความเป็นมาของโครงงานระบบประวัติการศึกษาของนิสิตบนบล็อกเชน โดยเริ่มจากหลักการและเหตุผล วัตถุประสงค์ ขอบเขตของโครงงาน ขั้นตอนการดำเนินงาน และประโยชน์ที่ได้รับ ตามลำดับ

1.1 หลักการและเหตุผล

ในปัจจุบันนี้ โลกของเรากำลังก้าวเข้าสู่สังคมยุคดิจิทัล ทุกคนให้ความสำคัญกับอินเทอร์เน็ตและสื่อต่าง ๆ ในโลกออนไลน์มากขึ้น ทางผู้จัดทำจึงเกิดแนวคิดในการสร้างเว็ปแอปพลิเคชันขึ้นมาเพื่อเพิ่มช่องทางให้นิสิตสามารถดำเนินการขอประวัติการศึกษาได้อย่างง่ายดายและสะดวกสบายมากขึ้น อีกทั้งในเอกสารประวัติการศึกษานั้นจะมีเพียงชื่อวิชาและเกรดเฉลี่ยของรายวิชานั้นๆ เท่านั้น ทางผู้จัดทำเห็นว่าข้อมูลเพียงเท่านั้นไม่เพียงพอต่อการที่จะเอกสารไปสมัครงงานหรือสมัครเรียนต่อต่าง ๆ จึงได้มีการเก็บข้อมูลเพิ่มเติม เช่น หัวข้อที่เรียนในรายวิชานั้น ๆ เนื่องจากในสถาบันการศึกษาต่างๆ ถึงจะมีชื่อวิชาที่เหมือนกัน แต่ในหัวเรื่องย่อยของรายวิชาอาจจะไม่เหมือนกัน การที่เพิ่มการเก็บข้อมูลพวกนี้จะทำให้เกิดการเปรียบเทียบได้

การที่ทางผู้จัดทำนำบล็อกเชนเข้ามาใช้แทนการเก็บข้อมูลแบบเดิม ๆ นั้นเนื่องจากการจะทำอะไรผ่านอินเทอร์เน็ตข้อมูลมีความสุ่มเสี่ยงที่จะเกิดการบิดเบือน หรือเจาะระบบเข้าไปเพื่อทำการเปลี่ยนข้อมูล บล็อคเชนสามารถสร้างความปลอดภัยให้กับข้อมูลได้เพราะบล็อกเชนจะเก็บข้อมูลเหล่านี้อยู่ในแต่ละบล็อก ที่นำมาเชื่อมต่อกันเป็นลูกโซ่ โดยตรวจสอบความถูกต้องและรักษาความปลอดภัยผ่านการเข้ารหัส แต่ละบล็อกจะมีค่าแฮชของบล็อกก่อนหน้าซึ่งใช้ยืนยันความถูกต้องของบล็อกก่อนหน้า บล็อกเชนจึงเป็นระบบการบันทึกข้อมูลที่ไม่สามารถแก้ไขได้อีกทั้งสามารถบันทึกข้อมูลแบบกระจายโดยสามารถยืนยันได้ว่าข้อมูลทุก ๆ ที่นั้นเหมือนกัน

ผู้จัดทำได้ทำการเลือกแพลตฟอร์ม Hyperledger ซึ่งเป็น Permissioned Blockchain ซึ่งมีลักษณะที่เหมาะกับระบบนี้ ในการเก็บรักษาข้อมูลและการเข้าถึงหรือเข้าร่วมบล็อกเชนจำเป็นจะต้องได้รับอนุญาตก่อน และแพลตฟอร์มนี้สามารถเขียน Smart Contract ซึ่งจะช่วยในการออกใบแสดงผลการศึกษาโดยอัตโนมัติผ่านการร้องขอของนิสิตได้เลย ดังนั้นการยื่นใบแสดงผลการศึกษาให้สถานที่ทำงานต่าง ๆ จะง่ายและรวดเร็วยิ่งขึ้น

1.2 วัตถุประสงค์

1. เพื่อศึกษาทำความเข้าใจเทคโนโลยีบล็อกเชน
2. เพื่อออกแบบบล็อกเชนการเก็บประวัติการศึกษา
3. เพื่อศึกษา Hyperledger Fabric และ Hyperledger Composer เพื่อนำไปประยุกต์ใช้ต่อไป

1.3 ขอบเขตของโครงงาน

1. สามารถบันทึกข้อมูลการลงทะเบียนของนิสิตมหาวิทยาลัยได้ เช่น รหัสวิชา ชื่อวิชา เป็นต้น
2. สามารถบันทึกข้อมูลของนิสิตได้โดยข้อมูลที่จะเก็บ เช่น ชื่อนิสิต วันเดือนปีเกิด ที่อยู่เบอร์ติดต่อ เป็นต้น
3. สามารถบันทึกหัวข้อต่าง ๆ ในแต่ละวิชาได้ ซึ่งวิชาแต่ละวิชาในมหาวิทยาลัยอาจมีชื่อเหมือนกันแต่หัวข้อในแต่ละวิชาไม่เหมือนกัน
4. สามารถออกใบแสดงผลการศึกษาได้
5. สามารถเชื่อมระบบกันระหว่างมหาวิทยาลัยและสถาบันการศึกษาอื่น ๆ เช่น Coursera Udemy เป็นต้น

1.4 วิธีการดำเนินงาน

เริ่มจากการศึกษาเอกสารและงานวิจัยต่าง ๆ ที่เกี่ยวข้องกับระบบของมหาวิทยาลัยว่าเดิมแล้วมีระบบเป็นอย่างไรมีข้อดีข้อเสียอย่างไร และศึกษาระบบบล็อกเชนที่จะนำมาใช้ในโครงงาน รวมถึงศึกษาเครื่องมือที่เกี่ยวข้องในการพัฒนาระบบบล็อกเชน เช่น VScode Docker เป็นต้นจากนั้นก็ทำการวิเคราะห์และออกแบบภาพรวมของระบบ เพื่อง่ายต่อการวางแผนพัฒนาระบบต่อไป หลังจากที่เราพัฒนาระบบเรียบร้อยแล้วก็ต้องตรวจสอบความถูกต้องของระบบ หากเกิดข้อผิดพลาดก็ต้องทำการแก้ไขปรับปรุง ขั้นตอนสุดท้ายคือจัดทำเอกสารประกอบโครงงานเพื่อบอกถึงวิธีการดำเนินงานต่างๆ เครื่องมือที่ใช้ และผลลัพธ์ที่ได้ เพื่อให้บุคคลที่สนใจได้ติดตามศึกษาค้นคว้าต่อไป

ตารางเวลาการดำเนินงาน

ขั้นตอนการดำเนินงาน	2018					2019			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	ม.ค.	เม.ย.
1. ศึกษาเอกสารและงานวิจัยต่างๆ ที่เกี่ยวข้องกับระบบมหาวิทยาลัยและบล็อกเชน	■	■							
2. ศึกษาเครื่องมือที่ใช้ในการพัฒนาระบบบล็อกเชน		■	■						
3. วิเคราะห์และออกแบบระบบ			■	■					
4. พัฒนาระบบบล็อกเชน				■	■	■	■		
5. ตรวจสอบความถูกต้องของระบบและแก้ไขข้อผิดพลาด					■	■	■	■	
6. จัดทำเอกสาร							■	■	■

1.5 ประโยชน์ที่ได้รับ

1. ประโยชน์ต่อผู้พัฒนา
    - ได้ศึกษาเทคโนโลยีบล็อกเชน ขั้นตอนการพัฒนาและการนำไปประยุกต์ใช้
    - ได้พัฒนาทักษะในการวิเคราะห์ระบบ
    - ได้ฝึกทักษะการทำงานเป็นกลุ่ม
2. ประโยชน์ต่อผู้ใช้
    - มหาวิทยาลัยสามารถนำระบบไปเป็นตัวต้นแบบในการพัฒนาระบบต่อไปได้
    - นิสิตสามารถขอข้อมูลจากทางมหาวิทยาลัยได้โดยข้อมูลมีความถูกต้องและตรวจสอบได้

1.6 โครงสร้างของรายงาน

สำหรับเนื้อหาในโครงงานฉบับนี้ จะประกอบไปด้วยเนื้อหา 5 บท ดังนี้

บทที่ 1 กล่าวถึงบทนำในภาพรวมของโครงงาน สาเหตุและความเป็นมา วัตถุประสงค์ ขอบเขต ขั้นตอนรวมถึงประโยชน์ที่ได้รับ

บทที่ 2 จะกล่าวถึงความรู้พื้นฐานที่เกี่ยวข้องที่ใช้ในการพัฒนาระบบบล็อกเชนและระบบเว็บแอปพลิเคชันนี้

บทที่ 3 จะกล่าวถึงวิธีการเก็บข้อมูล การวิเคราะห์และการออกแบบระบบ

บทที่ 4 จะกล่าวถึงเทคนิคการเขียนโปรแกรม การวางขอบเขตการทดสอบระบบ การวางแผนการทดสอบ วิธีการทดสอบและผลของการทดสอบ

บทที่ 5 จะกล่าวถึงการสรุปผล ที่ได้จากการพัฒนาระบบบล็อกเชนและระบบเว็ปแอปพลิเคชัน ปัญหาและอุปสรรคที่พบขณะดำเนินงาน วิธีการแก้ไขปัญหาและข้อเสนอแนะ

บทที่ 2
ความรู้พื้นฐานที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงความรู้พื้นฐานที่ใช้ในการพัฒนาระบบบล็อกเชนและระบบเว็ปแอปพลิเคชันซึ่งประกอบไปด้วยความรู้พื้นฐานทางด้านการเขียนโปรแกรมต่าง ๆ ดังนี้

2.1 JavaScript

ภาษาจาวาสคริปต์ เป็นภาษาคอมพิวเตอร์ที่ใช้ในการเขียนโปรแกรมบนระบบอินเทอร์เน็ตที่ออกแบบมาเพื่อให้นักพัฒนาใช้งานได้ง่ายโดยไม่ต้องพึ่งภาษาอื่น มีคำสั่งต่าง ๆ ที่ตอบสนองต่อนักพัฒนาได้มากขึ้น ภาษาจาวาสคริปต์มีลักษณะเป็นภาษาสคริปต์แบบ lightweight programming language และยังสามารถใช้ร่วมกับ HTML และ CSS เพื่อสร้างและพัฒนาเว็บไซต์ที่ใช้ได้กับ Web Browser รุ่นใหม่ๆ รวมถึง PCs, Laptops, Tablets และ Smart Phones

รูปแบบคำสั่งเบื้องต้น

การประกาศตัวแปรในภาษาจาวาสคริปต์

1. let ใช้ในการประกาศตัวแปรที่จะใช้เฉพาะในขอบเขตนั้นหรือเฉพาะในบล็อกนั้น ๆ คือ ถ้าตั้งค่าตัวแปรไว้ในขอบเขต แต่เรียกค่าจากภายนอกเข้ามาจะไม่สามารถเรียกมาได้

```
let x = 0;
if (x == 0) {
   let x = 1;
}
console.log(x);
// result จะเท่ากับ 0
```

2. var ใช้ในการประกาศตัวแปรภายในฟังก์ชัน จะไม่มีผลกับค่า var ที่อยู่ภายใน global เพราะคือขอบเขตของฟังก์ชัน

```
var x = 0;
if (x == 0) {
   var x = 1;
}
console.log(x);
// result จะเท่ากับ 1
```

3. const ใช้ในการประกาศตัวแปรแบบเฉพาะในขอบเขตของบล็อกนั้น ๆแต่จะต่างกับlet ตรงที่ไม่สามารถเปลี่ยนค่าได้ แต่สามารถเปลี่ยนค่าใน object ได้

```
const A = 234;
A =111 // A จะไม่สามารถเปลี่ยนค่าได้


obj = { a : 1 };
obj.a = 2;
console.log(obj);
// result จะเท่ากับ { a : 2 }
```

ฟังก์ชันการจัดการข้อมูลเชิงแถวระดับที่ใช้ในภาษาจาวาสคริปต์

1. shift( ) จะเอาค่าที่อยู่ด้านซ้ายสุดที่อยู่ในอาเรย์ออก

```
var list = [“Ploy”, “Aye”, “Tor”];
list.shift();
list;
// result [“Aye”, “Tor”];
```

2. pop( ) จะเอาค่าที่อยู่ด้านขวาสุดที่อยู่ในอาเรย์ออก

```
var list = [“Ploy”, “Aye”, “Tor”];
list.pop();
list;
// result [“Ploy”, “Aye”];
```

3. push( ) จะเอาค่าใหม่ที่ใส่เข้าไปไว้ที่ตำแหน่งสุดท้ายในอาเรย์

```
var list = [“Ploy”, “Aye”, “Tor”];
list.push(Petch);
list;
// result [“Ploy”, “Aye”, “Tor”, “Petch”];
```

4. concat( ) จะเป็นการเชื่อม 2 อาเรย์เข้าด้วยกัน

```
var list = [“Ploy”, “Aye”, “Tor”];
var newlist = [“Mint”, “Kaopode”];
var allList = list.concat(newlist);
allList;
// result [“Ploy”,“Aye”, “Tor”, “Mint”, “Kaopode”];
```

5. sort( ) จะเรียงค่าในอาเรย์ตามตัวอักษร a-z

```
var list = [“Ploy”, “Aye”, “Tor”];
list.sort();
list;
// result [“Aye”,“Ploy”, “Tor”];
```

2.2 React

React เป็น JavaScript Library ที่เอาไว้สำหรับทำ UI เอาไว้ render พวก Component หรือ Element ต่าง ๆ ในหน้า HTML เท่านั้น Component คือ ส่วนต่าง ๆ ในเว็บ State คือข้อมูลที่อยู่ใน Component แต่ละชิ้น และ Props (Properties) คือ ข้อมูลที่ถูกส่งต่อจาก Component ชั้นบนลงไปชั้นล่าง React ใช้ JSX ในการแสดงผลเว็บไซต์ คล้าย HTML แต่จะแตกต่างตรงที่จะเขียนเข้าไปในไฟล์จาวาสคริปต์แทนไฟล์ HTML การแทรกจาวาคริปต์เข้าไปใน React นั้นก็สามารถเขียนเหมือน HTML Attribute ได้เลย

2.3 Blockchain

Blockchain เป็นการเก็บข้อมูล (Data Structure) รูปแบบหนึ่ง ซึ่งมีรูปแบบการเก็บข้อมูลแบบไม่มีศูนย์กลาง (Distributed) โดยให้ทุกคนที่อยู่ในเครือข่ายเก็บข้อมูลชุดเดียวกัน เมื่อมีการแก้ไขอัปเดตก็จะทำเหมือนๆ กันทุกคนในเครือข่าย เพื่อให้ทุกคนในเครือข่ายถือข้อมูลชุดเดียวกันตลอด จึงทำให้มั่นใจได้ว่าข้อมูลจะเชื่อถือได้แน่นอน โดยเทคโนโลยีนี้เป็นต้นกำเนิดของ Bitcoin หรือสกุลเงินดิจิตอลสกุลแรกของโลก

2.3.1 แนวคิดของ Blockchain

แนวคิดของ Blockchain เกิดจากการที่มีการจัดการของข้อมูลถูกจัดการอยู่ที่เดียว หรือ การมีองค์กรใดองค์กรหนึ่งจัดการมูลนั้นอยู่เพียงองค์กรเดียว เรียกว่า ระบบรวมศูนย์กลาง (Centralized) ทำให้ความน่าเชื่อถือของข้อมูลหรือการฝากข้อมูลต่างๆ ลดลง เนื่องจากหากองค์กรที่เราฝากข้อมูลเปลี่ยนแปลงข้อมูลโดยมิได้รับอนุญาต หรือองค์กรถูกโจมตีก็จะทำให้ข้อมูลหายไป ยกตัวอย่างเช่น ธนาคารคือองค์กร และเงินคือข้อมูลที่เราฝากให้ธนาคารดูแล ในระบบปัจจุบันคือการที่เราเชื่อมั่นในธนาคารว่าจะดูแลเงินของเรา แต่ถ้าหากธนาคารถูกปล้นหรือถูกโจมตีทางไซเบอร์ เงินที่เราฝากไว้ก็จะหายไปและจะได้รับการรับผิดชอบเพียงแค่ที่ธนาคารกำหนดไว้

ทำให้เกิดแนวคิดในการรักษาข้อมูลแบบกระจาย เพื่อให้การข้อมูลมีความน่าเชื่อถือและเป็นการป้องกันข้อมูล เพราะเมื่อข้อมูลไม่มีตัวกลางในการถือข้อมูล ให้ทุกคนได้มีสิทธิ์ถือข้อมูลนั้นๆ หากถูกโจมตี ก็สามารถนำข้อมูลของคนอื่นๆ มายืนยันกันได้ ทำให้การโจมตีต้องโจมตีทุกคนพร้อมกันในการแก้ไขข้อมูลบนระบบนี้

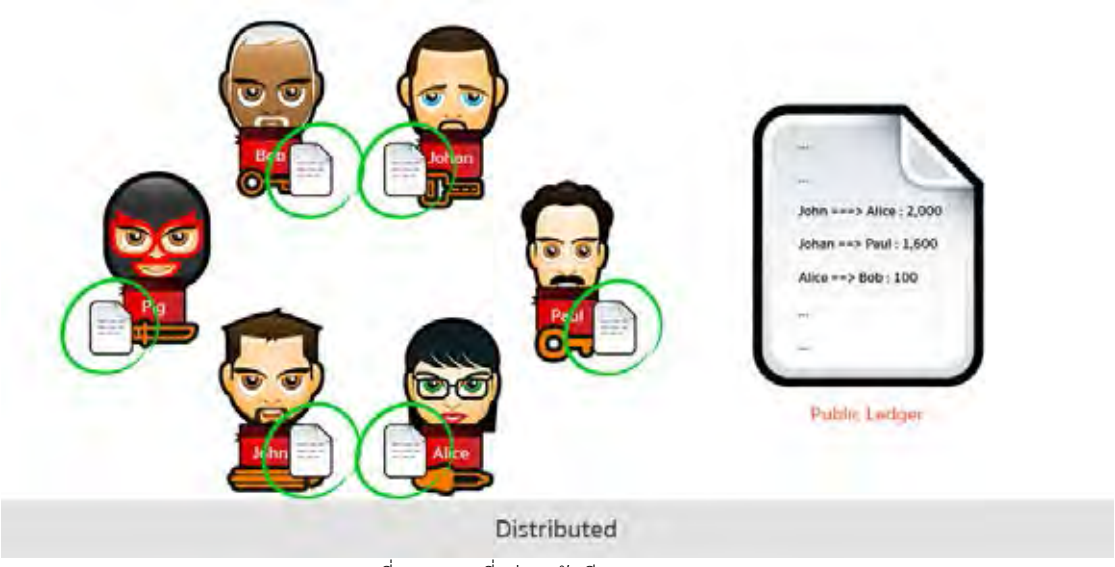
2.3.2 องค์ประกอบของ Blockchain

1. โหนด (Node)

โหนด คือ เครื่องที่อยู่ในระบบเครือข่ายของ Blockchain โดยจะมีกุญแจ 2 อัน ซึ่งจะกล่าวในบทต่อไป

2. บัญชี (Ledger)

บัญชีที่โหนดทุกๆ อันจะเก็บไว้ในการจดบันทึก Transaction ต่างๆ โดยในระบบ Blockchain จะเป็น Public Ledger คือ บัญชีที่จะเปิดเผยข้อมูลสู่สาธารณะ เนื่องจากต้องใช้ในการยืนยันข้อมูลกัน จึงจำเป็นต้องเปิดเผยข้อมูลกันและกัน



ภาพที่ 2.1 แสดงที่อยู่ของบัญชีแบบ Public Ledger

ที่มา: https://nuuneoi.com/blog/blog.php?read_id=900

3. กุญแจ (Key)

กุญแจมี 2 รูปแบบ คือ Private Key และ Public Key ซึ่งจะอยู่ในรูปแบบลายเซ็น (Signature) และกระเป๋า (Wallet) ใช้ในการทำธุรกรรมต่างๆ

Bitcoin Address: " 1HpfUoWJBAFApCE8mnHpCi8vVPRYRRLDpE "

Private Key: " KxxPMPbAqBEwQCKgtUgCymRBiEJqGuf6bigRcVxqycCL4N3qNFLi "

ตัวอย่างค่าของ Public Key และ Private Key

4. รายการบัญชี (Transaction)

รายการบัญชี คือ รายการบัญชีการกระทำต่างๆ ซึ่งทำให้เกิดการเปลี่ยนแปลงหรือเพิ่มเติมของข้อมูล เช่น Alice ยื่นความต้องการโอนเงินให้ Bob จำนวน 50 บาท เมื่อข้อมูลนี้ถูกบันทึกลงใน Blockchain แล้วนับเป็น 1 รายการบัญชี

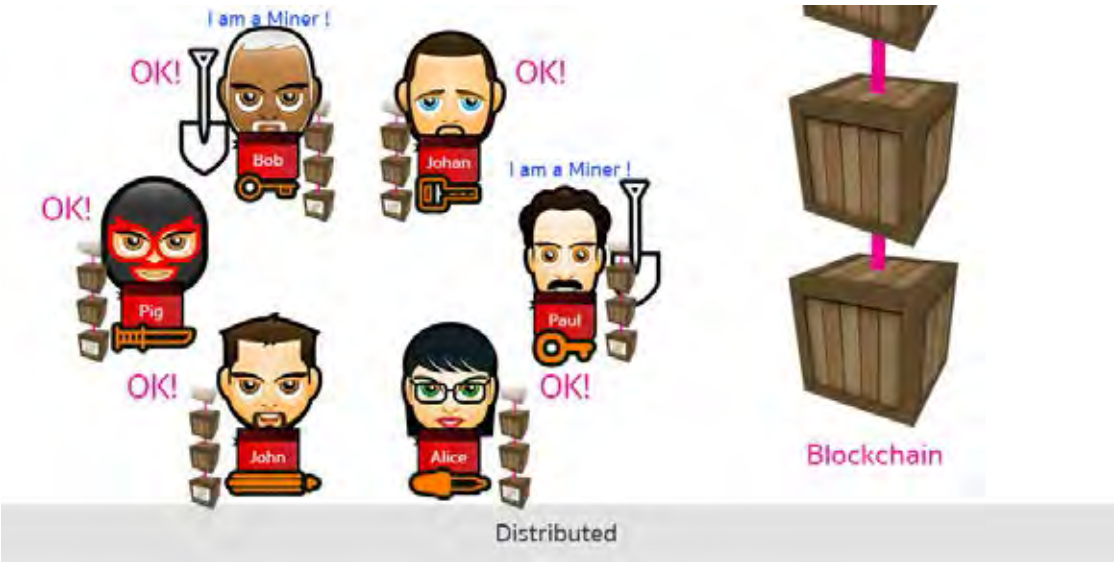
5. ข้อตกลงร่วมกัน (Consensus)

ข้อตกลงร่วมกันมีขึ้นในเครือข่ายบล็อกเชนเพื่อตัดสินว่าใครจะเป็นผู้กำหนดข้อมูลในบล็อกถัดไป ข้อตกลงร่วมกันภายในระบบเครือข่ายมีหลายรูปแบบ

ตัวอย่างข้อตกลงร่วมกันภายในเครือข่ายบล็อกเชน

- Proof of Work (PoW) เป็นการพิสูจน์โดยใช้แรงงานเพื่อให้ได้ซึ่งคำตอบมา โดยในการเป็นคนยืนยันความถูกต้องของรายการบัญชีในบล็อกนั้นจะต้องหาค่าๆ หนึ่งจากระบบ ที่ถูกฟังก์ชันแฮชเข้ารหัสไว้ (Cryptographic Hash Function) เนื่องจาก ฟังก์ชันแฮชเป็นฟังก์ชันทางเดียวกล่าวคือไม่

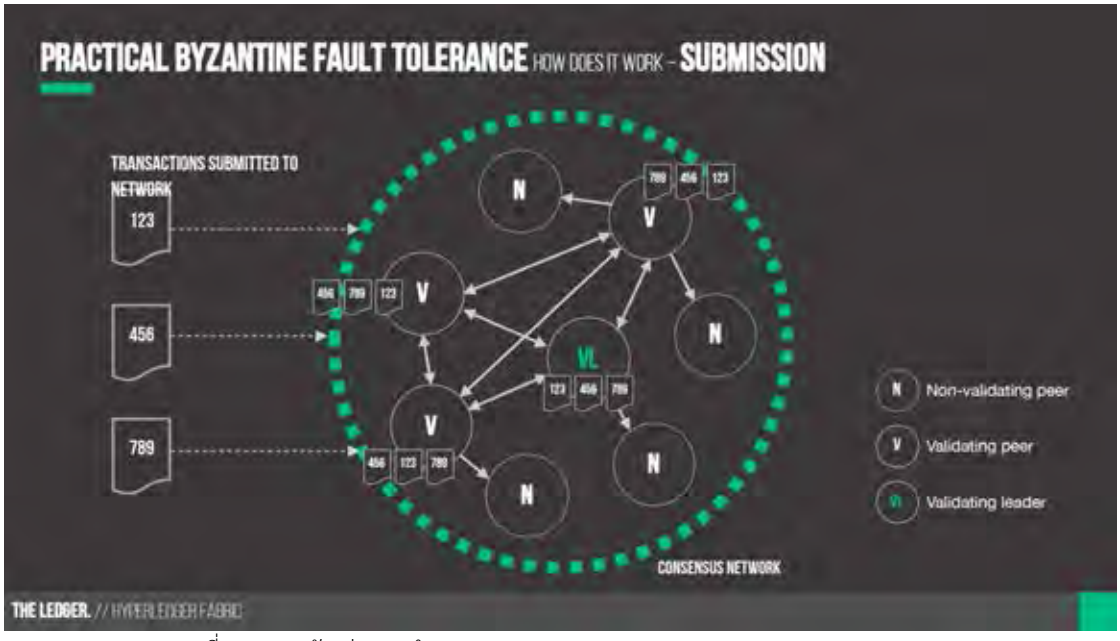
สามารถหาค่าย้อนกลับได้ วิธีเดียวที่จะทำการถอดรหัสหาค่าข้างในได้คือการค้นหารูปแบบที่เป็นไปได้ทั้งหมด (Brute-Force) จึงต้องใช้แรงประมวลผลจากคอมพิวเตอร์อย่างมากในการหาค่าที่จะแก้ไขสมการได้ ข้อตกลงร่วมกันประเภทนี้ใช้ใน บิทคอยน์ซึ่งเป็นสกุลเงินดิจิตอลบนระบบเครือข่ายบล็อกเชน



ภาพที่ 2.2 แสดงการเลือกผู้ยืนยันบล็อก

ที่มา: https://nuuneoi.com/blog/blog.php?read_id=900

- Practical Byzantine Fault Tolerance (PBFT) ในข้อตกลงนี้ทุกโหนดจะเชื่อมต่อกันแบบ Peer-to-Peer ทำให้สามารถส่งข้อมูลหากันได้หมดทุกโหนด โดยเมื่อมีรายการบัญชีเข้ามาในเครือข่าย โหนดต่างๆจะได้รับรายการบัญชีนั้นๆ และได้ยืนยันรายการบัญชีนั้นๆ ทุกโหนด โดยแต่ละโหนดจะส่งผลการยืนยันให้กับโหนดๆ อื่นๆ เช่นกัน โดยเมื่อรายการบัญชีนั้นๆ ได้รับการยืนยันจากแต่ละโหนดมากกว่า (2n+1)/3 โหนด (n คือโหนด) รายการบัญชีนั้นจะได้รับการยืนยันว่าถูกต้อง และจะมีโหนดใดโหนดหนึ่งที่ทุกโหนดร่วมกันโหวตให้เป็นโหนดเรียงรายการบัญชีและจัดเก็บลงในบล็อก ข้อตกลงร่วมกันประเภทนี้ใช้ในระบบเครือข่ายบล็อกเชนแบบส่วนตัว เช่น Hyperledger Fabric เป็นต้น



ภาพที่ 2.3 แสดงตัวอย่างการทำงานของ Practical Byzantine Fault Tolerance Consensus

ที่มา: https://www.altoros.com/blog/hyperledger-fabric-chaincode-practical-byzantine-fault-tolerance-and-v1-0/

6. การตรวจสอบความถูกต้อง (Validation)

การตรวจสอบความถูกต้องของข้อมูลภายในเครือข่ายบล็อกเชน จะขึ้นอยู่กับข้อตกลงร่วมกันในแต่ละเครือข่าย

2.3.3 การทำงานของ Blockchain

1. เกิดความต้องการในการส่งข้อมูลกันระหว่าง X และ Y

ผู้ส่งข้อมูลจะสร้างรายการบัญชี (Transaction) โดยการนำ Public Key ของคนที่เราต้องการส่งข้อมูลไปให้ ใส่ลงไปในรายการบัญชีพร้อมนำ Private Key ของเราเซ็น (Sign) ลงไปในรายการบัญชีซึ่งเป็นการยืนยันว่า รายการบัญชีนี้เราเป็นคนกระทำโดยใช้ Digital Signature Method แล้วจึงส่ง Public Key, Signature และ รายการบัญชี เข้าไปใน Blockchain Network

2. การยืนยันรายการบัญชี Transaction ภายใน Network

เมื่อรายการบัญชีถูกส่งเข้ามาภายในเครือข่าย จะทำการบันทึกลงในทุกๆ บัญชีภายในเครือข่าย โดยรายการนั้นๆ จะมีสถานะเป็นยังไม่ได้ยืนยันความถูกต้อง (Unconfirmed Transaction)

3. ยืนยันความถูกต้องของรายการบัญชีตามข้อตกลงร่วมกัน (Consensus)

ในแต่ละเครือข่าย Blockchain อาจจะมีข้อตกลงร่วมกันที่ไม่เหมือนกัน ซึ่งส่งผลให้การยืนยันความถูกต้องในแต่ละเครือข่ายไม่เหมือนกัน ดังนั้นจะขอยกตัวอย่างการยืนยันความถูกต้องขอเครือข่ายของ Bitcoin

ข้อตกลงร่วมกันของ Bitcoin คือ Proof of Work หรือการพิสูจน์ด้วยการทำงาน โดยผู้ทำการยืนยันความถูกต้องจะถูกเรียกว่า Miner โดย Miner จะใช้แรงประมวลผลจากอุปกรณ์ประมวลผลต่างๆ เช่น คอมพิวเตอร์, เครื่องเซิฟเวอร์ และเครื่องที่ทำมาโดยเฉพาะในการขุดเหรียญ (ASIC) ในการหาค่าบางค่าที่ถูก Hash ไว้โดยใช้การค้นหารูปแบบที่เป็นไปได้ทั้งหมด (Brute-Force) เพื่อใช้ในการเป็นผู้สร้างบล็อก เพื่อใส่รายการที่มีสถานะยังไม่ได้ยืนยันลงไป และเมื่อบล็อกเชื่อมกับบล็อกก่อนหน้าไปเรื่อยๆ ก็จะเกิด Chain ขึ้นมา โดยการเชื่อมแต่ละบล็อกจะมีการอ้างอิงถึงค่าแฮชของบล็อกก่อนหน้า ซึ่งค่าแฮชนี้หากมีการเปลี่ยนแปลงข้อมูลก่อนผ่านฟังก์ชันแม้แต่นิดเดียว ค่าแฮชที่ได้จะเปลี่ยนไปจนไม่เหมือนกันเลย ทำให้เมื่อเกิดการแก้ข้อมูลภายในบล็อกก่อนหน้า ค่าแฮชจะเปลี่ยนแล้วเครือข่ายก็จะปฏิเสธข้อมูลที่ถูกแก้ เมื่อรายการอยู่ในบล็อกเชนแล้วก็จะมีสถานะเป็นยืนยัน (Confirmed Transaction)

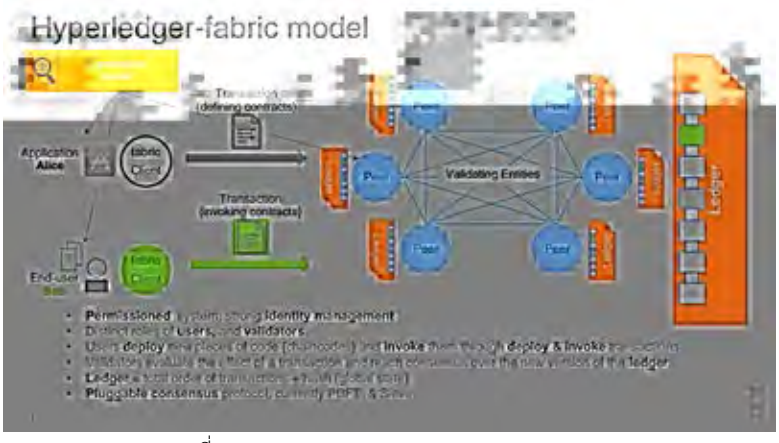
tor→BD144D4B9250B5395989271BF922DC5EFCC4F66C8EE2E180E1A29C96DB47F8B5

Tor→E7121E9FF0BCBE21DBF73A10D4AB7A7F4DDAFF759C00C8309FD64CBBC85A7995

ตัวอย่างค่าแฮช SHA-256 ที่เปลี่ยนเพียงแค่ตัวอักษรเพียงตัวเดียว

2.4 Hyperledger Fabric

Hyperledger Fabric เป็น Permission Blockchians คือการที่ผู้เข้าร่วมในระบบจะต้องเป็นผู้ที่ได้รับอนุญาตจากภายในระบบเท่านั้น ซึ่งทำให้การทำงานของระบบรวดเร็ว ลดความเสี่ยงในระบบ และสามารถพัฒนาระบบได้อย่างรวดเร็ว Hyperledger Fabric ถูกออกแบบมาให้มีสถาปัตยกรรมแบบแยกส่วน (Modular Architecture) โดยทำให้ส่วนประกอบต่างๆ สามารถประกอบแยกส่วนกันได้อย่างอิสระ ทำให้การพัฒนาสามารถแยกส่วนพัฒนาได้ อีกทั้งยังออกแบบโดยใช้แนวคิดเชิงธุรกิจโดยเฉพาะ จึงเหมาะอย่างมากในการนำมาพัฒนาเพื่อในธุรกิจต่างๆ Hyperledger Fabric เป็น Open Source ซึ่งถูกพัฒนาโดย Linux Foundation ที่เกิดจากการรวมตัวของบริษัทเทคโนโลยีชั้นนำทั่วโลก



ภาพที่ 2.4 แสดง Model ของ Hyperledger Fabric

ที่มา: https://blockgeeks.com/guides/hyperledger/

2.4.1 ส่วนประกอบในการทำงาน Hyperledger Fabric

- บัญชี (Ledger)

บัญชีจะถูกแยกเป็น 2 ประเภทได้แก่ สถานะปัจจุบัน (World State) และบล็อกเชน (Blockchain) โดย สถานะปัจจุบันเมื่อมีการแก้ไขเปลี่ยนแปลงและผ่านการยืนยันเรียบร้อยแล้วจะถูกบรรจุเข้าบล็อกเชน

- สัญญาอิเล็กทรอนิกส์ (Smart contract)

สัญญาอิเล็กทรอนิกส์หรือ Chaincode เป็นโค้ด (code) ที่จะถูกทำงานจากผู้ใช้ผ่านแอพลิเคชันภายนอก โดยจะเป็นการจัดการแก้ไขเปลี่ยนแปลงข้อมูลของสถานะปัจจุบัน โดยสัญญาอิเล็กทรอนิกส์จะถูกติดตั้งลงบนโหนดและถูกนำไปใช้กับช่องสื่อสารที่กำหนด

- โหนด (Node)

โหนดใน Hyperledger Fabric ถูกแบ่งออกเป็น 3 ประเภท

1. โหนดผู้ใช้ (Client or Submitting-client) ทำหน้าที่ส่งคำร้องในการทำรายการต่างๆ (Transaction) และ แพร่กระจายคำขอของรายการไปที่ระบบการจัดเรียง

2. เพียร์ (Peer) โหนดที่ทำหน้าที่ยืนยันรายการและเก็บรักษาสถานะและสำเนาของบัญชี นอกจากนั้นเพียร์สามารถเพิ่มหน้าที่พิเศษที่สามารถรับรองรายการซึ่งใช้ในสัญญาอิเล็กทรอนิกส์ได้อีกด้วย (Endorsing peer or Endorser)

3. โหนดจัดเรียง (Ordering-service-node or Orderer) โหนดซึ่งทำหน้าที่ดำเนินงานระบบติดต่อสื่อสารภายในเครือข่าย ซึ่งใช้ในการแพร่กระจายข้อความของรายการที่ใช้ในการดำเนินการของโหนดอื่นๆ

- หน่วยออกคำสั่ง (Ordering service)

หน่วยออกคำสั่งจะเป็นหน่วยคอยติดต่อสื่อสารผ่านช่องสื่อสารไปถึงโหนดผู้ใช้และเพียร์ โดยจะเป็นช่องทางการแพร่กระจายของคำสั่งการทำงานของรายการ

- ช่องสื่อสาร (Channel)

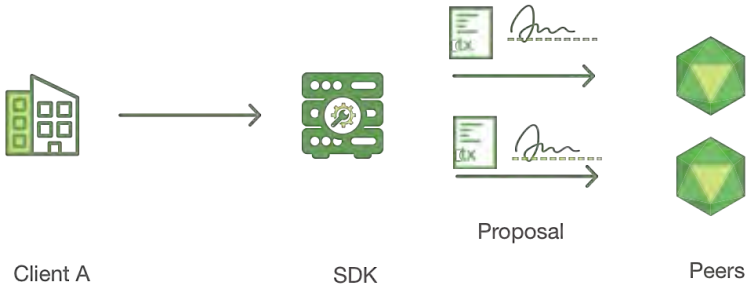
ช่องสื่อสารจะเป็นกลไกในการแยกเป็นส่วนๆ ซึ่งทำให้รายการบัญชีมองเห็นเฉพาะผู้ที่อยู่ในช่องสื่อสารเท่านั้น โดยบัญชีก็จะแบ่งปันกันแค่ผู้ที่อยู่ในช่องสื่อสารเดียวกันเท่านั้น

- ระบบให้บริการสมาชิก (Membership Service Provider)

ระบบให้บริการสมาชิกเป็นระบบจัดการข้อมูลประจำตัวให้แก่ผู้ใช้และเพียร์เข้าร่วมในเครือข่าย Hyperledger Fabric หากไม่มีข้อมูลประจำตัวนี้ก็จะไม่สามารถเข้าร่วมเครือข่ายได้ และไม่สามารถทำรายการบัญชีใดๆ ได้

2.4.2 การทำงานของ Hyperledger Fabric

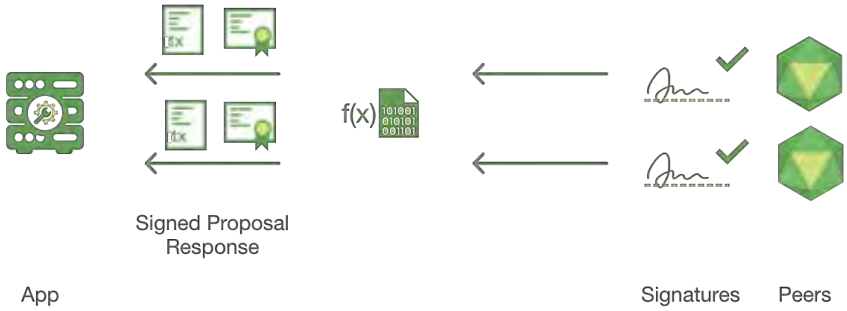
1. ส่งคำสั่งเข้าสู่เครือข่ายโดยจะส่งคำสั่งผ่านแอพลิเคชันที่เชื่อมต่อกับเครือข่ายบล็อกเชน โดยคำสั่งจะถูกส่งไปยัง Endorsing Peers



ภาพที่ 2.5 แสดงการส่งคำสั่งเข้าสู่เครือข่ายบล็อกเชน

ที่มา: https://hyperledger-fabric.readthedocs.io/en/release-1.4/txflow.html

2. Endorsing Peers จะทำการตรวจเช็คคำขอที่ส่งมา เมื่อผ่านจะทำการคัดลองค่าของ World State ทำการทำตามคำขอที่ได้มาแล้วลงลายเซ็นส่งกลับไปยังแอพลิเคชันผ่าน SDK (Software Developer Kit) โดยยังไม่บันทึกค่าลงใน World State



ภาพที่ 2.6 แสดงการตอบกลับของบล็อกเชนไปสู่แอปพลิเคชัน

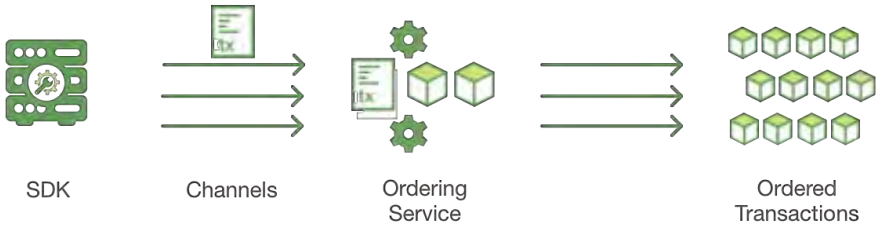
ที่มา: https://hyperledger-fabric.readthedocs.io/en/release-1.4/txflow.html

3. SDK ทำการตรวจสอบลายเซ็นจาก Endorsing Peer และเปรียบเทียบว่าถ้าคำสั่งเป็นแค่การ Query และได้คำตอบแล้ว SDK จะไม่ส่งรายการบัญชีไปให้ Ordering Service แต่ถ้าคำสั่งเป็นแก้ไขเปลี่ยนแปลงบัญชี SDK จะเช็คว่ามี Peer ที่ต้องเซ็นครบตามข้อกำหนดหรือยัง ถ้าครบแล้วจะส่งไปให้ Ordering Service

ภาพที่ 2.7 แสดงการตรวจสอบของ SDK

ที่มา: https://hyperledger-fabric.readthedocs.io/en/release-1.4/txflow.html

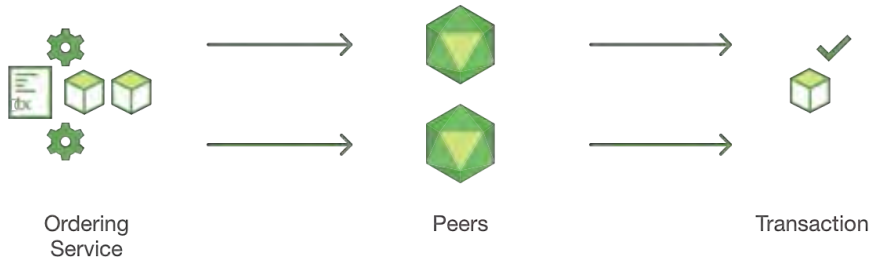
4. แอปพลิเคชันจะแพร่กระจายคำขอของรายการบัญชีและตอบกลับ Ordering Service ด้วย “Transaction Message” โดยรายการบัญชีจะบรรจุ ชุดข้อมูลอ่าน/เขียน ลายเซ็นของ Endorsing Peer และ รหัส ID ของช่องสื่อสาร Ordering Service จะทำการรับรายการบัญชีจากทุกๆ ช่องสื่อสาร แล้วเรียงอย่างเป็นลำดับตามช่องสื่อสาร และสร้างบล็อกของรายการบัญชีแต่ละช่องสื่อสาร



ภาพที่ 2.8 แสดงการส่งคำสั่งไปยัง Ordering Service

ที่มา: https://hyperledger-fabric.readthedocs.io/en/release-1.4/txflow.html

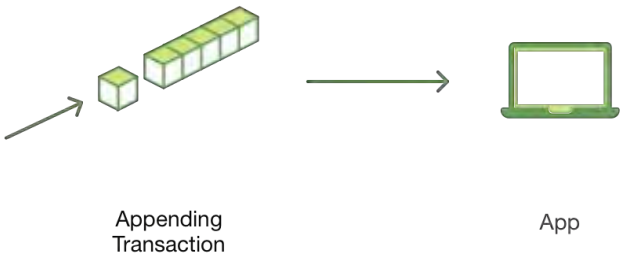
5. บล็อกของรายการบัญชีจะถูกส่งไปให้ทุก Peers ในช่องสื่อสาร รายการบัญชีในบล็อกจะถูกตรวจสอบเพื่อให้มั่นใจได้ว่าถูกต้องตามข้อกำหนดและไม่ถูกเปลี่ยนแปลงค่าใดๆ และรายการเดินบัญชีจะถูกติดป้ายว่ารายการบัญชีนี้ถูกต้องหรือไม่ถูกต้อง



ภาพที่ 2.9 แสดงการทำงานของ Ordering Service

ที่มา: https://hyperledger-fabric.readthedocs.io/en/release-1.4/txflow.html

6. แต่ละ Peer จะต่อท้ายบล็อกในห่วงโซ่ของช่องสื่อสารนั้นๆ และแต่ละรายการบัญชีที่ถูกนำไปเขียนบนสถานะของฐานข้อมูล และจะเกิดเหตุการณ์ให้ไปแจ้งเตือนแอปพลิเคชันว่ารายการบัญชีถูกบันทึกลงในบล็อกเชนเรียบร้อยแล้วไม่ว่ารายการบัญชีจะถูกต้องหรือไม่ก็ตาม



ภาพที่ 2.10 แสดงการเสร็จสิ้นของการทำงาน

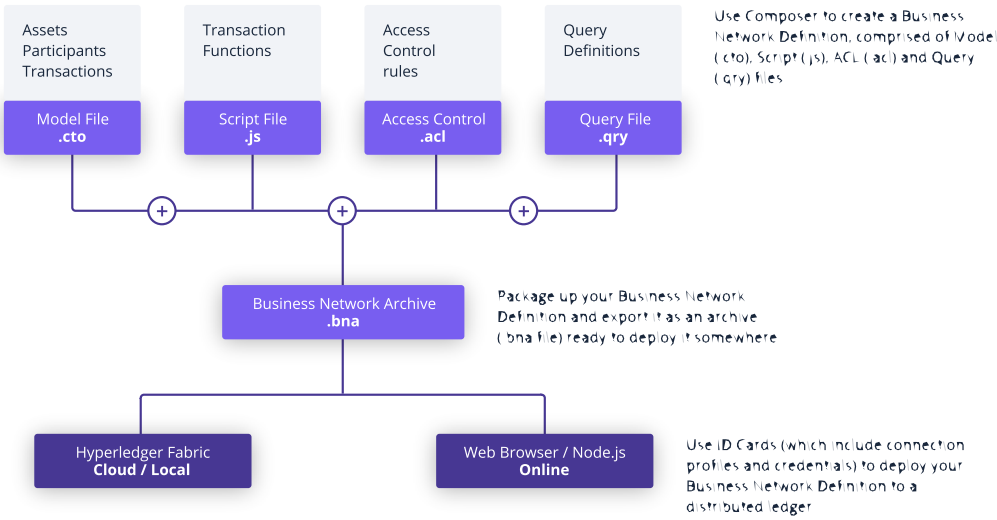
ที่มา: https://hyperledger-fabric.readthedocs.io/en/release-1.4/txflow.html

2.5 Hyperledger Composer

Hyperledger Composer เป็นทั้งเครื่องมือและโครงร่างซอฟต์แวร์ ซึ่งช่วยให้การพัฒนาเครือข่าย Blockchain เป็นไปได้ง่ายขึ้น โดย Hyperledger Composer ออกแบบมาจาก Use case จากธุรกิจต่างๆ จึงใช้ Modelling Language ซึ่งง่ายต่อการเข้าใจในการเขียนโค้ดมากขึ้น สามารถเขียนโดยอิงจากระบบธุรกิจจริงๆ ได้ และสนับสนุน Hyperledger Fabric ให้สามารถทำงานร่วมกันได้ และเป็นอีกหนึ่งโปรเจกต์ซึ่งพัฒนาโดย Linux Foundation

Hyperledger Composer มีไฟล์ประกอบสำคัญได้แก่

1. Model File เป็นไฟล์ที่ใช้ในการกำหนดองค์ประกอบต่างๆ ภายในระบบ เขียนโดยภาษาที่ object-oreinted ประกอบด้วย
   - สินทรัพย์ (Asset)
   - ผู้เข้าร่วม (Participant)
   - การดำเนินการ (Transaction)
2. Script File เป็นไฟล์ที่กำหนดการทำงานของการดำเนินการจาก Model File
3. Access Control เป็นไฟล์ที่สร้างกฎต่างๆ ว่าผู้เข้าร่วมคนไหนสามารถจัดการกับสินทรัพย์ชนิดใดได้บ้าง
4. Query File เป็นไฟล์ที่สร้างคำสั่ง Query ที่ใช้ในการหาข้อมูลต่างๆ ภายในดาต้าเบสของบล็อกเชน

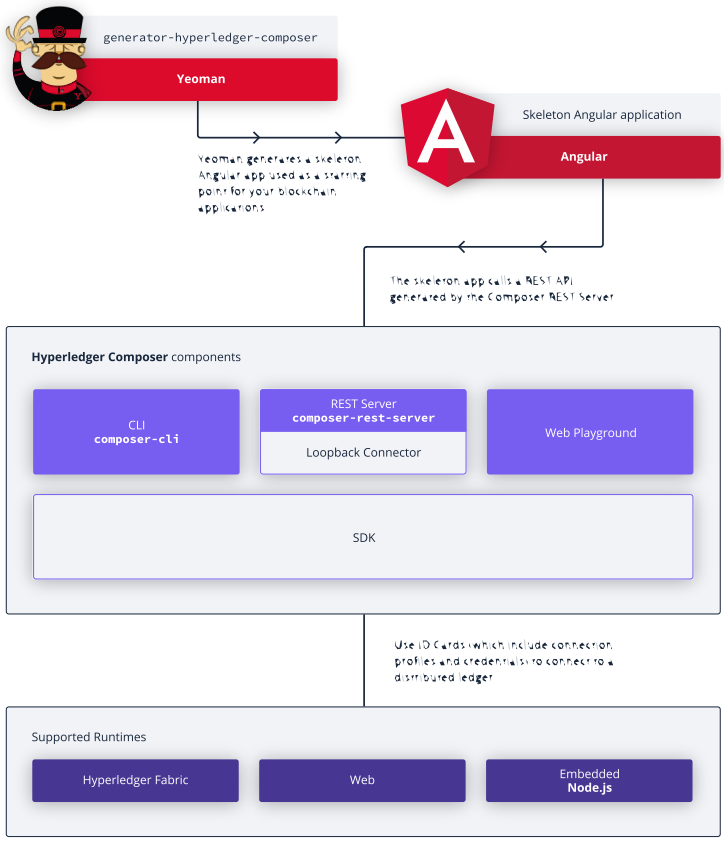


ภาพที่ 2.11 แสดงหน้าที่ของแต่ละไฟล์ใน Hyperledger Composer

ที่มา: https://hyperledger.github.io/composer/latest/introduction/introduction.html

เนื่องจาก Hyperledger Composer เป็นเพียงโครงร่างซอฟต์แวร์ทำให้ไม่สามารถดำเนินการเพียงซอฟต์แวร์เดียวได้ ต้องพึ่งโปรแกรมอื่นๆ ในการช่วยทำงาน (Runtime) ดังนั้น Hyperledger Fabric จึงเป็น Runtime ให้แก่ Hyperledger Composer ในการทำงาน โดย Hyperledger Fabric จะทำหน้าที่ด้านเครือข่ายบล็อกเชน การบันทึกข้อมูล การปรับปรุงแก้ไขข้อตกลงร่วมกันต่างๆ แต่ Hyperledger Composer จะทำหน้าที่เป็นตัวกลางรับคำสั่งจาก Application ผ่าน API รัน Chaincode ของตัวเอง แล้วส่งข้อมูลให้เครือข่ายบล็อกเชนของ Hyperledger Fabric

และเนื่องจากการจะนำไปติดตั้งใน Runtime ของโปรแกรมอื่นจึงต้องทำการรวบรวมไฟล์เป็น Business Network Archive เป็นไฟล์ที่เกิดจากการร่วมกันจาก 4 ไฟล์ข้างต้น โดยไฟล์นี้สามารถนำไปติดตั้งกับซอฟต์แวร์เครือข่าย Blockchain ที่ไหนก็ได้ ซึ่งซอฟต์แวร์ที่สนับสนุนมากที่สุดคือ Hyperledger Fabric ได้



ภาพที่ 2.12 แสดง Architecture ของ Hyperledger Composer

ที่มา: https://hyperledger.github.io/composer/latest/introduction/solution-architecture.html

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 การเก็บรวบรวมข้อมูล

ในส่วนการเก็บรวบข้อมูลที่เกี่ยวข้องกับระบบระบบประวัติการศึกษาของนิสิตบนบล็อกเช ผู้พัฒนาได้ศึกษาและรวบรวมข้อมูลจากเว็บไซต์ระบบประวัติการศึกษาและเว็บไซต์ข้อมูลการศึกษาที่เกี่ยวข้อง ทำแบบสำรวจสอบถามความคิดเห็นและความต้องการของนิสิตในมหาวิทยาลัยและผู้ใช้งานอื่นๆ เพื่อนำข้อมูลมาวิเคราะห์และออกแบบโปรแกรมประยุกต์ให้ตรงตามความต้องการของผู้ใช้งาน

3.2 การเก็บวิเคราะห์ระบบ

จากการศึกษารูปแบบของเว็บไซต์ของมหาวิทยาลัยต่างๆ เช่น RegCu, RegTu เป็นต้น และ วิธีการขอประวัติการศึกษาของนิสิตในปัจจุบัน ทางผู้พัฒนาจึงได้สรุปข้อดีข้อเสียได้ดังนี้

ข้อดีของระบบในปัจจุบัน

มีเอกสารเป็นแผ่นกระดาษของจริง ซึ่งในปัจจุบันระบบต่างๆ ยังรอบรับเพียงเอกสารซึ่งเป็นแผ่นกระดาษเท่านั้น

การทำประวัติต่างๆ ภายในมหาวิทยาลัย มีความรวดเร็วเนื่องจากเป็นระบบภายใน

ข้อเสียของระบบในปัจจุบัน

ในการขอประวัติการศึกษาของนิสิตในแต่ละครั้งเสียทั้งเวลา และค่าใช้จ่าย ในการเดินทางมารับประวัติการศึกษา

ยังไม่มีการแชร์ข้อมูลประวัติการศึกษาของนิสิตกันระหว่างมหาลัยในปัจจุบัน ทำให้การทำงานต่างๆ ล่าช้า

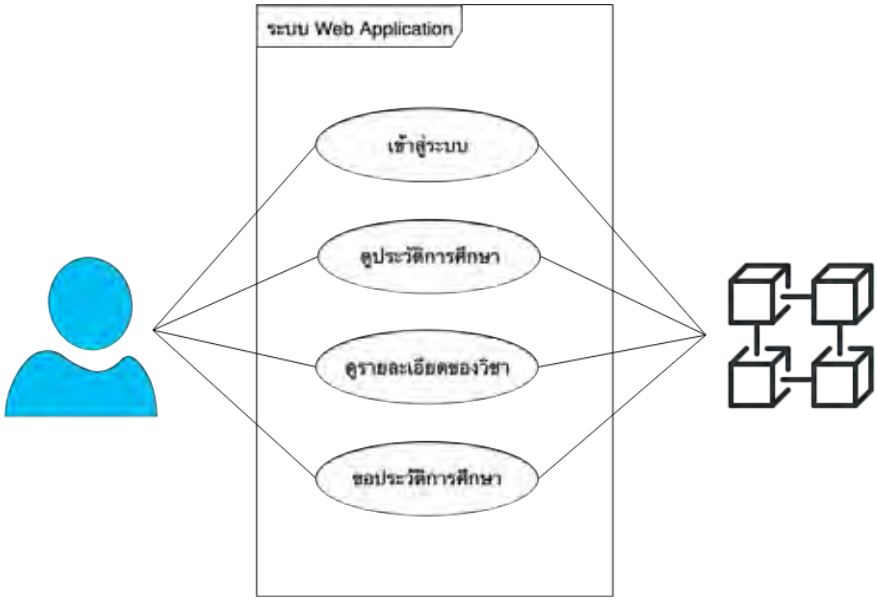
3.3 การกำหนดความสามารถของระบบ

จากการวิเคราะห์ระบบการเก็บและการออกประวัติการศึกษาของนิสิตปัจจุบัน ทำให้ผู้พัฒนาแบ่งการทำงานเป็น ดังนี้

1. สามารถดูข้อมูลประวัติการศึกของนิสิตในปัจจุบันได้
2. สามารถดูว่าวิชาที่เรียนมีหัวข้อการสอนอย่างไร
3. สามารถขอประวัติการศึกษาได้

3.4 การออกแบบระบบ

3.4.1 แผนภาพยูสเคส



ภาพที่ 3.1 แผนภาพยูสเคสของระบบเว็บแอปพลิเคชัน

แผนภาพยูสเคสแสดงให้เห็นว่าระบบ Web Application นั้นเชื่อมกับเครือข่ายบล็อกเชน โดยบล็อกเชนที่ใช้คือ Hyperledger Composer และ Hyperledger Fabric โดยมีฟังก์ชันต่างๆ ประกอบด้วยฟังก์ชันหลัก 4 ฟังก์ชัน ได้แก่ การเข้าสู่ระบบ การดูประวัติการศึกษา การดูรายละเอียดของวิชา และการขอประวัติการศึกษา

Use case name	เข้าสู่ระบบ
Participation Actor	ผู้ใช้งานระบบ
Purpose	เพื่อให้ผู้ใช้สามารถใช้งานฟังก์ชันต่างๆ ในเว็บแอปพลิเคชันนี้ได้
Overview	ผู้ใช้เข้าสู่ระบบเพื่อใช้งานฟังก์ชันต่างๆ ในเว็บแอปพลิเคชันได้
Entry Condition	ผู้ใช้กดเลือก เข้าสู่ระบบ
Flow of Events	1. ผู้ใช้เข้าสู่หน้าเว็บ 2. ผู้ใช้กรอก Username และ Password 3. ผู้ใช้กดปุ่ม “Login”
Exit Condition	ผู้ใช้เข้าสู่ระบบ แสดงหน้า Homepage
Special Requirements	ผู้ใช้งานต้องเป็นสมาชิกของระบบ

ตารางที่ 3.1 แสดงรายละเอียดฟังก์ชัน “เข้าสู่ระบบ”

Use case name	ดูประวัติการศึกษา
Participation Actor	ผู้ใช้งานระบบ
Purpose	เพื่อให้ผู้ใช้สามารถดูประวัติการศึกษาได้
Overview	ผู้ใช้สามารถดูประวัติการศึกษาได้
Entry Condition	ผู้ใช้กดเลือก “ดูประวัติการศึกษา”
Flow of Events	1. ผู้ใช้เข้าสู่หน้าเว็บ 2. ผู้ใช้เข้าสู่ระบบสำเร็จ 3. ผู้ใช้กดปุ่ม “ดูประวัติการศึกษา”
Exit Condition	เว็บแสดงผลประวัติการศึกษา
Special Requirements	ผู้ใช้งานต้องเป็นสมาชิกของระบบ

ตารางที่ 3.2 แสดงรายละเอียดฟังก์ชัน “ดูประวัติการศึกษาการศึกษา”

Use case name	ดูรายละเอียดของวิชา
Participation Actor	ผู้ใช้งานระบบ
Purpose	เพื่อให้ผู้ใช้สามารถดูรายละเอียดของแต่ละวิชาได้
Overview	ผู้ใช้สามารถดูรายละเอียดของแต่ละวิชาได้
Entry Condition	ผู้ใช้กดเลือก “ดูรายละเอียดวิชา”
Flow of Events	1. ผู้ใช้เข้าสู่หน้าเว็บ 2. ผู้ใช้เข้าสู่ระบบสำเร็จ 3. ผู้ใช้กดปุ่ม “ดูรายละเอียดวิชา” 4. ผู้ใช้เลือกวิชาที่ต้องการดูรายละเอียด
Exit Condition	เว็บแสดงผลรายละเอียดของวิชาที่ผู้ใช้เลือก
Special Requirements	ผู้ใช้งานต้องเป็นสมาชิกของระบบ

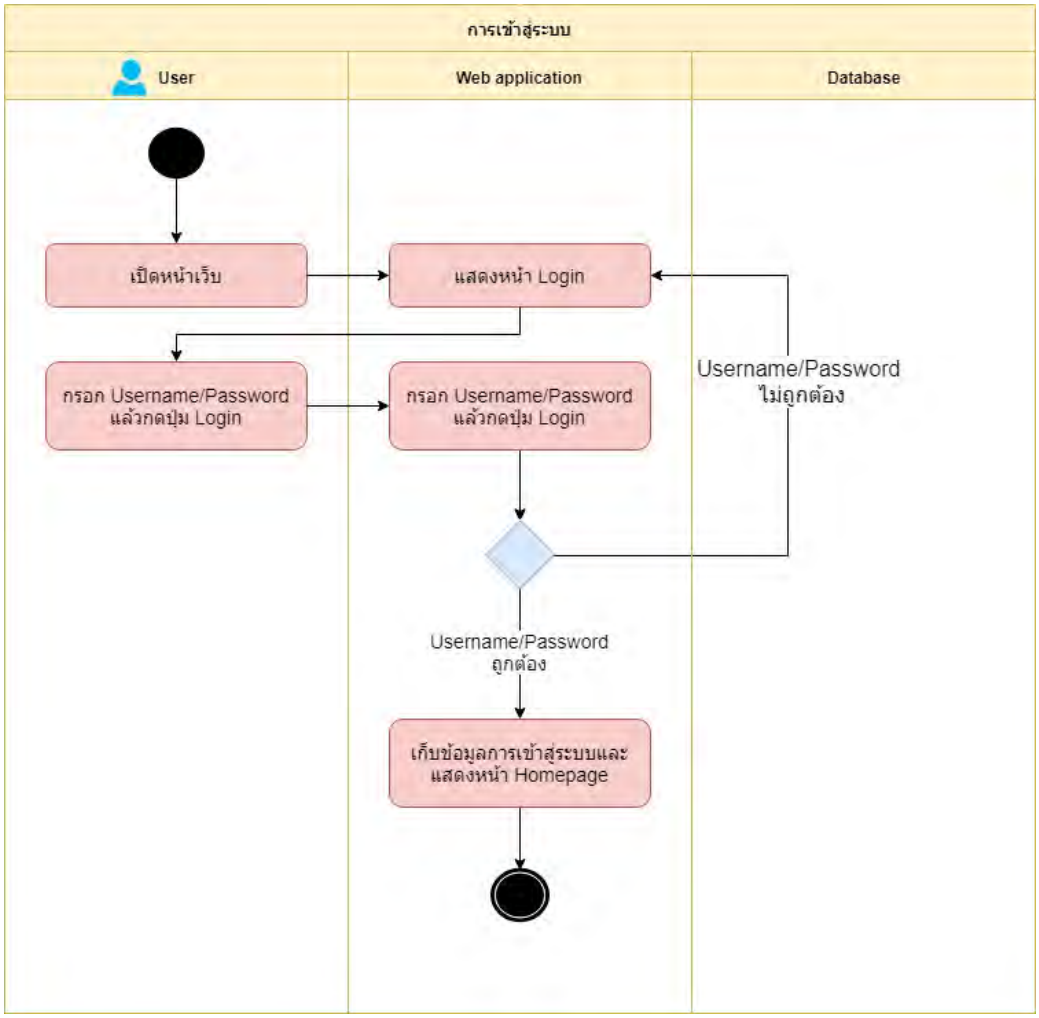
ตารางที่ 3.3 แสดงรายละเอียดฟังก์ชัน “ดูรายละเอียดของวิชา”

Use case name	ขอประวัติการศึกษา
Participation Actor	ผู้ใช้งานระบบ
Purpose	เพื่อให้ผู้ใช้สามารถขอประวัติการศึกษาได้

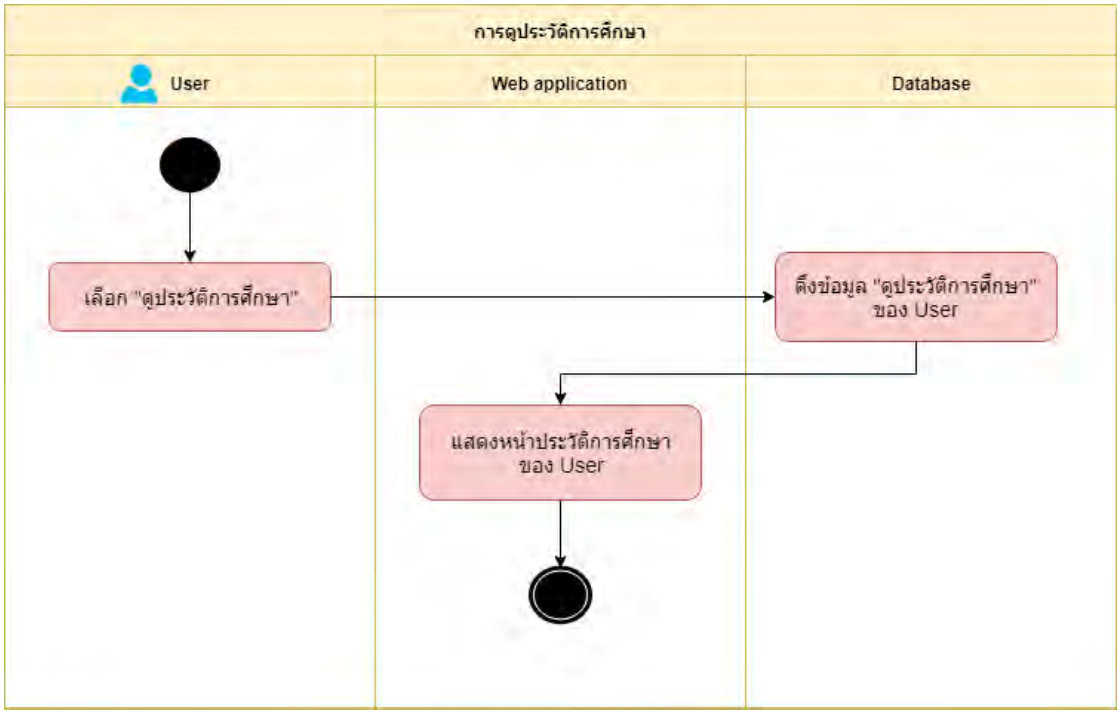
Overview	ผู้ใช้สามารถขอประวัติการศึกษาได้ โดยสามารถดาวน์โหลดออกมาเป็นไฟล์ .pdf
Entry Condition	ผู้ใช้กดเลือก “ขอประวัติการศึกษา”
Flow of Events	1. ผู้ใช้เข้าสู่หน้าเว็บ 2. ผู้ใช้เข้าสู่ระบบสำเร็จ 3. ผู้ใช้กดปุ่ม “ดูประวัติการศึกษา” 4. ผู้ใช้กดปุ่ม “ขอประวัติการศึกษา”
Exit Condition	เว็บแสดงผลประวัติการศึกษาเป็นไฟล์ pdf
Special Requirements	ผู้ใช้งานต้องเป็นสมาชิกของระบบ

ตารางที่ 3.4 แสดงรายละเอียดฟังก์ชัน “ขอประวัติการศึกษา”

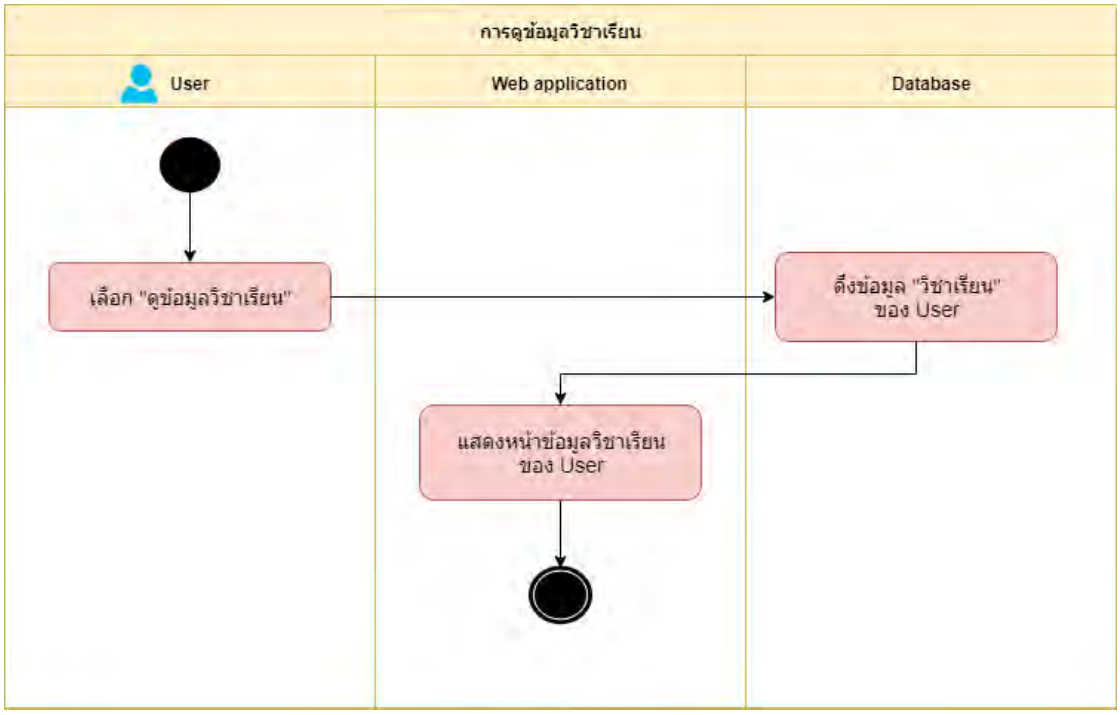
3.4.2 แผนภาพแสดงกิจกรรม



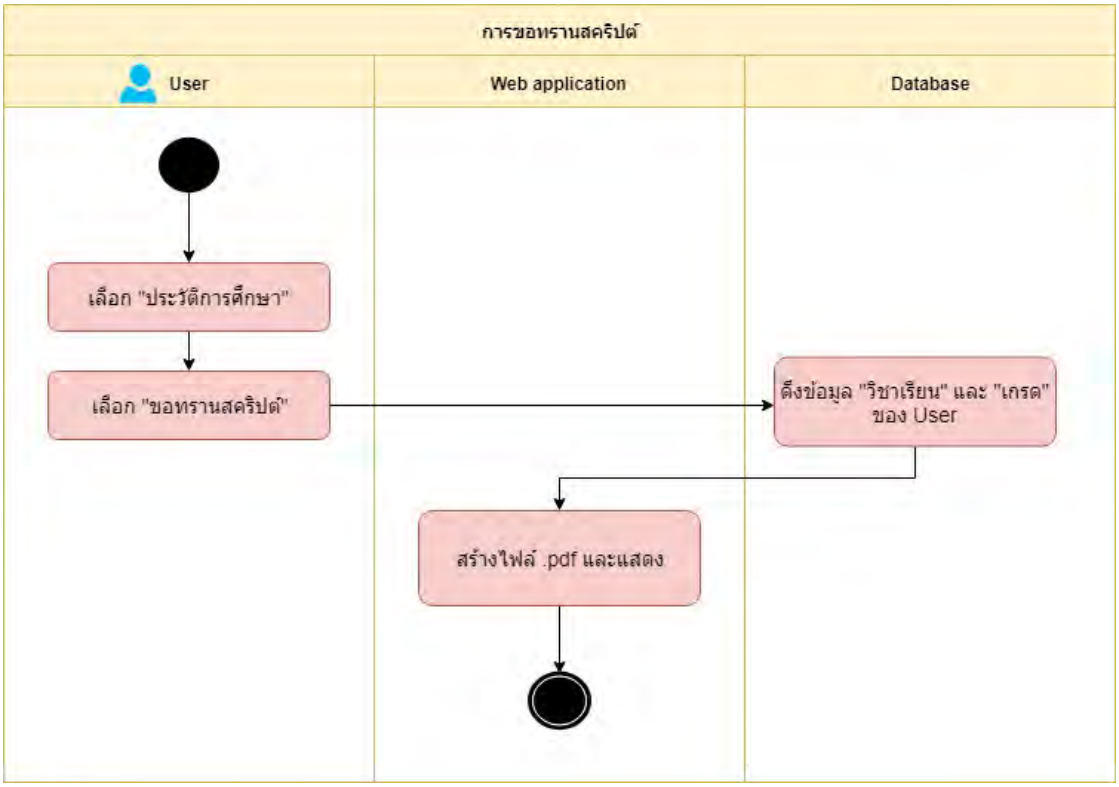
ภาพที่ 3.2 แสดงแผนภาพแสดงกิจกรรมการ “เข้าสู่ระบบ”



ภาพที่ 3.3 แสดงแผนภาพแสดงกิจกรรมการ “ดูประวัติการศึกษา”



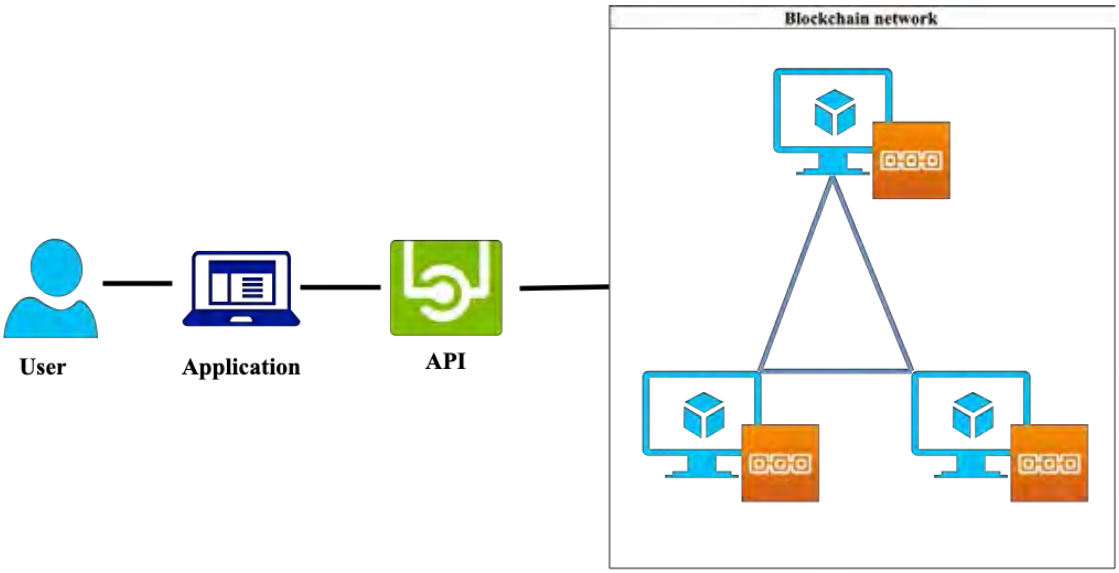
ภาพที่ 3.4 แสดงแผนภาพแสดงกิจกรรมการ “ดูรายละเอียดของวิชา”



ภาพที่ 3.5 แสดงแผนภาพแสดงกิจกรรมการ “ขอประวัติการศึกษา”

3.5 การออกแบบเครือข่ายบล็อกเชน

ในส่วนนี้จะกล่าวถึงแบบจำลองเครือข่ายบล็อกเชน



ภาพที่ 3.6 แบบจำลองเครือข่ายบล็อกเชน

ทางผู้พัฒนาได้ทำการจำลองโหนดภายในเครือข่ายจำนวน 3 โหนด เพื่อให้แสดงถึงจุดเด่นของบล็อกเชนคือการกระจายศูนย์กลางข้อมูล โดยการส่งผ่านข้อมูลจะทำผ่านทาง Application Programming Interface (API) หรือส่วนต่อประสานโปรแกรมประยุกต์ ซึ่งทำหน้าที่ประสานระหว่างเว็บแอปพลิเคชันกับเครือข่ายบล็อกเชน โดย Hyperledger Composer มี LoopBack ซึ่งเป็น Open Source ในการสร้างและจัดการ API เมื่อทำการติดตั้ง Hyperledger Composer จะมี API จาก LoopBack สร้างให้อัตโนมัติ โดยสามารถเชื่อม Web Application และเครือข่าย ผ่าน API นี้ได้เลย

3.6 การออกแบบส่วนตัวประสานผู้ใช้

ในส่วนนี้จะกล่าวถึงการออกแบบส่วนต่อประสานผู้ใช้งาน

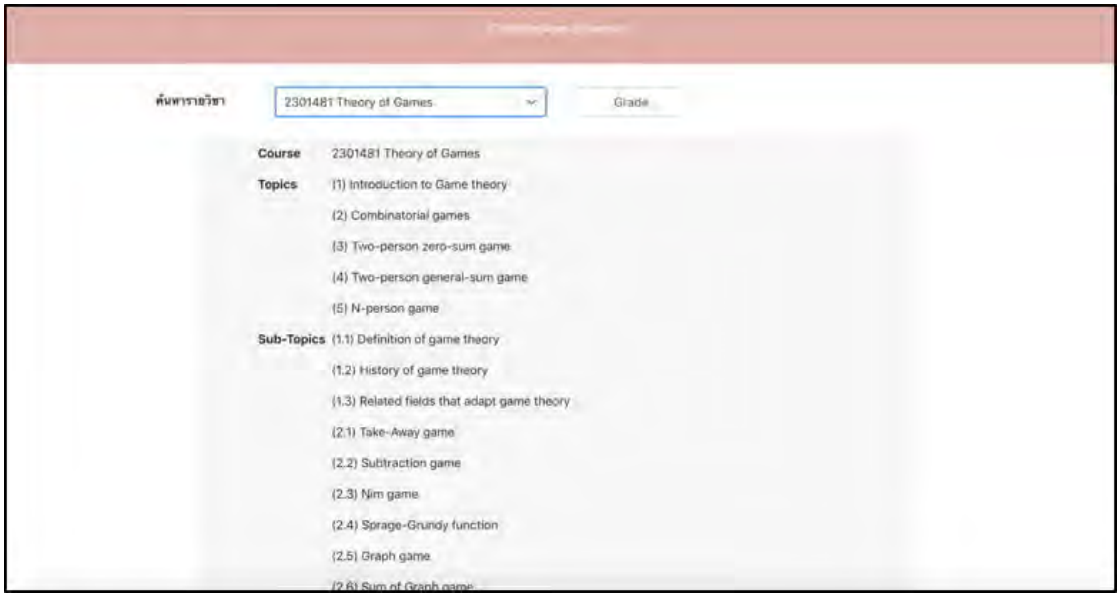
หน้าล็อคอิน

ภาพที่ 3.7 หน้าจอล็อคอินเข้าสู่ระบบ

หน้าเว็บหลัก

ภาพที่ 3.8 หน้าเว็บไซต์หลัก

หน้าแสดงรายละเอียดวิชา

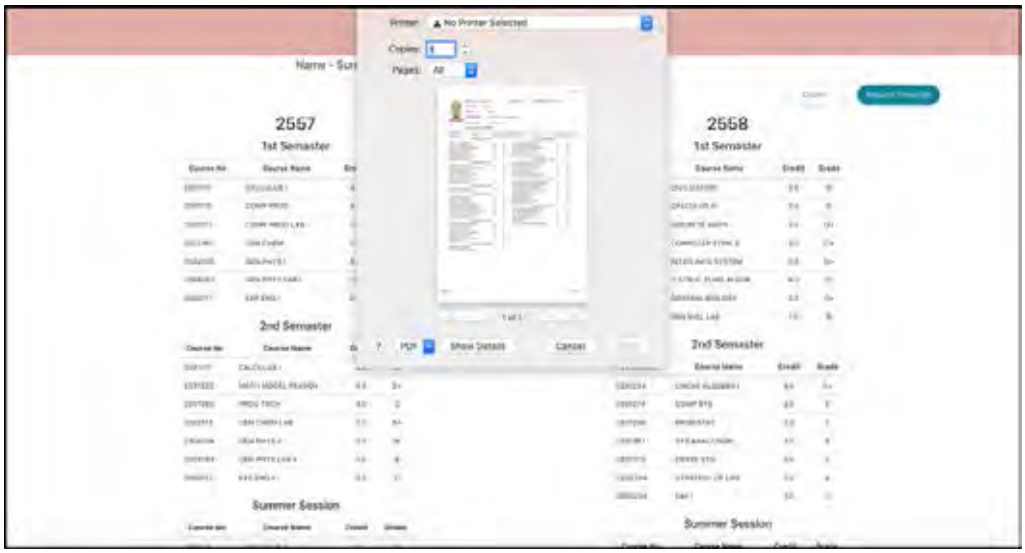


ภาพที่ 3.9 หน้าแสดงรายละเอียดวิชา

หน้าแสดงผลการศึกษา

ภาพที่ 3.10 หน้าแสดงผลการศึกษา

หน้าแสดงไฟล์ตัวอย่างเอกสารสำคัญทางการศึกษา



ภาพที่ 3.11 หน้าแสดงไฟล์ตัวอย่างเอกสารสำคัญทางการศึกษา

บทที่ 4
เทคนิคการเขียนโปรแกรมและการทดสอบระบบ

เพื่อให้ระบบทำงานได้อย่างมีประสิทธิภาพและสามารถทำงานได้ตรงตามความต้องการมากที่สุด การทดสอบจึงเป็นขั้นตอนที่มีความสำคัญ เพื่อเป็นการปรับปรุงคุณภาพให้ดีขึ้น โดยการหาข้อผิดพลาดที่อาจเกิดขึ้นกับระบบ ซึ่งในบทนี้จะกล่าวถึงขอบเขตการทดสอบระบบ การวางแผนการทดสอบ วิธีการทดสอบและผลของการทดสอบ

4.1 เทคนิคการเขียนโปรแกรม

ในการพัฒนาระบบบล็อกเชน ทางผู้พัฒนาเลือกใช้ Hyperledger Composer ในการออกแบบบล็อกเชน

1. Model File

Model File ใช้ภาษา Hyperledger Composer Modeling Language ซึ่งเป็นภาษาที่เป็นแบบ Object-oriented ใช้ในการออกแบบวิธีการเก็บข้อมูลต่างๆ ให้อยู่ในรูปแบบที่ต้องการ โดยจะเก็บเป็น Key-Value Pair การออกแบบจะออกแบบให้แต่ละวัตถุ (Object) แยกออกจากกันอย่างชัดเจน เพื่อให้ง่ายต่อการเข้าใจและการพัฒนาแก้ไข

```
//Participants
participant Student identified by studentId {
  o String studentId
  o String idNumber
  o String firstName
  o String lastName
  o String birthDay
  o String telNumber
  o String email
  o String address
  o String status
}
```

ภาพที่ 4.1 การออกแบบการเก็บข้อมูลนักเรียน

การเก็บข้อมูลนักเรียนจะเป็นการเก็บข้อมูลทั่วไปของนักเรียนได้แก่

- studentId = รหัสประจำตัวนักเรียน
- idNumber = รหัสประจำตัวประชาชน
- firstName = ชื่อจริง
- lastName = นามสกุล

- birthDay = วันเกิด
- telNumber = เบอร์โทรศัพท์
- email = อีเมล
- address = ที่อยู่
- status = สถานะของนักเรียน

```
asset Topic identified by topicId{
  o String topicId
  o String topicName
  o String subTopic
}
```

ภาพที่ 4.2 การออกแบบการเก็บข้อมูลหัวข้อวิชา

การเก็บหัวข้อวิชานั้น ทางผู้พัฒนาได้แยกออกมาเป็นวัตถุหนึ่ง เพื่อง่ายต่อการแก้ไขในภายภาคหน้า เนื่องจากทางผู้พัฒนาคิดว่าเนื้อหาในแต่ละวิชาจะเปลี่ยนแปลงตามยุคสมัย

- topicId = รหัสประจำตัวหัวข้อวิชา
- topicName = ชื่อหัวข้อวิชา
- subTopic = รายวิชาย่อยในหัวข้อวิชานี้

```
asset Subject identified by subjectId {
  o String subjectId
  o String nameSubject
  --> Topic[] topics
}
```

ภาพที่ 4.3 การออกแบบการเก็บข้อมูลรายวิชา

การเก็บข้อมูลรายวิชานั้น ทางผู้พัฒนาได้แยกออกมาเป็นวัตถุหนึ่งเช่นกัน เนื่องจากเหตุผลที่ว่าวิชามีการเปลี่ยนแปลงตามยุคสมัยนั้นๆ

- subjectId = รหัสประจำตัวรายวิชา
- nameSubject = ชื่อรายวิชา
- topics = หัวข้อวิชาในรายวิชานี้ เก็บเป็น Object อ้างอิงโดยใช้ topicId ในการเข้าถึงข้อมูลของเนื้อหาหัวข้อวิชานั้นๆ

```
//asset
asset Class identified by classId {
  o String classId
  o String classYear
  o String instructorName
  o String classSchedule
  o String numberStudent
  o String classGPA
  o String[] project
  o String[] learningTool
  --> Subject subject
  --> Subject[] preSubject //prerequisite subjects
}
```

ภาพที่ 4.4 การออกแบบการเก็บข้อมูลชั้นเรียน

การเก็บข้อมูลชั้นเรียนเป็นการเก็บข้อมูลวิชาเรียนของปีนั้นๆ โดยมีข้อมูลได้แก่

- classId = รหัสประจำตัวชั้นเรียน
- classYear = ปีของชั้นเรียนนั้นๆ เช่น 2019
- instructureName = ชื่อผู้สอน
- classSchedule = ช่วงเวลาของชั้นเรียน
- numberStudent = จำนวนนักเรียนในชั้นเรียนนี้
- classGPA = เกรดเฉลี่ยของนักเรียนในชั้นเรียนนี้
- project = โครงงานที่จะมีในชั้นเรียนนี้
- learningTool = เครื่องมือที่ใช้ในการเรียนชั้นเรียนนี้
- subject = วิชาเรียนในชั้นเรียนนี้ โดยอ้างอิงจาก subjectID ในการเข้าถึงข้อมูล
- preSubject = วิชาเรียนที่ต้องผ่านมาก่อนเรียนชั้นเรียนนี้ โดยอ้างอิงจาก subjectID ในการเข้าถึงข้อมูล

```
concept SubjectGrade {
  --> Class studiedClass
  o String grade
  o String semester
  o String collegeYear
}

asset Transcript identified by transcriptId {
  o String transcriptId
  o String studentGPA
  --> Student student
  o SubjectGrade[] subjectGrades optional
}
```

ภาพที่ 4.5 การออกแบบการเก็บข้อมูลใบแสดงผลการศึกษา

การเก็บข้อมูลใบแสดงผลการศึกษาได้ทำการเก็บแยกข้อมูลเป็น 2 ส่วนคือ เกรดในแต่ละรายวิชา และ ใบแสดงผลการศึกษาของนักเรียน

ข้อมูลเกรดในแต่ละวิชา

- studiedClass = ชั้นเรียนที่ได้เรียนผ่านมาแล้ว โดยจะอ้างอิงจาก classId ในการเข้าถึงข้อมูล
- grade = เกรดที่ได้รับในชั้นเรียน เช่น A B+ B เป็นต้น รวมถึง S/W V/U ด้วย
- semester = ภาคการศึกษาที่ได้เรียนชั้นเรียนนั้นๆ เช่น ภาคการศึกษาต้น ภาคการศึกษาปลาย ภาคการศึกษาฤดูร้อน เป็นต้น
- collegeYear = ปีที่ได้เรียนชั้นเรียน เช่น ปี 1 ปี 2 เป็นต้น

ข้อมูลใบแสดงผลการศึกษา

- transcriptId = รหัสประจำตัวใบแสดงผลการศึกษา
- studentGPA = เกรดเฉลี่ยของนักเรียน
- student = นักเรียนเจ้าของใบแสดงผลการศึกษา โดยจะอ้างอิงจาก studentId ในการเข้าถึงข้อมูล
- subjectGrades = เกรดของวิชา

```
//transaction
transaction RecordGrade {
  --> Transcript transcript
  o SubjectGrade subjectGrade
}

transaction CreateStudentInfo {
  o String id
  o String firstName
  o String lastName
}

transaction UpdateStudentInfo {
  --> Student student
  o String firstName optional
  o String lastName optional
}
```

ภาพที่ 4.6 การออกแบบรายการบัญชีในระบบ (Function)

การออกแบบรายการบัญชี เป็นการกำหนดชื่อในการเรียกใช้ทรัพยากรที่จำเป็นในการดำเนินการเท่านั้น ส่วนการกำหนดวิธีการของรายการบัญชีจะอยู่ใน Script File

2. Script File

Script File เป็น JavaScript File ใช้เขียนฟังก์ชันที่ประกาศใน Model File ให้สามารถทำงานตามที่ต้องการ

```
async function recordGrade(tx) {
    // Save the old value of the asset.
    const transcript = tx.transcript;

    if(transcript.subjectGrades){
        transcript.subjectGrades.push(tx.subjectGrade);
    } else {
        transcript.subjectGrades = [tx.subjectGrade];
    }

    // Get the asset registry for the asset.
    const assetRegistry = await getAssetRegistry('org.tp.transc2.Transcript');
    // Update the asset in the asset registry.
    await assetRegistry.update(tx.transcript);
}
```

ภาพที่ 4.7 การออกแบบฟังก์ชันการบันทึกเกรด

ฟังก์ชันบันทึกเกรดจะรับค่าจากรายการบัญชีที่ถูกส่งมาโดยมีค่าที่ต้องการเป็นข้อมูล subjectGrade ทั้งหมดและวัตถุ Transcipt ที่จะอ้างอิงในการบันทึกเกรดลงไป เมื่อได้รับข้อมูลทั้งหมดอย่างถูกต้อง ก็จะนำวัตถุ subjectGrades บันทึกลง Transcript ด้วยวิธีการบันทึกลงตัวแปรข้อมูลแบบลำดับ

```
async function createStudentInfo(tx) {

    const factory = getFactory();
    const NS = 'org.tp.transc2'

    const student = factory.newResource(NS, 'Student', tx.id);
    student.firstName = tx.firstName;
    student.lastName = tx.lastName;

    const transcript = factory.newResource(NS, 'Transcript', tx.id);
    transcript.student = factory.newRelationship(NS, 'Student', tx.id);

    const studentRegistry = await getParticipantRegistry(NS + '.Student');
    await studentRegistry.addAll([student]);

    const transcriptRegistry = await getAssetRegistry(NS + '.Transcript');
    await transcriptRegistry.addAll([transcript]);
}
```

ภาพที่ 4.8 การออกแบบฟังก์ชันสร้างใบแสดงผลการศึกษาเมื่อสร้างนักเรียน

ฟังก์ชันสร้างใบแสดงผลการศึกษาเมื่อสร้างนักเรียนสร้างขึ้นเพื่อให้นักเรียนที่อยู่ในระบบมีใบแสดงผลการศึกษาติดตัว ไม่ว่าจะมีผลการศึกษาแล้วหรือยังก็ตาม โดยเมื่อมีการ

สร้างผู้เข้าร่วมนักเรียนผ่านฟังก์ชันนี้ ระบบก็จะสร้างใบแสดงผลการศึกษาที่มีนักเรียนเป็นเจ้าของเลย

```
async function updateStudentInfo(tx) {
    const student = tx.student;

    if(tx.firstName !== "") {
       student.firstName = tx.firstName;
    }
    if(tx.lastName !== "") {
       student.lastName = tx.lastName;
    }

    const assetRegistry = await getParticipantRegistry('org.tp.transaction.Student');
    await assetRegistry.update(student);
}
```

ภาพที่ 4.9 การออกแบบฟังก์ชันการแก้ไขข้อมูลชื่อนักเรียน

ฟังก์ชันการแก้ไขข้อมูลชื่อนักเรียนมีขึ้นเพื่อในกรณีที่นักเรียนมีการเปลี่ยน ชื่อหรือนามสกุล จึงจำเป็นต้องแก้ไขข้อมูลในระบบ โดยรับค่าชื่อและนามสกุล แล้วนำไปบันทึกลงในค่าของนักเรียน

3. Access Control

Access Control เป็นไฟล์ที่ใช้สำหรับการควบคุมการเข้าถึงของสินทรัพย์ในระบบ โดยกำหนดว่า ผู้เข้าร่วมคนไหนมีสิทธิมากแค่ไหน ไฟล์นี้จะมีหรือไม่มีก็ได้

```
rule SystemACL {
  description:  "System ACL to permit all access"
  participant: "org.hyperledger.composer.system.Participant"
  operation: ALL
  resource: "org.hyperledger.composer.system.**"
  action: ALLOW
}
```

ภาพที่ 4.10 ตัวอย่างการเขียนไฟล์ Access Control

4. Query File

Query File เป็นไฟล์ที่ใช้ในการเขียนฟังก์ชันเรียกขอข้อมูล (Query) แต่เมื่อทำการเริ่มการทำงาน ทาง Hyperledger Composer จะทำการสร้างคำสั่ง Query พื้นฐานต่างๆ ของไฟล์โมเดลไว้ให้แล้ว ซึ่งก็เพียงพอต่อการใช้งานที่ไม่ซับซ้อน

```
query selectTranscript {
  description: "Select Student Transcript"
  statement:
      SELECT org.tp.transc.Transcript
      WHERE (student == _$student)
}
```

ภาพที่ 4.11 ตัวอย่างการ Query นักเรียนจากนักเรียนที่กำหนด

4.2 ขอบเขตการทดสอบ (Scope of Testing)

การทดสอบระบบเว็บแอปพลิเคชันและระบบการเก็บข้อมูลประวัติการศึกษาบนบล็อกเชนมีจุดประสงค์ระบบสามารถทำงานตามฟังก์ชันที่กำหนดได้เช่น การเข้าสู่ระบบ การดูประวัติการศึกษา การดูข้อมูลวิชา และการขอประวัติการศึกษา โดยการทดสอบระบบจะทดสอบบนเว็บเบราว์เซอร์ 3 โปรแกรมดังนี้

1. Google Chrome
2. Microsoft Edge
3. Mozilla Firefox

4.3 การทดสอบโดยผู้พัฒนา

ผู้พัฒนาได้ทำการตรวจสอบตามฟังก์ชันการทำงานของระบบว่าทำงานได้ถูกต้องตามที่กำหนดหรือไม่ โดยมีรายละเอียดดังต่อไปนี้

ตารางที่ 4.1 ตารางแสดงกรณีทดสอบและผลการทดสอบ

วิธีการทดสอบ	ผลที่คาดหวัง	ผลการทดสอบ
การเข้าสู่ระบบ	เมื่อผู้ใช้งานเข้าสู่ระบบได้สำเร็จจะไปสู่หน้า Homepage	ผ่าน
การดูประวัติการศึกษาของผู้ใช้งาน	ระบบแสดงประวัติการศึกาของผู้ใช้งานได้อย่างถูกต้อง	ผ่าน
การแสดงรายละเอียดของวิชา	ระบบแสดงรายละเอียดของวิชาได้อย่างถูกต้อง	ผ่าน
การขอประวัติการศึกษาของผู้ใช้งาน	ระบบแสดงประวัติการศึกษาออกมาในรูปแบบไฟล์ PDF และผู้ใช้งานสามารถดาวน์โหลดไปใช้ได้	ผ่าน
การออกจากระบบ	เมื่อผู้ใช้งานกดปุ่มออกจากระบบ ระบบจะพาไปสู่หน้า Login	ผ่าน

4.4 การทดสอบโดยผู้ใช้งาน

การทดสอบระบบบล็อกเชนและระบบเว็ปแอปพลิเคชันมีจุดประสงค์ให้ระบบสามารถทำงานในส่วนของการแสดงข้อมูลของนิสิตและการขอทรานสคริปต์ โดยจะให้นิสิตที่เรียนอยู่ในคณะวิทยาศาสตร์ ภาควิชาวิทยาการคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัยเป็นผู้ทดสอบระบบเท่านั้น

ตารางที่ 4.1 ผลการทดสอบของผู้ใช้งาน

หัวข้อ	ระดับความพึงพอใจ (คน)					ค่าเฉลี่ย
	ดีมาก (5)	ดี (4)	ปานกลาง (3)	พอใช้ (2)	ควรปรับปรุง (1)	
1. เว็บแอปพลิเคชันใช้งานง่าย	5	12	3	0	0	4.1
2. เว็บแอปพลิเคชันแสดงข้อมูลถูกต้องครบถ้วน	6	10	4	0	0	4.1
3. เว็ปแอพลิเคชันแสดงใบประวัติการศึกษาได้ถูกต้อง	5	14	1	0	0	4.2
4. ความสะดวกในการขอใบประวัติการศึกษา	1	18	1	0	0	3.9
5. เว็บแอปพลิเคชันมีความสวยงามมีความชัดเจน อ่านง่าย	3	17	0	0	0	4.15
6.ความพึงพอใจโดยรวมต่อเว็ปแอพลิเคชัน	5	11	4	0	0	4.05
					ค่าเฉลี่ย	4.08

สรุปผลการทดสอบ

จากตารางที่ 4.2 พบว่าเว็บแอปพลิเคชันอยู่ในระดับดี ด้วยคะแนนเฉลี่ย 4.08 และพบว่าผู้ใช้พึงพอใจที่จะใช้เว็บแอปพลิเคชัน

บทที่ 5
ข้อสรุปและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงการสรุปผลทีได้จากการพัฒนาระบบเว็ปแอปพลิเคชันสำหรับการขอทรานสคริปต์ในมหาวิทยาลัย ปัญหาและอุปสรรคที่พบขณะดำเนินงาน วิธีการแก้ไขปัญหา และข้อเสนอแนะเพื่อเป็นแนวทางในการพัฒนาและปรับปรุงระบบเว็ปแอปพลิเคชันให้มีความสามารถเพิ่มเติมต่อไปในอนาคต

5.1 ข้อสรุป

ผู้พัฒนาสามารถพัฒนาระบบประวัติการศึกษาของนิสิตบนบล็อกเชน ซึ่งสามารถแก้ปัญหาเรื่องความซับซ้อนในการดำเนินการขอทรานสคริปต์ได้ ทำให้ผู้ใช้งานทุกคนเกิดความสะดวกและประหยัดเวลาในการดำเนินการ

5.2 ปัญหาและอุปสรรค

1. การวางแผนระยะเวลาในการพัฒนามีความผิดพลาดบ้าง บางขั้นตอนไม่เสร็จตรงตามแผนที่ได้วางไว้ จึงต้องมีการปรับปรุงแผนการทำงานอยู่เสมอ
2. ผู้พัฒนาไม่มีประสบการณ์ในการพัฒนาคอมพิวเตอร์แม่ข่ายโดยใช้ภาษา JavaScript และ React มาก่อน ทำให้ใช้เวลาในการเริ่มต้นค่อนข้างมากเนื่องจากต้องศึกษาการพัฒนาใหม่ทั้งหมด

5.3 วิธีการแก้ปัญหา

1. ต้องหมั่นตรวจสอบแผนการดำเนินการอยู่เสมอ และปรับเปลี่ยนให้เข้ากับสถานการณ์ที่เปลี่ยนไป
2. ต้องศึกษาหาข้อมูลจากเว็บไซต์ รวมถึงข้อมูลจากแหล่งต่าง ๆ ให้มากที่สุดเท่าที่จะหาได้

5.4 ข้อเสนอแนะ

ระบบประวัติการศึกษาของนิสิตบนบล็อกเชนนี้ได้พัฒนาขึ้นภายใต้ขอบเขตของโครงงาน ทำให้ยังมีอีกหลายส่วนที่ต้องพัฒนา โดยสามารถนำไปพัฒนาต่อโดยใช้เครื่องคอมพิวเตอร์แม่ข่ายจริง และทำงานผ่านระบบออนไลน์จริงเพื่อตอบสนองความต้องการของผู้ใช้เว็บให้ดียิ่งขึ้น

รายการอ้างอิง

[1] THE LINUX FOUNDATION, “Hyperledger Fabric”.[Online]. Available from: https://www.hyperledger.org/projects/fabric[2018,September 13]

[2] Manav Gupta. (2018). Blockchain for Dummies. 2nd. Hoboken : John Wiley & Sons, Inc.

[3] Lucas Marx, “Storing Data on the Blockchain: The Developers Guide”.[Online]. Available from: https://malcoded.com/posts/storing-data-blockchain [2018,September 13]

[4] Pacharama,“การสรุปJavaScript”.[Online]. Available from: https://medium.com/open-source-technology/สรุปพื้นฐาน-javascript-ec02f18cfe47 [2018,June 5]

[5]ChaiPhonbopit,“React”.[Online]. Available from: https://devahoy.com/posts/getting-started-with-reactjs/ [2015, November 29]

[6]nuuneoi,“Blockchain คืออะไร? อธิบายแบบละเอียด แต่เข้าใจง่าย(มั้ง)”.[Online]. Available from: https://nuuneoi.com/blog/blog.php?read_id=900 [2016, December 21]

[7]nuuneoi,“Blockchain for Geek ... เบื้องหลังการทำงานฉบับ Technical ตัวอย่างจาก BitcoinOnline”. Available from: https://nuuneoi.com/blog/blog.php?read_id=901 [2016, December 26]

[8]nuuneoi,“ทำความรู้จัก Blockchain Consensus Protocol แบบต่าง ๆ: เมื่อโลกนี้ไม่ได้มีแค่ Proof of Work”. Available from: https://nuuneoi.com/blog/blog.php?read_id=933 [2018, March 8]

[9]THE LINUX FOUNDATION, “Hyperledger Composer”.[Online]. Available from:

https://hyperledger.github.io/composer/latest/introduction/introduction.html[2019,April 25]

[10] VEEDEVIL,“ทำความเข้าใจเทคโนโลยี Blockchain จุดเปลี่ยนของหลายอุตสาหกรรม”.[Online]. Available from: http://www.veedvil.com/news/blockchain/[2017, March 8]

[11]The Khaeng,“การทำงาน Blockchain ในระบบ Proof of work”.[Online]. Available from: https://medium.com/dcen/blockchain-pow-system-7a722f4b8261[2018, August 29]

ภาคผนวก

ภาคผนวก ก
แบบเสนอหัวข้อโครงงาน รายวิชา 2301399 Project Proposal
ปีการศึกษา 2561

ชื่อโครงงาน (ภาษาไทย)	ระบบประวัติการศึกษาของนิสิตบนบล็อกเชน	
ชื่อโครงงาน (ภาษาอังกฤษ)	An Education Record System on Blockchain	
อาจารย์ที่ปรึกษา	อาจารย์ ดร.อาธร เหลืองสดใส	
ผู้ดำเนินการ	1. นายทัชนนท์ กุมมาลือ	เลขประจำตัวนิสิต 5833630023
	2. นางสาวณัฏฐมณฑน์ กุลรัตนรักษ์	เลขประจำตัวนิสิต 5833619123
	สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย	

หลักการและเหตุผล

บล็อกเชนเป็นวิธีการบันทึกข้อมูลแบบหนึ่ง โดยจะเก็บข้อมูลเหล่านี้อยู่ในแต่ละบล็อก ที่นำมาเชื่อมต่อกันเป็นลูกโซ่ โดยตรวจสอบความถูกต้องและรักษาความปลอดภัยโดยการเข้ารหัส แต่ละบล็อกจะมีค่าแฮชของบล็อกก่อนหน้าซึ่งใช้ยืนยันความถูกต้องของบล็อกก่อนหน้า บล็อกเชนจึงเป็นระบบการบันทึกข้อมูลที่ไม่สามารถแก้ไขได้อีกทั้งสามารถบันทึกข้อมูลแบบกระจายโดยสามารถยืนยันได้ว่าข้อมูลทุก ๆ ที่นั้นเหมือนกัน

จากลักษณะการทำงานของบล็อกเชน จึงเหมาะที่จะนำมาทำระบบต่าง ๆ เช่น การทำระบบเลือกตั้ง การสร้างเอกสารต่าง ๆ การบันทึกประวัติทางการแพทย์ การบันทึกธุรกรรมทางการเงิน การโอนที่ดินโดยไม่ผ่านสำนักงานแต่ผ่านระบบบล็อกเชนที่มีการรับรองแล้วจากกรมที่ดิน เป็นต้น

ในระบบมหาวิทยาลัย การดำเนินงานหลายๆ อย่างในมหาวิทยาลัยจำเป็นจะต้องมีเอกสารรับรอง เช่น การขอใช้พื้นที่ การขอเบิกงบโครงการต่าง ๆ เป็นต้น และระบบต่าง ๆ ในมหาวิทยาลัยจำเป็นต้องใช้เอกสารมากมาย เช่น การลงทะเบียนประวัติ การลงทะเบียนวิชา การเพิ่ม/ลด/ถอนรายวิชา การออกใบทรานสคริปต์ เป็นต้น ซึ่งต้องใช้เวลาและเสียค่าใช้จ่ายต่าง ๆ มากมาย ผู้จัดทำจึงสนใจที่จะพัฒนาระบบบล็อกเชนเพื่อลดขั้นตอนในการยื่นเอกสารต่าง ๆ ให้สามารถกระทำผ่านระบบออนไลน์ได้โดยที่มีความถูกต้องและเชื่อถือได้ สามารถตรวจสอบข้อมูล และมีความปลอดภัย อีกทั้งผู้จัดทำเห็นว่าเทคโนโลยีบล็อกเชนสามารถเชื่อมระบบระหว่ามหาวิทยาลัยและสถาบันการศึกษาอื่น ๆ ให้เชื่อมต่อกันได้ เช่น การแชร์

ข้อมูลของนักศึกษาระหว่างมหาวิทยาลัยและแหล่งเรียนรู้ออนไลน์ ถ้าการเรียนรู้ผ่านทางออนไลน์มีมาตรฐาน เราสามารถให้หน่วยกิจผ่านทางออนไลน์ได้ ซึ่งจะช่วยพัฒนาระบบการศึกษาให้ดีขึ้น ลดการใช้เอกสาร เวลาที่ใช้ในการรับส่งและและเวลาที่ใช้ในการตรวจสอบ

ผู้จัดได้ทำการเลือกแพลตฟอร์ม Hyperledger ซึ่งเป็น Permissioned Blockchain ซึ่งมีลักษณะที่เหมาะกับระบบนี้ ในการเก็บรักษาข้อมูลและการเข้าถึงหรือเข้าร่วมบล็อกเชนจำเป็นจะต้องได้รับอนุญาตก่อน และแพลตฟอร์มนี้สามารถเขียน Smart Contract ซึ่งจะช่วยในการออกใบแสดงผลการศึกษาโดยอัตโนมัติผ่านการร้องขอของนักศึกษาได้เลย ดังนั้นการยื่นใบแสดงผลการศึกษาให้สถานที่ทำงานต่าง ๆ จะง่ายและรวดเร็วยิ่งขึ้น

วัตถุประสงค์

เพื่อพัฒนาระบบของมหาวิทยาลัยโดยใช้เทคโนโลยีบล็อกเชนที่สามารถเก็บข้อมูลเอกสารต่าง ๆ ของมหาวิทยาลัย และสามารถออกเอกสารต่าง ๆ จากข้อมูลในระบบบล็อกเชนได้

ขอบเขตของโครงงาน

1. สามารถบันทึกข้อมูลการลงทะเบียนเรียนของนักศึกษามหาวิทยาลัยได้ เช่น รหัสวิชา ชื่อวิชา เป็นต้น
2. สามารถบันทึกข้อมูลของนักศึกษาได้โดยข้อมูลที่จะเก็บ เช่น ชื่อนักศึกษา วันเดือนปีเกิด ที่อยู่ เบอร์ติดต่อ เป็นต้น
3. สามารถบันทึกหัวข้อต่าง ๆในแต่ละวิชาได้ ซึ่งวิชาแต่ละวิชาในมหาวิทยาลัยอาจมีชื่อเหมือนกันแต่หัวข้อในแต่ละวิชาไม่เหมือนกัน
4. สามารถออกใบแสดงผลการศึกษาได้
5. สามารถเชื่อมระบบกันระหว่างมหาวิทยาลัยและสถาบันการศึกษาอื่น ๆ เช่น Coursera Udemy เป็นต้น

วิธีการดำเนินงาน

1. ศึกษาเอกสารและงานวิจัยต่าง ๆ ที่เกี่ยวข้องกับระบบมหาวิทยาลัยและบล็อกเชน
2. ศึกษาเครื่องมือที่ใช้ในการพัฒนาระบบบล็อกเชน
3. วิเคราะห์และออกแบบระบบ
4. พัฒนาระบบ
5. ตรวจสอบความถูกต้องของระบบและแก้ไขข้อผิดพลาด
6. จัดทำเอกสาร

ตารางเวลาการดำเนินงาน

ขั้นตอนการดำเนินงาน	2018					2019			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	ม.ค.	เม.ย.
1. ศึกษาเอกสารและงานวิจัยต่างๆ ที่เกี่ยวข้องกับระบบมหาวิทยาลัยและบล็อกเชน	■	■							
2. ศึกษาเครื่องมือที่ใช้ในการพัฒนาระบบบล็อกเชน		■	■						
3. วิเคราะห์และออกแบบระบบ			■	■					
4. พัฒนาระบบบล็อกเชน				■	■	■	■		
5. ตรวจสอบความถูกต้องของระบบและแก้ไขข้อผิดพลาด					■	■	■	■	
6. จัดทำเอกสาร							■	■	■

ประโยชน์ที่คาดว่าจะได้รับ

1. ประโยชน์ต่อผู้พัฒนา
    - ได้ศึกษาเทคโนโลยีบล็อกเชน ขั้นตอนการพัฒนาและการนำไปประยุกต์ใช้
    - ได้พัฒนาทักษะในการวิเคราะห์ระบบ
    - ได้ฝึกทักษะการทำงานเป็นกลุ่ม
2. ประโยชน์ต่อผู้ใช้
    - มหาวิทยาลัยสามารถนำระบบไปเป็นตัวต้นแบบในการพัฒนาระบบต่อไปได้
    - นักศึกษาสามารถขอข้อมูลจากทางมหาวิทยาลัยได้โดยข้อมูลมีความถูกต้องและตรวจสอบได้

อุปกรณ์และเครื่องมือที่ใช้

1. ฮาร์ดแวร์
    - คอมพิวเตอร์

- Processor: Intel(R) Core(TM) i5-8400U CPU @ 2.80 GHz
- RAM: 16 GB
- OS: Microsoft Windows 10 Pro

- ยูเอสบีแฟลชไดรฟ์ SanDisk Ultra Dual Drive USB Type-C
  - Memory: 64 GB

2. ซอฟต์แวร์
   - Microsoft Office 2013
   - Docker
   - Hyperledger Fabric Platform
   - Visual Studio Code
3. วัสดุสำนักงาน
   - การดาษถ่ายเอกสารขนาด A4
   - อุปกรณ์จัดทำรูปเล่มรายงาน

งบประมาณ

1.	อุปกรณ์สำนักงาน	2,000.00 บาท
2.	Virtual Private Server (VPS)	3,000.00 บาท
3.	SanDisk Extreme Portable SSD 250G	3,190.00 บาท
4.	อุปกรณ์เสริมคอมพิวเตอร์ต่างๆ เช่น เม้าส์ คีย์บอร์ด	1,800.00 บาท
	รวม	9,990.00 บาท

เอกสารอ้างอิง

[1] THE LINUX FOUNDATION, “Hyperledger Fabric”.[Online]. Available from: https://www.hyperledger.org/projects/fabric[2018,September 13]

[2] Manav Gupta. (2018). Blockchain for Dummies. 2nd. Hoboken : John Wiley & Sons, Inc.

[3] Lucas Marx, “Storing Data on the Blockchain: The Developers Guide”.[Online]. Available from: https://malcoded.com/posts/storing-data-blockchain [2018,September 13]

ภาคผนวก ข
คู่มือการใช้งาน

เมื่อเข้าสู่เว็บไซต์จะพบกับหน้าล็อคอินซึ่งในหน้านี้ผู้ใช้งานจะต้องกรอกอีเมลและรหัสผ่าน และกดปุ่ม Login เพื่อทำการเข้าสู่เว็บไซต์

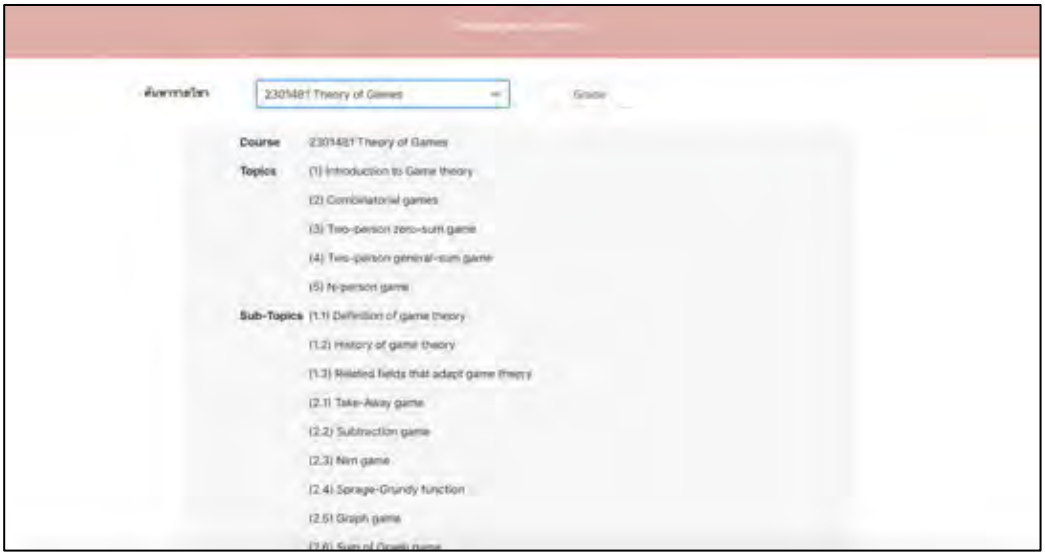
หน้าล็อคอินเข้าสู่เว็บไซต์

หลังจาดเข้าสู่เว็บไซต์เรียบร้อยแล้วจะพบกับหน้าหลักของเว็บไซต์ มีเมนูให้เลือก 2 ปุ่ม คือ “ดูรายละเอียด” และ “ผลการศึกษา”

หน้าหลักของเว็บไซต์

หากกดที่ปุ่ม “ดูรายละเอียด” จะพบกับหน้าที่ให้ผู้ใช้งานสามารถค้นหาชื่อรายวิชา และเมื่อค้นหาไปแล้วจะแสดงชื่อวิชา หัวข้อที่เรียนในรายวิชา รวมถึงรายวิชาย่อยที่เรียนในวิชานั้น ๆ และยังมีปุ่ม “Grade”

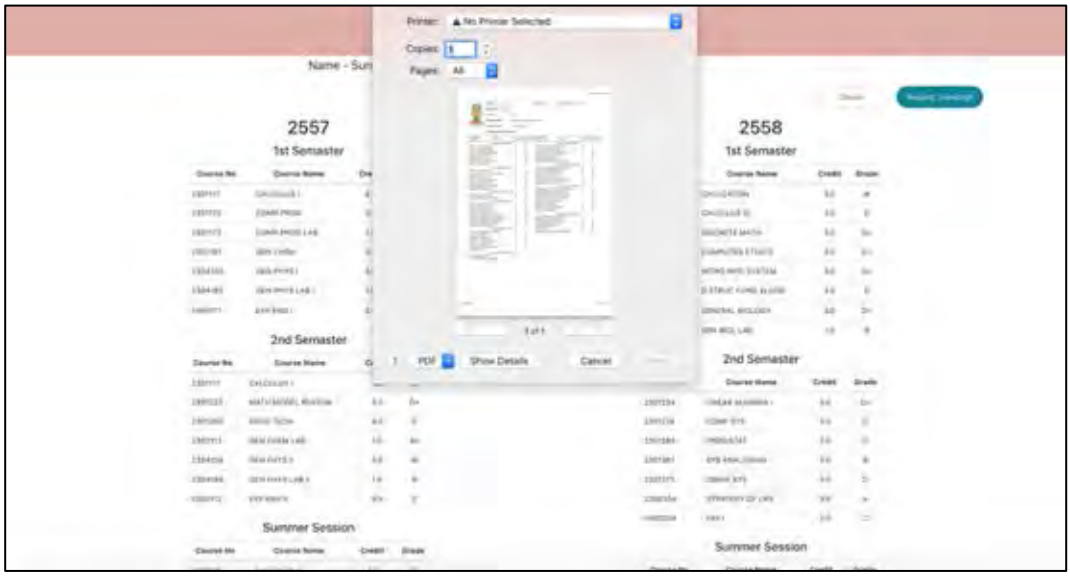
หน้าค้นหารายวิชา



หน้าต่างแสดงรายละเอียดของวิชานั้น ๆ

หน้าแสดงผลการศึกษา

จากหน้าหลักของเว็บไซต์ เมื่อกดปุ่ม “ผลการศึกษา”จะไปสู่หน้าแสดงผลการศึกษา ในหน้าแสดงผลการศึกษาจะมีปุ่ม “Course” ซึ่งจะนำไปยังหน้าค้นหารายวิชา ส่วนปุ่ม “Request Transcript” เมื่อกดเข้าไปจะขึ้นเป็นไฟล์ PDF มาให้สามารถบันทึก ส่งต่อ หรือพิมพ์ ออกมาได้



หน้าที่ขึ้นให้บันทึกหรือพิมพ์ผลการศึกษา

ประวัติผู้เขียน

นายทัชนนท์ กุมมาลือ

ประวัติ

เกิด วันที่ 25 เมษายน 2540

สถานที่เกิด จังหวัดกรุงเทพมหานคร

เชื้อชาติ ไทย สัญชาติ ไทย

การศึกษา สาขาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

อีเมล: tatchanon.k@gmail.com

นางสาวณัฏฐมณฑน์ กุลรัตนรักษ์

ประวัติ

เกิด วันที่ 28 ตุลาคม 2539

สถานที่เกิด จังหวัดกรุงเทพมหานคร

เชื้อชาติ ไทย สัญชาติ ไทย

การศึกษา สาขาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

อีเมล: ployseob@gmail.com