



รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบจัดการฐานข้อมูลสำหรับการตั้งค่าแอปพลิเคชัน K PLUS Configuration System for K PLUS Application

นางสาวนันทิกานต์ อวະกุลพานิชย์

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา ระบบจัดการฐานข้อมูลสำหรับการตั้งค่าแอปพลิเคชัน K PLUS

ชื่อ-สกุล นักศึกษา นางสาวนันทิกานต์ อาวะกุลพาณิชย์

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมคอมพิวเตอร์

ชื่อ-สกุล อาจารย์นิเทศ ผศ.บัณฑิต พัสยา

ชื่อ-สกุล ผู้นิเทศงาน นายประดิษฐ์ แก้วบริสุทธิ

ชื่อสถานประกอบการ กลิกร บิซิเนส เทคโนโลยี กรุ๊ป เซเครเทรียต จำกัด

บทคัดย่อ

โครงการนี้มีจุดประสงค์เพื่อทำการพัฒนาระบบเพิ่มเติมที่อยู่ภายใต้ระบบจัดการบัตรเดบิตและทำการปรับปรุงเว็บแอปพลิเคชัน (Web Application) โดยทำหน้าที่จัดการฐานข้อมูลสำหรับการตั้งค่าแอปพลิเคชัน K Plus ให้มีความสวยงาม ใช้งานง่าย และเป็นมิตรกับผู้ใช้งาน (User Friendly) มากยิ่งขึ้น และสามารถทำการเพิ่ม แก้ไข และลบข้อมูลภายในได้ รวมทั้งยังง่ายต่อผู้พัฒนาระบบ (Developer) และผู้ทดสอบระบบ (Tester) ที่จะสามารถตรวจสอบข้อมูลได้อย่างรวดเร็วหากเกิดข้อผิดพลาดใดๆ เกิดขึ้น

ในการพัฒนาเว็บแอปพลิเคชันในครั้งนี้ได้นำเอาหลักการวัฏจักรการพัฒนาซอฟต์แวร์ (Software Development Life Cycle: SDLC) เข้ามาใช้ โดยมีทั้งหมด 6 ขั้นตอน ได้แก่ การวางแผน, การวิเคราะห์ความต้องการ, การออกแบบ, การพัฒนาซอฟต์แวร์, การทดสอบและการนำไปใช้ และการบำรุงรักษาซอฟต์แวร์ โดยเว็บแอปพลิเคชันนี้ได้มีการแบ่งออกเป็น 2 เว็บไซต์ เว็บไซต์แรกได้ทำการพัฒนาระบบเพิ่มเติมด้วย ReactJS และ Java เว็บไซต์ที่สองนั้นได้ทำการพัฒนาเว็บไซต์ขึ้นใหม่จากเว็บไซต์แรกให้มีความสวยงามและใช้งานง่ายขึ้น ด้วย ReactJS, Typescript และ Golang และในระหว่างที่ทำการพัฒนาเว็บแอปพลิเคชันนั้น ได้มีการนำหลักการการทำงานแบบอไจล์ (Agile) เข้ามาใช้ภายในทีม

เว็บแอปพลิเคชันนี้สามารถช่วยให้ผู้ใช้งาน ผู้พัฒนาระบบ และผู้ทดสอบระบบ จัดการข้อมูลภายในระบบแอปพลิเคชันได้ง่ายขึ้น

คำสำคัญ : เว็บแอปพลิเคชัน วัฏจักรการพัฒนาซอฟต์แวร์ หลักการทำงานแบบอไจล์

Co-operative Title: Configuration System for K PLUS Application

Student Intern Name: Ms. Nanthikan Arwakulpanich

Faculty: Engineering

Department: Computer Engineering

Advisor Name: Asst. Prof. Bundit Pasaya

Mentor Name: Mr. Pradist Keawborisut

Company: Kasikorn Business-Technology Group

ABSTRACT

This project aims to implement additional system under debit management system and remodel K Plus's configuration system to more ease to use, more user friendly, able to add, edit and delete internal data and be convenient for developer and tester to quick verify when system is defective.

This development use Software Development Life Cycle: SDLC that has 6 processes include planning, analysis, design, implementation, testing and integration, maintenance. This application has 2 parts include first website that's implemented with ReactJS and Java, second website that's remodeled from first website to more ease to use with ReactJS, Typescript and Golang. During development, we use agile methodology with development team.

This web application able to make more ease to use for user, developer and tester who manage internal data inside application system.

Keywords: Web Application, Software Development Life Cycle, Agile

กิตติกรรมประกาศ

โครงการสหกิจศึกษานี้สำเร็จลุล่วงได้ด้วยความกรุณาจากหลายท่าน ซึ่งได้กรุณาให้การสนับสนุนและให้ความช่วยเหลือแก่ผู้จัดทำตลอดระยะเวลาที่ได้ทำโครงการฉบับนี้

ขอกราบขอบพระคุณบริษัทกลีกร บิซิเนส เทคโนโลยี กรุ๊ป เซक्टरเทรียต จำกัด ที่ให้โอกาสแก่ผู้จัดทำได้เข้ามาปฏิบัติงานสหกิจศึกษา และพัฒนาฝีมือ และขอขอบคุณนายประดิษฐ์ แก้วบริสุทธิ์, นายธนภัทร สุนทรลิมศิริ, นายจิตรกร ยี่สุนเทศ, นางสาวชนกานต์ ขอบรส และ นางสาวกฤษติญา บุญธีรนนท์ ที่ให้การดูแล และฝึกสอนงานตลอดระยะเวลาที่ได้ปฏิบัติงาน

ขอกราบขอบคุณอาจารย์บัณฑิต พัสยา และดร.จิระศักดิ์ สิทธิกร ซึ่งเป็นอาจารย์นิเทศโครงการสหกิจศึกษาที่ได้ให้ความอนุเคราะห์ในการให้คำปรึกษา และคอยสอบถามความคืบหน้าของโครงการสหกิจศึกษาฉบับนี้

ขอกราบขอบคุณคณาจารย์คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่าน ที่ได้กรุณาสั่งสอน ถ่ายทอด อบรมความรู้แก่ผู้จัดทำ ซึ่งทำให้ผู้จัดทำสามารถนำความรู้เหล่านั้นมาประยุกต์ใช้กับโครงการชิ้นนี้

ขอกราบขอบคุณพี่และเพื่อนทุกคนที่ได้มีส่วนช่วยให้ข้อคิดเห็น แลกเปลี่ยนความรู้ ตลอดจนให้กำลังใจแก่ผู้จัดทำเป็นอย่างดีเสมอมา

สุดท้ายนี้ขอกราบขอบพระคุณพระบิดา มารดา และสมาชิกครอบครัวทุกท่าน ที่ได้อบรม สั่งสอน เลี้ยงดูให้ได้รับการศึกษาที่ดี รวมทั้งให้การสนับสนุนและกำลังใจมาโดยตลอด จนสามารถประสบความสำเร็จในการจัดทำโครงการสหกิจศึกษาชิ้นนี้

นนทิกานต์ อวระกุลพาณิชย์

สารบัญ

บทที่	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	V
สารบัญรูปภาพ.....	VI
บทที่ 1 บทนำ.....	1
1.1. ความเป็นมาและความสำคัญ.....	1
1.2. วัตถุประสงค์ของโครงการ.....	2
1.3. ขอบเขตของโครงการ.....	2
1.4. วิธีการดำเนินงาน.....	2
1.5. ประโยชน์ที่คาดว่าจะได้รับ.....	4
บทที่ 2 ทฤษฎีและการวิจัย.....	5
2.1. ทฤษฎีที่เกี่ยวข้อง.....	5
บทที่ 3 วิธีการดำเนินงาน.....	20
3.1. ขั้นตอนการดำเนินงาน.....	20
3.2. สถาปัตยกรรมของระบบ.....	21
3.3. การออกแบบ.....	22
บทที่ 4 ผลการวิจัย.....	42
4.1. ภาพรวมของระบบ.....	42
4.2. ผลการพัฒนาระบบ.....	49
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	51
5.1. สรุปผลการดำเนินงาน.....	51
5.2. ประโยชน์ที่ได้รับจากการปฏิบัติงานสหกิจศึกษา.....	51
5.3. ปัญหาและอุปสรรคที่พบ.....	52
5.4. ข้อเสนอแนะ.....	52
บรรณานุกรม.....	53

สารบัญตาราง

ตารางที่	หน้า
3.1 รายละเอียดของข้อมูลในตาราง Campaign	30
3.2 รายละเอียดของข้อมูลในตาราง Category	30
3.3 รายละเอียดของข้อมูลในตาราง Card.....	31
3.4 รายละเอียดของข้อมูลในตาราง Category Order.....	31
3.5 รายละเอียดของข้อมูลในตาราง Card Order	32
3.6 รายละเอียดของข้อมูลในตาราง Card Fee.....	32
3.7 รายละเอียดของข้อมูลในตาราง Map Branch Address Card	32
3.8 รายละเอียดของข้อมูลในตาราง Branch Address	33
3.9 รายละเอียดของข้อมูลในตาราง Message.....	34
4.1 ตารางผลการพัฒนาระบบ.....	49

สารบัญรูปภาพ

ภาพที่	หน้า
ภาพที่ 1.1 เว็บไซต์จัดการฐานข้อมูลสำหรับการตั้งค่าแอปพลิเคชัน K PLUS แบบเก่า.....	1
ภาพที่ 1.2 เว็บไซต์จัดการฐานข้อมูลสำหรับการตั้งค่าแอปพลิเคชัน K PLUS แบบใหม่.....	2
ภาพที่ 1.3 วัฏจักรการพัฒนาซอฟต์แวร์	3
ภาพที่ 2.1 HTML5	5
ภาพที่ 2.2 CSS3	6
ภาพที่ 2.3 SASS	7
ภาพที่ 2.4 Preprocessor	7
ภาพที่ 2.5 กระบวนการแปลง SASS เป็น CSS	8
ภาพที่ 2.6 React.....	8
ภาพที่ 2.7 JavaScript.....	9
ภาพที่ 2.8 TypeScript.....	10
ภาพที่ 2.9 Golang	10
ภาพที่ 2.10 Java.....	11
ภาพที่ 2.11 Ant Design	12
ภาพที่ 2.12 Material UI.....	12
ภาพที่ 2.13 SQL Server.....	13
ภาพที่ 2.14 ความสัมพันธ์ระหว่าง Model View และ Presenter.....	14
ภาพที่ 2.15 ขั้นตอนการทำ Scrum.....	15
ภาพที่ 2.16 REST API.....	16
ภาพที่ 2.17 ตัวอย่างตารางภายใน Relational Database.....	18
ภาพที่ 2.18 Web Application.....	18
ภาพที่ 2.19 UX Design	19
ภาพที่ 3.1 สถาปัตยกรรมของเว็บแอปพลิเคชันตัวเก่า.....	21
ภาพที่ 3.2 สถาปัตยกรรมของเว็บแอปพลิเคชันตัวใหม่.....	22
ภาพที่ 3.3 การออกแบบระบบเพิ่มเติมภายในเว็บแอปพลิเคชันตัวเก่าโดยใช้โปรแกรม Figma.....	23
ภาพที่ 3.4 หน้าเว็บสำหรับระบบจัดการบัตรเดบิตที่ถูกเพิ่มเข้ามาในเว็บไซด์ตัวเก่า.....	23
ภาพที่ 3.5 หน้าเว็บสำหรับระบบจัดการหมวดหมู่ของบัตรเดบิตที่ถูกเพิ่มเข้ามาในเว็บไซด์ตัวเก่า.....	24

สารบัญรูปภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 3.6 หน้าเว็บสำหรับระบบจัดการแคมเปญของบัตรเดบิตที่ถูกเพิ่มเข้ามาในเว็บไซต์ตัวเก่า	24
ภาพที่ 3.7 การออกแบบเว็บแอปพลิเคชันตัวใหม่โดยใช้โปรแกรม Figma	25
ภาพที่ 3.8 หน้าเว็บสำหรับจัดการข้อความที่แสดงอยู่บนแอปพลิเคชัน K Plus	25
ภาพที่ 3.9 หน้าเว็บสำหรับจัดการข้อมูลของบัตรเดบิตภายในแอปพลิเคชัน K Plus.....	26
ภาพที่ 3.10 หน้าเว็บสำหรับจัดการหมวดของบัตรเดบิต	26
ภาพที่ 3.11 หน้าเว็บสำหรับจัดการหมวดหมู่ของบัตรเดบิตให้รวมอยู่เป็นแคมเปญ.....	27
ภาพที่ 3.12 Use case Diagram ของเว็บแอปพลิเคชัน Back Office.....	28
ภาพที่ 3.13 แผนภาพโครงสร้างฐานข้อมูลของ Debit.....	29
ภาพที่ 3.14 แผนภาพโครงสร้างฐานข้อมูลของ Message	33
ภาพที่ 3.15 Flowchart การทำงานเมื่อมีการเรียกดูข้อมูล	35
ภาพที่ 3.16 Flowchart แสดงการทำงานเมื่อผู้ใช้ทำการสร้างข้อมูล	36
ภาพที่ 3.17 Flowchart แสดงการทำงานเมื่อผู้ใช้ทำการแก้ไขข้อมูล	37
ภาพที่ 3.18 Flowchart แสดงการทำงานเมื่อผู้ใช้ทำการลบข้อมูล.....	38
ภาพที่ 3.19 Flowchart แสดงการทำงานเมื่อผู้ใช้งานทำการจัดการบัตรเดบิต (ส่วนที่ 1).....	39
ภาพที่ 3.20 Flowchart แสดงการทำงานเมื่อผู้ใช้งานทำการจัดการบัตรเดบิต (ส่วนที่ 2).....	40
ภาพที่ 3.21 Flowchart แสดงการทำงานเมื่อผู้ใช้งานทำการจัดการบัตรเดบิต (ส่วนที่ 3).....	41
ภาพที่ 4.1 ระบบจัดการบัตรเดบิตในเว็บแอปพลิเคชันตัวเก่า.....	42
ภาพที่ 4.2 ระบบจัดการหมวดหมู่ของบัตรเดบิตในเว็บแอปพลิเคชันตัวเก่า.....	43
ภาพที่ 4.3 ระบบจัดการแคมเปญของบัตรเดบิตในเว็บแอปพลิเคชันตัวเก่า	43
ภาพที่ 4.4 ระบบดูรายละเอียดภายในแคมเปญของบัตรเดบิตในเว็บแอปพลิเคชันตัวเก่า	44
ภาพที่ 4.5 ระบบดูรายละเอียดภายในบัตรเดบิตในเว็บแอปพลิเคชันตัวใหม่.....	45
ภาพที่ 4.6 ระบบดูรายละเอียดภายในหมวดหมู่ของบัตรเดบิตในเว็บแอปพลิเคชันตัวใหม่	45
ภาพที่ 4.7 ระบบดูรายละเอียดภายในแคมเปญของบัตรเดบิตในเว็บแอปพลิเคชันตัวใหม่	46
ภาพที่ 4.8 ระบบจัดการข้อมูลภายในแคมเปญของบัตรเดบิตในเว็บแอปพลิเคชันตัวใหม่	46
ภาพที่ 4.9 ภาพเปรียบเทียบของหน้าจัดการข้อความภายในเว็บแอปพลิเคชันตัวเก่า	47
ภาพที่ 4.10 ภาพเปรียบเทียบของหน้าจัดการข้อความภายในเว็บแอปพลิเคชันตัวใหม่	47
ภาพที่ 4.11 ภาพเปรียบเทียบของหน้าจัดการเมนูภายในเว็บแอปพลิเคชันตัวเก่า	48

สารบัญรูปร่าง (ต่อ)

ภาพที่

หน้า

ภาพที่ 4.12 ภาพเปรียบเทียบของหน้าจัดการเมนูภายในเว็บแอปพลิเคชันตัวใหม่48



บทที่ 1

บทนำ

1.1. ความเป็นมาและความสำคัญ

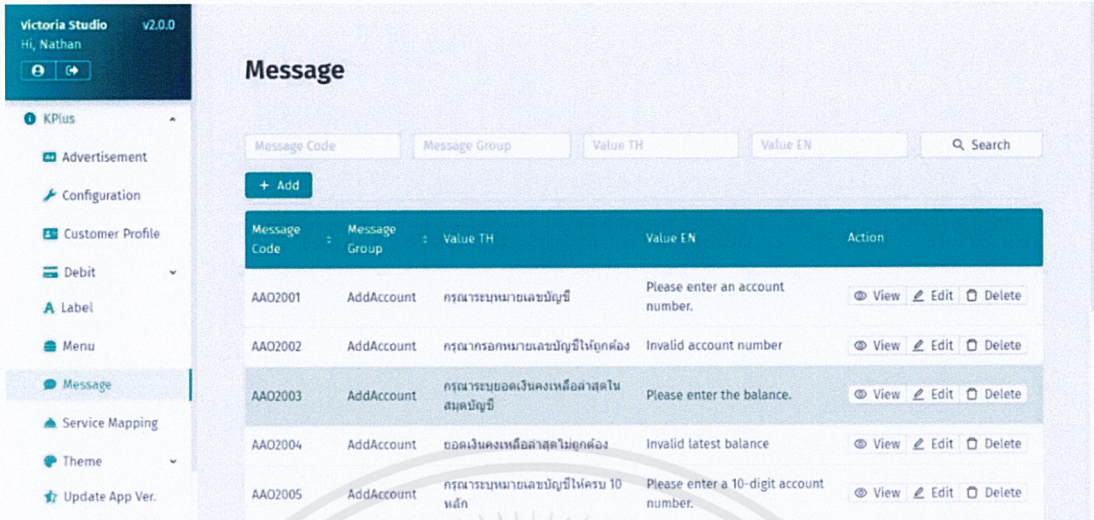
เนื่องจากแอปพลิเคชัน K Plus mobile banking เป็นแอปพลิเคชันที่ทำงานบนระบบแอนดรอยด์ (Android) และ บนระบบไอโอเอส (IOS) ดังนั้นหากมีส่วนที่จำเป็นจะต้องทำการแก้ไข เช่น การแก้ไขจำนวนเมนูที่แสดง แก้ไขข้อความของเมนูต่างๆ จัดการแก้ไขธีมต่างๆ หรือ จัดการการแสดงผลโฆษณาภายในแอปพลิเคชัน จะทำได้ยากลำบากเนื่องจากการแก้ไขส่วนต่างๆ จะเกิดตาม requirement ที่ได้รับมาจาก BU จึงทำให้ตัว BU เอง ไม่สามารถทำการแก้ไขได้ด้วยตัวเองได้ ดังนั้นหากต้องการแก้ไขจึงต้องทำการพึ่ง Developer เพื่อให้แก้ไขในส่วนต่างๆ และถ้าหากทำการแก้ไขภายใน code โดยตรง อาจทำให้เกิดข้อผิดพลาดที่อาจส่งผลกระทบต่อผู้ใช้งานได้

ดังนั้นเว็บไซต์ Back Office จึงเป็นซอฟต์แวร์ที่ให้บริการเกี่ยวกับการจัดการระบบภายในของแอปพลิเคชัน K Plus เช่น การจัดการส่วนการแสดงผลข้อความบนเมนู การแสดงผลโฆษณา หรือ การกำหนดเวลาการใช้งานของธีม เป็นต้น โดยเว็บไซต์ Back Office ถูกออกแบบมาเพื่อรองรับการใช้งานสำหรับพนักงานภายใน โดยเน้นไปที่ฝ่าย Business User เพื่อให้ง่ายต่อการใช้งาน หรือปรับเปลี่ยนระบบต่างๆภายในของแอปพลิเคชันได้อย่างง่ายดาย โดยไม่จำเป็นที่จะต้องมีการเข้ามาช่วยในการแก้ไข แต่สามารถแก้ไขได้ด้วยตนเอง และยังทำให้ Develop และ Business analysis สามารถตรวจสอบแก้ไขได้ง่ายขึ้นหากมีข้อผิดพลาดของการแสดงผลเกิดขึ้น



ภาพที่ 1.1 เว็บไซต์จัดการฐานข้อมูลสำหรับการตั้งค่าแอปพลิเคชัน K PLUS แบบเก่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 1.2 เว็บไซต์จัดการฐานข้อมูลสำหรับการตั้งค่าแอปพลิเคชัน K PLUS แบบใหม่

1.2. วัตถุประสงค์ของโครงการ

- 1.1.1. เพื่อทำการเพิ่มระบบการจัดการในส่วนของบัตรเครดิตบนเว็บไซต์เก่า
- 1.1.2. เพื่อทำการออกแบบแก้ไขเว็บไซต์ให้มีความสวยงามมากขึ้น
- 1.1.3. เพื่อทำการออกแบบแก้ไขเว็บไซต์ให้มีการใช้งานที่ง่ายขึ้นและเป็นมิตรต่อผู้ใช้งานมากขึ้น

1.3. ขอบเขตของโครงการ

- 1.1.4. ข้อมูลที่แสดงบนเว็บไซต์นี้เป็นข้อมูลจำลองเพื่อความเข้าใจเท่านั้น ไม่ใช่ข้อมูลจริงของบริษัท
- 1.1.5. เว็บไซต์นี้สามารถใช้ได้แค่ในเครือข่ายอินเทอร์เน็ตของบริษัทเท่านั้น ไม่สามารถนำมาใช้งานข้างนอกได้
- 1.1.6. เว็บไซต์ตัวใหม่ที่ถูกออกแบบใหม่ขึ้นมาแล้วยังไม่สามารถนำไปใช้งานได้ทันที เนื่องจากยังมีส่วนที่จะต้องทดสอบ และทำการปรับปรุงอีกครั้ง

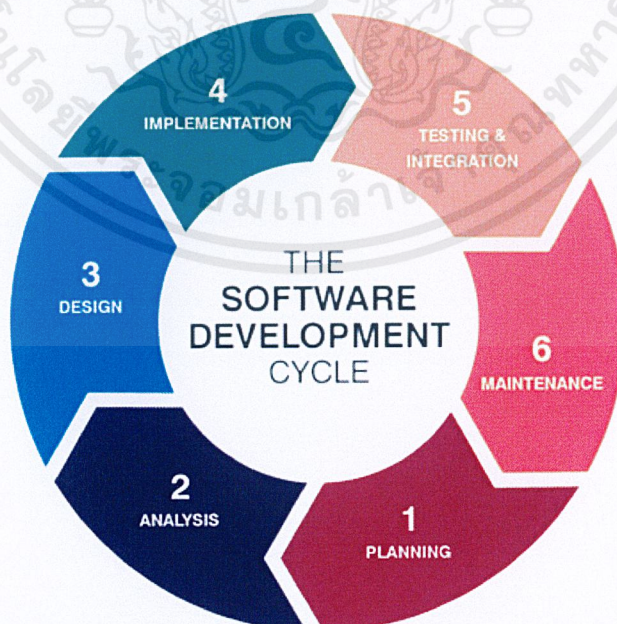
1.4. วิธีการดำเนินงาน

มีการนำหลักการวัฏจักรการพัฒนาซอฟต์แวร์ เข้ามาช่วยในการพัฒนาเว็บไซต์ ซึ่งมีทั้งหมด 6 ขั้นตอน ดังต่อไปนี้

- 1.1.7. การวางแผน (Planning)
 - 1.1.7.1. ได้รับ Requirement จาก Business User
- 1.1.8. การวิเคราะห์ความต้องการ (Analysis)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.1.8.1. วิเคราะห์ความต้องการของ Business User เพื่อนำมาใช้ในการออกแบบเว็บไซต์
- 1.1.9. การออกแบบ (Design)
 - 1.1.9.1. ทำการเลือกใช้เครื่องมือที่จะนำมาใช้ในการออกแบบ
 - 1.1.9.2. ทำการออกแบบหน้าของระบบใหม่ที่ต้องทำการเพิ่มเข้าไปในส่วนการจัดการของบัตรเดบิต โดยคำนึงถึง UI และ UX
 - 1.1.9.3. ทำการออกแบบเว็บไซต์ Back Office ใหม่ เพื่อให้ง่ายต่อการใช้งานมากยิ่งขึ้น และมีความสวยงามเพิ่มขึ้น โดยคำนึงถึง UI และ UX
- 1.1.10. การพัฒนาซอฟต์แวร์ (Implementation)
 - 1.1.10.1. ทำการเลือกเครื่องมือ และภาษาที่จะใช้ในการพัฒนาโปรแกรม
 - 1.1.10.2. ทำการพัฒนาโปรแกรมตามสิ่งที่ออกแบบไว้
- 1.1.11. การทดสอบและนำไปใช้ (Testing and Integration)
 - 1.1.11.1. ทำการทดสอบระบบต่างๆ เพื่อให้มั่นใจว่าไม่มีข้อผิดพลาดเกิดขึ้น
 - 1.1.11.2. นำตัวเว็บไซต์ให้กับ Tester เพื่อให้ Tester ทำการตรวจสอบเพิ่มเติม
- 1.1.12. การบำรุงรักษาซอฟต์แวร์ (Maintenance)
 - 1.1.12.1. ทำการแก้ไขปรับปรุงระบบที่ตรวจพบความผิดพลาดให้สามารถใช้งานได้ถูกต้อง



ภาพที่ 1.3 วัฏจักรการพัฒนาซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5. ประโยชน์ที่คาดว่าจะได้รับ

- 1.1.13. สามารถช่วยให้การจัดการระบบภายในแอปพลิเคชัน K Plus ได้ง่ายขึ้น และสะดวกมากยิ่งขึ้น
- 1.1.14. ได้ศึกษา และลองใช้งาน ภาษาใหม่ๆ และเครื่องมือใหม่ๆ
- 1.1.15. ได้เรียนรู้การทำงานอย่างเป็นระบบ และการดำเนินงานอย่างมีแบบแผน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและการวิจัย

2.1. ทฤษฎีที่เกี่ยวข้อง

2.1.1. HTML5 (Hyper Text Markup Language 5)

HTML เป็นภาษาคอมพิวเตอร์ที่ใช้ในการแสดงผลเอกสารลงบนเว็บไซต์ (Website) หรือที่ถูกเรียกว่าเว็บเพจ มีแม่แบบมาจากภาษา SGML (Standard Generalized Markup Language) ที่ถูกตัดความสามารถบางส่วนออกไป เพื่อให้สามารถทำความเข้าใจได้ง่าย ซึ่งถูกพัฒนาและกำหนดมาตรฐานโดยองค์กรเวปด์เวิลด์เว็บคอนซอร์เทียม หรือ ดับเบิลยูทีซี (World Wide Web Consortium: W3C)

HTML มีโครงสร้างการเขียนโดยอาศัย Tag ในการควบคุมการแสดงผลของข้อความ รูปภาพ เป็นต้น และแต่ละ Tag อาจมีส่วนขยายที่เรียกว่า Attribute สำหรับจัดการรูปแบบต่างๆ

HTML5 เป็นภาษาที่ถูกพัฒนาต่อยอดมาจากภาษา HTML และถูกพัฒนาขึ้นมาโดย WHATWG (The Web Hypertext Application Technology Working Group) โดยได้มีการปรับปรุงแก้ไข และเพิ่มคุณลักษณะหลายๆ อย่างเข้ามาเพื่อให้ผู้พัฒนาสามารถใช้งานได้ง่ายมากยิ่งขึ้น และสามารถทำงานควบคู่กับ CSS3 ได้ดี

HTML



ภาพที่ 2.1 HTML5

2.1.2. CSS3 (Cascading Style Sheets)

เป็นภาษาคอมพิวเตอร์ในกลุ่มภาษาสไตลชีต ซึ่งเป็นภาษาที่มีการใช้งานมาเป็นเวลานาน ในวงการการพิมพ์ โดยภาษาสไตลชีตจะเป็นโครงสร้างเอกสารต้นฉบับที่มีการจัดรูปแบบและตัวอักษรไว้ ด้วยเหตุนี้จึงนำภาษา CSS มาใช้ในการจัดรูปแบบและโครงสร้างของเอกสารที่เขียนมาจากภาษา HTML อันได้แก่ สีของข้อความ สีพื้นหลัง ประเภทตัวอักษร การจัดวางข้อความ โดยภาษา CSS นั้นสามารถใช้งานได้หลากหลายและมีความยืดหยุ่นให้สามารถใช้งานได้กับภาษา XML SVG และ XUL ได้อีกด้วย

ภาษา CSS ถูกกำหนดมาตรฐานโดยองค์กร World Wide Web Consortium ซึ่งเป็นองค์กรระหว่างประเทศที่ทำหน้าที่จัดระบบมาตรฐานที่ใช้งานบนอินเทอร์เน็ต (World Wide Web: WWW) ภาษา CSS ได้ถูกพัฒนาอย่างต่อเนื่องจนในปัจจุบันมีทั้งหมด 4 รุ่นด้วยกัน แต่ CSS4 ยังไม่มีบราวเซอร์ไหนที่สามารถรองรับได้



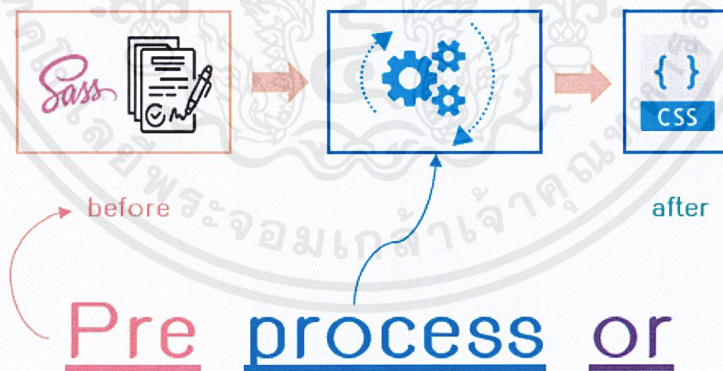
ภาพที่ 2.2 CSS3

2.1.3. SASS (Syntactically Awesome Style Sheets)

SASS คือ CSS Preprocessor ซึ่งทำงานเหมือนกับ LESS CSS โดยมีหลักการเขียนคล้ายกับ CSS ทั่วไป แต่มีการทำงานที่เพิ่มขึ้นซึ่งช่วยให้การเขียนนั้นง่ายยิ่งขึ้น เช่น ความสามารถในการใส่ตัวแปร หรือ การสร้างฟังก์ชัน เป็นต้น

ภาพที่ 2.3 SASS

Preprocessor นั้นประกอบด้วย 3 ส่วน ได้แก่ Pre- มีความหมายเดียวกับคำว่า “before” หรือ “ก่อน” ซึ่งมีนัยเวลาเข้ามาเกี่ยวข้อง, Process คือ กระบวนการใดกระบวนการหนึ่งที่เกิดขึ้น และ -or มีความหมายว่า “เป็นผู้กระทำ”

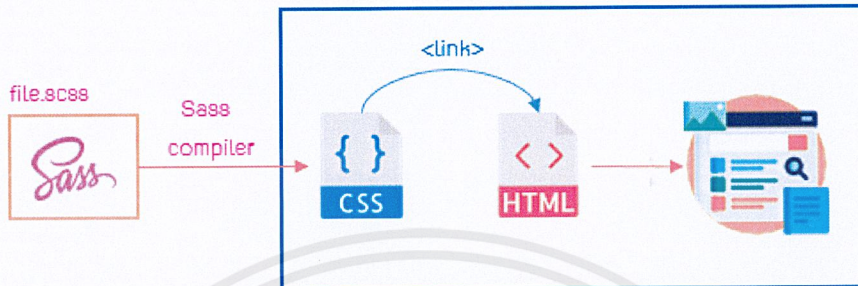


ภาพที่ 2.4 Preprocessor

เนื่องจากการทำงานของ SASS มีการทำงานที่ต่างจาก CSS โดยหลังจากทำการเขียน SASS ลงไฟล์ SASS และหลังจากนั้นคอมไพเลอร์ (Compiler) จะทำการคอมไพล์ (Compile) เพื่อทำการแปลงไฟล์ SASS ให้เป็นไฟล์ CSS ตามปกติ เพื่อนำไปใช้งานจริงในควบคุมการแสดงผลบนหน้าเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการที่ทำการเขียน SASS ก่อนที่ได้ CSS ออกมานั้นจึงเป็นเหตุผลที่ทำให้ SASS เป็น CSS Preprocessor



ภาพที่ 2.5 กระบวนการแปลง SASS เป็น CSS

2.1.4. React

React คือ JavaScript Library ที่ถูกสร้างโดย Facebook ไว้จัดการกับ Model หรือ View ซึ่งมีพื้นฐานแนวความคิดแบบ MVC (Model View Controller) และรองรับการเขียนด้วย JSX (JavaScript syntax extension) เนื่องจาก JSX ทำให้สามารถเขียน HTML ลงไปใน JavaScript ได้ ทำให้ตัว JSX ถูกใช้เพื่อแสดงผลเว็บไซต์ และภายใน React ยังมี Virtual DOM ที่จะสร้าง Tree-Object ซึ่งเป็นตัวแทนของ Real DOM ทำให้การอ้างอิงต่าง ๆ ทำผ่าน Memory โดยตรง ทำให้ React มีการทำงานที่ไวกว่าภาษาอื่น ๆ



ภาพที่ 2.6 React

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5. JavaScript

JavaScript เป็นภาษาที่ถูกพัฒนาโดย Sun Microsystems เป็นภาษาที่สำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOP) คล้ายกับภาษา C, C++ ใช้งานบนเว็บเพจต่างๆ ถูกสร้างขึ้นเพื่อให้เว็บเพจสามารถโต้ตอบกับผู้ใช้งานได้ดียิ่งขึ้น มักใช้ JavaScript เขียนเป็นฟังก์ชันสำหรับใช้งานต่างๆ เช่น ตรวจสอบความถูกต้องของข้อมูลในแบบฟอร์ม, ตรวจสอบชนิดและรุ่นของโปรแกรมเว็บเบราว์เซอร์, สร้างไฟล์ cookie, สร้างลูกเล่นต่างๆ เช่น ปฏิทิน, หิมะตก เป็นต้น

JavaScript เป็นภาษาประเภท Interpreted Language ไม่ต้องมีการคอมไพล์ก่อน ทำให้คอมพิวเตอร์จะแปลและทำงานตามคำสั่งแบบที่ละบรรทัด

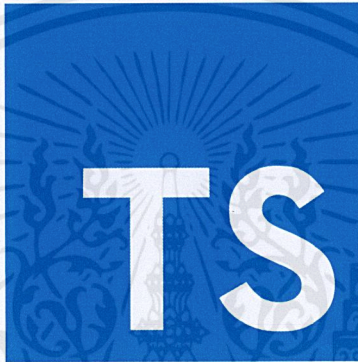
JavaScript เป็น Client-side Script ซึ่งจะประมวลผลบนเครื่องคอมพิวเตอร์ของผู้เยี่ยมชมเว็บไซต์ การที่ JavaScript ไม่ได้ถูกประมวลผลบนเครื่อง Web Server จะช่วยแบ่งเบาภาระการทำงานของเซิร์ฟเวอร์ และทำให้ทำงานได้รวดเร็ว



ภาพที่ 2.7 JavaScript

2.1.6. Typescript

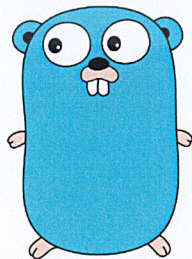
TypeScript เป็นภาษาที่มีความสามารถของ ES2015 (ES6) และอื่นๆ รวมอยู่ แต่สิ่งที่แตกต่างคือ TypeScript มีการสนับสนุน Type System รวมถึงคุณสมบัติอื่น ๆ ที่เพิ่มมากขึ้น เช่น Enum และความสามารถที่เพิ่มขึ้นของการเขียนเชิงวัตถุ ด้วยเหตุนี้ทำให้ TypeScript เหมาะกับการใช้พัฒนาระบบที่มีขนาดใหญ่ และเนื่องจาก TypeScript มีคุณสมบัติของ Transpiler (Source to Source) ทำให้ TypeScript สามารถแปลงเป็น JavaScript จึงทำให้สามารถมั่นใจได้ว่า TypeScript จะสามารถใช้งานได้บนเว็บเบราว์เซอร์ทั่วไป



ภาพที่ 2.8 TypeScript

2.1.7. Golang

Golang เป็นภาษาที่ถูกสร้างโดย Google และเป็น static language สามารถตรวจสอบชนิดของข้อมูลได้ว่าได้ทำการกำหนดถูกต้องหรือไม่ และยังเป็นภาษาประเภท compiled language ซึ่งทำให้สามารถสร้างผลลัพธ์ในรูปแบบของโปรแกรม (Executable File) ที่ทำงานบนแพลตฟอร์มนั้น ๆ ได้โดยไม่ต้องทำงานอยู่บน VM และ Go ยังมีไวยากรณ์ที่ไม่ซับซ้อน มี library ที่ครบครัน และยังใช้เวลาในการประมวลผลที่น้อย ด้วยเหตุนี้ทำให้ Go มีการใช้งานอย่างแพร่หลาย



Golang

ภาพที่ 2.9 Golang

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.8. Java

Java ถูกพัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่บริษัท ซัน ไมโครซิสเต็มส์ ภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทน

Java คือ ภาษา computer ชั้นสูงชนิดหนึ่ง ที่มีการทำงานเป็นลักษณะเชิงวัตถุ (OOP : Object-Oriented Programming) ถูกออกแบบให้มีการเชื่อมโยงกันภายในซับซ้อนน้อยที่สุดเท่าที่จะทำได้ เพื่อให้ application ที่ได้ compile เพียงครั้งเดียว แต่สามารถนำไปใช้ได้ทุกที่ หรือก็คือ รองรับทุกระบบปฏิบัติการ โดยไม่มีความจำเป็นต้องคอมไพล์ใหม่ เนื่องจากข้อมูลที่คอมไพล์แล้วจะถูกเก็บในรูปแบบของ bytecode ที่ใช้รันบน Java virtual machine (JVM) ตามสถาปัตยกรรมของเครื่องที่รัน ตั้งแต่ช่วง 2016 Java ถูกนำมาใช้งานสูงมาก โดยเฉพาะงาน web application แบบ client-server



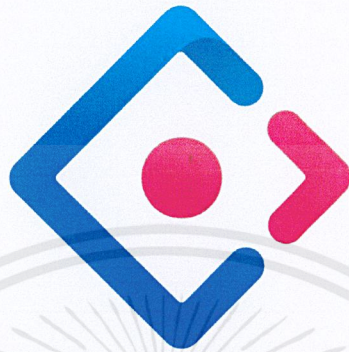
ภาพที่ 2.10 Java

2.1.9. Front-End Framework

Front-End Framework คือ ชุดโค้ดที่ช่วยให้นักพัฒนาส่วน Front-End สามารถทำงานได้สะดวกยิ่งขึ้น โดยภายในจะมี HTML/CSS สำเร็จรูปในการวางโครงสร้าง, ตัวอักษร, ปุ่ม หรือ เมนู เป็นต้น และยังมีส่วนของ JavaScript สำหรับใช้ในส่วนต่างๆ เช่น การทำกล่องโผล่ขึ้นมา ทำแกลอรีรูปภาพ เป็นต้น โดยจะขึ้นอยู่กับแต่ละ Framework ว่าจะมีอะไรให้บ้าง ซึ่งแต่ละ Framework จะมีความสวยงามและการใช้งานที่แตกต่างกัน มีข้อดีคือช่วยทำให้เข้าใจโครงสร้างต่างๆ ได้ง่าย สามารถแก้ไขได้ง่าย และยังสามารถนำไปใช้ต่อได้ในทันที และยังลดเวลาในการเขียนโปรแกรมลง ทำให้นักพัฒนาสามารถเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมได้รวดเร็วยิ่งขึ้น ยกตัวอย่าง Front-End Framework เช่น Material UI, Ant Design, Bootstrap เป็นต้น



ภาพที่ 2.11 Ant Design



ภาพที่ 2.12 Material UI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.10. SQL Server

SQL Server หรือ Microsoft SQL Server คือระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (relational database management system หรือ RDBMS) ผลิตโดยบริษัท Microsoft เป็นระบบฐานข้อมูลแบบ Client/Server และทำงานอยู่บน Window NT ซึ่งใช้ภาษา T-SQL ในการดึงข้อมูล เนื่องจากข้อมูลส่วนใหญ่เก็บไว้ในเครื่องที่ใช้ Microsoft Windows เป็น Operating System จึงทำให้ง่ายต่อ Microsoft SQL ที่จะนำข้อมูลที่อยู่ในรูปของ Window Based มาเก็บและประมวลผล จึงทำให้เป็นระบบฐานข้อมูลที่ถูกเลือกใช้งาน



2.1.11. MVP (Model-View-Presenter)

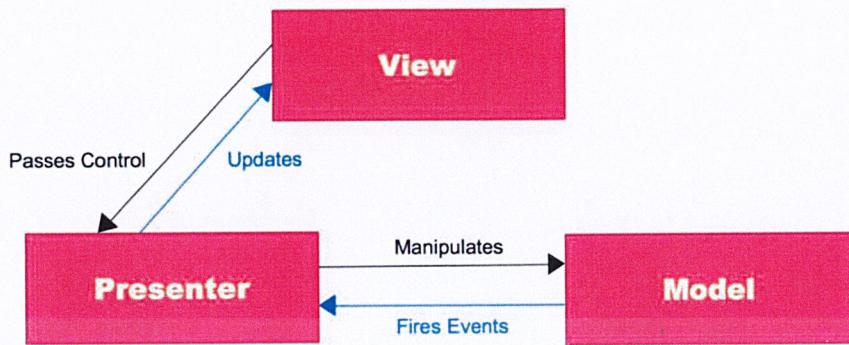
MVP เป็นดีไซน์แพตเทิร์น (Design Pattern) รูปแบบหนึ่ง ซึ่งใช้สำหรับการสร้างหน้าส่วนที่ต่อประสานกับผู้ใช้ (User Interface: UI) โดยเน้นไปที่การแยกหน้าที่ของแต่ละส่วนประกอบ โดย MVP จะประกอบไปด้วย Model, View และ Presenter

View คือ ส่วนที่ไว้ใช้สำหรับแสดงข้อมูล การเขียนโปรแกรมทุกอย่างที่เกี่ยวกับส่วนที่ต่อประสานกับผู้ใช้จะอยู่ในส่วนของ View

Presenter คือส่วนที่ทำหน้าที่เป็นตัวกลางระหว่าง View และ Model เพื่อไม่ให้ View และ Model ทำการติดต่อกันโดยตรง

Model เป็นส่วนที่ทำการเก็บ Business logic หรือส่วนที่ติดต่อกับเน็ตเวิร์ค

ความสัมพันธ์ระหว่าง View และ Presenter จะเป็นแบบ One-to-one โดยที่ Presenter จะทำหน้าที่รับเหตุการณ์ที่ได้มาจาก View และ Presenter จะทำหน้าที่อัปเดต View ตามการกระทำที่ได้จาก View โดยที่ Presenter จะทำการดึงข้อมูลจาก Model ซึ่งจะทำให้ View ไม่รับรู้ถึงการมีอยู่ของ Model



ภาพที่ 2.14 ความสัมพันธ์ระหว่าง Model View และ Presenter

2.1.12. Agile

Agile อยู่ในกลุ่ม Software methodologies ซึ่งมีแนวคิดเดียวกัน มีลักษณะการพัฒนาแบบทำซ้ำเป็นรอบๆ คือการแบ่งส่วนออกเป็นส่วนเล็กๆ และนำไปพัฒนาต่อเรื่อยๆ โดยการทำงานจะให้ความสำคัญกับคนที่ทำมากกว่าวิธีการ ซึ่ง Agile มีหลักการสำคัญหลักๆ ทั้งหมด 4 ข้อดังนี้

1. Individuals interactions over process and tool คือการให้ความสำคัญกับบุคคลมากกว่ากระบวนการและเครื่องมือ
2. Working software over comprehensive documentation คือการส่งมอบซอฟต์แวร์ที่นำไปใช้งานได้จริง มากกว่าการทำเอกสารที่ครบสมบูรณ์
3. Customer collaboration over contract negotiation คือการพบเจอกับลูกค้าตลอดระยะเวลาการทำงาน มากกว่าการทำตามเอกสารความต้องการของลูกค้า
4. Responding to change over following a plan คือการยอมรับการเปลี่ยนแปลง มากกว่าการทำตามแผนที่กำหนดไว้

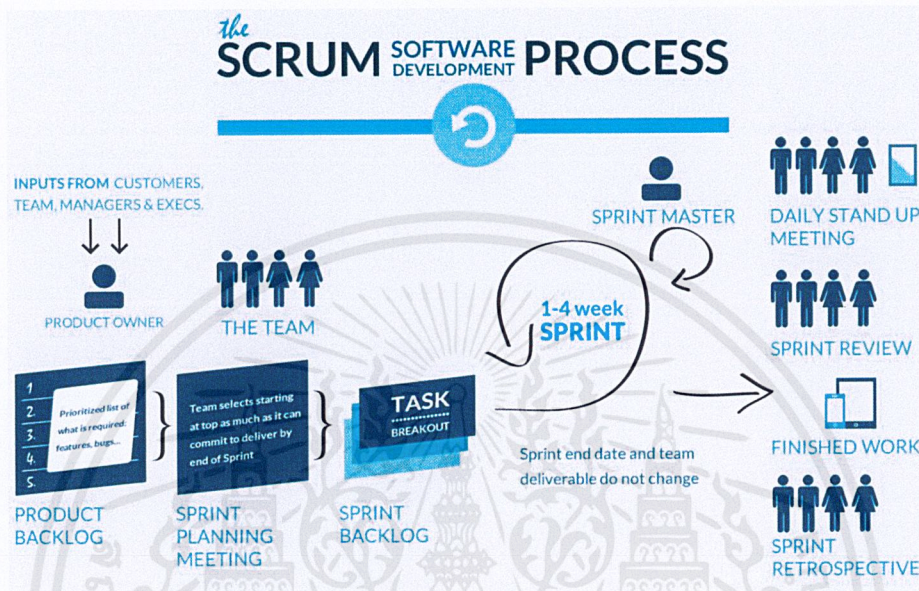
Agile หลายกระบวนการให้เลือกใช้ในการพัฒนา ซึ่งกระบวนการที่ได้รับความนิยมมากที่สุดคือ Scrum

Scrum คือเฟรมเวิร์คของ Agile สำหรับการพัฒนา software ชนิดหนึ่ง ที่มีลักษณะ cross-functional teams ซึ่งหมายถึงกลุ่มคนที่สามารถทำให้เกิดชิ้นงานออกมา โดยไม่มีการแบ่งหน้าที่กัน แต่ไม่ว่าใครก็สามารถทำหน้าที่นั้นๆ ได้ขั้นตอนการทำ Scrum โดยย่อคือ

1. Product backlog คือการทำบอร์ดที่รวบรวมความต้องการทั้งหมดของลูกค้า
2. Sprint backlog คือการทำบอร์ดที่รวบรวมงานของรอบนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทุกวันๆ วัน จะทำการ daily scrum คือการทำ stand-up meeting หน้า Sprint backlog เพื่ออธิบายว่าใครทำอะไรบ้าง และทำถึงไหนแล้ว
4. Sprint ที่เกิดขึ้นจะเกิดขึ้นทุกๆ 30 วัน หรืออาจจะน้อยกว่านี้ตามความเหมาะสม



ภาพที่ 2.15 ขั้นตอนการทำ Scrum

2.1.13. REST API

REST (Representational state transfer) เป็นรูปแบบของสถาปัตยกรรมซอฟต์แวร์ โดยใช้ประโยชน์จากเทคโนโลยี Web protocol เพื่อใช้ในการสร้าง Web Service โดยข้อกำหนดของ REST architectural style ประกอบไปด้วย

2.1.13.1. Performance คือ ประสิทธิภาพในการโต้ตอบ

2.1.13.2. Scalability คือ การมีความสามารถในการปรับขยายได้เพื่อรองรับส่วนประกอบจำนวนมาก

2.1.13.3. Simplicity คือ ความเรียบง่ายของอินเทอร์เฟซที่เหมือนกัน

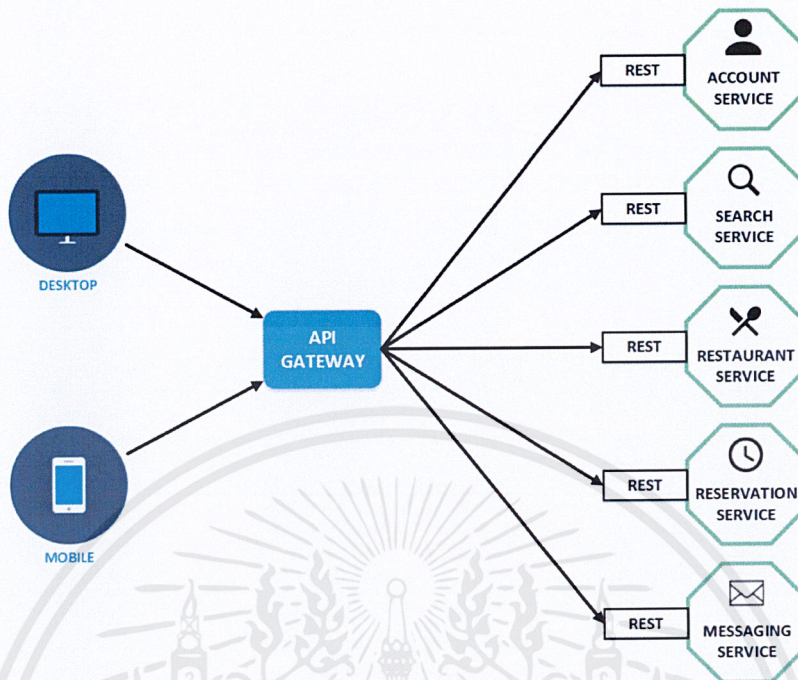
2.1.13.4. Modifiability คือ ความสามารถในการปรับเปลี่ยนส่วนประกอบเพื่อตอบสนองความต้องการที่เปลี่ยนแปลง

2.1.13.5. Visibility คือ การมองเห็นการสื่อสารระหว่าง Component ผ่าน Service agents

2.1.13.6. Portability คือ ความง่ายต่อการโยกย้ายโปรแกรมและข้อมูล

2.1.13.7. Reliability คือ ความน่าเชื่อถือในคาร์ด้านทานความล้มเหลวภายในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.16 REST API

โดย REST API จะถูกเรียกใช้ผ่านทาง HTTP Method ดังนี้

1. GET คือ การดึงข้อมูล
2. POST คือ การสร้างข้อมูล
3. PUT คือ การแก้ไขข้อมูล
4. DELETE คือ การลบข้อมูล

2.1.14. SQL database (Structured Query Language)

คือ ภาษาที่ถูกออกแบบมาเพื่อทำการจัดการข้อมูลที่มีรูปแบบ relational database management system (RDBMS) หรือก็คือมีไว้สำหรับค้นหาข้อมูล เปลี่ยนแปลง เพิ่ม และ ลบ ข้อมูลที่ถูกเก็บอยู่ในฐานข้อมูลในรูปแบบตารางที่มีลักษณะเป็น column และ row เรียกข้อมูลเหล่านี้ว่าถูกเก็บอยู่ใน table ด้วยความสามารถของ SQL สามารถสร้างตารางขึ้นมาใหม่ (create) รวมถึง ลบ (drop) และเปลี่ยนแปลงค่า (alter) ของ table ได้ ซึ่งในความเป็นจริงแล้วคำสั่ง SQL ประกอบไปด้วย

1. Data definition language คือ DDL ใช้ในการสร้างฐานข้อมูล
2. Data manipulation language คือ DML ใช้ เพิ่ม ลบ และเปลี่ยนแปลงข้อมูลใน table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Data control language คือ DCL ใช้ในการกำหนดสิทธิการอนุญาต ทั้งหมดนี้อยู่ในขอบเขตของการทำ insert, query, update, delete, schema creation and modification และ data access control

ภาษา SQL ถูกแบ่งออกมาเป็นส่วนประกอบต่างๆ เรียกส่วนเหล่านี้ตามรูปแบบเช่น

1. Clauses คือองค์ประกอบหนึ่งของ statement และ query (ส่วนนี้เป็น Optional)
2. Expressions คือการสร้างผลลัพธ์ออกมาในรูปแบบ table ที่ประกอบด้วย column และ row จากข้อมูล
3. Predicates คือรูปแบบเงื่อนไขที่มีผลลัพธ์เป็น true/false/unknown หรือก็คือ Boolean นั่นเอง
4. Queries คือการดึงข้อมูลตามเงื่อนไข (clause) เป็นส่วนสำคัญใน SQL
5. Statements คือสามารถมีผลต่อโครงสร้างข้อมูล, จัดการข้อมูล transactions, program flow, session หรือแม้กระทั่ง วิเคราะห์ปัญหา โดยจำเป็นต้องจบด้วย semicolon (;) ซึ่งเป็นสิ่งจำเป็นที่ต้องมีทุกครั้งสำหรับ SQL
6. Insignificant whitespace หรือช่องว่าง สำหรับใน SQL statement และ query จะไม่สนใจ ทำให้ SQL สามารถเขียนในรูปแบบที่หลากหลายสวยงาม SQL จะมีคำสั่งอยู่ 4 คำสั่งคือ
 1. Select query ใช้สำหรับดึงข้อมูลที่ต้องการ
 2. Update query ใช้สำหรับแก้ไขข้อมูล
 3. Insert query ใช้สำหรับการเพิ่มข้อมูล
 4. Delete query ใช้สำหรับลบข้อมูลออกไป

ID	First_Name	Title	Team
10011	Debra	Programmer	Eng
10018	Yolanda	Programmer	Eng
10019	Glen	Product Designer	Mkt
10028	Casey	Account Exec	Sal
10049	Tang	Support Tech	Sup
10051	Serge	UX Designer	Eng
10059	María	Sales Director	Sal

ภาพที่ 2.17 ตัวอย่างตารางภายใน Relational Database

2.1.15. Web Application

คือ แอปพลิเคชันที่ถูกเขียนขึ้นมาเพื่อเป็นเบรเซอร์สำหรับการใช้งานเว็บเพจต่างๆ ซึ่งถูกปรับแต่งให้แสดงผลแต่ส่วนที่จำเป็น เพื่อเป็นการลดทรัพยากรในการประมวลผล ของตัวเครื่อง สมาร์ทโฟน หรือ แท็บเล็ต ทำให้โหลดหน้าเว็บไซต์ได้เร็วขึ้น อีกทั้งผู้ใช้งานยังสามารถใช้งานผ่าน อินเทอร์เน็ต และอินทราเน็ต ในความเร็วต่ำได้



ภาพที่ 2.18 Web Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.16. UI (User Interface)

คือ ส่วนต่อประสานกับผู้ใช้งาน หรือ ส่วนที่ใช้ในการเชื่อมต่อกับผู้ใช้งาน กล่าวคือ ส่วนที่ให้ผู้ใช้งานสามารถโต้ตอบกับการใช้งานผลิตภัณฑ์ได้ ซึ่งจะมุ่งเน้นไปที่เรื่องของหน้าตา การออกแบบ และการดีไซน์ ยกตัวอย่างเช่น หน้าจอ แพลตฟอร์ม เมนู ฟอรัมต่าง ๆ การวางภาพ ขนาด ตัวอักษร ปุ่ม แป้นพิมพ์ เสียง หรือแม้แต่แสงไฟ เป็นต้น

สิ่งสำคัญสำหรับ UI ก็คือดีไซน์ที่ดูสะอาด สวยงาม ดึงดูดใจ อีกทั้งต้องเข้าใจง่าย ใช้งานง่าย มีมาตรฐานและเป็นมิตรต่อผู้ใช้งาน นอกจากนี้ยังต้องมีฟังก์ชันที่น่าสนใจ มีภาพกราฟที่ทำให้คนเกิดความรู้สึกอยากใช้งาน และที่สำคัญจะต้องมีความเป็นเอกลักษณ์ โดดเด่น แต่ก็ไม่ล้นหรือต่างมากเกินไป

2.1.17. UX (User Experience)

คือ ประสบการณ์ของผู้ใช้งาน ต่อการใช้งานของผู้ใช้งาน (Usability) และการเข้าถึง (Accessibility) โดยทั่วไปมักจะโยงในความหมายของการใช้งานของระบบงาน ที่มองถึงประสบการณ์การสร้างปฏิสัมพันธ์ของผู้ใช้งานต่อการใช้งานระบบงาน และต่อ User Interface (UI) ที่ซึ่งจะหมายความถึงความง่าย ความยากในการใช้งานของผู้ใช้งาน การเข้าถึงทั้งที่เป็นรูปแบบของ Web Site, Web Application หรือ Apps เป็นต้น



ภาพที่ 2.19 UX Design

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงาน

รายงานสหกิจฉบับนี้เป็นการพัฒนาเว็บแอปพลิเคชันจัดการฐานข้อมูลสำหรับการตั้งค่าแอปพลิเคชัน K PLUS ซึ่งในบทนี้จะกล่าวถึงขั้นตอนการดำเนินงาน รวมถึงการวิเคราะห์และออกแบบโครงสร้างและการทำงานของระบบ โดยจะแบ่งเป็นชิ้นงาน 2 ชิ้นงาน เนื่องจากผู้จัดทำได้มีการทำส่วนขยายในเว็บแอปพลิเคชันตัวเก่า และทำการนำแอปพลิเคชันตัวเก่ามาปรับปรุงให้มีความสวยงาม และใช้ง่ายยิ่งขึ้นในเว็บแอปพลิเคชันตัวใหม่ โดยมีรายละเอียด ดังนี้

3.1. ขั้นตอนการดำเนินงาน

3.1.1. เว็บแอปพลิเคชันตัวเก่า

- 3.1.1.1. ทำการเก็บข้อมูลความต้องการของระบบที่จะทำการเพิ่มเข้ามาในเว็บแอปพลิเคชันตัวเก่า
- 3.1.1.2. วางแผนตารางการพัฒนาระบบตัวเก่า
- 3.1.1.3. ออกแบบหน้าเว็บแอปพลิเคชันของระบบตัวเก่า และนำตัวต้นแบบให้กับผู้ใช้งานได้ลองใช้งาน
- 3.1.1.4. ทำการปรับปรุงตัวต้นแบบตามคำแนะนำของผู้ใช้งาน
- 3.1.1.5. พัฒนาระบบให้มีรูปร่างตามแบบที่ทำไว้
- 3.1.1.6. ทดสอบและปรับปรุงระบบ

3.1.2. เว็บแอปพลิเคชันตัวใหม่

- 3.1.2.1. ทำการเก็บข้อมูลความต้องการจากผู้ใช้งานที่เคยใช้เว็บแอปพลิเคชันตัวเก่าเพื่อนำมาปรับปรุง
- 3.1.2.2. วางแผนตารางการพัฒนาโครงการให้สอดคล้องกับงานเพื่อให้โครงการสำเร็จในระยะเวลาที่กำหนด
- 3.1.2.3. เลือกเทคโนโลยีสำหรับพัฒนาโครงการให้เหมาะสมกับงาน
- 3.1.2.4. ศึกษาเทคโนโลยีที่เลือกใช้สำหรับการพัฒนาโครงการ
- 3.1.2.5. ออกแบบเว็บแอปพลิเคชัน และนำตัวต้นแบบให้กับผู้ใช้งานได้ลองทดสอบ
- 3.1.2.6. ทำการปรับปรุงรูปแบบเว็บแอปพลิเคชันตามคำแนะนำของผู้ใช้งาน
- 3.1.2.7. พัฒนาเว็บแอปพลิเคชันให้มีรูปร่างตามแบบที่ได้ออกแบบไว้
- 3.1.2.8. ทดสอบและปรับปรุงเว็บแอปพลิเคชัน

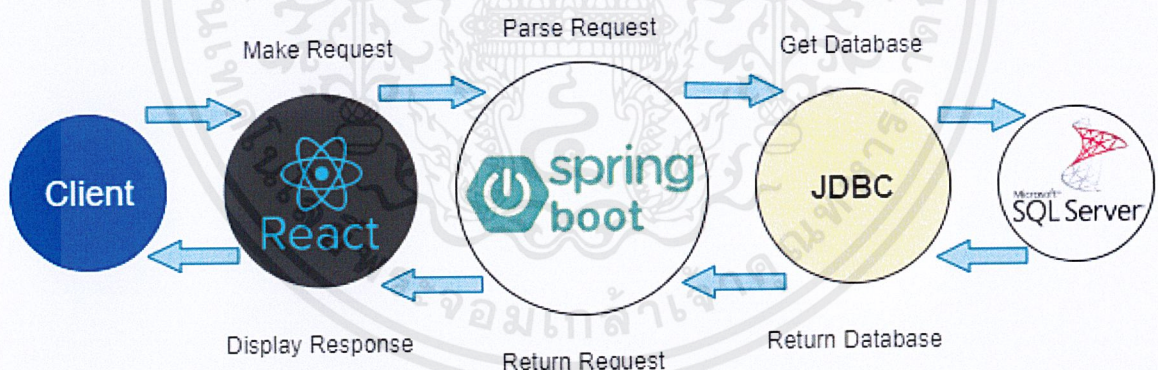
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.9. สรุปผลและจัดทำเอกสารอธิบายกระบวนการทำงานและโครงสร้างของ โครงการ

3.2. สถาปัตยกรรมของระบบ

ในโครงการนี้จะมีการพูดถึงเว็บแอปพลิเคชัน 2 เว็บ เนื่องจากมีการพัฒนาระบบใหม่ในเว็บแอปพลิเคชันตัวเก่า และนำเว็บแอปพลิเคชันตัวเก่ามาพัฒนาให้มีความสวยงาม และใช้งานได้ง่ายยิ่งขึ้นในเว็บแอปพลิเคชันตัวใหม่

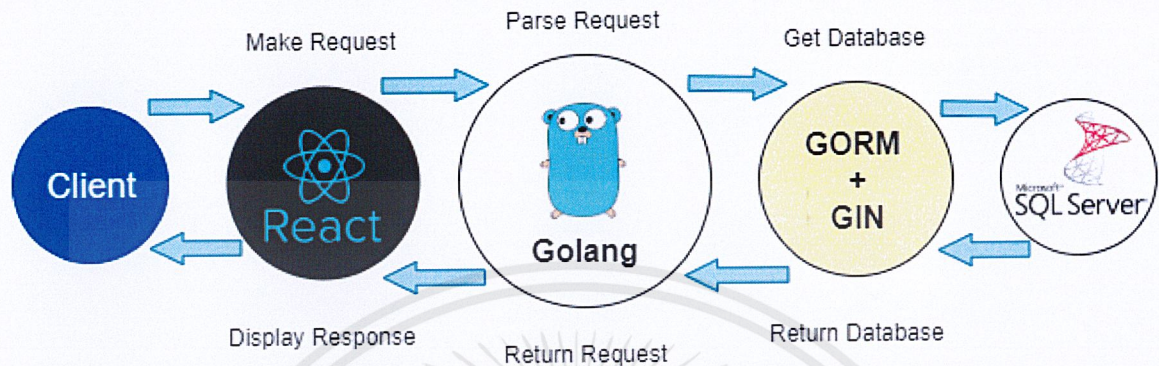
โดยเว็บแอปพลิเคชันตัวเก่านั้นจะถูกพัฒนาด้วย React ซึ่งเป็น JavaScript Library พัฒนา ร่วมกับ Material UI, CSS และ JavaScript ในส่วนของระบบจัดการเว็บไซต์จะถูกพัฒนาด้วยภาษา Java โดยข้อดีของการเลือกใช้ React เนื่องจากเป็น JavaScript Library ที่ทำให้การพัฒนาเว็บแอปพลิเคชันนั้นง่ายขึ้นอย่างมาก สามารถพัฒนาและออกแบบเว็บแอปพลิเคชันได้อย่างรวดเร็ว และทำความเข้าใจได้ง่าย พร้อมทั้งยังสามารถทำการแยกส่วนประกอบ ต่าง ๆ ภายในเว็บแอปพลิเคชันเพื่อนำส่วนเหล่านั้นกลับมาใช้ใหม่ได้ ภายในเว็บไซต์นั้นจะใช้ Material UI ซึ่งเป็น Front-end Framework ที่รวม HTML, CSS และ JavaScript เข้าด้วยกันสำหรับพัฒนาเว็บแอปพลิเคชัน ที่รองรับ Smart Device หรือที่เรียกกันว่า Responsive Web เพื่อให้สามารถใช้งานได้บน ทุก ๆ อุปกรณ์



ภาพที่ 3.1 สถาปัตยกรรมของเว็บแอปพลิเคชันตัวเก่า

ในส่วนเว็บแอปพลิเคชันตัวใหม่นั้นจะถูกพัฒนาด้วย React เช่นกัน แต่มีการพัฒนาร่วมกับ Ant Design, SASS และ Typescript ภายในเว็บไซต์ตัวใหม่นี้มีการใช้ Ant Design แทน Material UI ซึ่งเป็น Front-end Framework เช่นกัน ซึ่งมีความแตกต่างกันในเรื่องของดีไซน์ ในส่วนของระบบจัดการเว็บไซต์จะถูกพัฒนาด้วย Golang เนื่องจากเป็นภาษาที่มีความเรียบง่าย ตัวภาษาเป็น Static Type

System ทำให้เหมาะกับการพัฒนาระบบที่มีขนาดใหญ่ และยังทำให้ตัวโปรแกรมมีประสิทธิภาพการทำงานที่ดีขึ้น



ภาพที่ 3.2 สถาปัตยกรรมของเว็บแอปพลิเคชันตัวใหม่

3.3.การออกแบบ

3.3.1. การนำเข้าสู่ข้อมูล

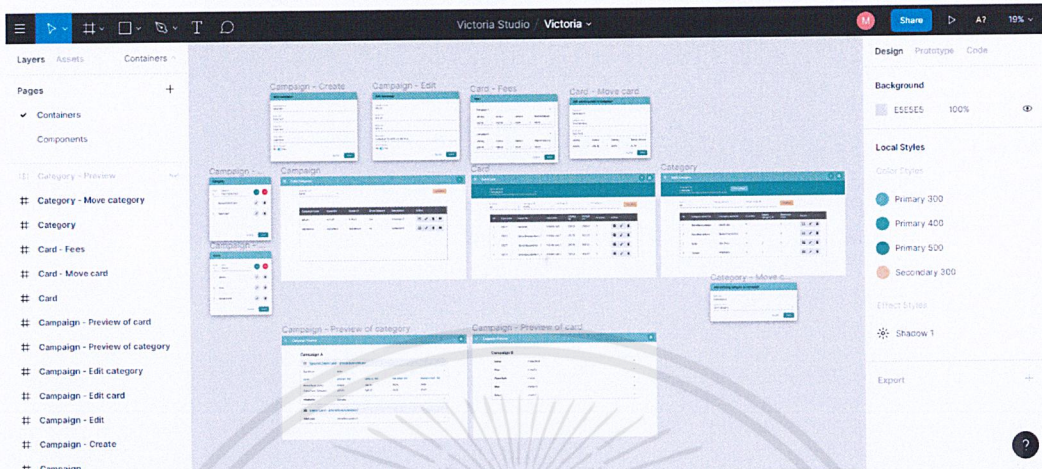
การนำเข้าสู่ข้อมูลของเว็บแอปพลิเคชัน ประกอบด้วยส่วนย่อยดังนี้

3.3.1.1. ส่วนที่ติดต่อกับผู้ใช้ (User Interface)

ส่วนที่ติดต่อกับผู้ใช้งานในเว็บแอปพลิเคชัน Back Office นั้น ประกอบด้วยการใช้งานหลาย ๆ ส่วนโดยแต่ละหน้าจะมีความซับซ้อนที่แตกต่างกันไป ทำให้มีส่วนที่ติดต่อกับผู้ใช้ (User Interface) ต้องแบ่งออกเป็นหลายหน้า โดยผู้จัดทำได้นำความรู้ทางด้าน User Interface และ User Experience มาใช้ในการออกแบบและพัฒนาส่วนที่ติดต่อกับผู้ใช้งาน เพื่อให้ผู้ใช้งานสามารถใช้งานได้ อย่างเข้าใจง่าย โดยมีตัวอย่างหน้าต่าง ๆ ซึ่งจะทำการแบ่งเป็น 2 เว็บไซต์ ได้แก่ เว็บแอปพลิเคชันตัวเก่า และเว็บแอปพลิเคชันตัวใหม่ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.1.1. เว็บแอปพลิเคชันตัวเก่า



ภาพที่ 3.3 การออกแบบระบบเพิ่มเติมภายในเว็บแอปพลิเคชันตัวเก่าโดยใช้โปรแกรม Figma



ภาพที่ 3.4 หน้าเว็บสำหรับระบบจัดการบัตรเดบิตที่ถูกเพิ่มเข้ามาในเว็บไซต์ตัวเก่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ID	Category name TH	Category name EN	Is active	Parent category ID	Sequence order	Action
1	บัตรเดบิตแบบธรรมดา	Debit Card	Y		1	[Menu] [Edit] [Delete]
2	บัตรเดบิตพิเศษ	Special Debit Card	Y		2	[Menu] [Edit] [Delete]
3	วันพีซ	One Piece	Y	2	1	[Menu] [Edit] [Delete]
4	โปเกมอน	POKEMON	Y	2	1	[Menu] [Edit] [Delete]

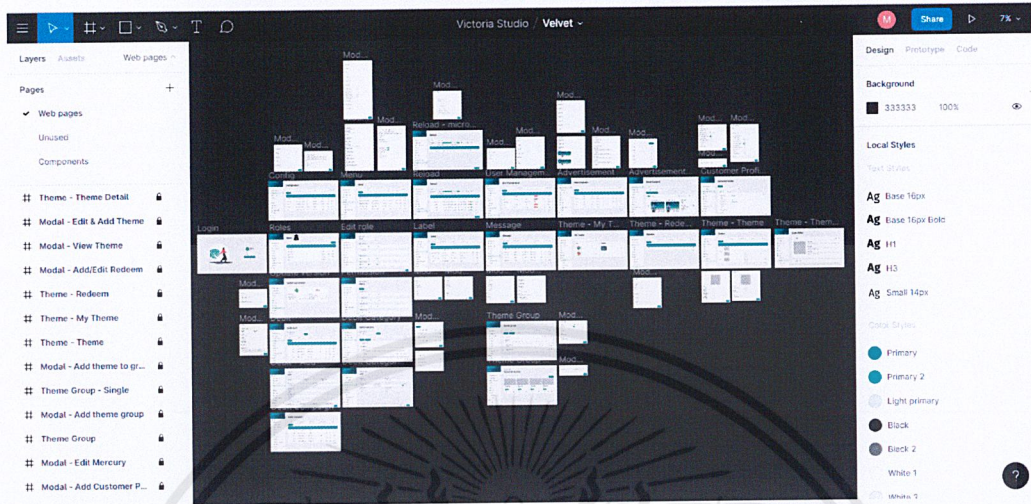
ภาพที่ 3.5 หน้าเว็บสำหรับระบบจัดการหมวดหมู่ของบัตรเดบิตที่ถูกเพิ่มเข้ามาในเว็บไซต์ตัวเก่า

Campaign Code	Name EN	Name TH	Show Category	Description	Action
KPLUS	K PLUS	K PLUS	Yes	Campaign A	[Menu] [Edit] [Delete] [Add]
OLD KPLUS	OLD KPLUS	OLD KPLUS	No	Campaign B	[Menu] [Edit] [Delete] [Add]

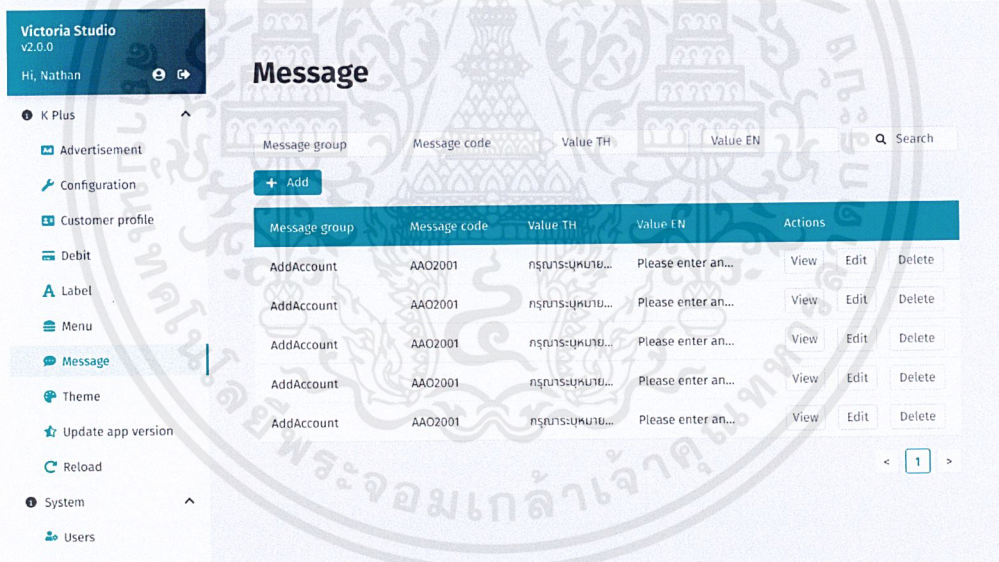
ภาพที่ 3.6 หน้าเว็บสำหรับระบบจัดการแคมเปญของบัตรเดบิตที่ถูกเพิ่มเข้ามาในเว็บไซต์ตัวเก่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.1.2. เว็บไซต์พลิกเคชั่นตัวใหม่

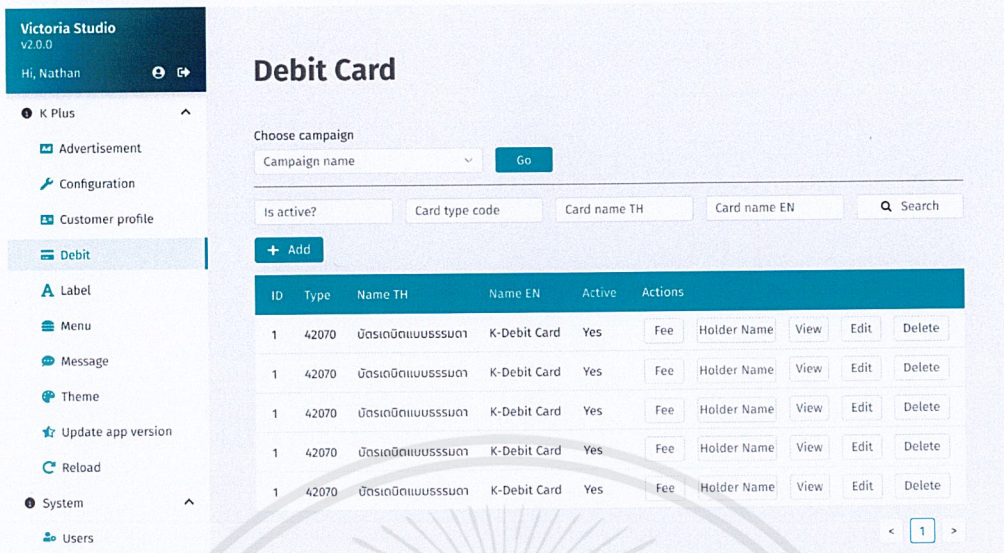


ภาพที่ 3.7 การออกแบบเว็บไซต์พลิกเคชั่นตัวใหม่โดยใช้โปรแกรม Figma

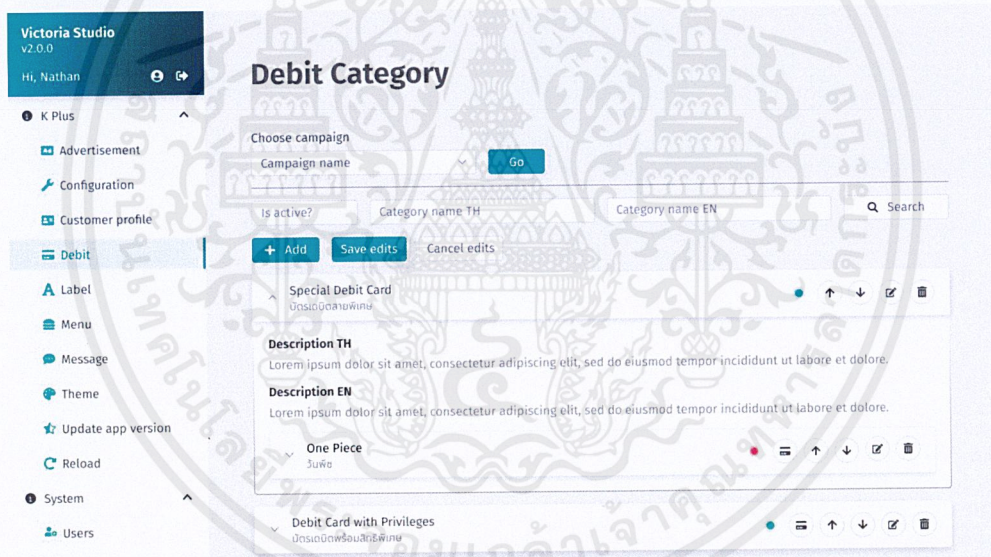


ภาพที่ 3.8 หน้าเว็บสำหรับจัดการข้อความที่แสดงอยู่บนแอปพลิเคชัน K Plus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

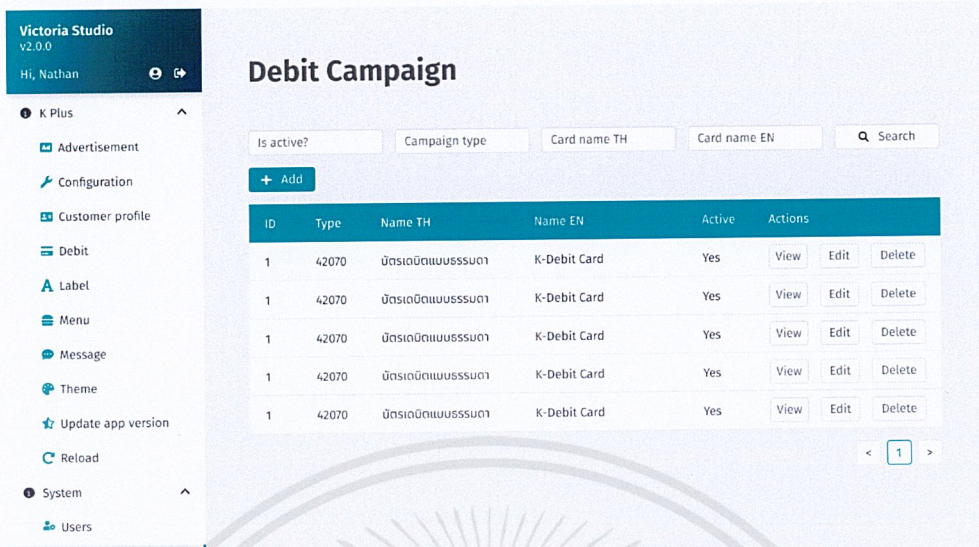


ภาพที่ 3.9 หน้าเว็บสำหรับจัดการข้อมูลของบัตรเดบิตภายในแอปพลิเคชัน K Plus



ภาพที่ 3.10 หน้าเว็บสำหรับจัดการหมวดของบัตรเดบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

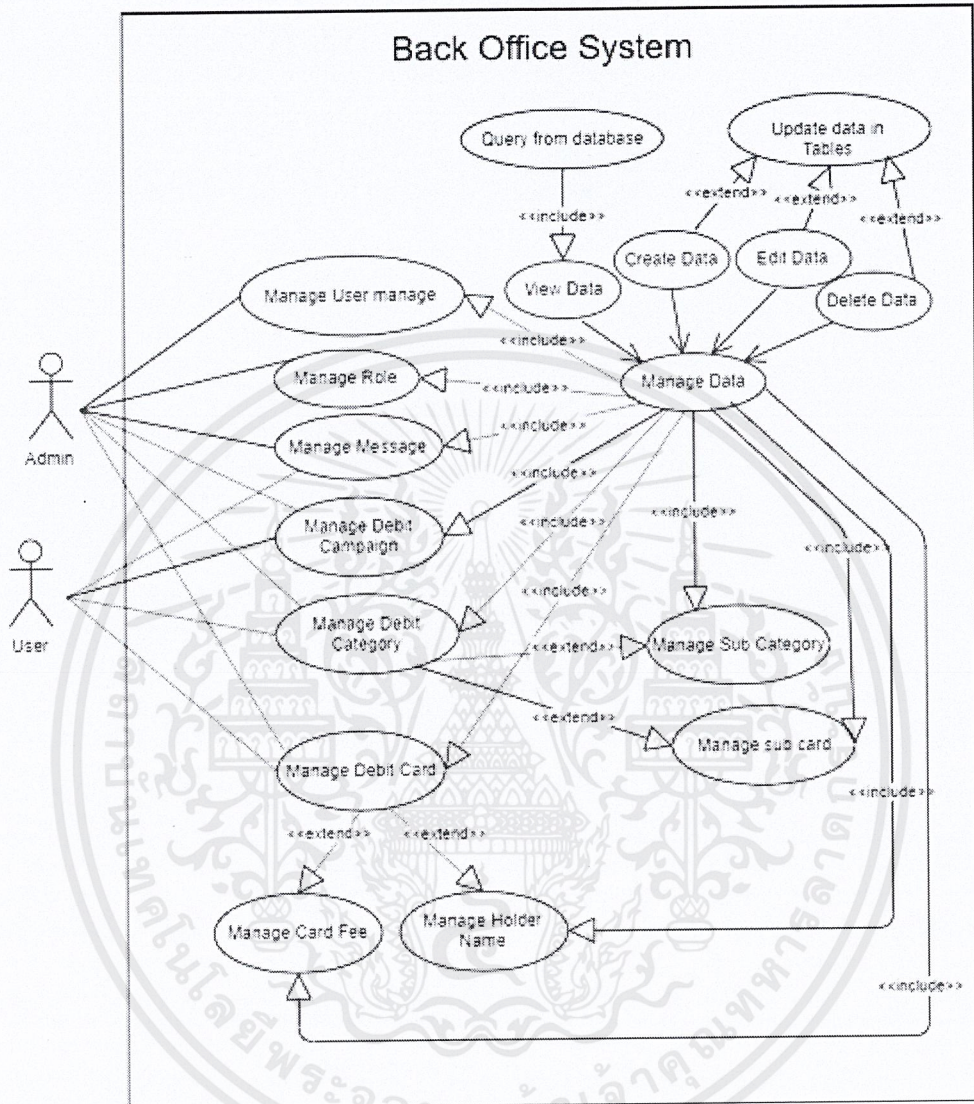


ภาพที่ 3.11 หน้าเว็บสำหรับจัดการหมวดหมู่ของบัตรเดบิตให้รวมอยู่เป็นแคมเปญ

3.3.1.2. ส่วนที่ติดต่อกับฐานข้อมูล

ในการพัฒนาส่วนของการนำเข้าข้อมูลจากฐานข้อมูลในเว็บแอปพลิเคชันตัวเก่าได้เลือกใช้ภาษา Java ในการพัฒนาส่วนต่อกับฐานข้อมูล แต่ในเว็บแอปพลิเคชันตัวใหม่นั้นได้เลือกใช้ภาษา Golang ในการพัฒนาส่วนต่อกับฐานข้อมูล และเลือกใช้ library GORM ซึ่งมีข้อดีก็คือ GORM เป็น ORM บนภาษา Go ซึ่งทำให้ผู้ใช้ไม่ต้องเขียน sql query string ทั้งหมดด้วยตนเอง และยังลด redundant, duplication ได้อย่างมากในการช่วยให้ต่อกับฐานข้อมูลของ SQL Server ได้ง่ายขึ้น และในการพัฒนาทั้งสองเว็บแอปพลิเคชันนั้นได้เลือกใช้โปรแกรม DBeaver ในการดูข้อมูลใน Database และทั้งสองเว็บแอปพลิเคชันนั้นใช้ฐานข้อมูลเดียวกัน ทำให้มี Use case diagram และ Database schema diagram ที่เหมือนกัน

3.3.2. Use Case Diagram



ภาพที่ 3.12 Use case Diagram ของเว็บแอปพลิเคชัน Back Office

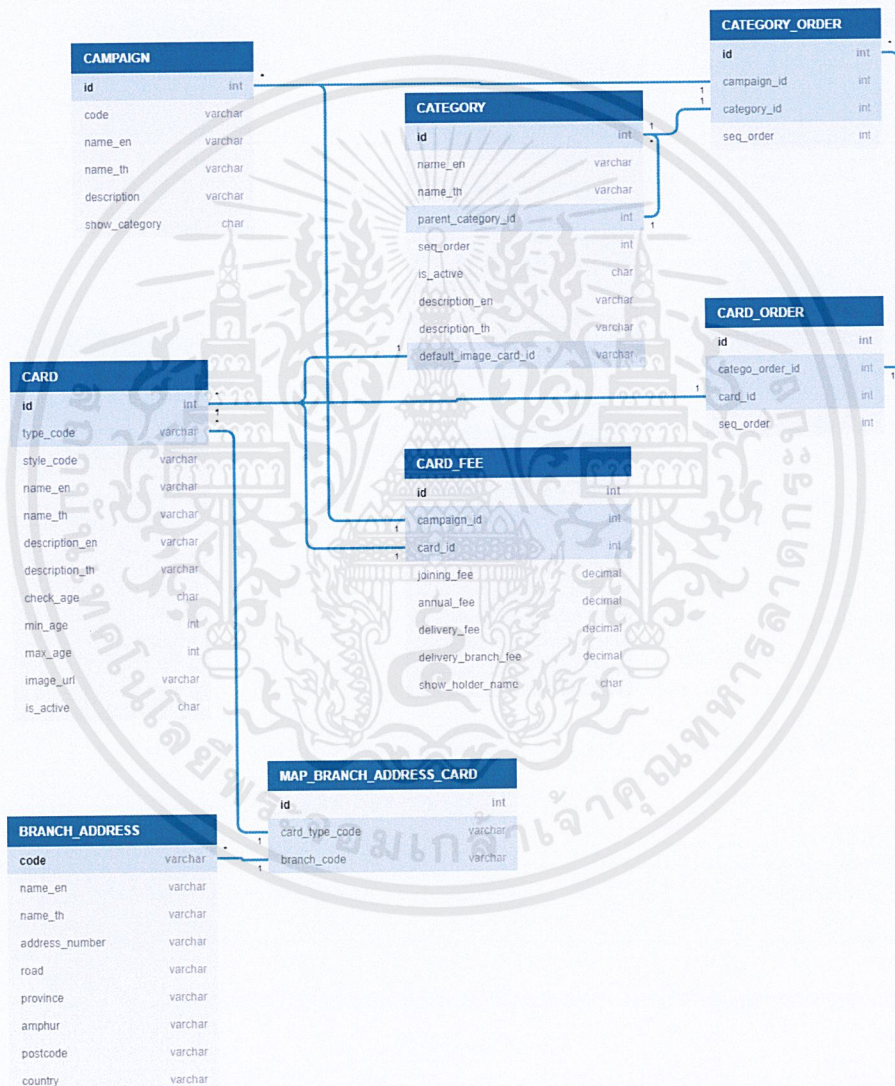
จาก Requirement ที่ได้มา จะสามารถนำมาเขียนเป็น Use case diagram ได้ดังภาพ

ที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3. Database Schema Diagram

ในส่วนของการออกแบบรายละเอียดภาพรวมทั้งหมดของโครงสร้างฐานข้อมูล เนื่องจากตัวระบบมีขนาดใหญ่ทำให้ฐานข้อมูลมีหลายส่วน ผู้จัดทำจึงได้นำเอาตัวอย่าง Schema Diagram ของ Debit ที่มีความซับซ้อนและ Message ที่ไม่ซับซ้อนมากนักมาอย่างละหนึ่งตัวอย่าง เพื่ออธิบายโครงสร้างของฐานข้อมูล



ภาพที่ 3.13 แผนภาพโครงสร้างฐานข้อมูลของ Debit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 3.13 แสดงถึงโครงสร้างฐานข้อมูลของ Debit ซึ่งแต่ละส่วนมีรายละเอียดดังนี้

- ตาราง Campaign ใช้สำหรับเก็บข้อมูลของ Campaign มีข้อมูลย่อยดังนี้

ตารางที่ 3.1 รายละเอียดของข้อมูลในตาราง Campaign

Field	Key	Datatype	Description
id	PK	int	เลขรหัสสำหรับเก็บลงฐานข้อมูล
code		varchar	เลขรหัสของ Campaign สำหรับให้ผู้ใช้งาน
name_en		varchar	ชื่อ Campaign ภาษาอังกฤษ
name_th		varchar	ชื่อ Campaign ภาษาไทย
description		varchar	คำอธิบายเพิ่มเติมเกี่ยวกับ Campaign
show_category		char	กำหนดว่า Campaign นี้จะเก็บ Category หรือ Card

- ตาราง Category ใช้สำหรับเก็บข้อมูลของ Category มีข้อมูลย่อยดังนี้

ตารางที่ 3.2 รายละเอียดของข้อมูลในตาราง Category

Field	Key	Datatype	Description
id	PK	int	เลขรหัสของ category
name_en		varchar	ชื่อ category ภาษาอังกฤษ
name_th		varchar	ชื่อ category ภาษาไทย
parent_category_id	FK	int	รหัสของ category ที่เป็น parent category ของ category นี้
seq_order		int	ลำดับของ category เมื่อเป็น sub category
is_active		char	สถานะการเปิดใช้งาน
description_en		varchar	คำอธิบายเพิ่มเติมภาษาอังกฤษ
description_th		varchar	คำอธิบายเพิ่มเติมภาษาไทย
default_image_card_id		varchar	รูปเริ่มต้นของ category นี้ โดยขึ้นอยู่กับ card

- ตาราง Card ใช้สำหรับเก็บข้อมูลของ Card มีข้อมูลย่อยดังนี้

ตารางที่ 3.3 รายละเอียดของข้อมูลในตาราง Card

Field	Key	Datatype	Description
id	PK	int	เลขรหัสของ card
type_code		varchar	เลขรหัสเพื่อให้ BU ใช้งาน
style_code		varchar	รหัสรูปแบบของแต่ละการ์ด
name_th		varchar	ชื่อภาษาไทย
name_en		varchar	ชื่อภาษาอังกฤษ
description_en		varchar	รายละเอียดเพิ่มเติมภาษาอังกฤษ
description_th		varchar	รายละเอียดเพิ่มเติมภาษาไทย
check_age		char	สถานะเพื่อบอกว่าการ์ดนี้มีอายุการใช้งานหรือไม่
min_age		int	อายุขั้นต่ำที่เป็นไปได้
max_age		int	อายุมากที่สุดที่เป็นไปได้
image_url		varchar	รูปภาพของบัตร
is_active		char	สถานะการเปิดใช้งาน

- ตาราง Category Order ใช้สำหรับเก็บข้อมูลของ Category ที่มี campaign เพื่อให้สามารถเช็คได้ง่ายว่า category นั้นอยู่ใน campaign อะไร มีข้อมูลย่อยดังนี้

ตารางที่ 3.4 รายละเอียดของข้อมูลในตาราง Category Order

Field	Key	Datatype	Description
id	PK	int	เลขรหัสสำหรับเก็บลงฐานข้อมูล
campaign_id	FK	int	เลขรหัสของ Campaign
category_id	FK	int	เลขรหัสของ Category
seq_order		int	ลำดับของ Category

- ตาราง Card Order ใช้สำหรับเก็บข้อมูลของ Card ที่มี campaign และ category เพื่อให้สามารถเช็คได้ง่ายว่า card นั้นอยู่ใน category และ campaign อะไร และมีลำดับแสดงผลที่เท่าไร มีข้อมูลย่อยดังนี้

ตารางที่ 3.5 รายละเอียดของข้อมูลในตาราง Card Order

Field	Key	Datatype	Description
id	PK	int	เลขรหัสสำหรับเก็บลงฐานข้อมูล
catego_order_id	FK	int	เลขรหัสของ Category Order
card_id	FK	int	เลขรหัสของ Card
seq_order		int	ลำดับของ Card

- ตาราง Card Fee ใช้สำหรับเก็บข้อมูลของค่า Fee ของ Card นั้น ๆ โดยขึ้นอยู่กับ Campaign มีข้อมูลย่อยดังนี้

ตารางที่ 3.6 รายละเอียดของข้อมูลในตาราง Card Fee

Field	Key	Datatype	Description
id	PK	int	เลขรหัสสำหรับเก็บลงฐานข้อมูล
campaign_id	FK	int	เลขรหัสของ campaign
card_id	FK	int	เลขรหัสของ card
joining_fee		decimal	
annual_fee		decimal	
delivery_fee		decimal	
delivery_brance_fee		decimal	
show_holder_name		char	

- ตาราง Map Branch Address Card ใช้สำหรับดูว่า Type Code อันไหนที่สามารถส่งไปได้ที่สาขาไหนได้บ้าง มีข้อมูลย่อยดังนี้

ตารางที่ 3.7 รายละเอียดของข้อมูลในตาราง Map Branch Address Card

Field	Key	Datatype	Description
id	PK	int	เลขรหัสสำหรับเก็บลงฐานข้อมูล
card_type_code	FK	varchar	รหัสของ type code
branch_code	FK	varchar	รหัสสาขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตาราง Branch Address ใช้สำหรับเก็บข้อมูลของสาขาแต่ละแห่ง มีข้อมูลย่อยดังนี้

ตารางที่ 3.8 รายละเอียดของข้อมูลในตาราง Branch Address

Field	Key	Datatype	Description
code	PK	varchar	เลขรหัสของสาขา
name_en		varchar	ชื่อภาษาอังกฤษ
name_th		varchar	ชื่อภาษาไทย
address_number		varchar	บ้านเลขที่
road		varchar	ถนน
province		varchar	ตำบล
amphur		varchar	อำเภอ
postcode		varchar	รหัสไปรษณีย์
country		varchar	จังหวัด

MESSAGE	
message_code	varchar
message_text_th	varchar
message_text_en	varchar
message_desc	varchar
message_type	varchar
message_group	varchar
system	varchar

ภาพที่ 3.14 แผนภาพโครงสร้างฐานข้อมูลของ Message

จากภาพที่ 3.14 แสดงถึงโครงสร้างฐานข้อมูลของ Message ซึ่งแต่ละส่วนมีรายละเอียดดังนี้

- ตาราง Message ใช้สำหรับเก็บข้อมูลของ Message มีข้อมูลย่อยดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

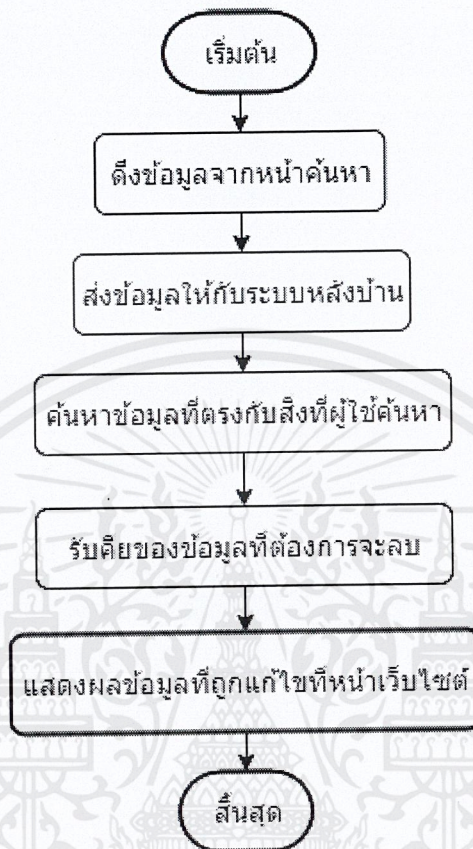
ตารางที่ 3.9 รายละเอียดของข้อมูลในตาราง Message

Field	Key	Datatype	Description
message_code	PK	varchar	รหัสสำหรับ message
message_text_th		varchar	ข้อความภาษาไทย
message_text_en		varchar	ข้อความภาษาอังกฤษ
message_desc		varchar	คำอธิบายเพิ่มเติม
message_type		varchar	ชนิดของข้อความ
message_group		varchar	จัดกลุ่มข้อความ
system		varchar	ระบบที่ใช้งาน



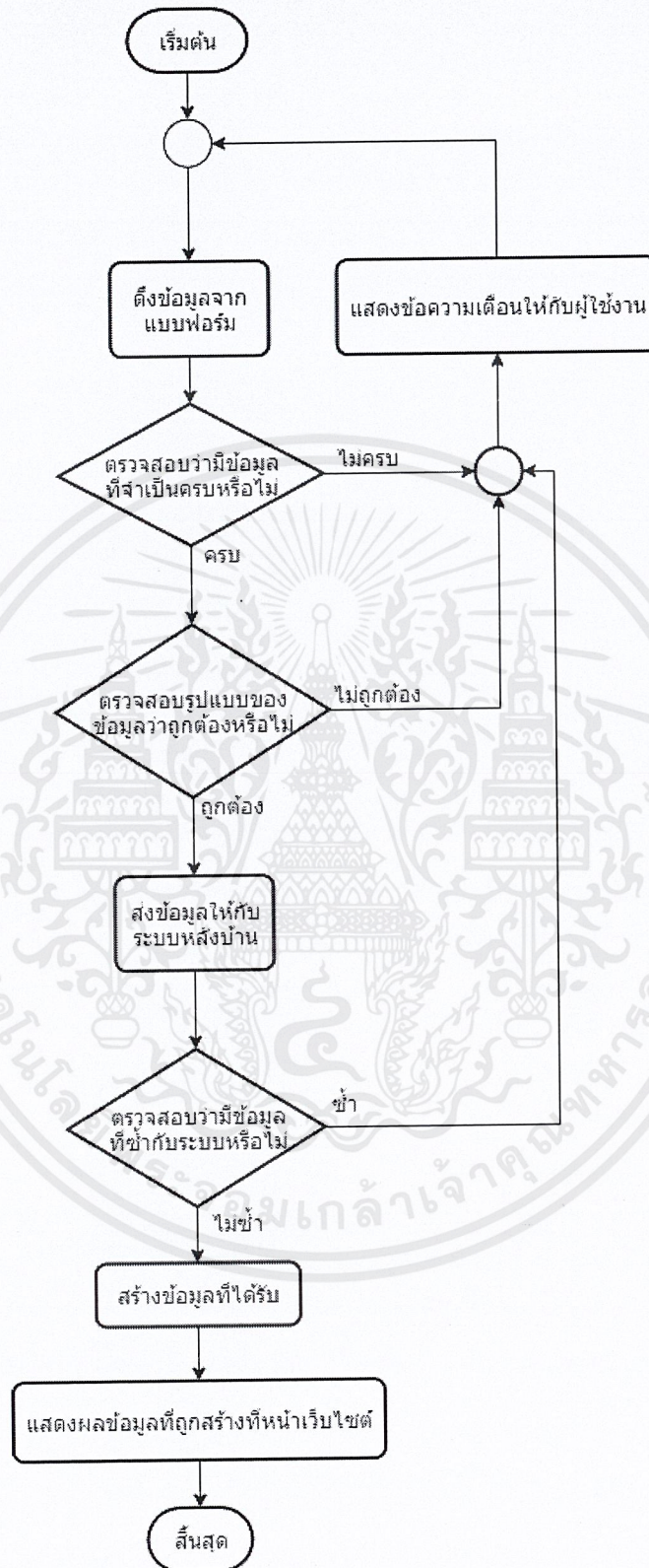
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3. Flowchart



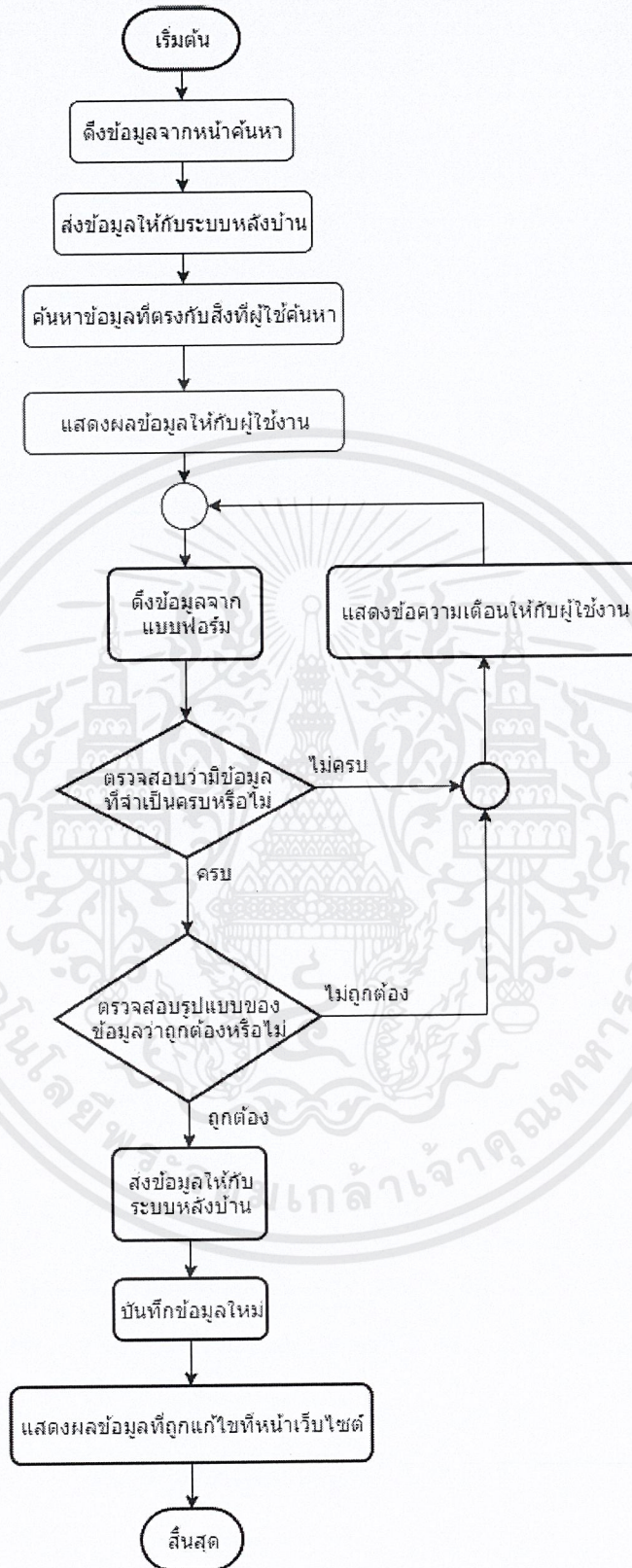
ภาพที่ 3.15 Flowchart การทำงานเมื่อมีการเรียกดูข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



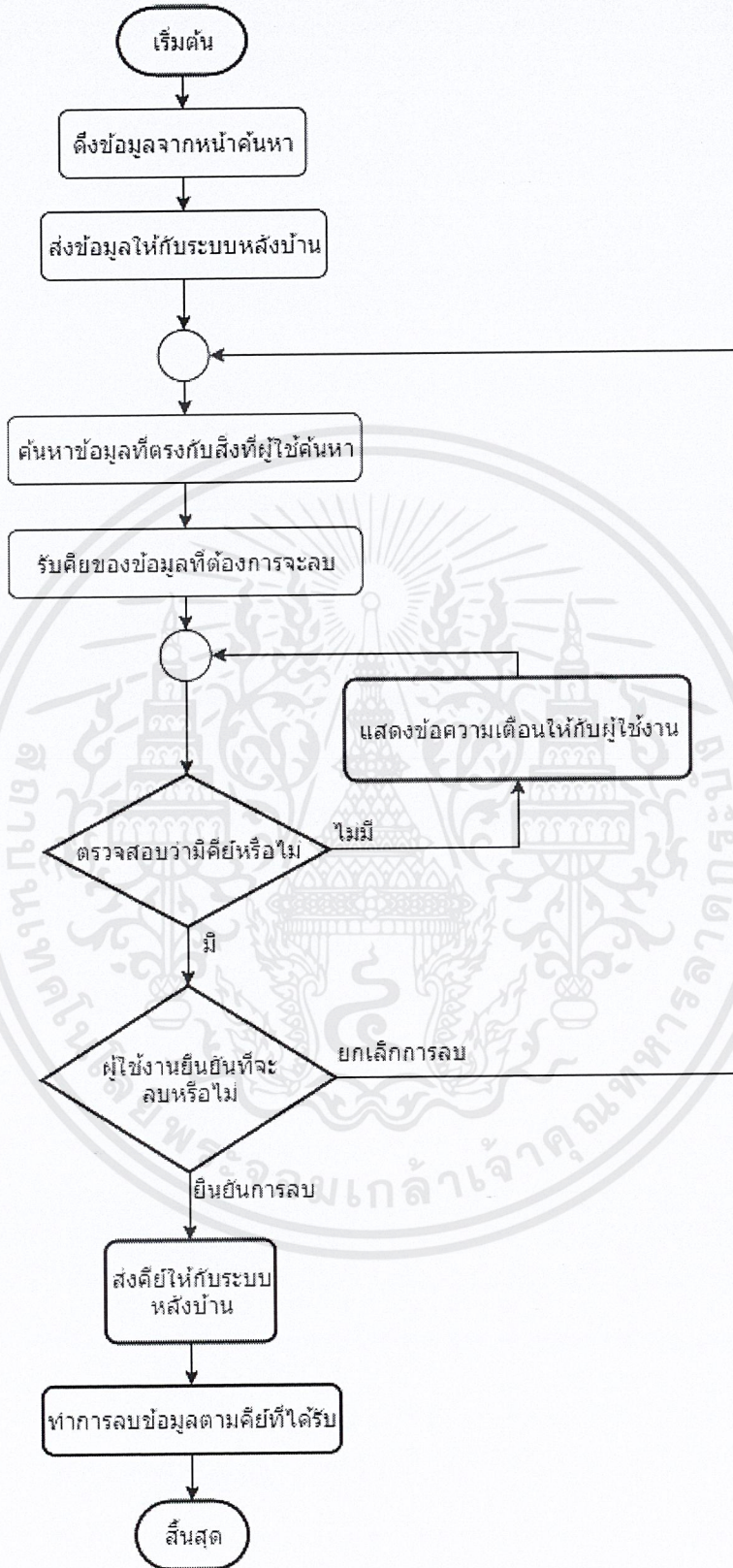
ภาพที่ 3.16 Flowchart แสดงการทำงานเมื่อผู้ใช้ทำการสร้างข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



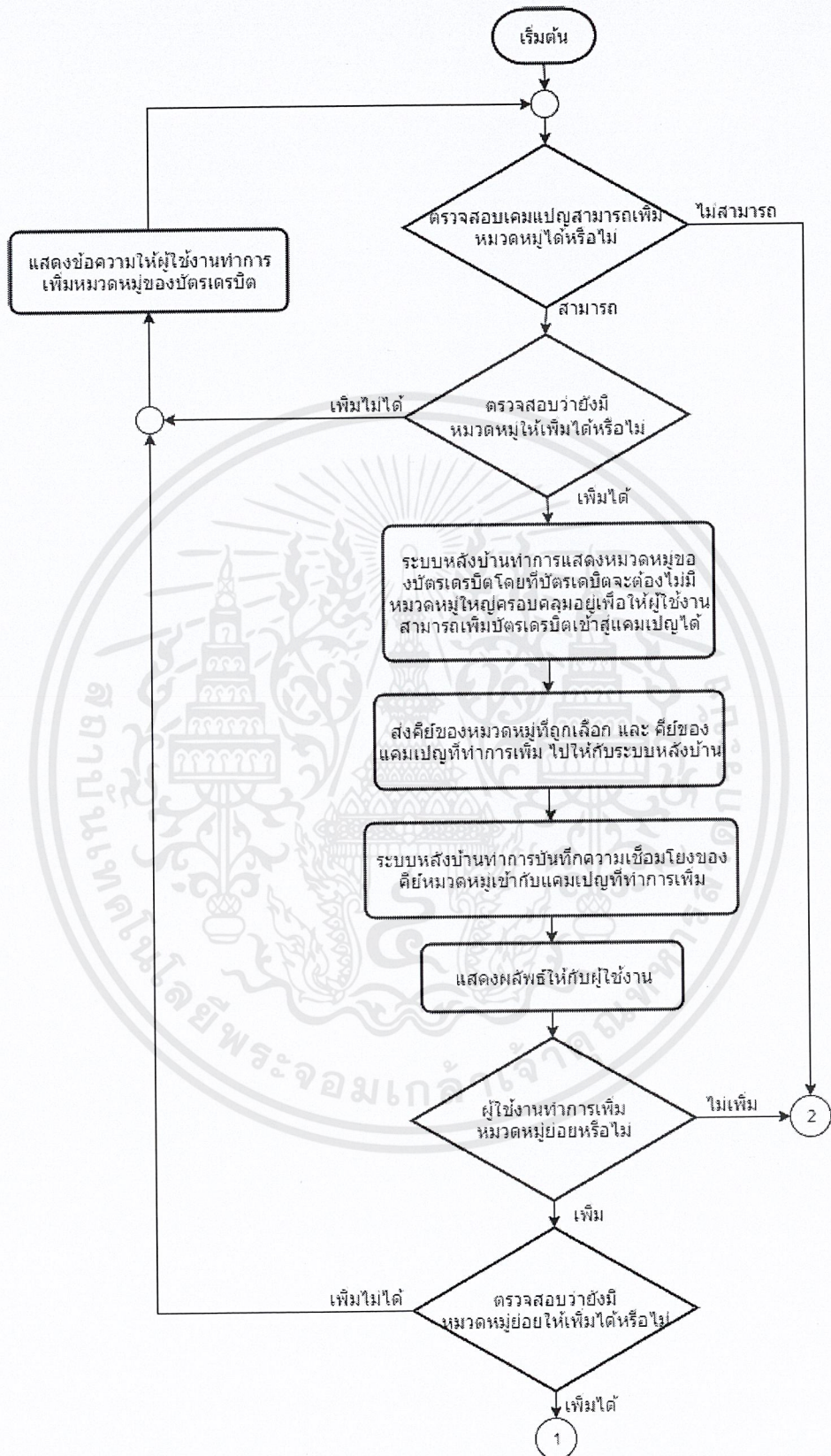
ภาพที่ 3.17 Flowchart แสดงการทำงานเมื่อผู้ใช้ทำการแก้ไขข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



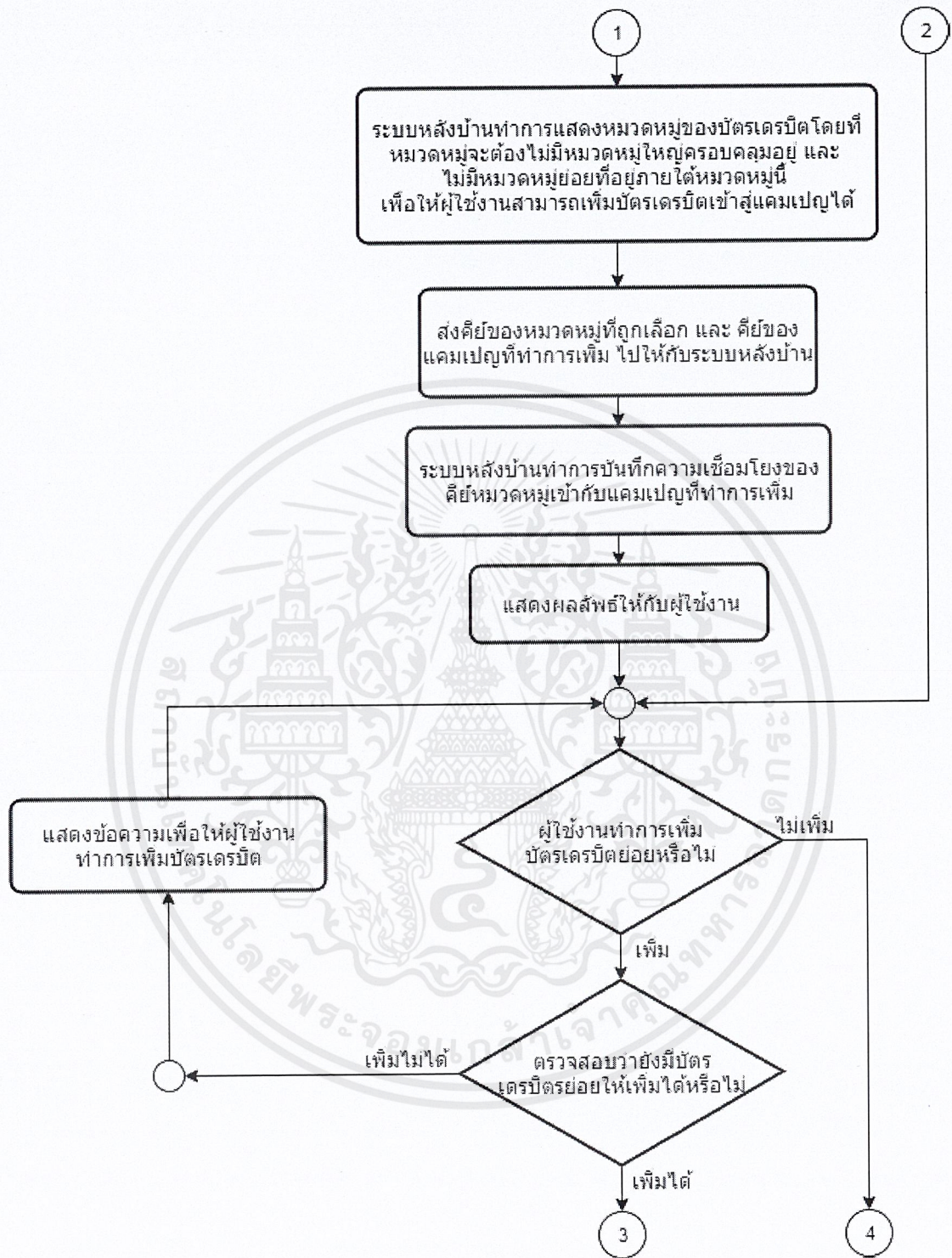
ภาพที่ 3.18 Flowchart แสดงการทำงานเมื่อผู้ใช้ทำการลบข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



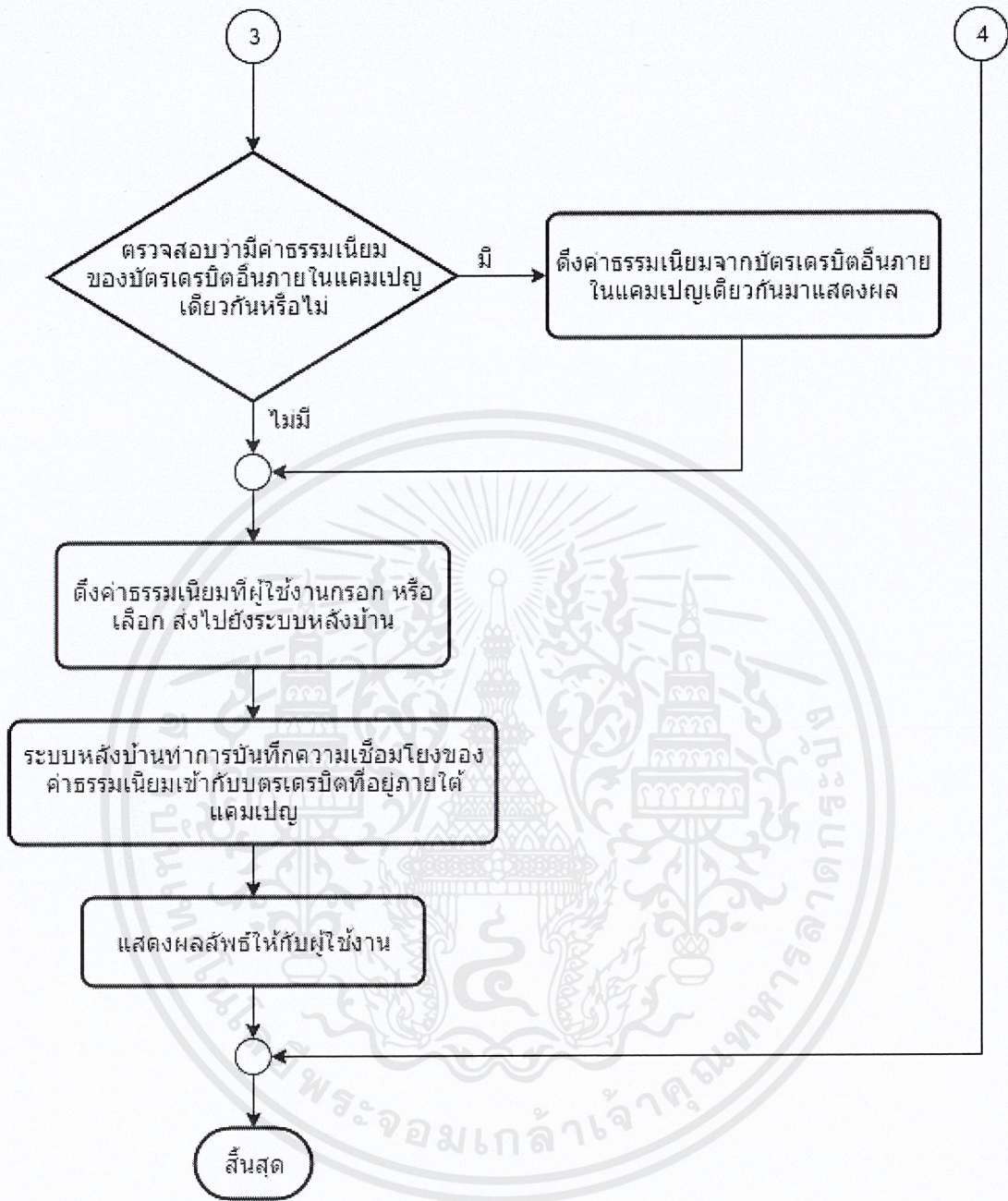
ภาพที่ 3.19 Flowchart แสดงการทำงานเมื่อผู้ใช้งานทำการจัดการบัตรเดบิต (ส่วนที่ 1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.20 Flowchart แสดงการทำงานเมื่อผู้ใช้งานทำการจัดการบัตรเดบิต (ส่วนที่ 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.21 Flowchart แสดงการทำงานเมื่อผู้ใช้งานทำการจัดการบัตรเครบิต (ส่วนที่ 3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 ผลการวิจัย

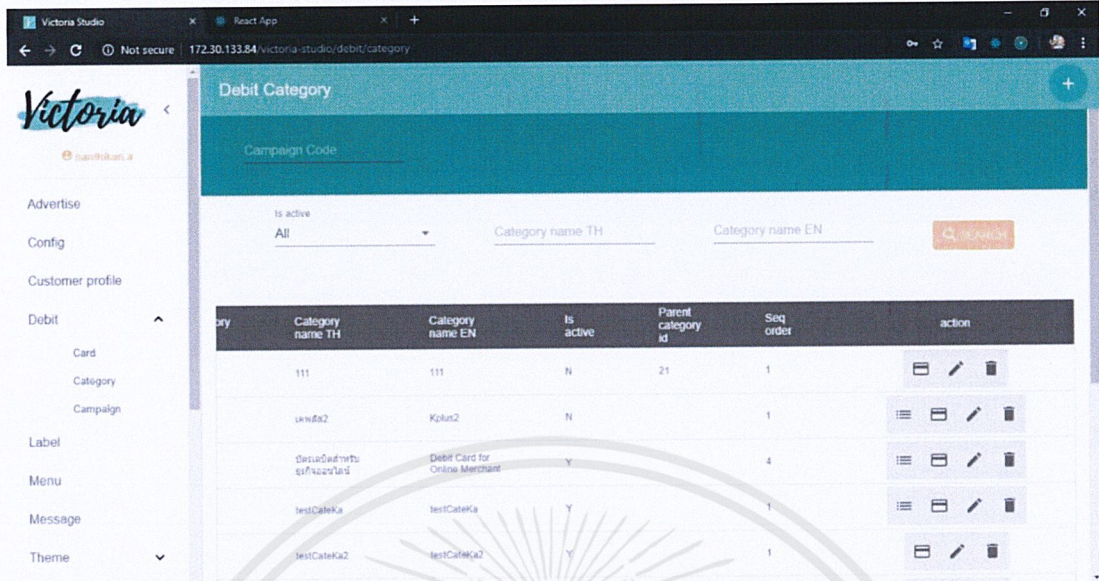
4.1.ภาพรวมของระบบ

ในส่วนของเว็บแอปพลิเคชันตัวเก่า เมื่อได้รับรายละเอียดความต้องการของผู้ใช้งาน และนำไปพัฒนาระบบเพิ่มเติมภายในเว็บแอปพลิเคชันตัวเก่าตามรายละเอียดดังกล่าว ทำให้ได้ภาพรวมระบบเป็นดังนี้

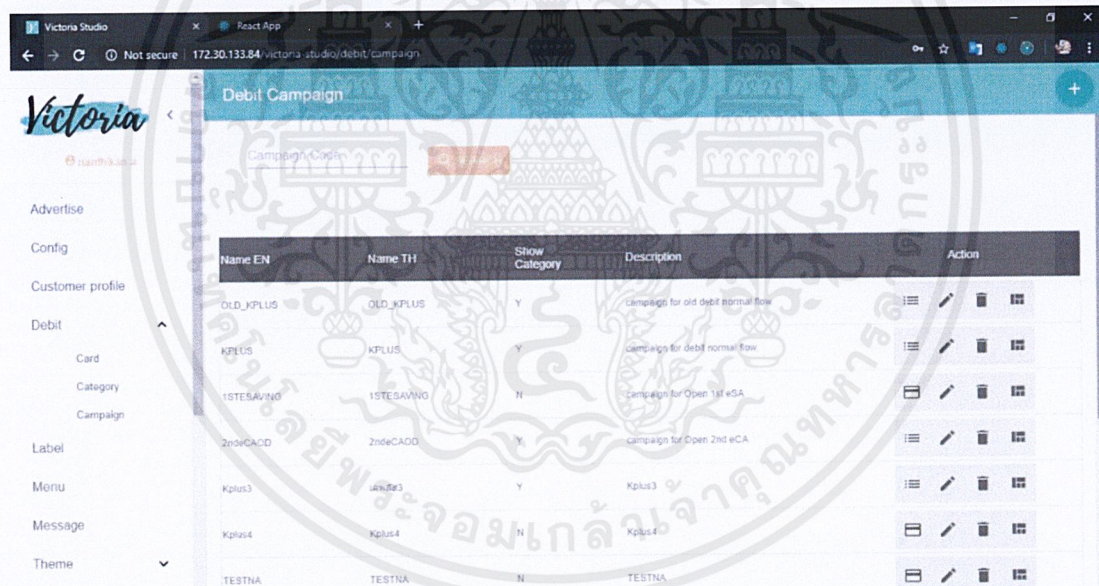


ภาพที่ 4.1 ระบบจัดการบัตรเดบิตในเว็บแอปพลิเคชันตัวเก่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.2 ระบบจัดการหมวดหมู่ของบัตรเดบิตในเว็บแอปพลิเคชันตัวเก่า



ภาพที่ 4.3 ระบบจัดการแคมเปญของบัตรเดบิตในเว็บแอปพลิเคชันตัวเก่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The screenshot shows a web browser window with the URL '172.30.133.84/victoria-studio/debit/campaign'. The page title is 'Campaign Preview'. Below the header, there is a section for 'OLD_KPLUS' with a sub-section 'Debit Card with Insurance'. A table lists fees for a 'K-Max Plus Contactless' card. Below this, there is another section for 'Debit Card with Privilege'.

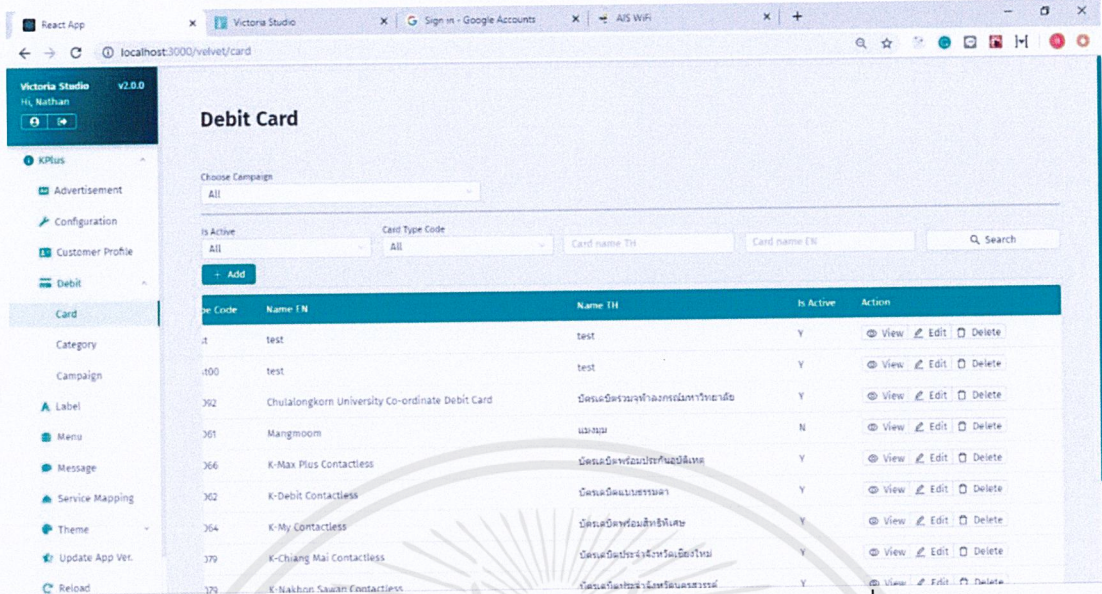
Debit Card with Insurance				
บัตรเดบิตพร้อมประกันอุบัติเหตุ				
CARD	JOINING FEE	ANNUAL FEE	DELIVERY FEE	BRANCH FEE
K-Max Plus Contactless	150.00	700.00	37.00	0.00

Below the table, there is another section for 'Debit Card with Privilege' with the sub-section 'บัตรเดบิตพร้อมสิทธิพิเศษ'.

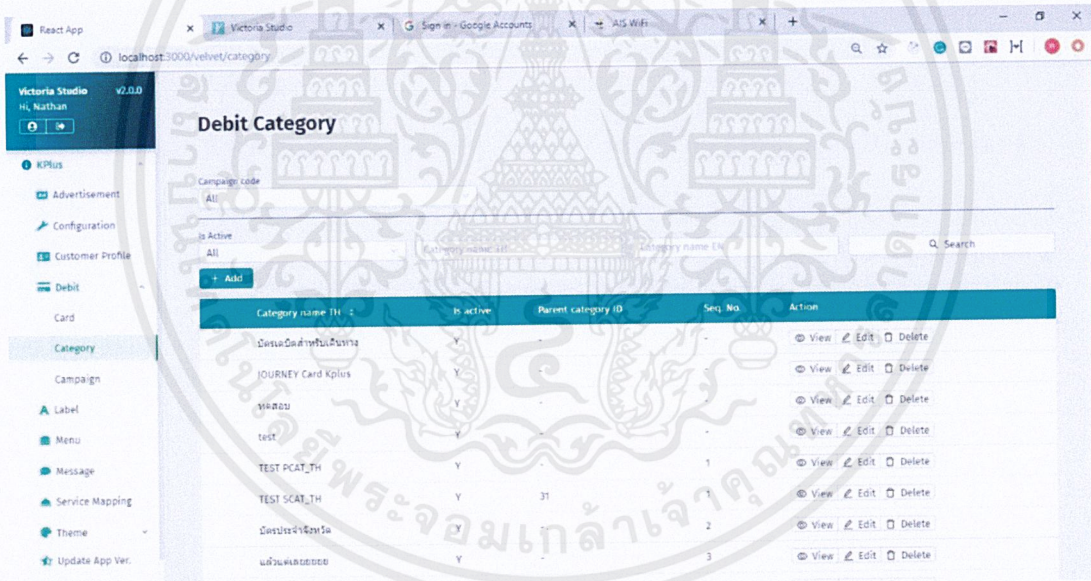
ภาพที่ 4.4 ระบบดูรายละเอียดภายในแคมเปญของบัตรเดบิตในเว็บแอปพลิเคชันตัวเก่า

จากภาพที่ 4.1 ถึง 4.4 เมื่อนำระบบเหล่านี้ไปให้กับผู้ใช้งานได้ลองใช้ พบว่าผู้ใช้งานส่วนใหญ่รู้สึกว่าการที่พัฒนาขึ้นภายในเว็บแอปพลิเคชันตัวเก่านั้นมีความสับสน ซับซ้อน ทำให้ใช้งานยาก เข้าใจยาก เช่น การลบหมวดหมู่ย่อยของบัตรเดบิตที่อยู่ภายใต้หมวดหมู่ของบัตรเดบิตนั้น จะต้องลบภายในหน้าจัดการหมวดหมู่ของบัตรเดบิต ซึ่งภายในหน้านี้ยังสามารถลบหมวดหมู่หลักของบัตรเดบิตได้ โดยความแตกต่างก็คือ หากลบหมวดหมู่ย่อย หมวดหมู่ย่อยจะยังแสดงอยู่ภายในแอปพลิเคชัน K PLUS แต่ถ้าหากทำการลบหมวดหมู่หลัก จะทำการลบหมวดหมู่นั้นๆ และหมวดหมู่ที่อยู่ภายใต้หมวดหมู่หลักออกจากแอปพลิเคชัน K PLUS ออกอย่างถาวร และด้วยเหตุนี้ทำให้เกิดข้อผิดพลาดในการใช้งาน และต้องใช้เวลาในการทำความเข้าใจในการใช้งานเว็บแอปพลิเคชัน ดังนั้นผู้จัดทำจึงได้นำค่าเหล่านี้ไปปรับปรุง พัฒนาในเว็บแอปพลิเคชันตัวใหม่ เพื่อให้มีการใช้งานที่ง่าย และทำความเข้าใจได้ง่ายยิ่งขึ้น ซึ่งได้หน้าตาดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

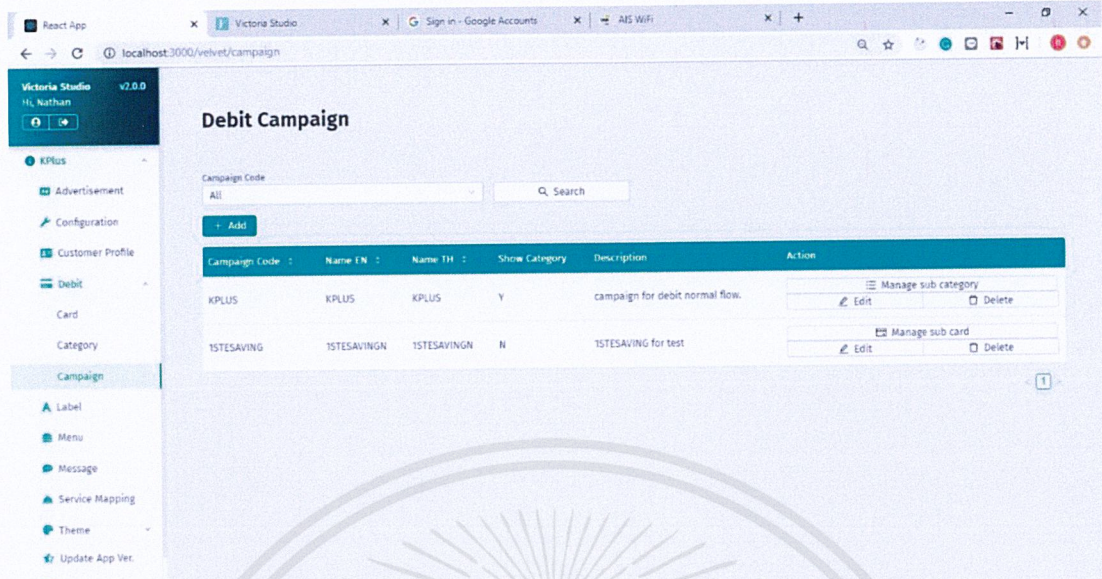


ภาพที่ 4.5 ระบบดูรายละเอียดภายในบัตรเดบิตในเว็บแอปพลิเคชันตัวใหม่

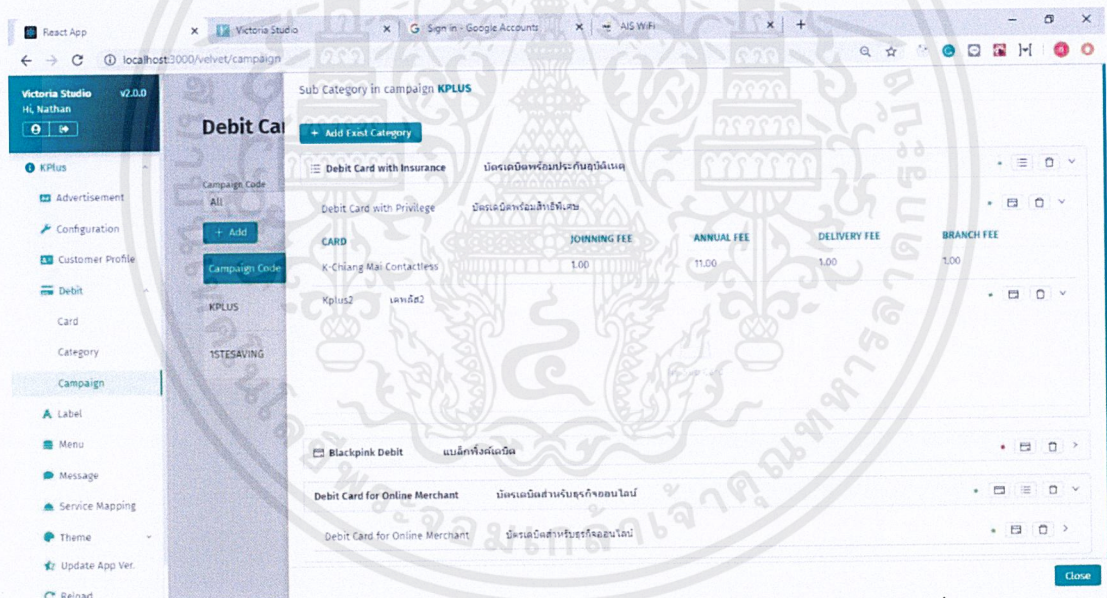


ภาพที่ 4.6 ระบบดูรายละเอียดภายในหมวดหมู่ของบัตรเดบิตในเว็บแอปพลิเคชันตัวใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.7 ระบบรายละเอียดภายในแคมเปญของบัตรเดบิตในเว็บแอปพลิเคชันตัวใหม่

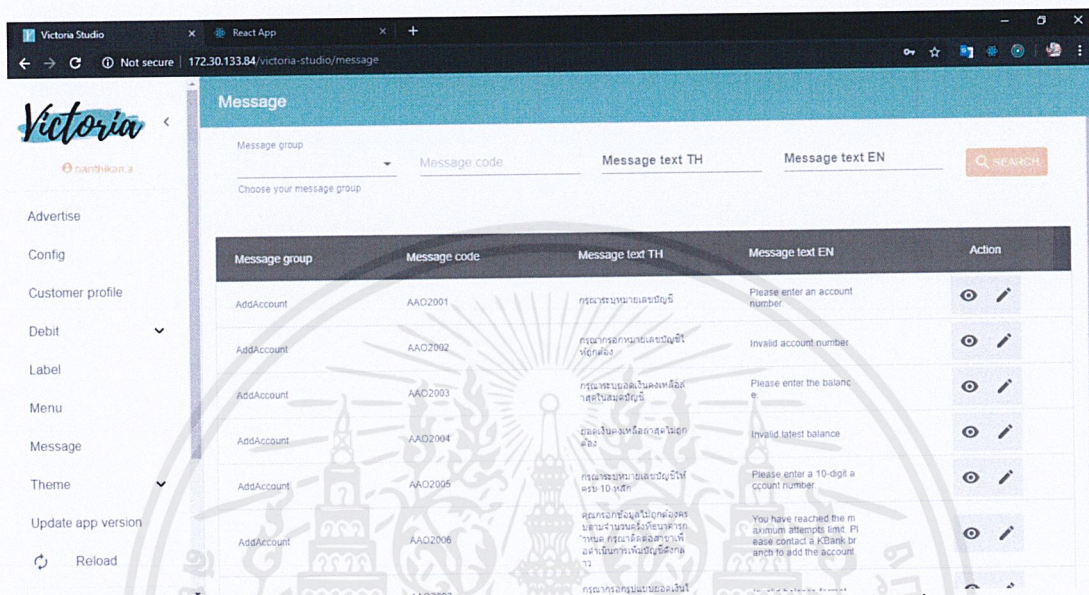


ภาพที่ 4.8 ระบบจัดการข้อมูลภายในแคมเปญของบัตรเดบิตในเว็บแอปพลิเคชันตัวใหม่

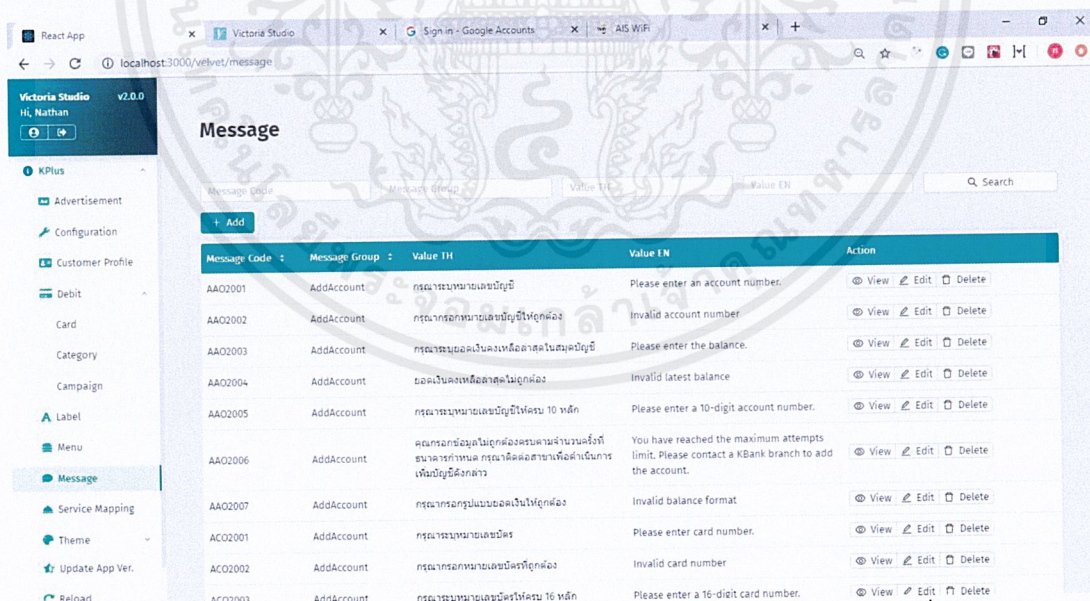
จากภาพที่ 4.5 ถึง ภาพที่ 4.8 จะเห็นว่ารูปแบบหน้าเว็บแอปพลิเคชันตัวใหม่ มีการเปลี่ยนแปลงไปจากเว็บแอปพลิเคชันตัวเก่า เพื่อให้ผู้ใช้งานสามารถใช้ได้ง่ายมากยิ่งขึ้น เข้าใจมากยิ่งขึ้น และไม่ต้องใช้เวลานานในการทำความเข้าใจการใช้งาน เช่น การลบหมวดหมู่ย่อยของบัตรเดบิตได้ทำการย้ายไปภายในหน้าจัดการแคมเปญของบัตรเดบิต เพื่อให้การลบหมวดหมู่ย่อยของบัตรเดบิตและการลบหมวดหมู่หลักของบัตรเดบิตอยู่แยกกัน เพื่อลดความสับสนในการจัดการหมวดหมู่ เป็นต้น และได้มีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปรับเปลี่ยนหน้าอื่นๆ ให้มีความสวยงาม และใช้งานง่ายมากยิ่งขึ้น เนื่องจากภายในเว็บแอปพลิเคชันมีข้อมูลที่ไม่สามารถเผยแพร่ได้ จึงได้นำบางหน้าภายในเว็บแอปพลิเคชันมาแสดง โดยจะนำหน้าจากเว็บแอปพลิเคชันเก่ามาเทียบกับหน้าจากเว็บแอปพลิเคชันอันใหม่เพื่อให้เห็นความเปลี่ยนแปลงดังนี้

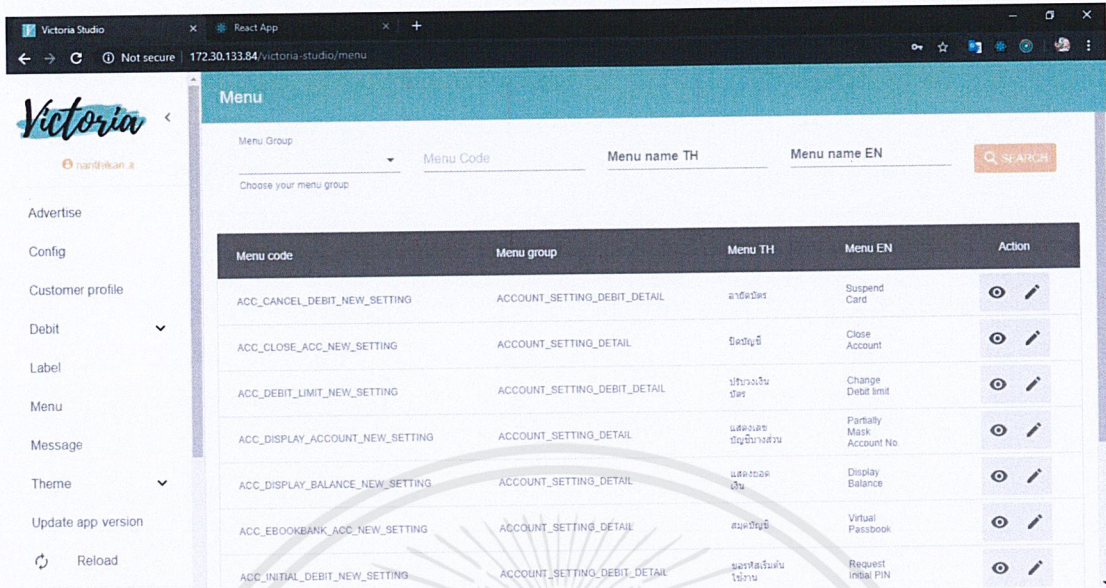


ภาพที่ 4.9 ภาพเปรียบเทียบของหน้าจัดการข้อความภายในเว็บแอปพลิเคชันตัวเก่า

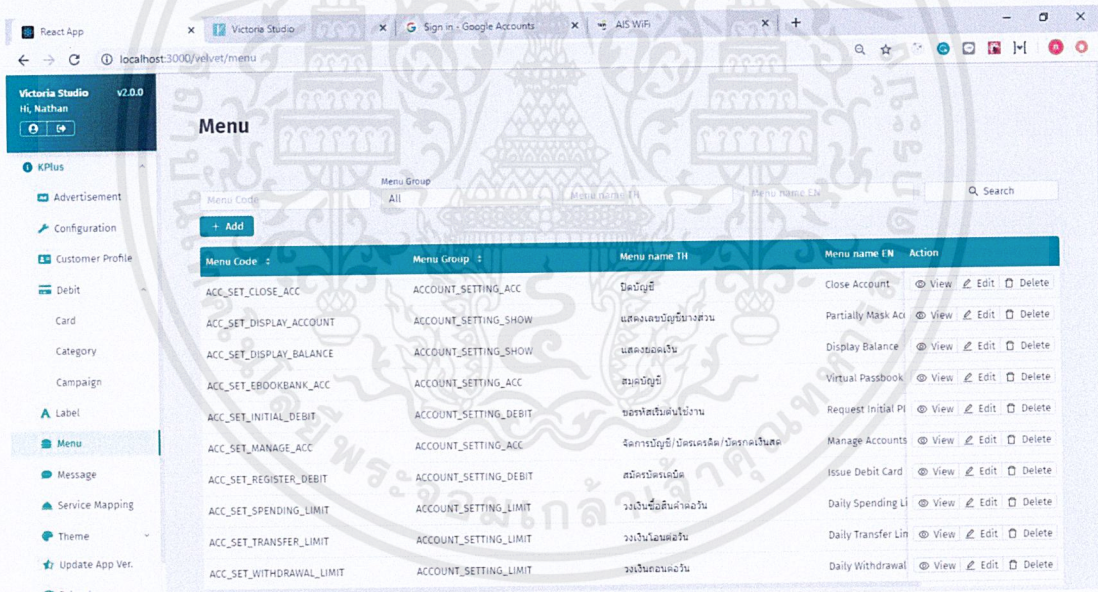


ภาพที่ 4.10 ภาพเปรียบเทียบของหน้าจัดการข้อความภายในเว็บแอปพลิเคชันตัวใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.11 ภาพเปรียบเทียบของหน้าจัดการเมนูภายในเว็บแอปพลิเคชันตัวเก่า



ภาพที่ 4.12 ภาพเปรียบเทียบของหน้าจัดการเมนูภายในเว็บแอปพลิเคชันตัวใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2. ผลการพัฒนาระบบ

เมื่อทำการพัฒนาระบบที่ถูกเพิ่มเติมลงในเว็บแอปพลิเคชันจนเสร็จและนำไปให้ผู้ใช้งานได้ลองใช้งานระบบ พบว่ามีการใช้งานที่ซับซ้อน ทำให้มีความสับสนในการใช้งาน จึงทำให้เกิดข้อผิดพลาดเกิดขึ้นในระหว่างที่ทำการจัดการระบบบัตรเดบิต และต้องใช้เวลาในการทำความเข้าใจในส่วนนี้ค่อนข้างมาก ผู้จัดทำจึงได้นำคำแนะนำเหล่านี้มาทำการปรับปรุงเว็บแอปพลิเคชันตัวใหม่ เพื่อให้ไม่เกิดปัญหาเช่นเดียวกันเกิดขึ้น

เมื่อทำการพัฒนาเว็บแอปพลิเคชันตัวใหม่สำเร็จและได้นำไปให้ผู้ใช้งานได้ลองใช้ระบบ พบว่าตัวเว็บแอปพลิเคชันตัวใหม่นั้นมีสวยงามมากยิ่งขึ้น ใช้งานง่าย และไม่ซับซ้อนเหมือนเว็บแอปพลิเคชันตัวเก่า ทำให้ลดข้อผิดพลาดที่จะเกิดขึ้น และช่วยให้ผู้ใช้งานสามารถจัดการฐานข้อมูลสำหรับการตั้งค่าแอปพลิเคชัน K PLUS ได้ดียิ่งขึ้น

ตารางที่ 4.1 ตารางผลการพัฒนาระบบ

ที่คาดว่าจะพัฒนา	ผลการพัฒนา	
	สำเร็จ	ไม่สำเร็จ
หน้าจัดการบัตรเดบิตบนตัวเว็บแอปพลิเคชันตัวเก่า	/	
หน้าจัดการหมวดหมู่ของบัตรเดบิตบนตัวเว็บแอปพลิเคชันตัวเก่า	/	
หน้าจัดการแคมเปญของบัตรเดบิตบนตัวเว็บแอปพลิเคชันตัวเก่า	/	
หน้าจัดการบัตรเดบิตบนตัวเว็บแอปพลิเคชันตัวใหม่	/	
หน้าจัดการหมวดหมู่ของบัตรเดบิตบนตัวเว็บแอปพลิเคชันตัวใหม่	/	
หน้าจัดการแคมเปญของบัตรเดบิตบนตัวเว็บแอปพลิเคชันตัวใหม่	/	
หน้าจัดการเมนูบนตัวเว็บแอปพลิเคชันตัวใหม่	/	
หน้าจัดการข้อความบนตัวเว็บแอปพลิเคชันตัวใหม่	/	
หน้าจัดการข้อมูลลูกค้าบนตัวเว็บแอปพลิเคชันตัวใหม่	/	
หน้าจัดการค่าตั้งค่าบริการบนตัวเว็บแอปพลิเคชันตัวใหม่	/	
หน้าจัดการป้ายบนตัวเว็บแอปพลิเคชันตัวใหม่	/	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจัดการธีมบนตัวเว็บแอปพลิเคชันตัวใหม่		/
--	--	---

ผู้จัดทำได้กำหนดส่วนที่คาดว่าจะเสร็จทั้งหมด 12 ส่วน ซึ่งเมื่อดำเนินการสามารถทำสำเร็จทั้งหมด 11 ส่วน และส่วนที่ไม่ได้ดำเนินการทั้งหมด 1 ส่วน คิดเป็นร้อยละ 91 ที่ทำสำเร็จ และอีกร้อยละ 9 ที่ไม่ได้ดำเนินการ เนื่องจากระบบส่วนของบัตรเดบิตทั้งเว็บแอปพลิเคชันเก่า และเว็บแอปพลิเคชันใหม่มีระบบที่ค่อนข้างใหญ่ มีความซับซ้อน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1. สรุปผลการดำเนินงาน

ในส่วนของผู้ใช้งานนั้นเมื่อได้ลองใช้งานระบบที่ทำการเพิ่มเติมในเว็บไซต์ตัวเก่าก่อนการทำเว็บไซต์ตัวใหม่แล้วพบว่าการใช้งานค่อนข้างมีความซับซ้อน และสับสน ทำให้บางครั้งเกิดการใช้งานที่ผิดพลาดทำให้เกิดข้อผิดพลาดเกิดขึ้นในขณะที่ทำการจัดการข้อมูล แต่เมื่อได้ลองใช้งานเว็บแอปพลิเคชันตัวใหม่ ผู้ใช้งานพบว่ามีความสวยงามมากยิ่งขึ้น และใช้งานง่ายขึ้นอย่างมาก ทำให้ไม่เกิดข้อผิดพลาดที่เคยเกิดขึ้นในเว็บไซต์ตัวเก่า และทำให้การจัดการฐานข้อมูลสำหรับการตั้งค่าแอปพลิเคชัน K PLUS นั้นทำได้ง่ายขึ้นมากกว่าเว็บแอปพลิเคชันตัวเก่า

หลังจากที่ผู้จัดทำได้พัฒนาเว็บแอปพลิเคชันตัวเก่าและตัวใหม่ให้ผู้จัดทำได้ศึกษาภาษาใหม่ๆ และได้ลองใช้เทคโนโลยีใหม่ๆ ทั้ง TypeScript, Golang และ React โดยระหว่างการพัฒนายังได้นำ Agile เข้ามาใช้งานภายในทีม และได้ฝึกฝนการทำงานร่วมกับผู้อื่น และได้รับความรู้ และประสบการณ์จากผู้อื่น จนทำให้โครงการนี้สามารถสำเร็จตามวัตถุประสงค์ที่ได้กำหนดไว้ ทั้งนี้ ระบบยังไม่ได้ถูกนำไปใช้งานจริง จึงอาจจะยังไม่พบปัญหาจากการใช้งานจริงซึ่งในส่วนนี้จะต้องทำการแก้ไขและพัฒนาให้ดียิ่งขึ้นในโอกาสต่อไป

5.2. ประโยชน์ที่ได้รับจากการปฏิบัติงานสหกิจศึกษา

- 5.2.1. ได้รับความรู้จากผู้มีประสบการณ์ ทั้งด้านประสบการณ์ที่คนๆ นั้นเคยได้รับมา และความรู้ใหม่ๆ ที่สามารถนำไปใช้ในชีวิตประจำวัน
- 5.2.2. ได้เรียนรู้การทำงานในองค์กรใหญ่ๆ ขั้นตอนการทำงานภายใน และระบบการทำงานภายในองค์กรใหญ่ๆ
- 5.2.3. ได้เรียนรู้การทำงานเป็นทีมร่วมกับผู้อื่น การทำงานอย่างเป็นระบบ เป็นขั้นเป็นตอน และรูปแบบการเขียนโปรแกรมที่ดี
- 5.2.4. ได้เรียนรู้ความรู้ใหม่ๆ เกี่ยวกับเทคโนโลยี หรืออุปกรณ์ที่ภายในองค์กรได้ใช้งาน
- 5.2.5. ได้เรียนรู้เทคโนโลยีใหม่ๆ จากเซสชันที่จัดขึ้นโดยพนักงานภายในเพื่อถ่ายทอดให้แก่พนักงานภายในด้วยกัน และเซสชันที่จัดขึ้นภายในทีมเพื่อแบ่งปันความรู้ให้กันและกัน
- 5.2.6. ได้ฝึกฝนการทำงานแบบ Agile ทั้งการทำงานเป็น sprint การมี standup meeting ทุกๆ เช้า เพื่อบอกว่าทำอะไรเสร็จไปแล้วบ้าง และตอนนี้จะทำอะไร
- 5.2.7. ได้นำความรู้ที่ได้รับภายในห้องเรียนมาใช้ในการทำงานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5.2.8. ได้ฝึกการทำงานทั้งทักษะด้านสังคม (Soft Skill) และทักษะด้านความรู้ (Hard Skill)
- 5.2.9. ได้ฝึกการเขียนทดสอบระบบในส่วนเล็กๆ (Unit Test) ทำให้สามารถลดข้อผิดพลาดเพื่อทำการพัฒนาระบบให้มีขนาดใหญ่ยิ่งขึ้น หรือลดการเกิดข้อผิดพลาดเวลาทำการแก้ไขระบบ
- 5.2.10. ได้รู้จักและได้รับมิตรภาพจากเพื่อนต่างสถาบัน ได้แลกเปลี่ยนความรู้และประสบการณ์ให้แกกัน
- 5.2.11. ได้รับประสบการณ์จริงจากชีวิตการทำงาน ทำให้ตนเองมีความรับผิดชอบทั้งต่อตนเองและผู้อื่นเพิ่มขึ้น ทั้งเรื่องการบริหารเวลา การตรงต่อเวลา มีความรอบคอบในการทำงานเพิ่มขึ้น

5.3. ปัญหาและอุปสรรคที่พบ

- 5.3.1. ในช่วงแรกของการทำงาน ผู้จัดทำมีประสบการณ์ทำงานด้านเว็บแอปพลิเคชันที่ค่อนข้างน้อย ทำให้การทำงาน และการศึกษางานต่างๆ ค่อนข้างช้า
- 5.3.2. หลายๆ เครื่องมือที่ถูกใช้ภายในบริษัท เป็นเครื่องมือที่ผู้จัดทำไม่เคยใช้หรือรู้จัก ทำให้ต้องใช้เวลาในการศึกษา
- 5.3.3. เนื่องจากไม่เคยทำงานจริงมาก่อน เมื่อเริ่มทำงานทำให้เริ่มต้นมีความรู้สึกกังวลในการทำสิ่งต่างๆ จึงต้องใช้เวลาในการปรับตัวกับที่ทำงาน และพี่เลี้ยงที่คอยดูแลเรา
- 5.3.4. ช่วงระหว่างทำงานเนื่องจากผู้จัดทำไม่เคยพัฒนาโปรแกรมให้ผู้อื่นได้ใช้งาน จึงทำให้เวลาพัฒนาเว็บไซต์ไม่มีความรอบคอบเพียงพอ ทำให้พบข้อผิดพลาดระหว่างที่ผู้ใช้งานทดสอบระบบ

5.4. ข้อเสนอแนะ

- 5.4.1. ปรับให้มีการรองรับหน้าจอหลายขนาดมากขึ้น
- 5.4.2. ปรับให้มีการดึงข้อมูล และแสดงผลได้ไวขึ้น เพื่อรองรับระบบในอนาคตที่อาจจะมียุติข้อมูลที่มากขึ้น
- 5.4.3. มีการปรับแต่งรูปร่างของเว็บแอปพลิเคชันให้มีความสวยงามมากขึ้น และเข้าใจง่ายยิ่งขึ้น
- 5.4.4. มีระบบสอนใช้งานให้กับระบบที่มีความซับซ้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- Nutti Saelor. (2560). MVC MVP MVVM คืออะไร และต่างกันอย่างไร. [ออนไลน์]. ได้จาก: <https://medium.com/@leelorz6/mvc-mvp-mvvm-คืออะไร-และต่างกันอย่างไร-ca16a19631dc> [สืบค้นเมื่อ วันที่ 10 ธันวาคม 2562].
- Travis P. (2560). MVP คืออะไร แล้วเกี่ยวอะไรกับ Android. [ออนไลน์]. ได้จาก: <https://blacklenspub.com/mvp-คืออะไร-แล้วเกี่ยวอะไรกับ-android-7a0460d7cd49> [สืบค้นเมื่อ วันที่ 10 ธันวาคม 2562].
- SQL Server คืออะไร เอสคิวแอลเซิร์ฟเวอร์ คือระบบจัดการฐานข้อมูลเชิงสัมพันธ์. (2560). [ออนไลน์]. ได้จาก: <https://www.mindphp.com/คู่มือ/73-คืออะไร/2266-sql-server-คืออะไร.html> [สืบค้นเมื่อ วันที่ 4 ธันวาคม 2562].
- Designil. (2556). Front-End Framework คืออะไร? เปรียบเทียบ Twitter Bootstrap VS ZURB Foundation !!. [ออนไลน์]. ได้จาก: <https://www.designil.com/front-end-framework-twitter-bootstrap-zurb-foundation.html> [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2556].
- Nattavut Thongjor. (2560). เรียนรู้การใช้ภาษา Go ใน 15 นาที. [ออนไลน์]. ได้จาก: <https://www.babelcoder.com/blog/posts/intro-to-golang> [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].
- somkiat. (2560). ทำไมถึงใช้ภาษา Go ?. [ออนไลน์]. ได้จาก: <http://www.somkiat.cc/why-i-use-go/> [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].
- Palm Korrawit. (2555). TypeScript ภาษาตัวแทน JavaScript จาก Microsoft. [ออนไลน์]. ได้จาก: <https://blog.sogoodweb.com/Article/Detail/54455> [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].
- Nuttavut Thongjor. (2559). [TypeScript#1] TypeScript คืออะไร? เรียนรู้ชนิดข้อมูลพื้นฐานของ TypeScript. [ออนไลน์]. ได้จาก: <https://www.babelcoder.com/blog/posts/typescript-data-types> [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].
- Enjoyday. (2552). JavaScript คืออะไร. [ออนไลน์]. ได้จาก: <http://www.enjoyday.net/javascript-คืออะไร.html> [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562]
- Enjoyday. (2552). HTML คืออะไร. [ออนไลน์]. ได้จาก: http://www.enjoyday.net/webtutorial/html/html_chapter01.html [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].

- JavaScript คืออะไร จาวา สคริปต์ คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบ
อินเทอร์เน็ต. (2560). [ออนไลน์]. ได้จาก: <https://www.mindphp.com/คู่มือ/73-คืออะไร/2187-java-javascript-คืออะไร.html> [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].
- สวัสดีครับ ผมมีชื่อว่า “React”. (2560). [ออนไลน์]. ได้จาก: <https://microbenz.in.th/สวัสดีครับ-ผมมีชื่อว่า-react-3e8fd72ccdbb> [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].
- Chai Phonbopit. (2558). React คืออะไร? + เริ่มต้นเขียน React. [ออนไลน์]. ได้จาก:
<https://devahoy.com/blog/2015/11/getting-started-with-reactjs/> [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].
- Thamonwan Maneechan. (2562). Sass คืออะไร? (overview). [ออนไลน์]. ได้จาก:
<https://www.thamonwan.top/what-is-css-sass/> [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].
- Designil. (2562). SASS คืออะไร? เทคนิคช่วยให้การเขียน CSS เป็นเรื่องง่าย ๆ ใช้กันทั่วโลก !!
[ออนไลน์]. ได้จาก: <https://www.designil.com/sass-css-scss-tutorial.html> [สืบค้นเมื่อ
วันที่ 11 ธันวาคม 2562].
- อัจฉราพรรณ เชี่ยวภู. (2559). CSS3 คืออะไร. [ออนไลน์]. ได้จาก: [https://www.designil.com/sass-
css-scss-tutorial.html](https://www.designil.com/sass-css-scss-tutorial.html) [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].
- HTML5 คืออะไร. (2554). [ออนไลน์]. ได้จาก: <https://www.softmelt.com/article.php?id=404>
[สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].
- วัฏจักรการพัฒนาซอฟต์แวร์: Software Development Life Cycle. (2560). [ออนไลน์]. ได้จาก:
[https://www.9chor.com/2017/11/19/software-development-life-
cycle/#comments](https://www.9chor.com/2017/11/19/software-development-life-cycle/#comments) [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].
- ter@digitalcooking. (2560). W3C คืออะไร. [ออนไลน์]. ได้จาก:
https://medium.com/@ter_dgc/w3c-คืออะไร-b5336f7318c9 [สืบค้นเมื่อ วันที่ 11
ธันวาคม 2562].
- SDCL (Software Development Life Cycle) using Agile. (2558). [ออนไลน์]. ได้จาก:
<http://www.riverpark.co.th/blog/agilesdlc.html> [สืบค้นเมื่อ วันที่ 11 ธันวาคม 2562].
- Sakul Montha. (2560). REST กับ RESTful API ต่างกันนะรู้ยัง. [ออนไลน์]. ได้จาก:
[https://medium.com/@iamgique/restful-api-กับ-rest-api-ต่างกันนะรู้ยัง-
2c70c42990e3](https://medium.com/@iamgique/restful-api-กับ-rest-api-ต่างกันนะรู้ยัง-2c70c42990e3) [สืบค้นเมื่อ วันที่ 14 ธันวาคม 2562].
- NICH. (2562). RESTful API คืออะไร. [ออนไลน์]. ได้จาก: [http://www.4x-treme.com/restful-api-
คืออะไร/?lang=th](http://www.4x-treme.com/restful-api-คืออะไร/?lang=th) [สืบค้นเมื่อ วันที่ 14 ธันวาคม 2562].

Todspol Wonhchomphu. (2560). SQL คืออะไร? และ NOSQL คืออะไร?. [ออนไลน์]. ได้จาก: <https://medium.com/@todspolwonhchomphu/sql-คืออะไร-และ-nosql-คืออะไร-561c750dbe4d> [สืบค้นเมื่อ วันที่ 14 ธันวาคม 2562].

Saixiii. (2560). SQL คืออะไร Structured Query Language คือ คำสั่งบริหารจัดการ database. [ออนไลน์]. ได้จาก: <https://saixiii.com/sql-statement/> [สืบค้นเมื่อ วันที่ 14 ธันวาคม 2562].

Mdsoft. (2559). ทำความรู้จักกับ Web Application (เว็บแอปพลิเคชัน). [ออนไลน์]. ได้จาก: <https://mdsoft.co.th/ความรู้/359-web-application.html> [สืบค้นเมื่อ วันที่ 14 ธันวาคม 2562].

Zendesk. (2562). UX / UI คืออะไร. [ออนไลน์]. ได้จาก: <https://www.dmit.co.th/th/blog/2019/05/29/ux-vs-ui/> [สืบค้นเมื่อ วันที่ 14 ธันวาคม 2562].

Diakunwadee. (2561). UI vs. UX. [ออนไลน์]. ได้จาก: <https://www.thaiprogrammer.org/2018/12/ui-vs-ux-อะไรคือความแตกต่างระหว่าง/> [สืบค้นเมื่อ วันที่ 14 ธันวาคม 2562].

9Expert. (2019). UX Design คืออะไร. [ออนไลน์]. ได้จาก: <https://www.9experttraining.com/articles/ux-design-คืออะไร> [สืบค้นเมื่อ วันที่ 14 ธันวาคม 2562].

Saixiii. (2560). ทำความรู้จัก Java คืออะไร. [ออนไลน์]. ได้จาก: <https://saixiii.com/java-programming/> [สืบค้นเมื่อ วันที่ 14 ธันวาคม 2562].

mindphp. (2560). Java คืออะไร. [ออนไลน์]. ได้จาก: <https://www.mindphp.com/คู่มือ/73-คืออะไร/2185-java-คืออะไร.html> [สืบค้นเมื่อ วันที่ 14 ธันวาคม 2562].