

ระบบจองคิวร้านเสริมสวย
Beauty salon queue reservation system

นางสาว ปิยพร อามรศรี

โครงงานนี้เป็นส่วนหนึ่งของการศึกษา
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยอุบลราชธานี
ปีการศึกษา 2562
ลิขสิทธิ์ของมหาวิทยาลัยอุบลราชธานี

โครงการ : ระบบจองคิวร้านเสริมสวย
Beauty salon queue reservation system
โดย : นางสาว ปิยพร อามรศรี
อาจารย์ที่ปรึกษา : ดร.ทศพร จูนิม
ระดับการศึกษา : วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์
ปีการศึกษา : 2562

ได้รับการพิจารณาให้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรวิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์

คณะกรรมการสอบประเมินความรู้โครงการคอมพิวเตอร์

..... อาจารย์ที่ปรึกษา
(ดร.ทศพร จูนิม)

..... กรรมการ
(ดร.วรารุณี ฟ้าเจริญ)

..... กรรมการ
(ดร.ไพชยนต์ คงไชย)

..... หัวหน้าภาควิชา
(ผศ.ดร. สุพจน์ สืบคร)

วันที่ ... / ... / ...

กิตติกรรมประกาศ

การพัฒนาโครงการระบบการจองคิวร้านเสริมสวย สำเร็จลุล่วงได้ด้วยความรู้และความช่วยเหลือจากหลายๆ ฝ่าย ถึงแม้จะประสบกับปัญหาในการทำโครงการก็ได้รับคำแนะนำและการช่วยเหลือที่ดีเสมอมา ข้าพเจ้าจึงใคร่ขอขอบคุณทุกๆ ท่าน ที่มีส่วนร่วมในการพัฒนาโครงการนี้

ขอขอบพระคุณอาจารย์ ดร.ทศพร จูฉิม อาจารย์ที่ปรึกษาโครงการที่ได้แนะนำทฤษฎีและแนวทางในแก้ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการพัฒนาและคอยตรวจสอบความก้าวหน้าของโครงการ รวมทั้งยังคอยให้กำลังใจในการพัฒนาโครงการนี้ตลอดจนโครงการเสร็จสมบูรณ์

ขอขอบพระคุณอาจารย์ประจำสาขาวิทยาการคอมพิวเตอร์ อาจารย์ประจำภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ และอาจารย์ในคณะวิทยาศาสตร์ทุก ๆ ท่าน ที่คอยให้คำแนะนำ อบรมสั่งสอน และคอยช่วยเหลือข้าพเจ้าในการศึกษาตลอดมาขอบคุณเจ้าหน้าที่และบุคลากรของคณะวิทยาศาสตร์ ที่ได้อำนวยความสะดวกทางด้านอุปกรณ์และเครื่องมือต่าง ๆ

ขอขอบพระคุณบิดา มารดา ที่เป็นกำลังใจ คอยให้ความรักและความห่วงใยเสมอมา ตลอดจนคอยช่วยเหลือทุนทรัพย์ทางการศึกษา และอุปกรณ์ในการพัฒนาโครงการ

ขอบคุณเพื่อน ๆ สาขาวิทยาการคอมพิวเตอร์ชั้นปีที่ 4 ที่ได้คอยช่วยแก้ไขปัญหาและให้คำปรึกษาในการพัฒนาโครงการครั้งนี้จนเสร็จสิ้น

นางสาวปิยพร อารศรี

วันที่ 28 กุมภาพันธ์ 63

โครงการ	:	ระบบจองคิวร้านเสริมสวย
โดย	:	นางสาว ปิยพร อามรศรี
อาจารย์ที่ปรึกษา	:	ดร.ทศพร จูนิม
ระดับการศึกษา	:	วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา	:	2562

บทคัดย่อ

การเสริมสวยเป็นที่นิยมและเป็นธุรกิจที่ได้รับความนิยมจากลูกค้าเป็นจำนวนมากในปัจจุบัน โดยปกติเมื่อลูกค้าต้องการใช้บริการจะมาที่ร้านโดยไม่ได้นัดหมาย ซึ่งปัญหาคือร้านมีลูกค้าที่ใช้บริการอยู่ในขณะนั้นทำให้ต้องผู้ที่เข้ามาโดยไม่ได้นัดต้องรอคิวหรือถ้ามีลูกค้ากำลังรอรับบริการอยู่เป็นจำนวนมากอาจทำให้ต้องมาใช้บริการในวันอื่นแทน แม้ในบางกรณีที่ลูกค้าโทรมาสอบถามเพื่อทำการจองคิวล่วงหน้า แต่ช่างไม่สะดวกรับโทรศัพท์เนื่องจากกำลังให้บริการลูกค้าคนอื่นอยู่ ดังนั้นผู้พัฒนาจึงมีแนวคิดสร้างเว็บแอปพลิเคชันระบบการจองคิวร้านเสริมสวยขึ้น เพื่อช่วยจัดการปัญหาดังกล่าว ระบบนี้ถูกพัฒนาด้วย React framework , material ui , nodejs และ MySQL โดยระบบสามารถให้ลูกค้าทำการจองคิว ดูคิวว่างของร้านเสริมสวย หาดำแหน่งของร้าน ดูข้อมูลทั่วไปของร้าน และเขียนรีวิวได้ ในส่วนของเจ้าของร้าน สามารถเพิ่มข้อมูลทั่วไปของร้าน และจัดการการจองคิวของร้านเสริมสวยได้ ระบบการจองคิวรองรับการแสดงผลบนอุปกรณ์สมาร์ตโฟนและเว็บเบราว์เซอร์ ระบบที่พัฒนาขึ้นจะช่วยอำนวยความสะดวกในการนัดหมายล่วงหน้า ลดการรอคิวของผู้ใช้บริการ และช่วยให้การจองคิวมีระเบียบมากขึ้น

คำสำคัญ: เว็บแอปพลิเคชัน , ระบบจองคิวร้านเสริมสวย

Topic	:	Beauty salon queue reservation system
Author	:	PIYAPHORN ARPHORNSRI
Advisor	:	Tossaporn Joochim, Ph.D.
Degree	:	Bachelor of Science (Computer Science)
Academic Year	:	2019

Abstract

Beauty is popular and is a business that has been popular with many customers today. Usually when customers want to use the service, will come to the shop without an appointment. The problem is that the shop has customers who use the service at that time, causing those who enter without an appointment to wait in the queue. Or if there are many customers waiting to receive the service, may cause to use the service on another day instead Even in some cases where users call to inquire to reserve a queue in advance But the technician was not comfortable to answer the phone because he was serving other customers Therefore, the developer has the idea to create a salon queue reservation system. To help manage the said problem This system was developed with the React framework, material ui, nodejs and MySQL. The system allows customers to reserve a queue. See the salon queue. Find a store location See general information of the store And can write a review In the part of the store owner Can add general information of the shop And can manage the beauty salon queue reservations The queue reservation system supports display on Smart Phon devices and web browsers. The developed system will help facilitate advance appointments. Reduce user waiting And helps to make queue reservations more organized.

Keywords: Web Application,Beauty salon queue reservation system

สารบัญ

	หน้า
กิตติกรรมประกาศ	ค
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
สารบัญ	ฉ
สารบัญตาราง	ญ
สารบัญภาพ	ฎ
บทที่	
1 บทนำ	1
1.1 ที่มาและเหตุผล	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
1.5 เครื่องมือที่ใช้ในการพัฒนา (Development tools)	3
1.5.1 ฮาร์ดแวร์	3
1.5.2 ซอฟต์แวร์ (Software)	3
1.6 แผนการดำเนินการ	4
2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 ความรู้พื้นฐานเกี่ยวกับ React	5
2.1.1 React Life Cycle	6
2.1.2 React Life Cycle	7
2.2 ความรู้พื้นฐานเกี่ยวกับ Node.js	8
2.2.1 node.js ทำงานแบบ event driven	9
2.2.2 ข้อดีของ Node.js	10
2.3 ความรู้พื้นฐานเกี่ยวกับ JavaScript	10
2.3.1 ประโยชน์ของ JavaScript	11
2.3.2 ข้อดีและข้อเสียของ Java JavaScript	11
2.4 ความรู้พื้นฐานเกี่ยวกับ MySQL	12
2.4.1 ข้อดีของMySQL	12
2.5 ความรู้เกี่ยวกับ Visual Studio Code	12

2.6	ความรู้พื้นฐานเกี่ยวกับ Google Maps API	13
2.7	เอกสารและงานวิจัยที่เกี่ยวข้อง	15
2.7.1	เว็บแอปพลิเคชัน Gowabi	15
2.7.2	ข้อแตกต่างระหว่างเว็บแอปพลิเคชัน wongnai กับเว็บของโครงการ . . .	15
3	การวิเคราะห์และออกแบบระบบ	16
3.1	โครงสร้างภาพรวมของระบบ	16
3.2	System Requirements	19
3.2.1	ความต้องการหลักของโปรแกรม (Functional Requirements)	19
3.2.2	ความต้องการด้านคุณภาพ (Non-functional Requirements)	19
3.3	การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface Design)	20
3.3.1	ส่วนของผู้ใช้งาน	20
3.3.2	ส่วนของผู้ดูแลระบบ	33
3.4	Use Case Diagram	36
3.5	Class Diagram	46
3.6	Sequence Diagram	54
3.6.1	Sequence Diagram ของการสมัครสมาชิก	55
3.6.2	Sequence Diagram ของการเข้าสู่ระบบ	56
3.6.3	Sequence Diagram ของการดูข้อมูลรายละเอียดสถานที่	57
3.6.4	Sequence Diagram ของการเขียนรีวิวให้คะแนนสถานที่	58
3.6.5	Sequence Diagram ของการแจ้งแก้ไขข้อมูลสถานที่	59
3.6.6	Sequence Diagram ของการเพิ่มข้อมูลสถานที่	60
3.6.7	Sequence Diagram ของการกรอกแบบประเมิน	61
3.6.8	Sequence Diagram ของการดูผลลัพธ์การจัดอันดับ	62
3.6.9	Sequence Diagram ของการจัดการแก้ไขข้อมูลสถานที่	63
3.6.10	Sequence Diagram ของการจัดการลบข้อมูลสถานที่	64
3.6.11	Sequence Diagram ของการจัดการแจ้งแก้ไขข้อมูลสถานที่	65
3.6.12	Sequence Diagram ของการจัดการ train model	66
3.7	โครงสร้างฐานข้อมูลคาสเซิล(Kuzzle Database Structure)	67
4	การพัฒนาระบบ	85
4.1	การพัฒนาในส่วนสมัครสมาชิก	85

4.2	การพัฒนาในส่วนการเข้าสู่ระบบ	87
4.3	การพัฒนาในส่วนการจัดอันดับสถานที่	89
4.4	การพัฒนาในส่วนของการเขียนรีวิว	93
4.5	การพัฒนาในส่วนของการเพิ่มสถานที่	99
4.6	การพัฒนาในส่วนของการลบรีวิว	103
4.7	การพัฒนาในส่วนของการบันทึกสถานที่	105
4.8	การพัฒนาในส่วนของการแจ้งแก้ไขข้อมูล	107
4.9	การพัฒนาในส่วนของการค้นหาสถานที่	109
4.10	การพัฒนาในส่วนของการทำแบบประเมิน	110
4.11	การพัฒนาในส่วนของ Server ที่ใช้ในการ train model	113
4.12	การพัฒนาในส่วนของ Server ที่ใช้ในการ predict ข้อมูล	117
5	การทดสอบระบบ	119
5.1	การทดสอบในส่วนฟังก์ชันของระบบ	119
5.1.1	ผลการทดสอบการสมัครสมาชิก	119
5.1.2	ผลการทดสอบการเข้าสู่ระบบ	120
5.1.3	ผลการทดสอบหน้าจัดอันดับสถานที่ยอดนิยม	121
5.1.4	ผลการทดสอบหน้าหมวดที่เกี่ยวข้อง	122
5.1.5	ผลการทดสอบหน้าหมวดที่กิน	123
5.1.6	ผลการทดสอบหน้าหมวดที่พัก	124
5.1.7	ผลการทดสอบหน้าแสดงสถานที่	125
5.1.8	ผลการทดสอบหน้าแสดงรายละเอียดสถานที่	126
5.1.9	ผลการทดสอบหน้าเขียนรีวิว	127
5.1.10	ผลการทดสอบหน้ากรอกแบบประเมิน	127
5.1.11	ผลการทดสอบหน้าแสดงผลการค้นหาหมวดท่องเที่ยว	128
5.1.12	ผลการทดสอบหน้าเพิ่มสถานที่	129
5.1.13	ผลการทดสอบการแสดงผลหน้าโปรไฟล์	130
5.1.14	ผลการทดสอบการแก้ไขรหัสผ่าน	131
5.1.15	ผลการทดสอบลบสถานที่	132
5.1.16	ผลการทดสอบลบรีวิว	132
5.1.17	ผลการทดสอบการจัดการแก้ไขข้อมูลสถานที่	133

5.1.18	ผลการทดสอบการจัดการลบข้อมูลสถานที่	133
5.1.19	ผลการทดสอบการจัดการเพิ่มสถานที่	134
5.1.20	ผลการทดสอบจัดการข้อมูลแจ้งแก้ไข	134
5.1.21	ผลการทดสอบการ Train Model	135
5.2	การทดสอบประสิทธิภาพของโมเดล	135
5.2.1	ผลการทดสอบความถูกต้องของโมเดล	135
6	สรุปและข้อเสนอแนะ	138
6.1	สรุปความสามารถของระบบ	138
6.2	ปัญหาและอุปสรรคในการพัฒนา	139
6.3	แนวทางการพัฒนาต่อ	139
	บรรณานุกรม	140
	ภาคผนวก	143
	ภาคผนวก ก การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม	143
	ก.1 การติดตั้ง Visual Studio Code	143
	ก.2 การติดตั้ง Node.js	147
	ก.3 การติดตั้ง Angular	150
	ภาคผนวก ข คู่มือการติดตั้งระบบ	154
	ภาคผนวก ค คู่มือการใช้งานระบบ	156
	ค.1 ส่วนผู้ใช้งาน	156
	ค.2 ส่วนผู้ดูแลระบบ	165
	ประวัติผู้เขียน	170

สารบัญตาราง

ตารางที่	หน้า
1.1 ขั้นตอนการดำเนินงาน	4
3.1 สัญลักษณ์ของ Use case Diagram	36
3.2 อธิบาย Use Case หน้าที่ของระบบในภาพที่ 3.3	38
3.3 อธิบาย Use Case หน้าที่ของระบบในภาพที่ 3.3 (ต่อ)	39
3.4 Use Case สมัครสมาชิก	39
3.5 Use Case เข้าสู่ระบบ	40
3.6 Use Case เขียนรีวิวให้คะแนนสถานที่	40
3.7 Use Case เพิ่มข้อมูลสถานที่	41
3.8 Use Case กรอกแบบประเมิน	41
3.9 Use Case ดูผลลัพธ์การแนะนำ	42
3.10 Use Case ดูรายละเอียดสถานที่	42
3.11 Use Case ดูผลลัพธ์การจัดอันดับ	43
3.12 Use Case ค้นหาสถานที่	43
3.13 Use Case แก้ไขข้อมูลสถานที่	44
3.14 Use Case ลบข้อมูลสถานที่	44
3.15 Use Case แก้ไขข้อมูลสถานที่	44
3.16 Use Case ส่งเทรนโมเดล	45
3.17 สัญลักษณ์ของ Class Diagram	47
3.18 อธิบาย Class Diagram ผู้ใช้งานจากภาพที่ 3.20 ได้ดังนี้	50
3.19 อธิบาย Class Diagram ผู้ใช้งานจากภาพที่ 3.20 ได้ดังนี้ (ต่อ)	51
3.20 อธิบาย Class Diagram ผู้ดูแลระบบจากภาพที่ 3.21 ได้ดังนี้	52
3.21 อธิบาย Class Diagram ผู้ดูแลระบบจากภาพที่ 3.21 ได้ดังนี้ (ต่อ)	53
3.22 สัญลักษณ์ของ Sequence Diagram	54
3.23 อธิบายโครงสร้างฐานข้อมูลของ datamodel จากรูปที่ 3.36	70
3.24 อธิบายโครงสร้างฐานข้อมูลของ datamodel จากรูปที่ 3.36 (ต่อ)	71
3.25 อธิบายโครงสร้างฐานข้อมูลของ dataedit จากรูปที่ 3.37	73
3.26 อธิบายโครงสร้างฐานข้อมูลที่เก็บข้อมูลสถานที่กิน จากรูปที่ 3.38	75
3.27 อธิบายโครงสร้างฐานข้อมูลที่เก็บข้อมูลสถานที่เที่ยว จากรูปที่ 3.39	77
3.28 อธิบายโครงสร้างฐานข้อมูลที่เก็บข้อมูลสถานที่พัก จากรูปที่ 3.40	79
3.29 อธิบายโครงสร้างฐานข้อมูลที่เก็บข้อมูลการเขียนรีวิวสถานที่ จากรูปที่ 3.41	81
3.30 อธิบายโครงสร้างฐานข้อมูลของ admin ในส่วน security จากรูปที่ 3.42	82
3.31 อธิบายโครงสร้างฐานข้อมูลของ user ในส่วน security จากรูปที่ 3.43	84
5.1 ผลการทดสอบการสมัครสมาชิก	119
5.2 ผลการทดสอบการเข้าสู่ระบบ	120

5.3	ผลการทดสอบหน้าจัดอับดับสถานที่ยอดนิยม	121
5.4	ผลการทดสอบหน้าหมวดที่เที่ยว	122
5.5	ผลการทดสอบหน้าหมวดที่กิน	123
5.6	ผลการทดสอบหน้าหมวดที่พัก	124
5.7	ผลการทดสอบหน้าแสดงสถานที่	125
5.8	ผลการทดสอบหน้าแสดงรายละเอียดสถานที่	126
5.9	ผลการทดสอบหน้าเขียนรีวิว	127
5.10	ผลการทดสอบหน้ากรอกแบบประเมิน	127
5.11	ผลการทดสอบหน้าแสดงผลการแนะนำหมวดท่องเที่ยว	128
5.12	ผลการทดสอบหน้าเพิ่มสถานที่	129
5.13	ผลการทดสอบการแสดงผลหน้าโปรไฟล์	130
5.14	ผลการทดสอบการแก้ไขรหัสผ่าน	131
5.15	ผลการทดสอบลบสถานที่	132
5.16	ผลการทดสอบลบรีวิว	132
5.17	ผลการทดสอบการจัดการแก้ไขข้อมูลสถานที่	133
5.18	ผลการทดสอบการจัดการลบข้อมูลสถานที่	133
5.19	ผลการทดสอบการจัดการเพิ่มสถานที่	134
5.20	ผลการทดสอบจัดการข้อมูลแจ้งแก้ไข	134
5.21	ผลการทดสอบการ Train Model	135
5.22	ผลลัพธ์การทดสอบเทรนโมเดล 5 ครั้งต่อหนึ่งค่า Learning Rate	136
5.23	ผลการทดสอบค่าความถูกต้องและค่าความผิดพลาดของโมเดล	136

สารบัญภาพ

รูปที่		หน้า
2.1	concept หลักของ React	6
2.2	การทำงานแบบ event driven	9
2.3	หน้าเว็บบริการของ Google Maps	14
2.4	หน้าแรกของเว็บไซต์ Wongnai	15
3.1	System architecture ระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี . . . 17	
3.2	การทำงานในส่วนของการ train model และ predict	18
3.3	หน้าสมัครสมาชิก	20
3.4	หน้าเข้าสู่ระบบ	21
3.5	หน้าจัดอันดับ	22
3.6	หน้าจอหมวดที่เที่ยว	23
3.7	หน้าจอหมวดที่กิน	24
3.8	หน้าจอหมวดที่พัก	25
3.9	หน้าจอแสดงสถานที่	26
3.10	หน้าจอแสดงรายละเอียดสถานที่	27
3.11	หน้าจอเขียนรีวิว	28
3.12	หน้าจอแบบประเมิน	29
3.13	หน้าจอเพิ่มสถานที่	30
3.14	หน้าจอผลลัพธ์การแนะนำสถานที่	31
3.15	หน้าจอโปรไฟล์	32
3.16	หน้าจอจัดการข้อมูลสถานที่เที่ยว	33
3.17	หน้าจอจัดการข้อมูลสถานที่กิน	34
3.18	หน้าจอจัดการข้อมูลสถานที่พัก	35
3.19	Use Case Diagram ระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี 37	
3.20	Class Diagram ระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี 48	
3.21	Class Diagram ระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี 49	
3.22	Sequence Diagram ของการสมัครสมาชิก	55
3.23	Sequence Diagram ของการเข้าสู่ระบบ	56
3.24	Sequence Diagram ของการดูข้อมูลรายละเอียดสถานที่	57
3.25	Sequence Diagram ของการเขียนรีวิวให้คะแนนสถานที่	58
3.26	Sequence Diagram ของการแจ้งแก้ไขข้อมูลสถานที่	59
3.27	Sequence Diagram ของการเพิ่มข้อมูลสถานที่	60
3.28	Sequence Diagram ของการกรอกแบบประเมิน	61
3.29	Sequence Diagram ของการดูผลลัพธ์การจัดอันดับ	62
3.30	Sequence Diagram ของการจัดการแก้ไขข้อมูลสถานที่	63
3.31	Sequence Diagram ของการจัดการลบข้อมูลสถานที่	64

3.32	Sequence Diagram ของการจัดการแจ้งแก้ไขข้อมูลสถานที่	65
3.33	Sequence Diagram ของการจัดการ train model	66
3.34	สัญลักษณ์ของโครงสร้างฐานข้อมูลแบบ Kuzzle	67
3.35	โครงสร้างฐานข้อมูลภาพรวมใน Kuzzle	68
3.36	โครงสร้างฐานข้อมูล datamodel	69
3.37	โครงสร้างฐานข้อมูล dataedit	72
3.38	โครงสร้างฐานข้อมูล eat	74
3.39	โครงสร้างฐานข้อมูล travel	76
3.40	โครงสร้างฐานข้อมูล hotel	78
3.41	โครงสร้างฐานข้อมูล review	80
3.42	โครงสร้างฐานข้อมูล Admin	82
3.43	โครงสร้างฐานข้อมูล User	83
4.1	การทำงานของระบบเมื่อกดปุ่มสมัครสมาชิก	85
4.2	การทำงานของระบบเมื่อกดปุ่มสมัครสมาชิก (ต่อ)	86
4.3	การทำงานของระบบเมื่อกดเข้าสู่ระบบ	87
4.4	การทำงานของระบบเมื่อผู้ใช้กดเลือกหมวด	89
4.5	การทำงานของระบบเมื่อผู้ใช้กดเลือกหมวด (ต่อ)	91
4.6	การทำงานของระบบเมื่อผู้ใช้กดปุ่มเขียนรีวิว	93
4.7	การทำงานของระบบเมื่อผู้ใช้กดปุ่มเขียนรีวิว(ต่อ)	95
4.8	การทำงานของระบบเมื่อผู้ใช้กดปุ่มเขียนรีวิว(ต่อ)	97
4.9	การทำงานของระบบเมื่อผู้ใช้กดปุ่มเพิ่มสถานที่	99
4.10	การทำงานของระบบเมื่อผู้ใช้กดปุ่มเพิ่มสถานที่(ต่อ)	100
4.11	การทำงานของระบบเมื่อผู้ใช้กดปุ่มเพิ่มสถานที่(ต่อ)	101
4.12	การทำงานของระบบเมื่อผู้ใช้กดปุ่มลบรีวิว	103
4.13	การทำงานของระบบเมื่อผู้ใช้กดปุ่มบันทึกสถานที่	105
4.14	การทำงานของระบบเมื่อผู้ใช้กดปุ่มแจ้งแก้ไขสถานที่	107
4.15	การทำงานของระบบเมื่อผู้ใช้กดปุ่มแจ้งแก้ไขสถานที่	108
4.16	การทำงานของระบบเมื่อผู้ใช้กดปุ่มเลือกประเภท	109
4.17	การทำงานของระบบเมื่อผู้ใช้กดปุ่มค้นหา(ต่อ)	109
4.18	การทำงานของระบบเมื่อผู้ใช้กรอกแบบประเมิน	110
4.19	การทำงานของระบบเมื่อผู้ใช้กรอกแบบประเมินเสร็จกดปุ่มส่งข้อมูล(ต่อ)	111
4.20	การทำงานของระบบเมื่อผู้ใช้กรอกแบบประเมินเสร็จกดปุ่มส่งข้อมูล(ต่อ)	112
4.21	ขั้นตอนกระบวนการ train model	113
4.22	ขั้นตอนกระบวนการ train model(ต่อ)	114
4.23	ขั้นตอนกระบวนการ train model(ต่อ)	115
4.24	ขั้นตอนกระบวนการ predict ข้อมูล	117
5.1	กราฟเปรียบเทียบค่า LearningRate	137
ก.1	หน้าเว็บดาวน์โหลด Visual Studio Code	143

ก.2	หน้าต่างต้อนรับของ Visual Studio Code	144
ก.3	หน้าต่างข้อตกลงการใช้งาน Visual Studio Code	144
ก.4	หน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Visual Studio Code	145
ก.5	หน้าต่างการจัดการซอร์สโค้ด ของ Visual Studio Code	145
ก.6	หน้าต่างเริ่มทำการติดตั้งทำการกด ของ Visual Studio Code	146
ก.7	หน้าต่างเมื่อเข้าโปรแกรมหลังติดตั้งเสร็จ ของ Visual Studio Code	146
ก.8	หน้าเว็บดาวน์โหลด Node.js	147
ก.9	ไฟล์ติดตั้งสำหรับติดตั้ง Node.js	147
ก.10	หน้าต่างต้อนรับของ Node.js	148
ก.11	หน้าต่างข้อตกลงในการใช้ Node.js	148
ก.12	หน้าต่างเลือกโฟลเดอร์ที่จะทำการติดตั้ง Node.js	149
ก.13	หน้าต่างติดตั้ง Node.js	149
ก.14	หน้าเว็บ Angular	150
ก.15	ใช้คำสั่งติดตั้ง Angular	150
ก.16	ใช้คำสั่งติดตั้ง Angular	151
ก.17	ใช้คำสั่งสร้าง project	152
ก.18	ใช้คำสั่งสร้าง project	152
ก.19	ใช้คำสั่งสร้าง project	152
ก.20	การใช้คำสั่งรัน project	153
ก.21	หน้าแรกของเว็บไซต์	153
ข.1	หน้าเว็บ Repository ของโปรเจ็ค	154
ข.2	คำสั่งติดตั้ง Package	154
ข.3	คำสั่งเริ่มใช้งานโปรเจ็ค	155
ค.1	หน้าจอการสมัครสมาชิก	156
ค.2	หน้าจอการเข้าสู่ระบบ	157
ค.3	หน้าจอการจัดอันดับสถานที่ยอดนิยม	158
ค.4	หน้าจอหมวดที่กิน	160
ค.5	หน้าจอแสดงข้อมูลสถานที่	161
ค.6	หน้าจอแสดงรายละเอียดสถานที่	162
ค.7	หน้าจอแสดงข้อมูลโปรไฟล์ของผู้ใช้	163
ค.8	หน้าจอแสดงเขียนรีวิว	164
ค.9	หน้าจอจัดการข้อมูลสถานที่เที่ยว	165
ค.10	หน้าจอการเพิ่มสถานที่	167
ค.11	หน้าจอการแก้ไขข้อมูลสถานที่	168
ค.12	หน้าจอการแจ้งแก้ไข	169

บทที่ 1

บทนำ

1.1 ที่มาและเหตุผล

เนื่องจากปัจจุบัน การดำเนินธุรกิจร้านเสริมสวยที่ให้บริการเสริมความงาม เช่น การทำผม ตัดผม ออกแบบทรงผม อบไอน้ำ เวลาผู้ให้บริการมาให้บริการโดยจะมาที่ร้านเลยโดยไม่จองคิว พบว่าร้านที่มาใช้บริการมีลูกค้าเป็นจำนวนมาก อาจจะทำให้ต้องรอคิวนานหรือต้องเสียเวลามาใช้บริการในวันอื่นบางครั้งผู้ให้บริการมีเบอร์ของร้านเสริมสวยก็จะโทรมา สอบถามคิวและจองคิว แต่ช่างตัดผมให้ลูกค้าท่านอื่นก็ไม่สามารถรับโทรศัพท์ได้ผู้พัฒนาจึงมีแนวคิดว่าจะทำระบบการจอง คิวร้านเสริมสวยขึ้น เพื่อแก้ปัญหาการรอคิวนานและให้มีความทันสมัยตลอดจนสามารถรองรับการแสดงผลบนอุปกรณ์ สมาร์ทโฟนในปัจจุบัน ทำให้สามารถจองคิวหรือติดต่อสื่อสารในเรื่องของการจองคิวทำผมกับทางร้านได้สะดวกมากยิ่งขึ้น แนวทางการแก้ปัญหา จัดทำการพัฒนาเป็นเว็บแอปพลิเคชัน ระบบจองคิวร้านเสริมสวยที่ถูกพัฒนาขึ้นเป็นเว็บแอปพลิเคชัน จะช่วยเพิ่มระเบียบในการจัดการจองคิวให้เป็นระบบ ลดขั้นตอนการดำเนินงานที่ซับซ้อน ลดระยะเวลาในการดำเนินงาน ลดความผิดพลาดที่จะเกิดขึ้นในขั้นตอนการดำเนินงาน และช่วยเพิ่มประสิทธิภาพในการทำงาน

1.2 วัตถุประสงค์

1. เพื่อออกแบบและพัฒนาเว็บแอปพลิเคชัน จองคิวร้านเสริมสวย
2. เพื่อแก้ปัญหาการรอคิวร้านเสริมสวย

1.3 ขอบเขตของโครงการ

1.3.1 เจ้าของร้าน

- สามารถลงทะเบียนเข้าสู่ระบบด้วย Email ได้
- สามารถดูคิวที่ผู้ให้บริการได้ทำการจองคิวไว้
- สามารถจัดการคิวได้
- สามารถ post ภาพผลงานทั้งหมดของร้านได้

- สามารถเพิ่ม แก้ไข และลบรายการให้บริการประจำร้านได้
- สามารถเพิ่ม แก้ไข และลบ ข้อมูลร้านได้
- สามารถเพิ่ม แก้ไข และลบข้อมูลตำแหน่งร้านได้

1.3.2 ช่างประจำร้าน

- ลงทะเบียนใช้ web ด้วย Email ได้
- สามารถดูตารางการทำงานของตนเองได้
- สามารถ post ภาพผลงานของตัวเองได้
- สามารถแก้ไขข้อมูลส่วนตัวได้

1.3.3 ผู้ใช้บริการ

- สามารถลงทะเบียนเข้าสู่ระบบด้วย Email ได้
- สามารถค้นหาร้านเสริมสวยได้
- สามารถจองคิวของร้านเสริมสวยได้
- สามารถดูคิวว่างของร้านเสริมสวยได้
- สามารถดูข้อมูลต่างๆของร้านเสริมสวยได้
- สามารถดูตำแหน่งของทางร้านได้
- สามารถดูผลงานของร้านได้
- สามารถเขียนรีวิว ทิชชเม ได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยลดขั้นตอนการจองคิวร้านเสริมสวย
2. ช่วยลดปัญหาในการใช้บริการที่เกิดจากการรอคิวนาน
3. ช่วยให้การจองคิวมีระเบียบมากขึ้น

1.5 เครื่องมือที่ใช้ในการพัฒนา (Development tools)

1.5.1 ฮาร์ดแวร์

1. เครื่องคอมพิวเตอร์ส่วนบุคคล (Personal computer) เพื่อใช้ในการพัฒนาโมบายแอปพลิเคชัน โดยมีคุณสมบัติอย่างน้อยดังนี้
 - ทำงานบนระบบปฏิบัติการ Elementary OS พื้นฐานการทำงานบน Windows 10
 - หน่วยประมวลผลกลาง AMD Rezen(™) 5 3500U
 - หน่วยประมวลผลกราฟิก AMD Radeon(™) Vega(8) Mobile Graphics
 - หน่วยความจำหลักอย่างน้อย 8 กิกะไบต์ (Gigabyte, GB)
 - หน่วยความจำสำรองอย่างน้อย 256 กิกะไบต์ (Gigabyte, GB)

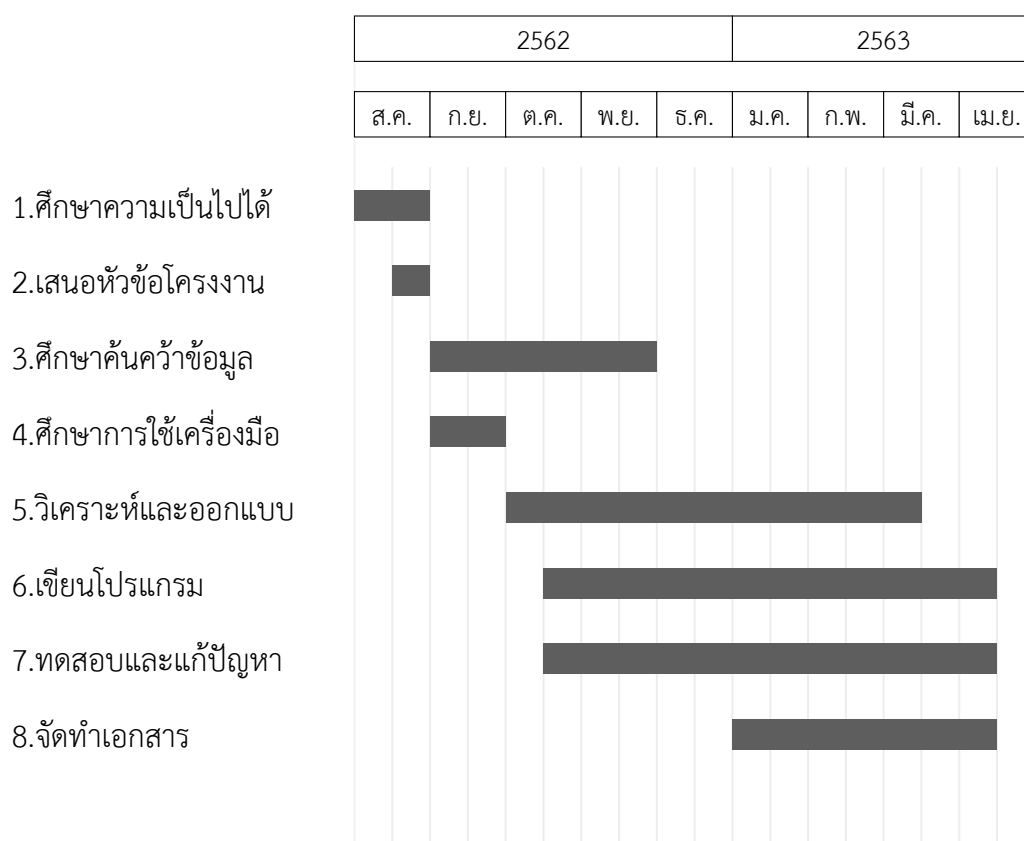
1.5.2 ซอฟต์แวร์ (Software)

1. React JS เป็น Java Script Framework ที่ใช้ในการพัฒนาเว็บไซต์ฝั่งผู้ใช้งาน
2. Node Js เป็น เทคโนโลยีที่ใช้ในการพัฒนาเว็บไซต์ฝั่งที่ติดต่อกับฐานข้อมูล
3. JavaScript เป็น ภาษาที่ใช้ในการพัฒนาเว็บ
4. MySQL เป็น โปรแกรมระบบจัดการฐานข้อมูล
5. Visual Studio Code เป็น Code Editor ที่ใช้ในการแก้ไขและปรับแต่ง Code
6. Google Map API เทคโนโลยีที่ใช้ในการใช้งานแผนที่

1.6 แผนการดำเนินการ

ในการสร้างระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี ผู้พัฒนาได้แบ่งขั้นตอนการดำเนินงานไว้ด้วยกัน 8 ขั้นตอน ดังตารางที่ 1.1

ตารางที่ 1.1: ขั้นตอนการดำเนินงาน



บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงรายละเอียดเกี่ยวกับทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการพัฒนาระบบจองคิวร้านเสริมสวย โดยแบ่งเนื้อหาออกเป็น 2 ส่วน ได้แก่ ส่วนที่หนึ่งเป็นเนื้อหาพื้นฐานเกี่ยวกับทฤษฎีการเขียนโปรแกรมและเทคโนโลยีที่นำมาใช้ในการพัฒนาในหัวข้อที่ 2.1 - 2.6 ได้แก่ ความรู้พื้นฐานเกี่ยวกับ React Node.js JavaScript MySQL Visual studio code Google maps API และในส่วนที่สองเป็นเนื้อหาเกี่ยวกับเว็บแอปพลิเคชันที่เกี่ยวข้องกับโครงงานนี้เว็บแอปพลิเคชัน Gowabi

2.1 ความรู้พื้นฐานเกี่ยวกับ React

React เป็น JavaScript Library ที่ถูกสร้างโดย Facebook ซึ่ง React ทำหน้าที่เป็นเพียง User Interface (UI) ที่สร้างมาจากพื้นฐานแนวความคิดแบบ Model View Controller (MVC)

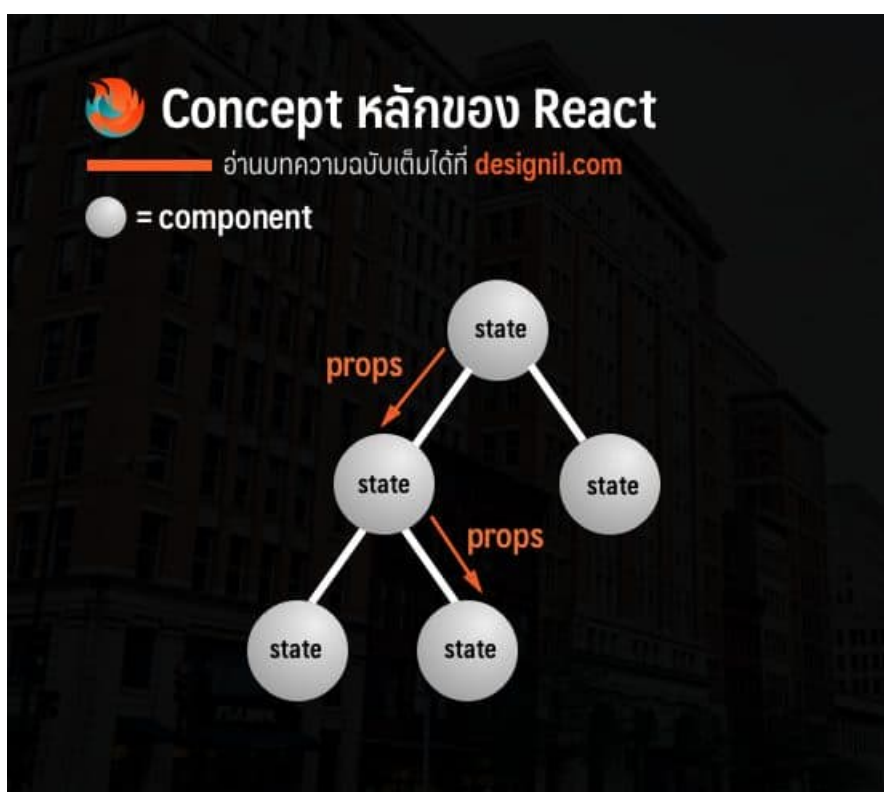
React ทำหน้าที่เฉพาะส่วน View (จาก Model View Controller) เหมาะกับงาน Web Front-End ที่สามารถแบ่งออกเป็น Web Component ย่อยๆ โดยหลักการวิเคราะห์ควรแยกให้ย่อยที่สุดเท่าที่จะทำได้ ซึ่งสามารถแบ่ง Component ออกเป็น 2 รูปแบบ คือ

- Container สำหรับบรรจุ Component หลักย่อยอื่นๆ ซึ่งไม่ควรมีการเก็บค่าใดๆ (สามารถทำหน้าที่เป็นตัวกลางในการส่งผ่านค่าได้) เน้นไปที่การจัด Layout
- Web Component คือ ส่วนที่ต้อง interact กับผู้ใช้จริงๆ เช่น ช่องกรอกข้อมูล ลิสต์แสดงข้อมูล ลาเบล (Label) และ ปุ่ม เป็นต้น ซึ่งอาจมีการเก็บค่าบางค่าเอาไว้ที่ สเตท (State) เพื่อนำมาแสดงผล

การเขียน React จำเป็นต้องมีความรู้ใน 3 ประเด็น ได้แก่

- Component – ส่วนประกอบต่างๆ ในเว็บ จะถูกมองเป็น Component
- State – ข้อมูลที่อยู่ใน Component แต่ละชิ้น เรียกว่า State
- Props – ข้อมูลที่ถูกส่งต่อจาก Component ขึ้นบนลงไปชั้นล่าง ซึ่งเรียกว่า Props (Properties)

อธิบายประเด็นหลักทั้ง 3 ประเด็น ดังรูปที่ 2.1



รูปที่ 2.1: concept หลักของ React

ที่มา: <https://www.designnil.com/wp-content/uploads/2017/07/react-concept-designnil.jpg>

2.1.1 React Life Cycle

การเขียน `render()` ฟังก์ชันใน component [?] นั้น ควรจะเขียนในแบบ pure function ซึ่งจะไม่มีการเปลี่ยน state และสร้าง side effect ต่อภายนอกทั้งสิ้น อย่างเช่น การ call external service แบบ Ajax request, Firebase calling เป็นต้น เพราะหน้าที่ของ `render()` มีแค่การ render UI เท่านั้น หากไม่สามารถทำสิ่งดังกล่าวภายใน `render()` แล้ว ก็กรรมเหล่านั้นจึงสามารถทำได้ที่ life cycle ของ React

ตลอดช่วงวงจรชีวิต [?] สามารถควบคุมเหตุการณ์ต่างๆ ที่เกิดขึ้นในการแสดงผล UI การอัปเดตข้อมูล และการ re-rendering จนกระทั่งข้อมูลนั้นหายไป โดยที่ React ได้มีการเตรียมฟังก์ชันต่างๆ ไว้ สามารถอธิบายการทำงานของฟังก์ชันได้ ดังนี้

- `componentWillMount()` : คุณสมบัติของ `componentWillMount` ไม่มีอะไรเกี่ยวกับ

การใช้งาน component เพราะยังไม่มี mount อะไรขึ้นมา โดยมีหน้าที่ คือ การกำหนดค่าเริ่มต้นสำหรับการใช้งาน

- `componentDidMount()` : เกิดขึ้นเมื่อทำการ Mount เรียบร้อย พร้อมที่จะใช้งาน โดยปกติจะใช้ในการกำหนดค่าทุกอย่างที่ต้องใช้ DOM และรับข้อมูลที่ต้องการมาแสดงผล
- `componentWillReceiveProps(nextProps)` : เมื่อ Component ทำงาน จนกระทั่งมี pro-ps ใหม่เข้ามา เพื่อทำการเปลี่ยนแปลงข้อมูล `componentWillReceiveProps` จะถูกเรียก โดยมี `nextProps` เป็นตัวแปรที่ถูกส่งเข้ามา
- `shouldComponentUpdate(nextProps, nextState)` : ถูกเรียกเมื่อ component มีการเปลี่ยนแปลงด้วย `nextProps` กับ `nextState`
- `componentWillUpdate(nextProps, nextState)` : ถูกเรียกก่อนที่จะ render หลังจากได้รับค่าใหม่ของ props หรือ state คุณสมบัติของคล้ายกับ `componentWillReceiveProps`
- `componentDidUpdate(prevProps, prevState)` : ถูกเรียกทันทีหลังจากเกิดการเปลี่ยนแปลงของ component แต่จะไม่ถูกเรียกตอนครั้งแรกที่ render โดยที่ `componentDidUpdate` สามารถใช้งานได้เหมือน `componentDidMount`
- `componentWillUnmount()` : ถูกเรียกก่อนที่ component ทำการ unmount และ destroy โดยปกติแล้วจะใช้เพื่อทำการรีเซ็ต (reset) ค่าต่างๆ

2.1.2 React Life Cycle

การเขียน `render()` ฟังก์ชันใน component [?] นั้น ควรจะเขียนในแบบ pure function ซึ่งจะไม่มีการเปลี่ยน state และสร้าง side effect ต่อภายนอกทั้งสิ้น อย่างเช่น การ call external service แบบ Ajax request, Firebase calling เป็นต้น เพราะหน้าที่ของ `render()` มีแค่การ render UI เท่านั้น หากไม่สามารถทำสิ่งดังกล่าวภายใน `render()` แล้ว กิจกรรมเหล่านั้นจึงสามารถทำได้ที่ life cycle ของ React

ตลอดช่วงวงจรชีวิต [?] สามารถควบคุมเหตุการณ์ต่างๆ ที่เกิดขึ้นในการแสดงผล UI การอัปเดตข้อมูล และการ re-rendering จนกระทั่งข้อมูลนั้นหายไป โดยที่ React ได้มีการเตรียมฟังก์ชันต่างๆ ไว้ สามารถอธิบายการทำงานของฟังก์ชันได้ ดังนี้

- `componentWillMount()` : คุณสมบัติของ `componentWillMount` ไม่มีอะไรเกี่ยวกับการใช้งาน component เพราะยังไม่มี mount อะไรขึ้นมา โดยมีหน้าที่ คือ การกำหนด

ค่าเริ่มต้นสำหรับการใช้งาน

- `componentDidMount()` : เกิดขึ้นเมื่อทำการ Mount เรียบร้อย พร้อมที่จะใช้งาน โดยปกติจะใช้ในการกำหนดค่าทุกอย่างที่ต้องใช้ DOM และรับข้อมูลที่ต้องการมาแสดงผล
- `componentWillReceiveProps(nextProps)` : เมื่อ Component ทำงาน จนกระทั่งมี pro-ps ใหม่เข้ามา เพื่อทำการเปลี่ยนแปลงข้อมูล `componentWillReceiveProps` จะถูกเรียก โดยมี `nextProps` เป็นตัวแปรที่ถูกส่งเข้ามา
- `shouldComponentUpdate(nextProps, nextState)` : ถูกเรียกเมื่อ component มีการเปลี่ยนแปลงด้วย `nextProps` กับ `nextState`
- `componentWillUpdate(nextProps, nextState)` : ถูกเรียกก่อนที่จะ render หลังจากได้รับค่าใหม่ของ props หรือ state คุณสมบัติของคล้ายกับ `componentWillReceiveProps`
- `componentDidUpdate(prevProps, prevState)` : ถูกเรียกทันทีหลังจากเกิดการเปลี่ยนแปลงของ component แต่จะไม่ถูกเรียกตอนครั้งแรกที่ render โดยที่ `componentDidUpdate` สามารถใช้งานได้เหมือน `componentDidMount`
- `componentWillUnmount()` : ถูกเรียกก่อนที่ component ทำการ unmount และ destroy โดยปกติแล้วจะใช้เพื่อทำการรีเซ็ต (reset) ค่าต่างๆ

2.2 ความรู้พื้นฐานเกี่ยวกับ Node.js

Node.js [3] เป็นภาษาที่ทำงานอยู่ในฝั่งเซิร์ฟเวอร์ (server) ซึ่ง syntax ที่ใช้ในการเขียนคือ JavaScript และเป็นภาษาที่ออกแบบมาให้ทำงานแบบ Event-Driven หรือทำงานเมื่อเกิดเหตุการณ์ตามที่กำหนดไว้ และการทำงานแบบ Asynchronous ซึ่งสามารถทำงานในลำดับต่อไป โดยที่ไม่ต้องรอให้งานก่อนหน้าเสร็จก่อนแล้วจึงทำงานขั้นต่อไป แต่ก็สามารถกำหนดให้ทำงานแบบ Synchronous ได้เช่นกัน โดยการกำหนด Callback เมื่องานแรกทำงานเสร็จแล้ว นอกจากนี้ Node.js นั้นจะใช้ Compiler จาก Google JavaScript Engine V8

ส่วนใหญ่จะนิยมใช้ node.js ในงานที่ทำเป็นเบื้องหลัง คือ งานที่ประมวลผลฝั่งเซิร์ฟเวอร์ ซึ่งเป็นงานที่อาจจะต้อง interface กับผู้ใช้ หรือไม่ต้อง interface กับผู้ใช้ ตัวอย่างงานที่ต้อง interface กับผู้ใช้ เช่น การทำตัวเองเป็น http server ในการดึงหน้าเว็บมาแสดงผลให้กับ user หรือว่า การเปิด socket เพื่อรับส่งข้อมูลกันระหว่างเซิร์ฟเวอร์กับผู้ใช้ งาน เช่น ทำเป็นห้อง chat ทำเกม ทำระบบที่ป้อนข้อมูลเพื่อคำนวณผลลัพธ์ เป็นต้น ตัวอย่างงานที่ไม่ต้อง interface กับผู้ใช้

เช่น ทำ spider crawler เว็บ คือ การเปิดเว็บแล้วเก็บข้อมูลไปเรื่อยๆ หรือ โปรแกรมที่ รอรับค่าจาก streaming ต่างๆ เพื่อนำมาบันทึกไว้ ซึ่งการทำงานเหล่านี้ไม่จำเป็นต้อง interface กับผู้ใช้

node.js มีส่วนเสริมที่ชื่อว่า node package management (npm) ซึ่งเปรียบเหมือน google play ใน android หรือ app store ใน iOS ที่สามารถเอา package ที่คนอื่นเขียนเอาไว้แล้ว เพื่อแจกฟรี (free) มาต่อยอดเพื่อใช้ในงานของตนได้ โดยตัวอย่างที่ได้รับความนิยมจะเป็น underscore, async, request และ express เป็นต้น สำหรับการติดตั้ง ใช้คำสั่ง npm install ตามด้วยชื่อ package ที่ต้องการติดตั้ง [?]

node.js มีการทำงานเป็น Asynchronous คือ การทำงานบางอย่างไม่ต้องรอให้บรรทัดนั้นทำงานเสร็จ เช่น ส่งคำสั่งไป query ข้อมูลจากฐานข้อมูล แล้วสามารถข้ามไปทำงานบรรทัดต่อไปโดยไม่ต้องรอผลจากฐานข้อมูล เมื่อการทำงานนั้นทำงานเสร็จจึงค่อยรอผลลัพธ์กลับมา ดังนั้นปัญหาจะเกิดขึ้นที่ ถ้าการทำงานต่อไปนำผลลัพธ์จากคำสั่งก่อนหน้านี้มาใช้ต่อ ซึ่งส่งผลให้เกิดการทำงานผิดพลาด เพราะผลลัพธ์ยังไม่ได้รับกลับมา

2.2.1 node.js ทำงานแบบ event driven

การทำงานของ node [?] เรียกว่าเป็นการขับเคลื่อนด้วย event ต่างๆ ที่เกิดขึ้น ทำให้สามารถข้ามจาก event หนึ่งทีเสร็จแล้วไปยัง event อื่นได้ด้วยการสั่งงานต่อเนื่องกันไป หรือการสั่งให้ event หลาย event เริ่มทำงานในเวลาใกล้เคียงกัน ประโยชน์ที่ได้จาก event driven คือ การสั่งให้รอรับ event นั้นไปตลอดการณ โดยไม่เปลืองทรัพยากร เช่น การเชื่อมต่อไปยัง streaming channel ที่หนึ่ง ซึ่งอาจเป็น text หรือข้อมูลบางอย่าง เช่น ปริมาณน้ำฝน เอาไว้ หากต้นทางของ streaming ยังไม่มีข้อมูลส่งมา จะไม่เกิด event ใดๆ และ node.js จะรออยู่ แต่หากต้นทาง streaming มีข้อมูลมา node.js จะทำงานเพื่อตอบสนองต่อ event ที่เกิดขึ้นนั้นทันที สามารถแสดงการทำงานดังกล่าวได้ ดังรูปที่ 2.2

รูปที่ 2.2: การทำงานแบบ event driven

ที่มา: http://meewebfree.com/u/i/nodejs/node_js_stage.png

จากรูปที่ 2.2 สามารถอธิบายการทำงานได้ดังนี้ การทำงานของ stage1 เมื่อเรียกใช้การทำงานของ stage2 แล้ว ไม่จำเป็นต้องรอให้ stage2 ทำงานเสร็จก่อน ซึ่ง stage1 สามารถเรียก

การทำงานของ stage3 ได้เลยโดยไม่จำเป็นต้องรอการทำงานของ stage2

2.2.2 ข้อดีของ Node.js

- มีการทำงานแบบ Event-Driven และ Asynchronous
- เหมาะกับการทำ Web แบบ Real time
- ประหยัดทรัพยากรในการทำงาน
- มีการประมวลผลที่รวดเร็ว

2.3 ความรู้พื้นฐานเกี่ยวกับ JavaScript

JavaScript [?] คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต (Internet) ที่กำลังได้รับความนิยมอย่างสูง JavaScript เป็น ภาษาสคริปต์ (script) เชิงวัตถุ ที่เรียกกันว่า "สคริปต์" ซึ่งการใช้ JavaScript ในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) จะช่วยให้เว็บไซต์มีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น โดยมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (interpret) หรือเรียกว่า โปรแกรมเชิงวัตถุ (Object Oriented Programming) ที่มีเป้าหมายในการ ออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (client) และ ทางฝั่งเซิร์ฟเวอร์

JavaScript ถูกพัฒนาขึ้นโดย บริษัท เน็ตสเคปคอมมิวนิเคชันส์ (Netscape Communications Corporation) โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator 2.0 เพื่อใช้สร้างเว็บเพจ (Web page) โดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมาเน็ตสเคปได้ร่วมมือกับ บริษัทซันไมโครซิสเต็มส์ (Sun Microsystems, Inc) ปรับปรุงระบบของเบราว์เซอร์ (Browser) เพื่อให้สามารถติดต่อกับภาษาจาวา (Java) ได้ และได้ปรับปรุง LiveScript ใหม่เมื่อ ปี พ.ศ. 2538 แล้วตั้งชื่อใหม่ว่า JavaScript ซึ่ง JavaScript ทำให้การสร้างเว็บเพจมีลูกเล่นต่างๆ มากมาย และยังสามารถโต้ตอบกับผู้ใช้ได้อย่างทันที เช่น การใช้เมาส์คลิก หรือ การกรอกข้อความในฟอร์ม เป็นต้น

เนื่องจาก JavaScript ช่วยให้ผู้พัฒนา สามารถสร้างเว็บเพจได้ตรงกับความต้องการ และมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิด ที่ทุกคนสามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย

European Computer Manufacturer's Association (ECMA) การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยเบราว์เซอร์ (เรียกว่าเป็น client-side script) ดังนั้น JavaScript จึงสามารถทำงานได้เฉพาะบนเบราว์เซอร์ที่สนับสนุน ซึ่งปัจจุบันเบราว์เซอร์เกือบทั้งหมดสามารถสนับสนุน JavaScript แล้ว อย่างไรก็ตาม สิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชันใหม่ๆ ออกมาด้วย ดังนั้น ถ้านำโค้ด (code) ของเวอร์ชันใหม่ ไปรันบนเบราว์เซอร์รุ่นเก่าที่ยังไม่สนับสนุน อาจจะทำให้เกิด error ได้

2.3.1 ประโยชน์ของ JavaScript

- JavaScript ทำให้สามารถใช้เขียนโปรแกรมแบบง่ายได้ โดยไม่ต้องพึ่งภาษาอื่น
- JavaScript มีคำสั่งที่ตอบสนองกับผู้ใช้งาน เช่น เมื่อผู้ใช้คลิกที่ปุ่ม หรือ Checkbox สามารถสั่งให้เปิดหน้าต่างใหม่ได้ ทำให้เว็บไซต์มีปฏิสัมพันธ์กับผู้ใช้งานมากขึ้น
- JavaScript สามารถเขียนหรือเปลี่ยนแปลง HTML Element ได้ นั่นคือสามารถเปลี่ยนแปลงรูปแบบการแสดงผลของเว็บไซต์ได้ หรือหน้าแสดงเนื้อหาสามารถซ่อนหรือแสดงเนื้อหาได้โดยง่าย
- JavaScript สามารถใช้ตรวจสอบข้อมูลได้ เช่น เมื่อผู้ใช้งานกรอกข้อมูลบางเว็บไซต์ เช่น Email เมื่อบันทึกข้อมูลผิดจะมีหน้าต่างฟ้องขึ้นมาว่ากรอกผิด หรือลิ้มบันทึกอะไรบางอย่าง เป็นต้น
- JavaScript สามารถใช้ในการตรวจสอบผู้ใช้ได้เช่น ตรวจสอบว่าผู้ใช้ ใช้ web browser อะไร
- JavaScript สร้าง Cookies (เก็บข้อมูลของผู้ใช้ในคอมพิวเตอร์ของผู้ใช้เอง) ได้

2.3.2 ข้อดีและข้อเสียของ JavaScript

JavaScript [?] ทำงานบนเว็บเบราว์เซอร์ (client-side script) จึงไม่มีข้อจำกัดว่าจะใช้เซิร์ฟเวอร์แบบไหนก็ตาม เพราะ JavaScript ทำงานเฉพาะในเครื่องของผู้ใช้งานเท่านั้น ซึ่งต่างกับภาษาสคริปต์อื่น เช่น PHP , ASP, JSP หรือ Perl ซึ่งต้องประมวลผลและทำงานที่เครื่องเซิร์ฟเวอร์ (server-side script) จึงจำเป็นต้องใช้บนเซิร์ฟเวอร์ ที่สนับสนุนภาษาเหล่านี้เท่านั้นจึงจะสามารถใช้งาน server-side script ได้ แต่อย่างไรก็ตาม จากลักษณะการทำงานที่กล่าวมาก็ทำให้ JavaScript มีข้อจำกัด กล่าวคือคือไม่สามารถรับและส่งข้อมูลต่างๆ กับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ชม เพื่อนำไปเก็บบน

เซิร์ฟเวอร์ เป็นต้น ดังนั้นงานลักษณะนี้ จึงยังคงต้องอาศัยภาษา server-side script อยู่ (ความจริงมี JavaScript ที่ทำงานบนเซิร์ฟเวอร์เช่นกัน ซึ่งต้องอาศัยเซิร์ฟเวอร์ที่สนับสนุนโดยเฉพาะเช่นกัน แต่ไม่เป็นที่นิยมนัก)

นักพัฒนาเว็บส่วนใหญ่จึงนิยมใช้ JavaScript ร่วมกับ ภาษา Server Script เพื่อทำการส่งข้อมูลระหว่าง เซิร์ฟเวอร์กับเครื่องของผู้ใช้งาน ซึ่งทำให้การแสดงผลของหน้าเว็บมีความสวยงามและราบรื่นมากยิ่งขึ้น

2.4 ความรู้พื้นฐานเกี่ยวกับ MySQL

MySQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) โดยใช้ภาษา SQL และเป็นซอฟต์แวร์โอเพนซอร์ส

2.4.1 ข้อดีของMySQL

- เป็นซอฟต์แวร์ฟรี สามารถใช้งานได้โดยไม่ต้องเสียค่าลิขสิทธิ์
- มีความเร็วในการทำงานสูง
- มีเสถียรภาพสูง
- ทำงานได้กับหลายระบบปฏิบัติการ เช่น UNIX, Linux, Windows 2000, Windows XP
- ติดตั้งและใช้งานง่ายมีคู่มือให้ดาวน์โหลดได้ฟรี
- เหมาะกับธุรกิจขนาดเล็กและขนาดกลาง

เนื่องจากภาษา SQL เป็นภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลโดยเฉพาะ ซึ่งในระบบงานจะใช้ 4 คำสั่งหลักๆคือ

- SELECT ใช้สำหรับดึงข้อมูลที่ต้องการ
- UPDATE ใช้สำหรับแก้ไขข้อมูล
- INSERT ใช้สำหรับการเพิ่มข้อมูล
- DELETE ใช้สำหรับลบข้อมูล

2.5 ความรู้เกี่ยวกับ Visual Studio Code

วิชวล สตูดิโอโค้ด (Visual Studio Code หรือ VSCode) [5] เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ดถูกพัฒนาโดยค่ายไมโครซอฟท์(Microsoft) มีการพัฒนาออกมา

ในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบไม่ต้องเสียค่าใช้จ่าย Visual Studio Code เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม (Cross-platform) โดยจะรองรับการใช้งานทั้งบนระบบปฏิบัติการ Windows, macOS และ Linux ซึ่งภาษาที่ Visual Studio Code รองรับการทำงานซึ่งมีมากกว่า 30 ภาษาโปรแกรม เช่น C++, C, CSS, Dockerfile, HTML, JavaScript, JSON, Less, Markdown, PHP, Python, Sass, TypeScript, Node.js และ Java เป็นต้น

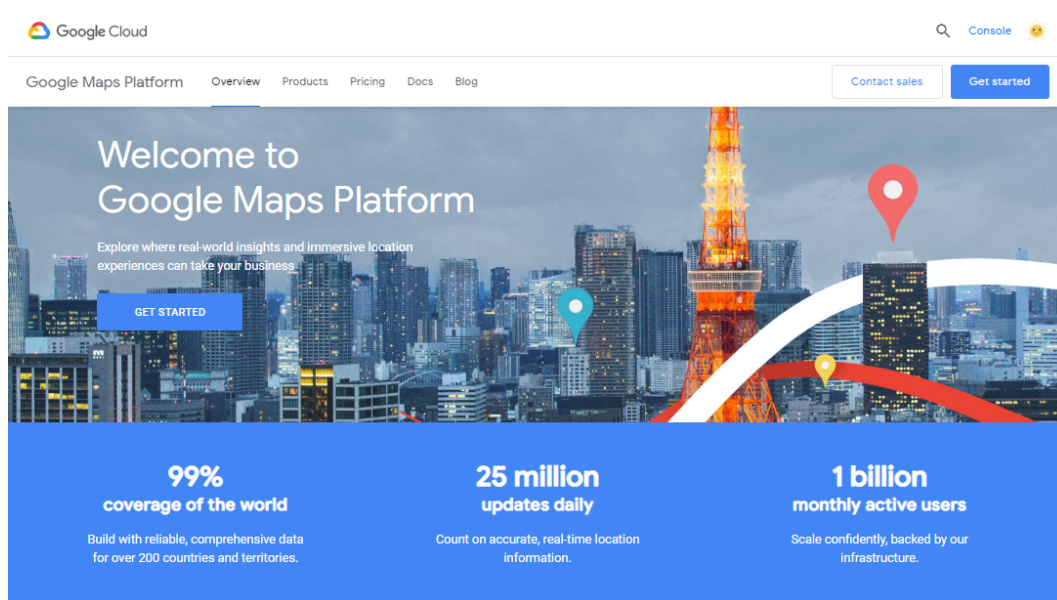
2.6 ความรู้พื้นฐานเกี่ยวกับ Google Maps API

Google Maps API [6] เป็นชุด API ของ Google สำหรับพัฒนา web application และ mobile application (Android, ios) มีไว้สำหรับเรียกใช้แผนที่และชุด service ต่างๆ มากมายให้เรียกใช้ เช่น

- ชุดควบคุมแผนที่ (Map Control)
- การปรับแต่งแผนที่ (Styled Map)
- การนำทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง (Directions Service)
- Street view
- การดึงข้อมูล POI (Point of Interest) คือข้อมูลสถานที่ต่างๆ ที่ Google รวบรวมไว้ เช่น โรงแรม ห้างสรรพสินค้า โรงเรียน สถานที่ราชการ เป็นต้น เพื่อให้ผู้ใช้งานนำมาใช้งานได้
- การแปลงที่อยู่เป็นพิกัด Latitude และ Longitude (Geocoding Service)
- แผนที่รองรับระบบ 3 มิติ ผู้ใช้งาน สามารถปรับมุมมองเป็นลักษณะ 3 มิติได้ ซึ่งระบบนี้ให้บริการบางพื้นที่

การใช้งาน Google Maps API สามารถขอใช้งานโดยใช้บัญชี Gmail และเข้าไปที่เว็บ

<https://cloud.google.com/maps-platform/> ดังรูปที่ 2.3



รูปที่ 2.3: หน้าเว็บบริการของ Google Maps

ที่มา : <https://cloud.google.com/maps-platform/>

2.7 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.7.1 เว็บแอปพลิเคชัน Gowabi

Wongnai[11] เป็นเว็บไซต์และแอปพลิเคชันที่ให้บริการเกี่ยวกับการค้นหาร้านอาหาร สถาน ที่ท่องเที่ยว ร้านเสริมสวย มีฟังก์ชันการทำงานพื้นฐานอันได้แก่ การค้นหา ดูข้อมูล โพสต์รีวิว บันทึกร้านที่ชอบ เป็นต้น

2.7.2 ข้อแตกต่างระหว่างเว็บแอปพลิเคชัน wongnai กับเว็บของโครงการ

เว็บแอปพลิเคชันของ wongnai ยังไม่มีฟังก์ชันการแนะนำสถานที่ที่เหมาะสมแก่ผู้ใช้ ผู้พัฒนาจึงได้ทำฟังก์ชันการแนะนำสถานที่ที่เหมาะสมตามรสนิยมของผู้ใช้เพิ่มลงในเว็บของโครงการ

รูปที่ 2.4: หน้าแรกของเว็บไซต์ Wongnai

ที่มา : <https://www.wongnai.com/>

บทที่ 3

การวิเคราะห์และออกแบบระบบ

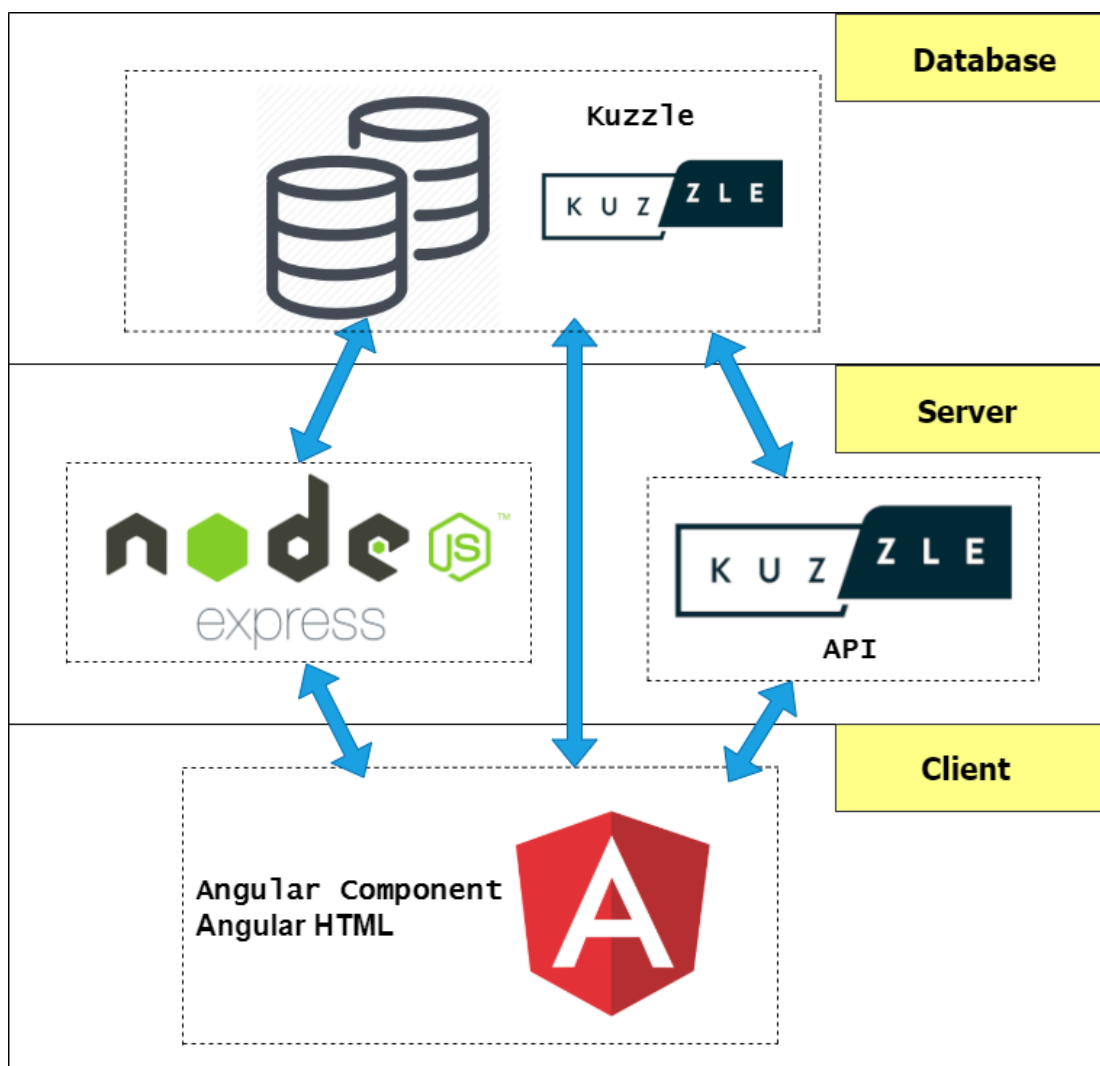
ในบทนี้จะกล่าวถึงการวิเคราะห์และออกแบบระบบของเว็บแอปพลิเคชันแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี ซึ่งการวิเคราะห์และออกแบบระบบ ที่ใช้ในการพัฒนาจะแบ่งออกเป็น 6 ขั้นตอนดังนี้

- 3.1 โครงสร้างภาพรวมของระบบ (System Architecture) เป็นการออกแบบภาพรวมและเทคโนโลยีของระบบ
- 3.2 System Requirements คือ ความต้องการหรือสิ่งที่ระบบควรจะทำ หรือหน้าที่หลักของระบบที่จะต้องทำ
- 3.3 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface Design)
- 3.4 Use Case Diagram เป็นแผนภาพที่ใช้แสดงให้เห็นว่าระบบทำงานหรือมีหน้าที่ใดบ้าง
- 3.5 Class Diagram เป็นแผนภาพที่ใช้แสดง Class และความสัมพันธ์ระหว่าง Class
- 3.6 Sequence Diagram เป็นแผนภาพที่ใช้แสดงให้เห็นถึงการตอบโต้ข้อมูลระหว่างคลาส เรียงตามลำดับของเวลาที่เกิดเหตุการณ์จากน้อยไปมาก

3.1 โครงสร้างภาพรวมของระบบ

ความหมายของ System Architecture [12] หมายถึง กรอบโครงสร้างของระบบที่อธิบายความสัมพันธ์ขององค์ประกอบต่าง ๆ ไปจนถึงขั้นการเชื่อมต่อกันของระบบย่อย โดยจัดกลุ่มองค์ประกอบไว้หลายลักษณะเพื่อให้ผู้เกี่ยวข้อง (Stakeholder) จากพื้นฐานสาขาอาชีพที่แตกต่างกันสามารถทำความเข้าใจได้ง่าย เช่น การจัดแบ่งองค์ประกอบตามลักษณะการทำงานของระบบ (functional components) เป็นต้น

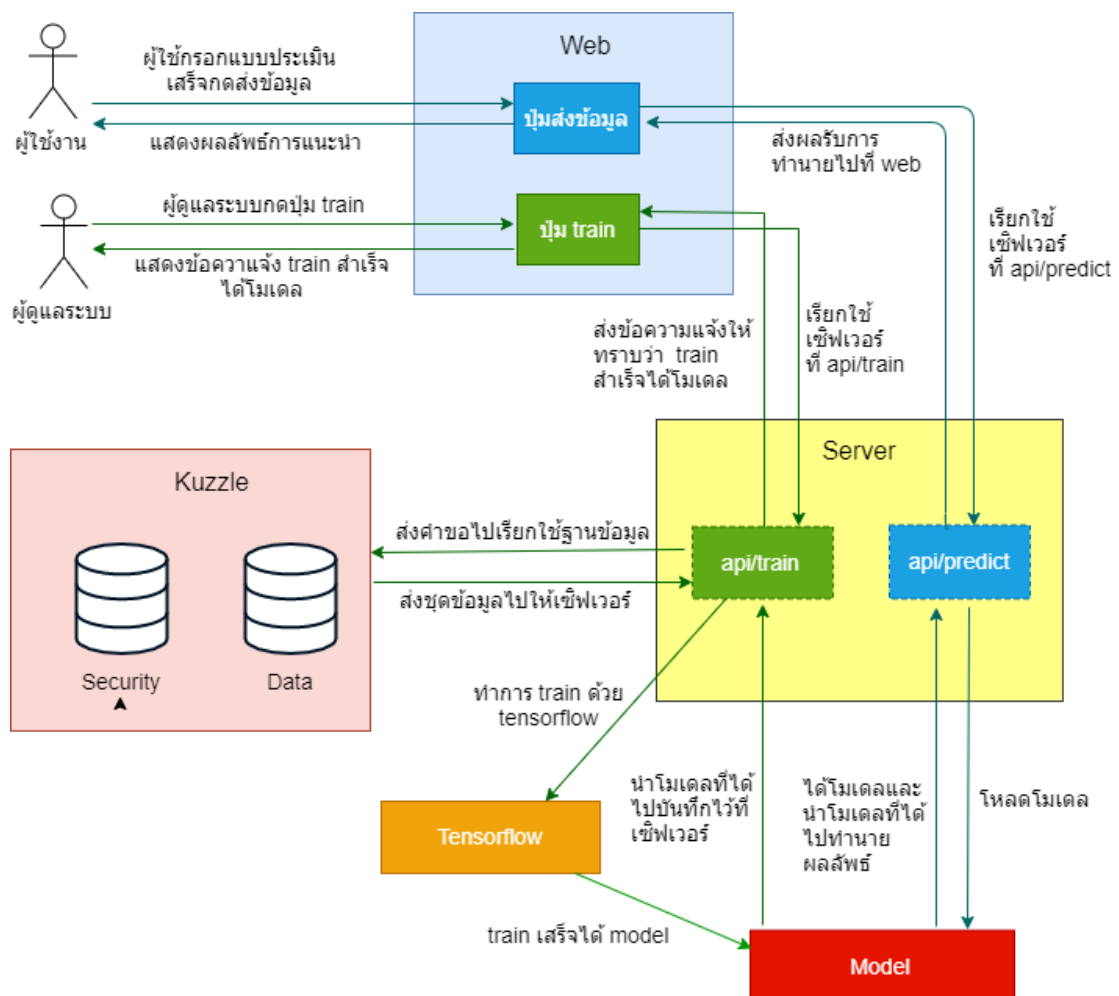
การออกแบบ System architecture แสดงภาพรวมและเทคโนโลยีของระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี ซึ่งมีรายละเอียดดังรูปที่ 3.1



รูปที่ 3.1: System architecture ระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี

จากรูปที่ 3.1 สามารถอธิบายโครงสร้างและเทคโนโลยีของระบบได้ดังนี้

1. Database ระบบใช้บริการ Backend Platform ของ Kuzzle ซึ่งเป็นฐานข้อมูล แบบ NoSQL
2. Server กระบวนการทำงานในส่วนของเซิร์ฟเวอร์ (server) แบ่งเป็น 2 ส่วนได้แก่
 - ชุดเผยแพร่สำหรับการพัฒนาเว็บไซต์ซึ่งในที่นี้ใช้ Node.js และ Express ในการพัฒนา
 - ใช้ชุดบริการ Kuzzle Api สำหรับการทำงานกับบริการต่างๆของ Kuzzle บนแพลตฟอร์มที่แตกต่างกัน
3. Client เว็บแอปพลิเคชันแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี มีวัตถุประสงค์เพื่อรองรับการทำงานของผู้ใช้งานเว็บเบราว์เซอร์พัฒนาด้วย Angular



รูปที่ 3.2: การทำงานในส่วนของการ train model และ predict

3.2 System Requirements

3.2.1 ความต้องการหลักของโปรแกรม (Functional Requirements)

ระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี แบ่งความสามารถของระบบตามประเภทของผู้ใช้งานดังนี้

1. ผู้ใช้งานทั่วไป

- สามารถสมัครสมาชิกและเข้าสู่ระบบได้
- สามารถค้นหาชื่อสถานที่ท่องเที่ยว ร้านอาหาร และที่พักได้
- สามารถดูผลลัพธ์การจัดอันดับสถานที่ท่องเที่ยว ร้านอาหาร และที่พักได้
- สามารถดูข้อมูลรายละเอียดของสถานที่ท่องเที่ยว ร้านอาหาร และที่พักได้
- สามารถเขียนรีวิวให้คะแนนสถานที่ได้
- สามารถทำแบบประเมินเพื่อวิเคราะห์หมวดท่องเที่ยวที่เหมาะสมกับผู้ใช้ได้
- สามารถเพิ่มข้อมูลสถานที่ได้
- สามารถแก้ไขข้อมูลสถานที่

2. ผู้ดูแลระบบ

- สามารถเข้าสู่ระบบได้
- สามารถเพิ่มข้อมูลสถานที่ท่องเที่ยว ร้านอาหารและที่พักได้
- สามารถแก้ไขข้อมูลสถานที่ท่องเที่ยว ร้านอาหารและที่พักได้
- สามารถลบข้อมูลสถานที่ท่องเที่ยว ร้านอาหารและที่พักได้
- สามารถสั่ง Train Model ได้

3.2.2 ความต้องการด้านคุณภาพ (Non-functional Requirements)

- ต้องเชื่อมต่ออินเทอร์เน็ตเพื่อใช้งานเว็บแอปพลิเคชัน
- สนับสนุนการแสดงผลในหน้าจอคอมพิวเตอร์เท่านั้น

3.3 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface Design)

ในการออกแบบแบ่งการออกเป็น 2 ส่วนคือ ส่วนของผู้ใช้งาน และส่วนของผู้ดูแลระบบ

3.3.1 ส่วนของผู้ใช้งาน

โดยประกอบด้วยส่วนการออกแบบหน้าจอทำงานต่างๆดังนี้

1. การออกแบบหน้าสมัครสมาชิก

The image shows a web form for registration on the 'UBON Trip' website. The header includes the site name and navigation links. The form itself is centered and contains the following elements:

- Header: UBON Trip, สถานที่ยอดนิยม, ที่เที่ยว, ที่กิน, ที่พัก, แนะนำสถานที่, เข้าสู่ระบบ
- Title: สมัครสมาชิก
- Fields:
 - ชื่อผู้ใช้ (Username)
 - ชื่อ - นามสกุล (Full Name)
 - รหัสผ่าน (Password)
 - ยืนยันรหัสผ่าน (Confirm Password)
 - เลือกเพศ (Gender dropdown)
 - เลือกรูปภาพโปรไฟล์ (Profile picture upload)
- Buttons:
 - สมัครสมาชิก (Register)
 - ยกเลิก (Cancel)

รูปที่ 3.3: หน้าสมัครสมาชิก

จากภาพที่ 3.3 การออกแบบหน้าสมัครสมาชิก หากผู้ใช้นี้ยังไม่มีบัญชีในระบบผู้ใช้งานสามารถทำการสมัครสมาชิกเพื่อเข้าใช้งานระบบได้จากหน้าสมัครสมาชิก โดยผู้ใช้งานจำเป็นต้องกรอกข้อมูลให้ครบก่อนกดปุ่มสมัครสมาชิก

2. การออกแบบหน้าจอเข้าสู่ระบบ

UBON Trip

สถานที่ยอดนิยม เที่ยว กิน พัก แนะนำสถานที่ [เข้าสู่ระบบ](#)

เข้าสู่ระบบ

[ยังไม่มีบัญชีใช้งาน? สมัครสมาชิกที่นี่](#)

รูปที่ 3.4: หน้าเข้าสู่ระบบ

จากภาพที่ 3.4 แสดงหน้าจอเข้าสู่ระบบเมื่อผู้ใช้อย่างยังไม่ได้ทำการเข้าสู่ระบบ โดยจำเป็นต้องกรอกข้อมูลชื่อผู้ใช้และรหัสผ่านเพื่อใช้ในการเข้าสู่ระบบ ซึ่งการเข้าสู่ระบบจะทำเพียงครั้งเดียวเท่านั้น เมื่อผู้ใช้เปิดการทำงานเว็บแอปพลิเคชันในครั้งถัดไประบบจะระบุข้อมูลของผู้ใช้งานอัตโนมัติ

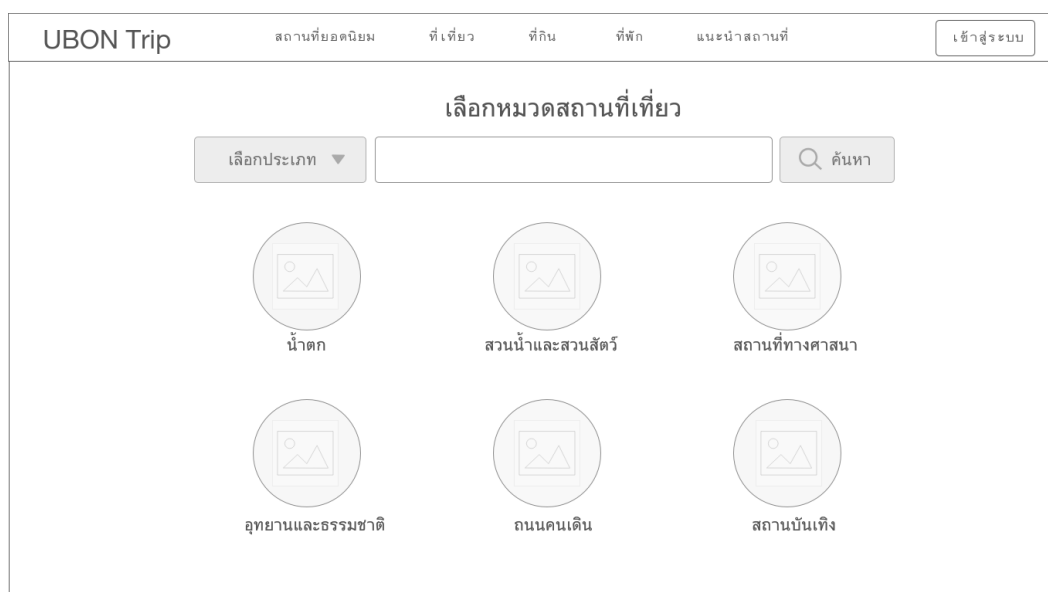
3. การออกแบบหน้าจอการจัดอันดับ



รูปที่ 3.5: หน้าจัดอันดับ

จากภาพที่ 3.5 แสดงหน้าจอการจัดอันดับสถานที่ตามหมวด หากผู้ใช้ต้องการดูข้อมูลการจัดอันดับของหมวดอื่นก็สามารถเลือกหมวดที่ต้องการได้

4. การออกแบบหน้าจอหมวดที่เที่ยว



รูปที่ 3.6: หน้าจอหมวดที่เที่ยว

จากภาพที่ 3.6 แสดงหน้าจอหมวดที่เที่ยว ซึ่งการออกแบบในหน้านี้จะมีประกอบด้วยหมวดที่เที่ยวทั้งหมด 6 หมวดได้แก่ หมวดน้ำตก หมวดสวนน้ำและสวนสัตว์ หมวดสถานที่ทางศาสนา หมวดอุทยานและธรรมชาติ หมวดถนนคนเดิน หมวดสถานบันเทิง และจะมีในส่วนของการค้นหาสถานที่ถ้าผู้ใช้ต้องการค้นหาสถานที่ก็สามารถค้นหาได้ด้วยการเลือกประเภท หลังจากนั้นก็จะกรอกชื่อสถานที่เพื่อค้นหา

5. การออกแบบหน้าจอหมวดที่กิน



รูปที่ 3.7: หน้าจอหมวดที่กิน

จากภาพที่ 3.7 แสดงหน้าจอหมวดที่กิน ซึ่งการออกแบบในหน้านี้จะมีประกอบด้วยหมวดที่กินทั้งหมด 6 หมวดได้แก่ หมวดบุฟเฟ่ต์ปิ้งย่าง หมวดกาแฟ หมวดอาหารประเภทเส้น หมวดอาหารท้องถิ่น หมวดอาหารญี่ปุ่น หมวดสเต็ก และจะมีในส่วนของการค้นหาสถานที่ถ้าผู้ใช้ต้องการค้นหาสถานที่ก็สามารถค้นหาได้ด้วยการเลือกประเภทหลังจากนั้นก็กรอกชื่อสถานที่เพื่อค้นหา

6. การออกแบบหน้าจอหมวดที่พัก

UBON Trip

สถานที่ยอดนิยม เที่ยว กิน พัก แนะนำสถานที่

เข้าสู่ระบบ

เลือกหมวดสถานที่พัก

เลือกประเภท ค้นหา

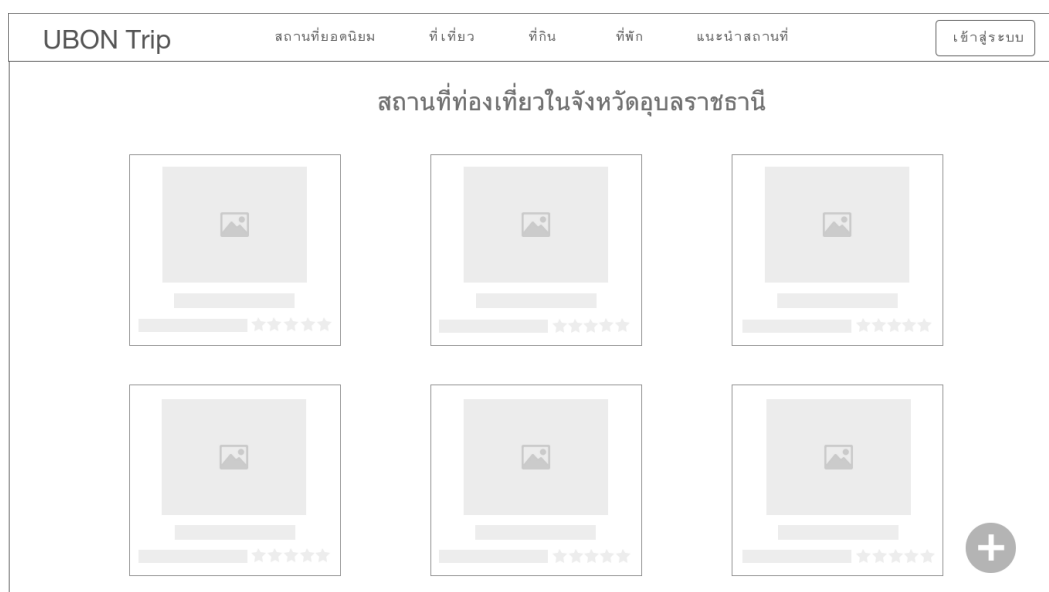
วิลล่า โรงแรม รีสอร์ท

กางเต็นท์ บังกะโล/เกสท์เฮาส์ โฮมสเตย์

รูปที่ 3.8: หน้าจอหมวดที่พัก

จากภาพที่ 3.8 แสดงหน้าจอหมวดที่พัก ซึ่งการออกแบบในหน้านี้จะมีประกอบด้วยหมวดที่พักทั้งหมด 6 ประเภทได้แก่ วิลล่า โรงแรม รีสอร์ท กางเต็นท์ บังกะโล/เกสท์เฮาส์ โฮมสเตย์ และจะมีในส่วนของการค้นหาสถานที่ถ้าผู้ใช้ต้องการค้นหาสถานที่ที่สามารถค้นหาได้ด้วยการเลือกประเภทหลังจากนั้นก็กรอกชื่อสถานที่เพื่อค้นหา

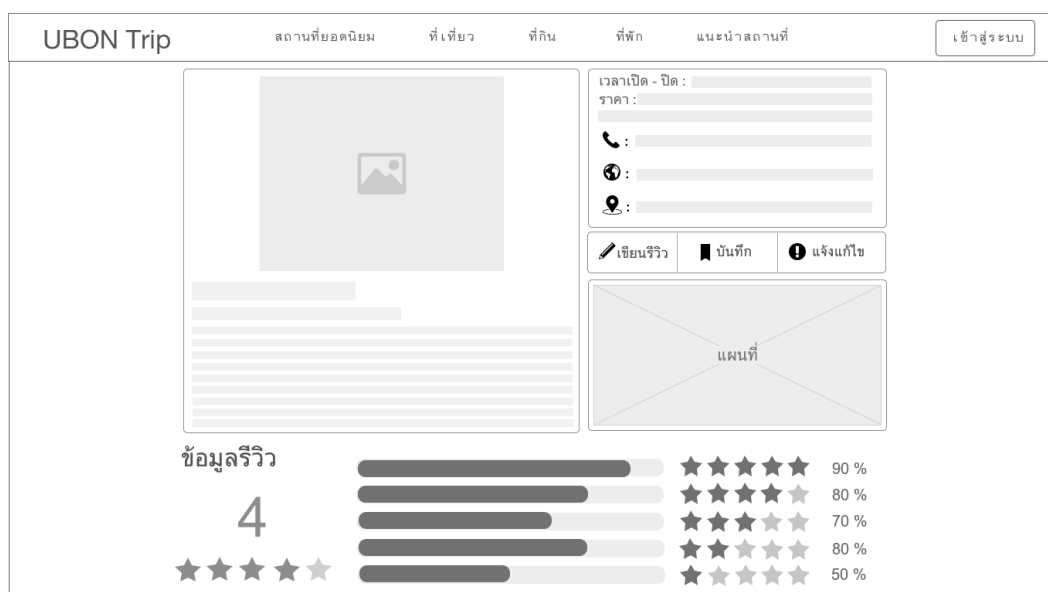
7. การออกแบบหน้าจอแสดงสถานที่



รูปที่ 3.9: หน้าจอแสดงสถานที่

จากภาพที่ 3.9 แสดงหน้าจอสถานที่ โดยจะแสดงสถานที่ตามหมวดที่ผู้ใช้เลือกซึ่งในหน้านี้จะแสดงรูปภาพสถานที่ ชื่อสถานที่ ที่อยู่สถานที่และจำนวนคะแนนที่เป็นรูปดาว

8. การออกแบบหน้าจอแสดงรายละเอียดสถานที่



รูปที่ 3.10: หน้าจอแสดงรายละเอียดสถานที่

จากภาพที่ 3.10 แสดงหน้าจอข้อมูลรายละเอียดของสถานที่ โดยข้อมูลที่แสดงในหน้านี้ ประกอบด้วย รูปภาพสถานที่ ชื่อสถานที่ รายละเอียดสถานที่ แผนที่แสดงตำแหน่งของสถานที่ ปุ่มบันทึกสถานที่ ปุ่มรีวิว ปุ่มแจ้งแก้ไข และมีในส่วนของการแสดงคะแนนเฉลี่ยจากการเขียนรีวิวของผู้ใช้

9. การออกแบบหน้าจอเขียนรีวิว

The image shows a web interface for 'UBON Trip'. At the top, there's a navigation bar with links: 'สถานที่ยอดนิยม' (Popular Places), 'เที่ยว' (Travel), 'กิน' (Eat), 'พัก' (Stay), and 'แนะนำสถานที่' (Recommend Places), along with a 'เข้าสู่ระบบ' (Login) button. The main content area displays a review submission modal. The modal has a title 'ให้คะแนนและเขียนรีวิว' and a 5-star rating system. Below the stars are input fields for 'หัวข้อรีวิว' (Review Title) and 'รายละเอียด' (Details). There's also a button 'เลือกรูปภาพ' (Select Photo) with a camera icon. At the bottom of the modal are buttons 'บันทึกรีวิว' (Save Review) and 'ยกเลิก' (Cancel). The background shows a review for a trip with a 4-star rating, a bar chart of review counts, and a 'แจ้งแก้ไข' (Report Issue) button.

รูปที่ 3.11: หน้าจอเขียนรีวิว

จากภาพที่ 3.11 แสดงหน้าจอเขียนรีวิวและให้คะแนน เมื่อผู้ใช้กดปุ่มรีวิว จากภาพที่ 3.10 โปรแกรมจะแสดงหน้าจอเขียนรีวิวโดยผู้ใช้งานจะต้องกรอกข้อมูลให้ครบก่อนกดปุ่มบันทึกรีวิว และถ้าผู้ใช้งานไม่ต้องการเขียนรีวิวก็สามารถกดปุ่ม "ยกเลิก" ได้

10. การออกแบบหน้าแนะนำสถานที่

UBON Trip		สถานที่ยอดนิยม	ที่เที่ยว	ที่กิน	ที่พัก	แนะนำสถานที่	เข้าสู่ระบบ
<div> <input checked="" type="checkbox"/> แบบประเมินส่วนที่ 1 ข้อมูลส่วนตัว </div> <div> 1 <div></div> <div> <input type="radio"/> <div></div> <input type="radio"/> <div></div> </div> <div> <input type="radio"/> <div></div> <input type="radio"/> <div></div> </div> </div> <div> <div> <input checked="" type="checkbox"/> แบบประเมินส่วนที่ 2 ข้อมูลการท่องเที่ยว </div> <div> 2 <div></div> <div> <input type="radio"/> <div></div> <input type="radio"/> <div></div> </div> <div> <input type="radio"/> <div></div> <input type="radio"/> <div></div> </div> </div> <div> <div> <input checked="" type="checkbox"/> แบบประเมินส่วนที่ 3 ปัจจัยที่มีผลต่อการตัดสินใจในการมาท่องเที่ยว </div> <div> 3 <div></div> <div> <input type="radio"/> <div></div> <input type="radio"/> <div></div> </div> <div> <input type="radio"/> <div></div> <input type="radio"/> <div></div> </div> </div> <div> <input type="button" value="ส่งข้อมูล"/> </div> </div> </div> <div data-bbox="647 936 1035 996" data-label="Caption"> <p>รูปที่ 3.12: หน้าจอบแบบประเมิน</p> </div> <div data-bbox="359 1084 1407 1261" data-label="Text"> <p>จากภาพที่ 3.12 แสดงหน้าจอบแบบประเมินโดยมีข้อมูลในการทำแบบประเมินแบ่งออกเป็น 3 ส่วนได้แก่ ข้อมูลส่วนตัว ข้อมูลการท่องเที่ยวและปัจจัยที่มีผลต่อการตัดสินใจในการมาท่องเที่ยว</p> </div>							

11. การออกแบบหน้าเพิ่มสถานที่

UBON Trip สถานที่ยอดนิยม ที่เที่ยว ที่กิน ที่พัก แนะนำสถานที่ เข้าสู่ระบบ

+ เพิ่มข้อมูลสถานที่

ชื่อสถานที่

ที่อยู่

อำเภอ - จังหวัด

เลือกหมวดสถานที่ ▼

รายละเอียด

เบอร์โทรศัพท์

ละติจูด ลองจิจูด

เว็บไซต์

เลือกรูปภาพ

บันทึก ยกเลิก

รูปที่ 3.13: หน้าจอเพิ่มสถานที่

จากภาพที่ 3.13 แสดงหน้าจอเพิ่มสถานที่โดยมีช่องให้กรอกชื่อสถานที่ ที่อยู่ รายละเอียด เบอร์โทร เว็บไซต์(website) เฟสบุค(facebook) ละติจูด(latitude) ลองจิจูด(longitude)และรูปภาพสถานที่

12. การออกแบบหน้าแสดงผลการแนะนำ

UBON Trip สถานที่ยอดนิยม ที่เที่ยว ที่กิน ที่พัก แนะนำสถานที่ เข้าสู่ระบบ

แบบประเมินส่วนที่ 1 ข้อมูลส่วนตัว

1

2

3

ที่เที่ยว

ส่งข้อมูล

ผลลัพธ์การแนะนำ

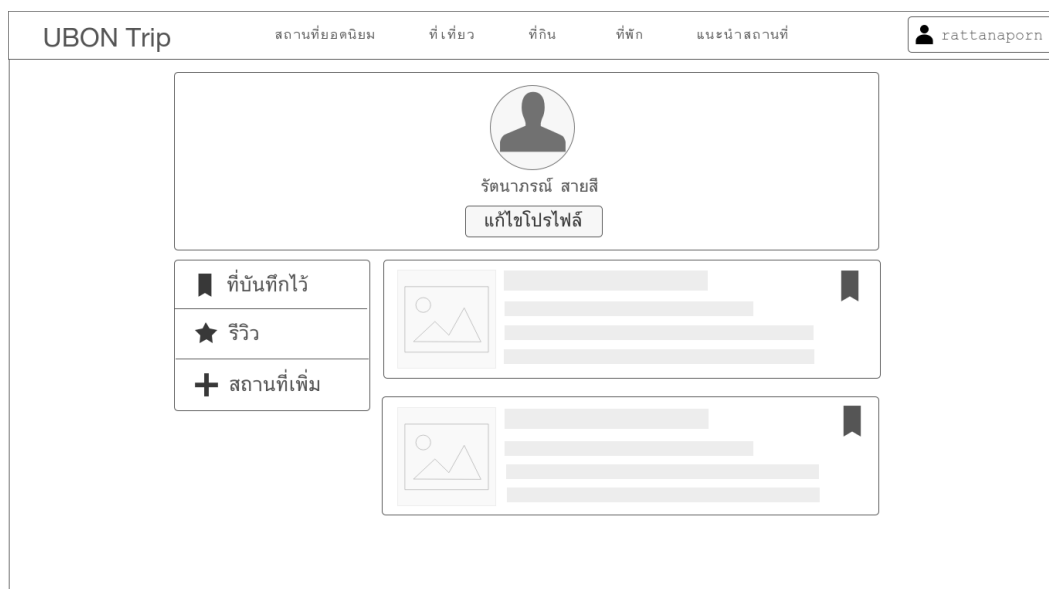
- ☐ สถานที่เที่ยง 40 %
- ☐ ถนนคนเดิน 15 %
- ☐ สวนน้ำ/สวนสัตว์ 15 %
- ☐ อุทยานและธรรมชาติ 10 %
- ☐ สถานที่ทางศาสนา 10 %
- ☐ น้ำตก 10 %

ใช่ ยกเลิก

รูปที่ 3.14: หน้าจอผลลัพธ์การแนะนำสถานที่

จากภาพที่ 3.14 แสดงหน้าจอผลลัพธ์การแนะนำหมวดท่องเที่ยวที่เหมาะสมกับผู้ใช้ ทั้งนี้ ผลลัพธ์การแนะนำได้มาจากการข้อมูลที่ใช้กรอกแบบประเมิน ซึ่งในหน้าจอแสดงผลการเรียงลำดับจากตัวที่มีค่าสูงที่สุดไปหาตัวที่ต่ำสุด และแสดงค่าเปอร์เซ็นต์ในรูปแบบของ progress bar

13. การออกแบบหน้าจoproไฟล์ของผู้ใช้งาน



รูปที่ 3.15: หน้าจoproไฟล์

จากภาพที่ 3.15 แสดงหน้าจoproไฟล์ของผู้ใช้โดยหน้านี้แสดงข้อมูลการเขียนรีวิว การบันทึกสถานที่และข้อมูลการเพิ่มสถานที่ของผู้ใช้งาน นอกจากนี้ผู้ใช้งานสามารถลบรีวิว แก้ไขรีวิว ลบข้อมูลสถานที่ แก้ไขข้อมูลสถานที่ได้

3.3.2 ส่วนของผู้ดูแลระบบ

1. การออกแบบหน้าจัดการข้อมูลสถานที่เที่ยว

Admin แจ้งแก้ไข จัดการข้อมูลเที่ยว จัดการข้อมูลที่พัก จัดการข้อมูลที่พัก Train

จัดการข้อมูลสถานที่เที่ยว

เลือกประเภท ▼ + เพิ่มข้อมูลสถานที่เที่ยว

ลำดับ	ชื่อสถานที่	ประเภท	ที่อยู่	หมายเหตุ
1				แก้ไข ลบ
2				แก้ไข ลบ
3				แก้ไข ลบ
4				แก้ไข ลบ
5				แก้ไข ลบ


<< Previous 1 2 3 4 Next >>

รูปที่ 3.16: หน้าจอจัดการข้อมูลสถานที่เที่ยว

จากภาพที่ 3.16 แสดงหน้าจอในส่วนของผู้ดูแลระบบ โดยในหน้านี้จะแสดงข้อมูลรายชื่อสถานที่ท่องเที่ยวทั้งหมดที่มีอยู่ในระบบผู้ดูแลระบบสามารถเลือกประเภทข้อมูลเพื่อดูข้อมูลเฉพาะประเภทนั้นๆได้ ซึ่งผู้ดูแลระบบสามารถกดปุ่มเพิ่มข้อมูลสถานที่เที่ยวได้ กดแก้ไขข้อมูลสถานที่หรือกดปุ่มลบข้อมูลสถานที่ได้

2. การออกแบบหน้าจัดการข้อมูลสถานที่กิน

Admin
แจ้งแก้ไข
จัดการข้อมูลที่เกี่ยวข้อง
จัดการข้อมูลที่พัก
จัดการข้อมูลที่พัก
Train



จัดการข้อมูลสถานที่กิน

เลือกประเภท

+ เพิ่มข้อมูลร้านอาหาร

ลำดับ	ชื่อสถานที่	ประเภท	ที่อยู่	หมายเหตุ
1				แก้ไข ลบ
2				แก้ไข ลบ
3				แก้ไข ลบ
4				แก้ไข ลบ
5				แก้ไข ลบ


<< Previous
1
2
3
4
Next >>

รูปที่ 3.17: หน้าจอจัดการข้อมูลสถานที่กิน

จากภาพที่ 3.17 แสดงหน้าจอในส่วนของผู้ดูแลระบบ โดยในหน้านี้จะแสดงข้อมูลรายชื่อสถานที่ร้านอาหารทั้งหมดที่มีอยู่ในระบบผู้ดูแลระบบสามารถกดเลือกประเภทข้อมูลเพื่อดูข้อมูลเฉพาะประเภทนั้นๆได้ ซึ่งผู้ดูแลระบบสามารถกดปุ่มเพิ่มข้อมูลสถานที่กินได้ กดแก้ไขข้อมูลสถานที่หรือกดปุ่มลบข้อมูลสถานที่ได้

3. การออกแบบหน้าจัดการข้อมูลสถานที่พัก

Admin
แจ้งแก้ไข
จัดการข้อมูลเที่ยว
จัดการข้อมูลที่พัก
จัดการข้อมูลที่พัก
Train



จัดการข้อมูลสถานที่พัก

เลือกประเภท ▼

+ เพิ่มข้อมูลสถานที่พัก

ลำดับ	ชื่อสถานที่	ประเภท	ที่อยู่	หมายเหตุ
1				แก้ไข ลบ
2				แก้ไข ลบ
3				แก้ไข ลบ
4				แก้ไข ลบ
5				แก้ไข ลบ

<< Previous
1
2
3
4
Next >>

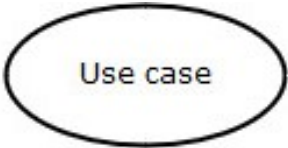


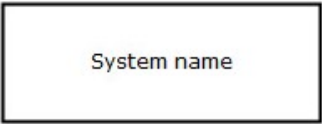
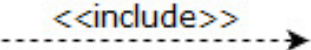
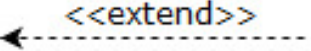
รูปที่ 3.18: หน้าจอจัดการข้อมูลสถานที่พัก

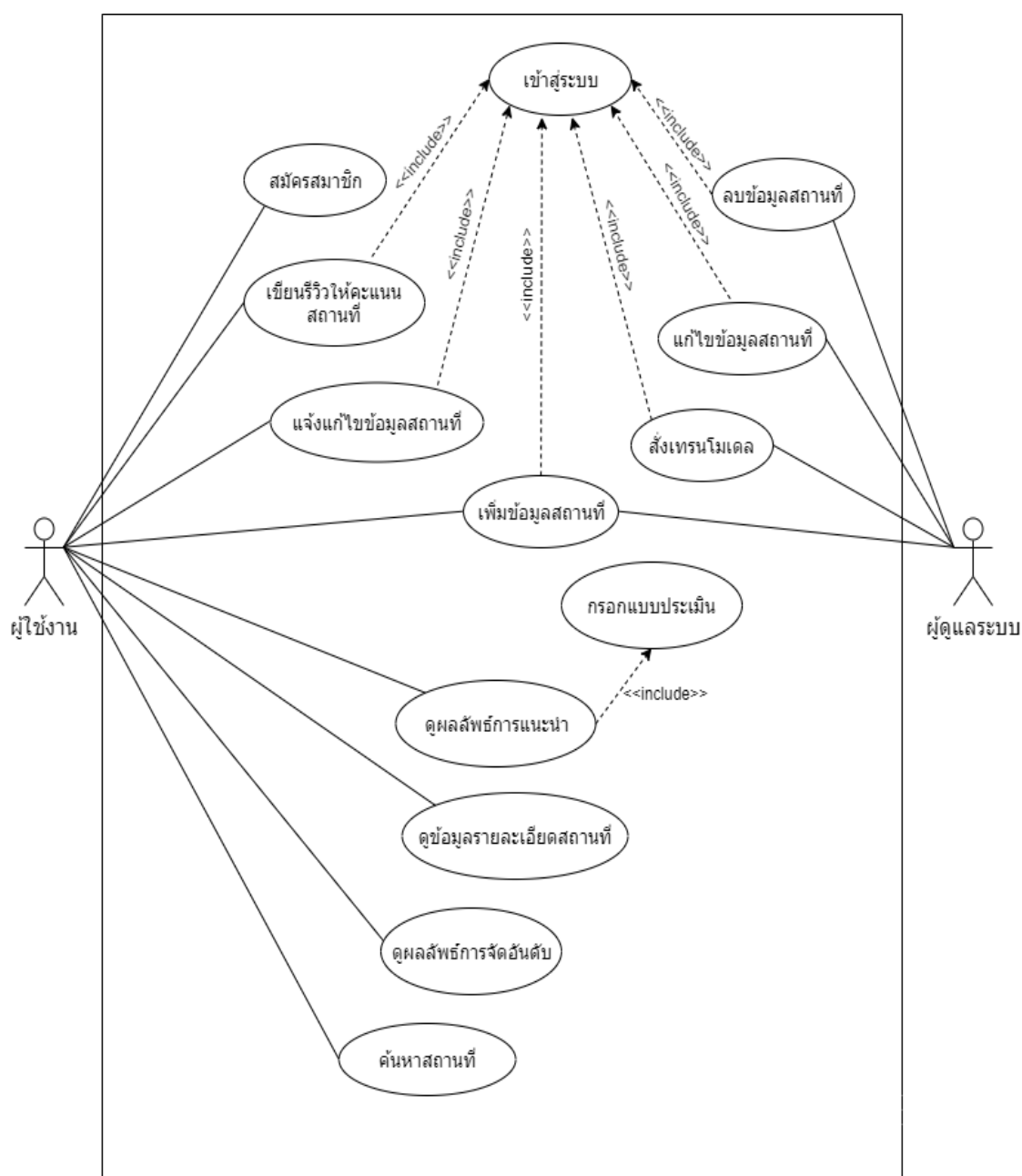
จากภาพที่ 3.18 แสดงหน้าจอในส่วนของผู้ดูแลระบบ โดยในหน้านี้จะแสดงข้อมูลรายชื่อสถานที่พักทั้งหมดที่มีอยู่ในระบบผู้ดูแลระบบสามารถกดเลือกประเภทข้อมูลเพื่อดูข้อมูลเฉพาะประเภทนั้นๆได้ ซึ่งผู้ดูแลระบบสามารถกดปุ่มเพิ่มข้อมูลสถานที่พักได้ กดแก้ไขข้อมูลสถานที่หรือกดปุ่มลบข้อมูลสถานที่ได้

3.4 Use Case Diagram

Use Case Diagram เป็นแผนผังเพื่อแสดงฟังก์ชันการทำงานของระบบโดยรวม แสดงส่วนประกอบในระบบและกิจกรรมที่เกิดขึ้นในระบบ สัญลักษณ์ที่ใช้ในการเขียน Use Case Diagram แสดงในตารางที่ 3.1

ตารางที่ 3.1: สัญลักษณ์ของ Use case Diagram

สัญลักษณ์	คำอธิบาย
	Use case คือส่วนย่อยของระบบงาน แทนด้วยวงรีและชื่อของ Use case ภายในวงรี
	Actor คือบุคคลหรือระบบงานอื่นที่ใช้งานระบบหรือได้รับประโยชน์จากระบบซึ่งอยู่ภายนอกระบบ แทนด้วยรูปคนและมีชื่อบทบาทการใช้งานระบบ
	เส้นตรงที่แสดงถึงการใช้งาน Use case ของผู้กระทำ
	กรอบสี่เหลี่ยม แสดง ถึง ขอบเขต ของ ระบบ โดย แสดง ชื่อ ระบบ ภายในหรือด้านบนกรอบสี่เหลี่ยม Use case อยู่ภายในกรอบสี่เหลี่ยม และ actor อยู่ภายนอกกรอบสี่เหลี่ยม
	ความสัมพันธ์แบบ «includes» แสดงว่า Use case ด้านหางลูกศร จำเป็นต้องมีการทำงานของ Use case ด้านหัวลูกศรทุกครั้งที่มีการทำงาน
	ความสัมพันธ์แบบ «extend» แสดงว่า Use case หนึ่งดำเนินการตามขั้นตอนของ Use case อื่น โดยแทนด้วยสัญลักษณ์ลูกศรเส้นประ ซึ่ง Use case ด้านหางลูกศรเป็นส่วนเสริมในการทำงานของ Use case ที่อยู่ด้านหัวลูกศร



รูปที่ 3.19: Use Case Diagram ระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี

ตารางที่ 3.2: อธิบาย Use Case หน้าที่ของระบบในภาพที่ 3.3

Use Case	คำอธิบาย
สมัครสมาชิก	ผู้ใช้งานสามารถสมัครสมาชิกเพื่อเข้าใช้งานในระบบได้
เข้าสู่ระบบ	ผู้ใช้ที่เป็นสมาชิกสามารถใช้ username และ password ของตัวเองเพื่อเข้าสู่ระบบได้
เขียนรีวิวให้คะแนนสถานที่	เมื่อผู้ใช้งานเข้าสู่ระบบเรียบร้อยแล้วจะสามารถเขียนรีวิวและให้คะแนนสถานที่ต่างๆได้
เพิ่มข้อมูลสถานที่	เมื่อผู้ใช้งานและผู้ดูแลระบบเข้าสู่ระบบเรียบร้อยแล้วจะสามารถเพิ่มข้อมูลสถานที่ลงในระบบได้
กรอกแบบประเมิน	ผู้ใช้งานสามารถกรอกข้อมูลแบบประเมินได้โดยไม่จำเป็นต้องเข้าสู่ระบบ
ดูผลลัพธ์การแนะนำ	ผู้ใช้งานสามารถดูผลลัพธ์การแนะนำหมวดท่องเที่ยวที่เหมาะสมกับผู้ใช้ได้โดยไม่ต้องเข้าสู่ระบบ ซึ่งข้อมูลจะแสดงชื่อหมวดและค่าเปอร์เซ็นต์ของแต่ละหมวดโดยเรียงจากมากไปน้อย
ดูรายละเอียดสถานที่	ผู้ใช้งานสามารถดูข้อมูลรายละเอียดของสถานที่ที่ผู้ใช้งานต้องการได้โดยไม่ต้องเข้าสู่ระบบ
ดูผลลัพธ์การจัดอันดับ	ผู้ใช้งานสามารถดูผลลัพธ์การจัดอันดับของสถานที่และสามารถเลือกหมวดที่ต้องการดูผลลัพธ์การจัดอันดับได้โดยไม่ต้องเข้าสู่ระบบ ซึ่งข้อมูลที่แสดงจะเรียงลำดับตามจำนวนดาวจากมากไปน้อย
ค้นหาสถานที่	ผู้ใช้งานสามารถค้นหาสถานที่ที่ผู้ใช้งานต้องการได้โดยไม่ต้องเข้าสู่ระบบ

ตารางที่ 3.3: อธิบาย Use Case หน้าหนึ่งของระบบในภาพที่ 3.3 (ต่อ)

Use Case	คำอธิบาย
แจ้งแก้ไขข้อมูลสถานที่	ผู้ใช้งานสามารถแจ้งแก้ไขข้อมูลสถานที่ได้โดยจะต้องเข้าสู่ระบบให้เรียบร้อยก่อน
ลบข้อมูลสถานที่	ใช้สำหรับผู้ดูแลระบบสามารถจัดการลบข้อมูลสถานที่ออกจากกรระบบได้
แก้ไขข้อมูลสถานที่	ใช้สำหรับผู้ดูแลระบบสามารถจัดการแก้ไขข้อมูลสถานที่ในระบบได้
สั่งเทรนโมเดล	ใช้เพื่อปรับโมเดลให้มีความแม่นยำมากยิ่งขึ้น

ตารางที่ 3.4: Use Case สมัครงาน

Use Case Title : สมัครงาน	Use case Id : 1
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : ผู้ใช้งานสามารถสมัครงานเพื่อเข้าใช้งานระบบได้	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ตจะไม่สามารถสมัครงานได้	
Exceptional Flow ที่ 2 : username ที่กรอกเข้ามาจะต้องไม่ซ้ำในระบบถึงจะสามารถสมัครงานได้	

ตารางที่ 3.5: Use Case เข้าสู่ระบบ

Use Case Title : เข้าสู่ระบบ	Use case Id : 2
Primary Actor : ผู้ใช้งาน,ผู้ดูแลระบบ	
Stakeholder Actor : -	
Main Flow : ในส่วนนี้ ผู้ใช้งานและผู้ดูแลระบบ สามารถใช้ username และ password ที่มีเพื่อลงชื่อเข้าใช้งานระบบได้	
Exceptional Flow ที่ 1 : หากผู้ใช้งานและผู้ดูแลระบบไม่เชื่อมต่ออินเทอร์เน็ตจะไม่สามารถเข้าสู่ระบบได้	
Exceptional Flow ที่ 2 : หากผู้ใช้กรอก username และ password ไม่ถูกต้องจะไม่สามารถเข้าสู่ระบบได้	

ตารางที่ 3.6: Use Case เขียนรีวิวให้คะแนนสถานที่

Use Case Title : เขียนรีวิวให้คะแนนสถานที่	Use case Id : 3
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : ในส่วนของรีวิวสถานที่ ผู้ใช้งานเข้าสู่ระบบเรียบร้อยแล้วจะรีวิวสถานที่ได้ก็ต่อเมื่อมาดูข้อมูลสถานที่และทำการกดปุ่มรีวิว เมื่อกดปุ่มรีวิวจะแสดงกล่องข้อความให้ผู้ใช้งานรีวิวและให้คะแนนสถานที่	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ตและยังไม่ได้เข้าสู่ระบบจะไม่สามารถเขียนรีวิวได้	

ตารางที่ 3.7: Use Case เพิ่มข้อมูลสถานที่

Use Case Title : เพิ่มข้อมูลสถานที่	Use case Id : 4
Primary Actor : ผู้ใช้งาน,ผู้ดูแลระบบ	
Stakeholder Actor : -	
Main Flow : ในส่วนของการเพิ่มสถานที่ ผู้ใช้งานและผู้ดูแลระบบเข้าสู่ระบบเรียบร้อยแล้วเมื่อกดปุ่มเพิ่มสถานที่ที่จะแสดงหน้าเพิ่มสถานที่ซึ่งประกอบไปด้วย ชื่อสถานที่ ประเภทสถานที่ ที่อยู่ของสถานที่ รายละเอียดของสถานที่ เวลาทำการของสถานที่ เบอร์โทรศัพท์ของสถานที่ เว็บไซต์ของสถานที่ ตำแหน่งละติจูดและลองจิจูดของสถานที่และรูปภาพของสถานที่	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ตและยังไม่ได้เข้าสู่ระบบจะไม่สามารถเพิ่มข้อมูลสถานที่ได้	

ตารางที่ 3.8: Use Case กรอกแบบประเมิน

Use Case Title : กรอกแบบประเมิน	Use case Id : 5
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : เมื่อผู้ใช้งานกดปุ่มแนะนำสถานที่ที่จะแสดงหน้าแบบประเมินเพื่อให้ผู้ใช้กรอกแบบประเมินซึ่งแบบประเมินมีทั้งหมด 3 ส่วนได้แก่ ข้อมูลส่วนตัว ข้อมูลการท่องเที่ยวและปัจจัยที่มีผลต่อการตัดสินใจในการมาท่องเที่ยว	
Exceptional Flow ที่ 1 : หากผู้ใช้งานไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถกรอกแบบประเมินได้	
Exceptional Flow ที่ 2 : ผู้ใช้งานจะต้องกรอกข้อมูลให้ครบทุกข้อถึงจะสามารถกดส่งข้อมูลแบบประเมินได้	

ตารางที่ 3.9: Use Case คู่มือการแนะนำ

Use Case Title : คู่มือการแนะนำ	Use case Id : 6
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : ส่วนคู่มือการแนะนำสถานที่ เมื่อผู้ใช้งานส่งข้อมูลระบบจะแสดงหน้าผลการแนะนำสถานที่โดยแสดงชื่อหมวดและค่าเปอร์เซ็นต์เรียงจากค่ามากไปน้อย หมวดที่ระบบจะแนะนำมีทั้งหมด 6 หมวดได้แก่ น้ำตก สวนน้ำและสวนสัตว์ สถานที่ทางศาสนา อุทยานและธรรมชาติ ถนนคนเดินและสถาบันเทิง	

ตารางที่ 3.10: Use Case ดูรายละเอียดสถานที่

Use Case Title : ดูรายละเอียดสถานที่	Use case Id : 7
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : ส่วนดูรายละเอียดสถานที่ เมื่อผู้ใช้งานที่รูปภาพหรือชื่อสถานที่ระบบจะแสดงหน้าข้อมูลสถานที่โดยข้อมูลที่แสดงประกอบด้วย ชื่อและข้อมูลทั่วไปของสถานที่ แผนที่แสดงตำแหน่งของสถานที่ ข้อมูลการรีวิวและคะแนนที่ได้	
Exceptional Flow ที่ 1 : หากผู้ใช้งานไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถดูรายละเอียดสถานที่ได้	

ตารางที่ 3.11: Use Case ดูผลลัพธ์การจัดอันดับ

Use Case Title : ดูผลลัพธ์การจัดอันดับ	Use case Id : 10
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : ส่วนดูผลลัพธ์การจัดอันดับ เมื่อใช้กดปุ่มสถานที่ยอดนิยมระบบจะแสดงหน้าข้อมูลการจัดอันดับสถานที่ซึ่งผู้ใช้สามารถเลือกประเภทเพื่อดูผลลัพธ์การจัดอันดับ โดยมีทั้งหมด 3 ประเภท ได้แก่ สถานที่ท่องเที่ยว ร้านอาหารและที่พัก ข้อมูลที่จะแสดงจะเรียงลำดับตามจำนวนดาวจากมากไปน้อย	
Exceptional Flow ที่ 1 : หากผู้ใช้งานไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถดูผลลัพธ์สถานที่ยอดนิยมได้	

ตารางที่ 3.12: Use Case ค้นหาสถานที่

Use Case Title : ค้นหาสถานที่	Use case Id : 11
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : ส่วนค้นหาสถานที่ เมื่อผู้ใช้เลือกประเภทและกรอกชื่อสถานที่ที่ต้องการค้นหา ระบบจะแสดงรายการสถานที่ค้นหาพบและเมื่อผู้ใช้กดสถานที่ที่ต้องการระบบจะแสดงหน้าข้อมูลของสถานที่นั้น	
Exceptional Flow ที่ 1 : หากผู้ใช้งานไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถค้นหาสถานที่ได้	
Exceptional Flow ที่ 2 : หากผู้ใช้งานไม่เลือกประเภทสถานที่และไม่กรอกชื่อสถานที่ จะไม่สามารถค้นหาสถานที่ได้	

ตารางที่ 3.13: Use Case แจ้งแก้ไขข้อมูลสถานที่

Use Case Title : แจ้งแก้ไข	Use case Id : 13
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : ในส่วนของการแจ้งแก้ไข เมื่อผู้ใช้งานกดปุ่มแจ้งแก้ไขระบบจะแสดงหน้าแจ้งแก้ไขข้อมูลซึ่งผู้ใช้งานสามารถพิมพ์ข้อความที่ต้องการแก้ไขได้	
Exceptional Flow ที่ 1 : ผู้ใช้งานจะต้องเข้าสู่ระบบให้เรียบร้อยก่อนถึงจะสามารถกดปุ่มแจ้งแก้ไขข้อมูลสถานที่ที่ต้องการแก้ไขได้	

ตารางที่ 3.14: Use Case ลบข้อมูลสถานที่

Use Case Title : ลบข้อมูลสถานที่	Use case Id : 14
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Main Flow : ผู้ดูแลระบบสามารถลบข้อมูลสถานที่ออกจากระบบได้	
Exceptional Flow ที่ 1 : หากผู้ดูแลระบบไม่เชื่อมต่ออินเทอร์เน็ตหรือไม่ได้เข้าสู่ระบบ จะไม่สามารถลบข้อมูลสถานที่ออกจากระบบได้	

ตารางที่ 3.15: Use Case แก้ไขข้อมูลสถานที่

Use Case Title : แก้ไขข้อมูลสถานที่	Use case Id : 15
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Main Flow : ผู้ดูแลระบบสามารถแก้ไขข้อมูลสถานที่ได้	
Exceptional Flow ที่ 1 : หากผู้ดูแลระบบไม่เชื่อมต่ออินเทอร์เน็ตหรือไม่ได้เข้าสู่ระบบ จะไม่สามารถแก้ไขข้อมูลสถานที่ได้	



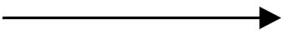

ตารางที่ 3.16: Use Case สั้งเทรนโมเดล

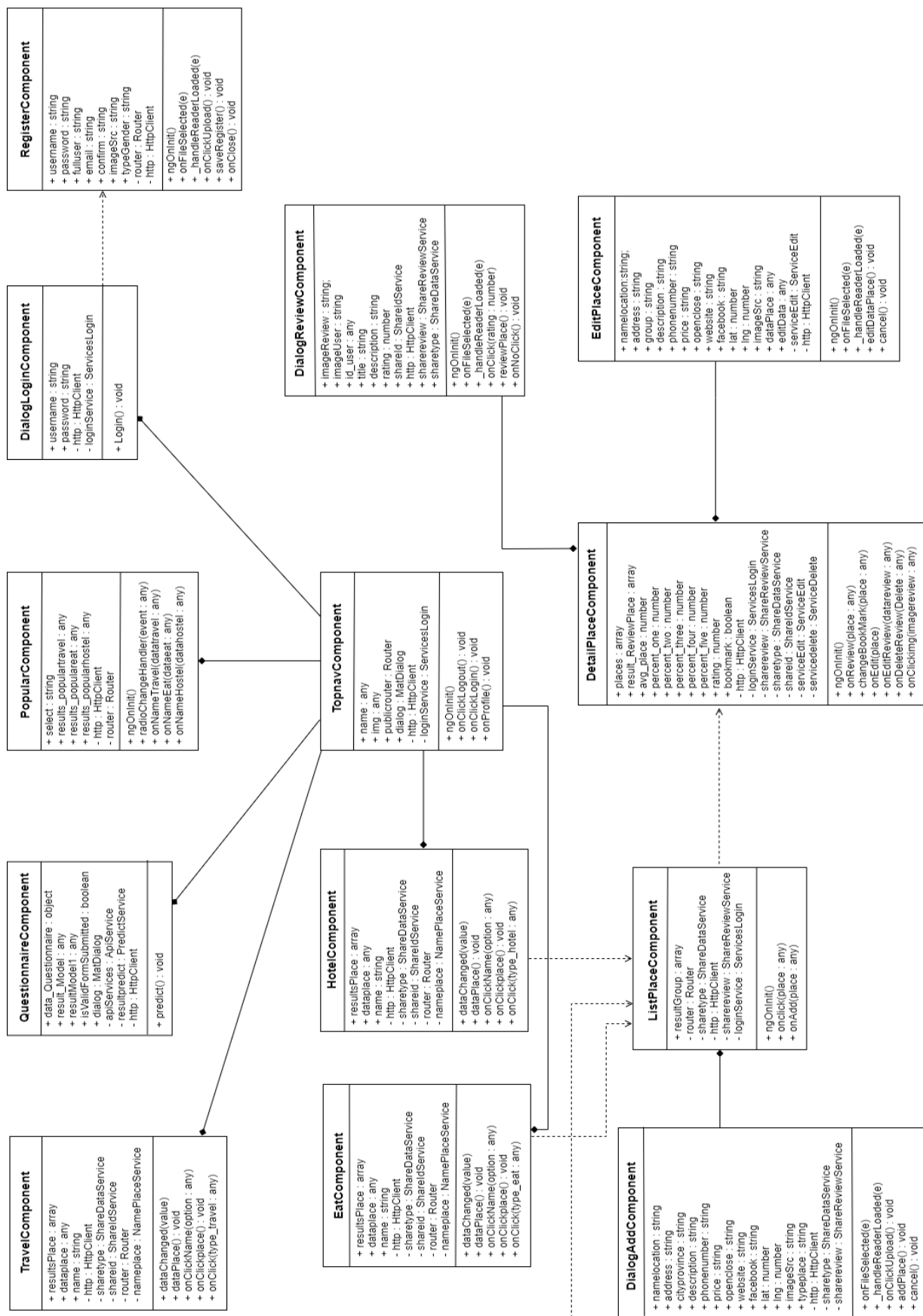
Use Case Title : Train model	Use case Id : 17
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Main Flow : ผู้ดูแลระบบสามารถกด Train model ได้	
Exceptional Flow ที่ 1 : หากผู้ดูแลระบบไม่เชื่อมต่ออินเทอร์เน็ตหรือไม่ได้เข้าสู่ระบบ จะไม่สามารถ Train model ได้	

3.5 Class Diagram

Class Diagram คือแผนภาพที่ใช้แสดงคลาสและความสัมพันธ์ในแบบต่างๆ ระหว่างคลาส สัญลักษณ์ที่ใช้ในการเขียน Class Diagram แสดงในตารางที่ 3.17

ตารางที่ 3.17: สัญลักษณ์ของ Class Diagram

สัญลักษณ์	คำอธิบาย
<div> <div>Class Name</div> <div>Attribute Name</div> <div>Operation Name()</div> </div>	Class สัญลักษณ์แทนด้วยสี่เหลี่ยมแบ่งเป็น 3 ส่วน ส่วนบนเป็นชื่อ Class ส่วนกลางเป็น Attribute และส่วนล่างเป็น Operation Name หรือ Method ซึ่งคลาสเป็นสิ่งที่เก็บรวบรวมข้อมูลที่แสดงถึงบุคคล สถานที่ เหตุการณ์หรือสิ่งต่างๆ ที่มีความเกี่ยวข้องกับระบบ Method เป็นการกระทำหรือฟังก์ชันที่คลาสนั้นสามารถทำได้
Method Name()	Method สามารถแบ่งการมองเห็น (Visibility) ได้ 3 ชนิด ได้แก่ <ul style="list-style-type: none"> - Public แทนสัญลักษณ์ด้วยเครื่องหมายบวก (+) - Private แทนสัญลักษณ์ด้วยเครื่องหมายลบ (-) - Protected แทนสัญลักษณ์ด้วยเครื่องหมายบวกร (#)
	Dependency Relationship หมายความว่า คลาสที่อยู่ฝั่งต้นลูกศรสามารถเรียกใช้คลาสที่อยู่ฝั่งหัวลูกศร
	Aggregation เป็นความสัมพันธ์ระหว่าง Object หรือ class แบบ "Whole-Part" หรือ "is part of" โดยจะมี class ที่เป็นใหญ่ที่สุดเป็น object หลัก และมี class อื่นเป็นส่วนประกอบ
	Association Relationship หมายความว่า คลาสที่อยู่ฝั่งต้นลูกศรทำการกำหนดคลาสอื่นในรูป Attribute ภายในคลาส และสามารถเรียกใช้ Method จากคลาสนั้นได้
	Connector เป็นสัญลักษณ์แทนด้วยรูปห้าเหลี่ยมและมีชื่ออยู่ตรงกลาง จะสร้างสัญลักษณ์นี้ไว้เมื่อต้องการเชื่อมต่อคลาสที่อยู่คนละหน้า



รูปที่ 3.20: Class Diagram ระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี



รูปที่ 3.21: Class Diagram ระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี

ตารางที่ 3.18: อธิบาย Class Diagram ผู้ใช้งานจากภาพที่ 3.20 ได้ดังนี้

Class Diagram	คำอธิบาย
QuestionnaireComponent	คลาส QuestionnaireComponent เป็น คลาส ที่ ใช้ แสดง หน้า ให้ ผู้ ใช้ กรอก แบบ ประเมิน เพื่อ ดู ผลลัพธ์ การ แนะนำ หมวด ท่อง เที่ยว ซึ่ง สามารถ เรียก ใช้ คลาส DialogResultComponent
PopularComponent	คลาส PopularComponent เป็นคลาสที่ใช้แสดงหน้าแรกเมื่อเข้ามาใช้งานโดยหน้านี้จะแสดงสถานที่ยอดนิยม
DialogLoginComponent	คลาส DialogLoginComponent เป็น คลาส ที่ ใช้ เพื่อให้ สมาชิก ที่ได้ ลง ทะเบียน กับ ระบบ เข้า สู่ ระบบ เพื่อ ใช้งาน บริการ ต่าง ๆ จาก ระบบ ซึ่ง สามารถ เรียก ใช้ คลาส RegisterComponent
RegisterComponent	คลาส RegisterComponent เป็นคลาสที่ใช้แสดงหน้าสมัครสมาชิกเพื่อให้ผู้ใช้ที่ต้องการลงทะเบียนเข้าสู่ระบบสามารถใช้งานได้
TopnavComponent	คลาส TopnavComponent เป็น คลาส ที่ จัดการ เกี่ยว กับ การแสดงแถบเมนูส่วนบนของเว็บไซต์ ซึ่งจะสามารถเรียกใช้ คลาสQuestionnaireComponent PopularComponent DialogLoginComponent ProfileComponent TravelComponent EatComponent และ คลาส HotelComponent
TravelComponent	คลาส TravelComponent เป็นคลาสที่ใช้แสดงหน้าหมวดสถานที่ ท่อง เที่ยว เพื่อให้ ผู้ ใช้ได้ เลือก ดู ข้อมูล สถานที่ ตาม ประเภท ของหมวดต่างๆได้ ซึ่งสามารถเรียกใช้ คลาส ListPlaceComponent ในการแสดงหน้ารายการสถานที่

ตารางที่ 3.19: อธิบาย Class Diagram ผู้ใช้งานจากภาพที่ 3.20 ได้ดังนี้ (ต่อ)

Class Diagram	คำอธิบาย
EatComponent	คลาส EatComponent เป็น คลาส ที่ ใช้ แสดง หน้า หมวดร้านอาหารเพื่อให้ผู้ใช้ได้เลือกดูข้อมูลสถานที่ตาม ประเภท ของ หมวด ต่างๆ ได้ ซึ่ง สามารถ เรียก ใช้ คลาส ListPlaceComponent ในการแสดงหน้ารายการสถานที่
HotelComponent	คลาส HotelComponent เป็น คลาส ที่ ใช้ แสดง หน้า หมวดสถานที่พักเพื่อให้ผู้ใช้ได้เลือกดูข้อมูลสถานที่ตาม ประเภท ของ หมวด ต่างๆ ได้ ซึ่ง สามารถ เรียก ใช้ คลาส ListPlaceComponent ในการแสดงหน้ารายการสถานที่
DialogAddComponent	คลาส DialogAddComponent เป็นคลาสที่ใช้ในการแสดง หน้าเพิ่มข้อมูลสถานที่
ListPlaceComponent	คลาส ListPlaceComponent เป็นคลาสที่ใช้ในการแสดง ข้อมูล สถานที่ ที่ มี การ ทำงาน หลัก คือ สืบค้น ฐาน ข้อมูล จาก Kuzzle เพื่อนำมาแสดง และสามารถเรียกใช้ คลาส DialogAddComponent เมื่อผู้ใช้กดปุ่มเพิ่มสถานที่
DialogReviewComponent	คลาส DialogReviewComponent เป็นคลาสที่ใช้ในการ แสดงหน้าเขียนรีวิวและให้คะแนนสถานที่
DetailPlaceComponent	คลาส DetailPlaceComponent เป็น คลาส ที่มีหน้าที่ใน การ แสดง ข้อมูล รายละเอียด ของ สถานที่ ที่ ได้รับ จาก หน้า ListPlaceComponent ที่ จะ ส่ง ข้อมูล เมื่อ ผู้ใช้ กด ที่ รูป- ภาพ ของ สถานที่ ซึ่ง ใน คลาส นี้ สามารถ เรียก ใช้ คลาส DialogReviewComponent เมื่อผู้ใช้กดปุ่มรีวิว
EditPlaceComponent	คลาส EditPlaceComponent เป็นคลาสที่ใช้ในการแสดง หน้าแก้ไขข้อมูลสถานที่ โดยจะดึงข้อมูลจากฐานข้อมูล มาแสดงในแบบฟอร์มที่กำหนดไว้ ในคลาสนี้จะถูกทำงานก็ ต่อเมื่อผู้ใช้กดปุ่มแก้ไข

ตารางที่ 3.20: อธิบาย Class Diagram ผู้ดูแลระบบจากภาพที่ 3.21 ได้ดังนี้

Class Diagram	คำอธิบาย
AdminloginComponent	คลาส AdminloginComponent เป็นคลาสที่ใช้แสดงหน้าเข้าสู่ระบบของผู้ดูแลระบบ โดยผู้ใช้จะต้องเข้าสู่ระบบก่อนถึงจะสามารถเข้าไปจัดการข้อมูลสถานที่ได้ เมื่อผู้ดูแลระบบกดปุ่มเข้าสู่ระบบ จะไปเรียกใช้คลาส PageEditComponent
PageEditComponent	คลาส PageEditComponent เป็นคลาสที่ใช้ในการแสดงหน้าข้อมูลการแก้ไขข้อมูลสถานที่จากผู้ใช้งาน ข้อมูลจะถูกแก้ไขก็ต่อเมื่อผู้ดูแลระบบกดแก้ไข และจะไปเรียกใช้คลาส DialogeditComponent
TopbarComponent	คลาส TopbarComponent เป็น คลาส ที่ จัดการ เกี่ยวกับการ แสดง แถบ เมนู ส่วน บน ของ เว็บไซต์ ของ ผู้ดูแลระบบซึ่งจะสามารถเรียกใช้คลาส PageEditComponent Adminroomcomponent AdmineatComponent และ คลาส AdmintourComponent
Adminroomcomponent	คลาส Adminroomcomponent เป็น คลาส ที่ ใช้ ใน การแสดง หน้า จัดการ ข้อมูล สถานที่ ที่ พัก ใน หน้า นี้ จะ แสดง ข้อมูล ราย ชื่อ สถานที่ ข้อมูล ที่ อยู่ ประเภท ของ สถานที่ เมื่อ ผู้ดูแล ระบบ กด ปุ่ม เพิ่ม สถานที่ จะ ไป เรียกใช้ คลาส DialogaddComponent เมื่อ ผู้ดูแล ระบบ กด ปุ่ม แก้ไข จะ ไป เรียก ใช้ คลาส DialogeditComponent และ เมื่อ ผู้ดูแล ระบบ กด ปุ่ม ลบ จะ ไป เรียก ใช้ คลาส DialogdeleteComponent
AdmineatComponent	คลาส AdmineatComponent เป็น คลาส ที่ ใช้ ใน การแสดง หน้า จัดการ ข้อมูล สถานที่ ที่ กิน ใน หน้า นี้ จะ แสดง ข้อมูล ราย ชื่อ สถานที่ ข้อมูล ที่ อยู่ ประเภท ของ สถานที่ เมื่อ ผู้ดูแล ระบบ กด ปุ่ม เพิ่ม สถานที่ จะ ไป เรียกใช้ คลาส DialogaddComponent เมื่อ ผู้ดูแล ระบบ กด ปุ่ม แก้ไข จะ ไป เรียก ใช้ คลาส DialogeditComponent และ เมื่อ ผู้ดูแล ระบบ กด ปุ่ม ลบ จะ ไป เรียก ใช้ คลาส DialogdeleteComponent

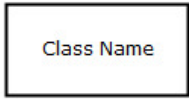



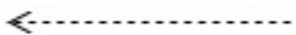
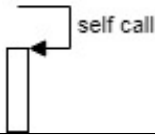

ตารางที่ 3.21: อธิบาย Class Diagram ผู้ดูแลระบบจากภาพที่ 3.21 ได้ดังนี้ (ต่อ)

Class Diagram	คำอธิบาย
AdmintourComponent	คลาส AdmintourComponent เป็น คลาส ที่ ใช้ ใน การ แสดง หน้า จัดการ ข้อมูล สถานที่ ที่ เกี่ยว ใน หน้า นี้ จะ แสดง ข้อมูล ราย ชื่อ สถานที่ ข้อมูล ที่ อยู่ ประเภท ของ สถานที่ เมื่อ ผู้ดูแล ระบบ กด ปุ่ม เพิ่ม สถานที่ จะ ไป เรียก ใช้ คลาส DialogaddComponent เมื่อ ผู้ดูแล ระบบ กด ปุ่ม แก้ไข จะ ไป เรียก ใช้ คลาส DialogeditComponent และ เมื่อ ผู้ดูแล ระบบ กด ปุ่ม ลบ จะ ไป เรียก ใช้ คลาส DialogdeleteComponent
DialogdeleteComponent	คลาส DialogdeleteComponent เป็น คลาส ที่ ใช้ ใน การ แสดง การ ยืนยัน ลบ ข้อมูล สถานที่ ที่ ออกจากระบบ คลาส นี้ จะ ทำงาน ก็ ต่อ เมื่อ ผู้ดูแล ระบบ กด ปุ่ม ลบ
DialogeditComponent	คลาส DialogeditComponent เป็น คลาส ที่ ใช้ ใน การ แสดง หน้า แก้ไข ข้อมูล สถานที่ โดย จะ ดึง ข้อมูล จาก ฐาน ข้อมูล มา แสดง ใน แบบ ฟอม ที่ กำหนด ไว้ คลาส นี้ จะ ทำงาน ก็ ต่อ เมื่อ ผู้ดูแล ระบบ กด ปุ่ม แก้ไข
DialogaddComponent	คลาส DialogaddComponent เป็น คลาส ที่ ใช้ ใน การ แสดง หน้า เพิ่ม ข้อมูล สถานที่ คลาส นี้ จะ ทำงาน ก็ ต่อ เมื่อ ผู้ดูแล ระบบ กด ปุ่ม เพิ่ม สถานที่

3.6 Sequence Diagram

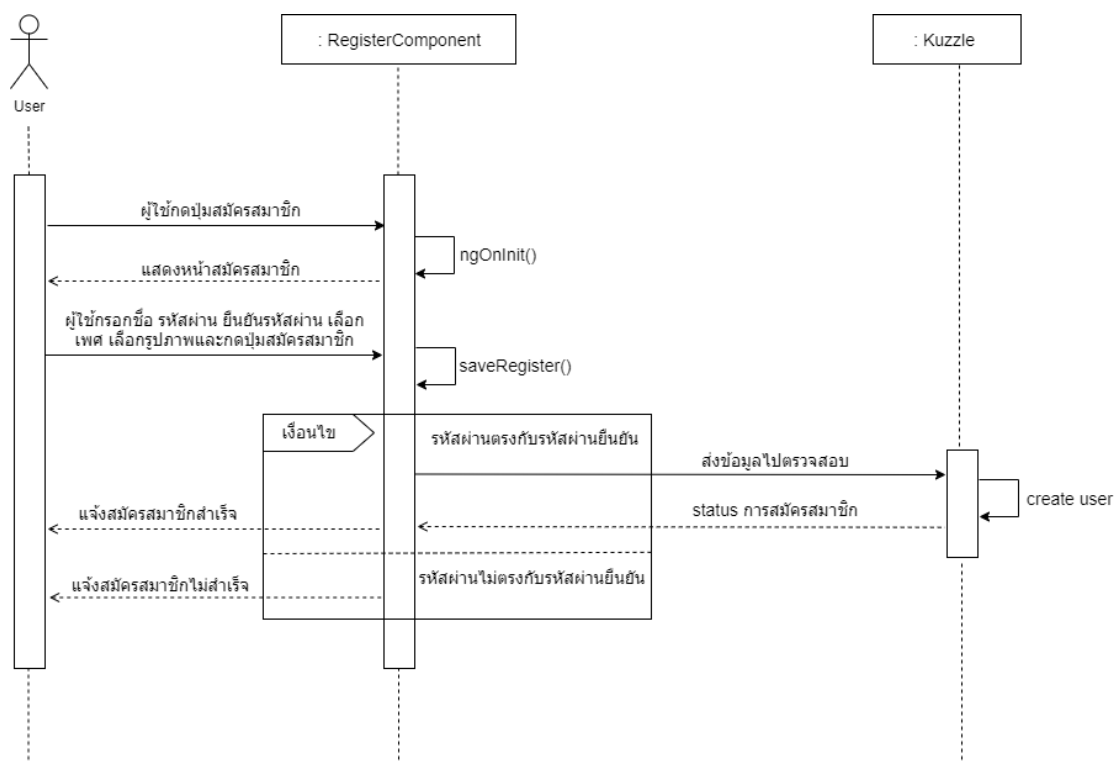
Sequence Diagram เป็น Diagram ที่แสดงขั้นตอนการทำงานของแต่ละ Use Case ระหว่าง Object ต่างๆ ที่ส่งข้อความถึงกันและกัน โดย Sequence Diagram จะช่วยให้มองเห็นการทำงานของภาพรวมของระบบ ส่วนประกอบสัญลักษณ์ที่ใช้ในการเขียน Sequence Diagram แสดงดังตารางที่ 3.22

ตารางที่ 3.22: สัญลักษณ์ของ Sequence Diagram

สัญลักษณ์	การใช้งาน
	Class แสดงถึงการทำงานของ Use Case ในการส่งหรือรับข้อความ แทนด้วยสัญลักษณ์สี่เหลี่ยมมีชื่อคลาสอยู่ภายใน
	Lifeline หรือ เส้น आयुขัย แสดง ช่วง เวลา ตั้งแต่ เริ่ม สร้าง object ในคลาสนั้น จนกระทั่ง object นั้นถูกทำลาย สัญลักษณ์แทนด้วยเส้นประ
	Focus of control หรือจุดควบคุม เป็นจุดควบคุมที่ object ใช้ทำการส่งหรือรับข้อความ สัญลักษณ์แทนด้วยสี่เหลี่ยม
	Message คือ ข้อความที่รับส่งระหว่าง Object สัญลักษณ์แทนด้วยลูกศรและประกอบด้วย 2 ส่วน คือ ข้อมูล (Data) และฟังก์ชัน (Function)
	Return Message เป็นข้อมูลที่ส่งกลับหลังจากทำงานเสร็จ
	Self call เป็นการเรียกฟังก์ชันการทำงานภายในตัวเอง
	สร้างกรอบการทำงานของโปรแกรม เพื่อให้รู้ขอบเขตของการทำงานเช่น ลูป(loop)

Sequence Diagram ที่ใช้อธิบายการทำงานของระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี มีรายละเอียดดังต่อไปนี้

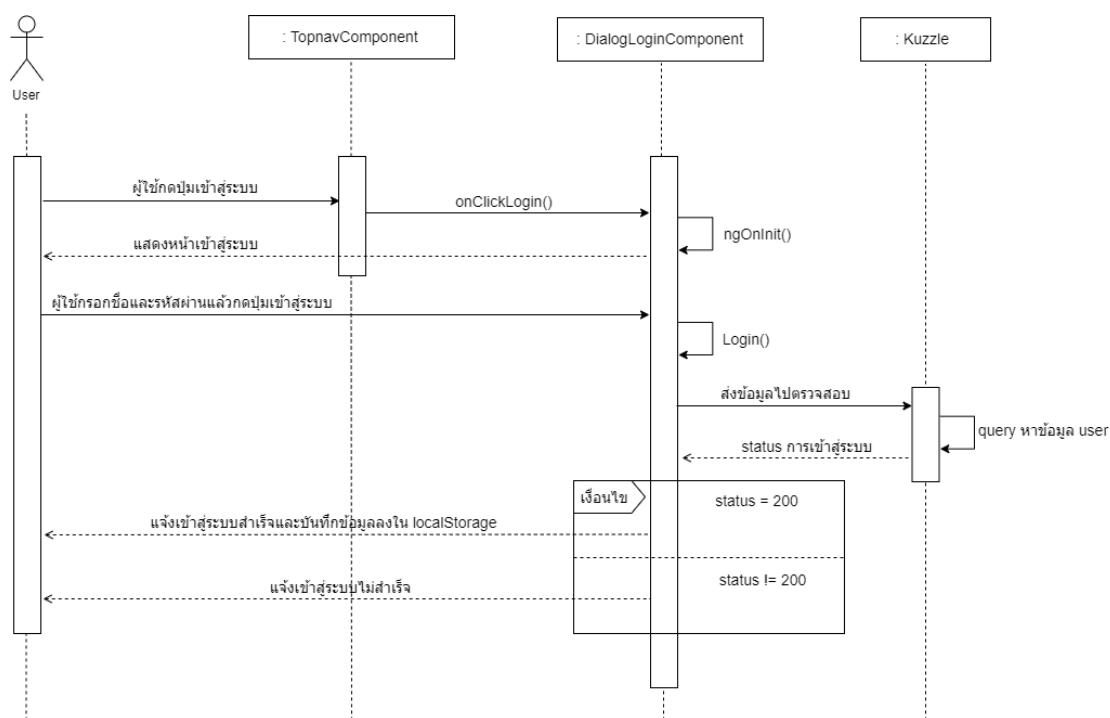
3.6.1 Sequence Diagram ของการสมัครสมาชิก



รูปที่ 3.22: Sequence Diagram ของการสมัครสมาชิก

จากรูปที่ 3.22 สามารถอธิบายแผนภาพ Sequence Diagram ของการสมัครสมาชิก ได้ดังต่อไปนี้ เมื่อผู้ใช้กดสมัครสมาชิกระบบจะไปเรียกใช้ฟังก์ชัน `ngOnInit()` หรือคลาส `RegisterComponent` โดยจะแสดงหน้าสมัครสมาชิกเมื่อผู้ใช้กรอกชื่อ รหัสผ่าน ยืนยันรหัสผ่าน เลือกเพศ เลือกรูปโปรไฟล์เสร็จและกดปุ่มสมัครสมาชิกระบบจะไปเรียกฟังก์ชัน `saveRegister()` ซึ่งจะทำให้การเช็คเงื่อนไขว่ารหัสผ่านตรงกับรหัสยืนยันหรือไม่ ถ้าตรงจะส่งข้อมูล user ไปตรวจสอบในส่วน security ในฐานข้อมูล Kuzzle และจะทำการสร้าง user เสร็จจะส่ง status การสมัครสมาชิกไปกลับไปให้คลาส `registerComponent` และจะแสดงข้อความแจ้งผู้ใช้ว่าสมัครสมาชิกสำเร็จ หรือในกรณีที่รหัสผ่านและรหัสยืนยันไม่ตรงกันระบบจะแจ้งผู้ใช้ว่า รหัสผ่านไม่ตรงกัน

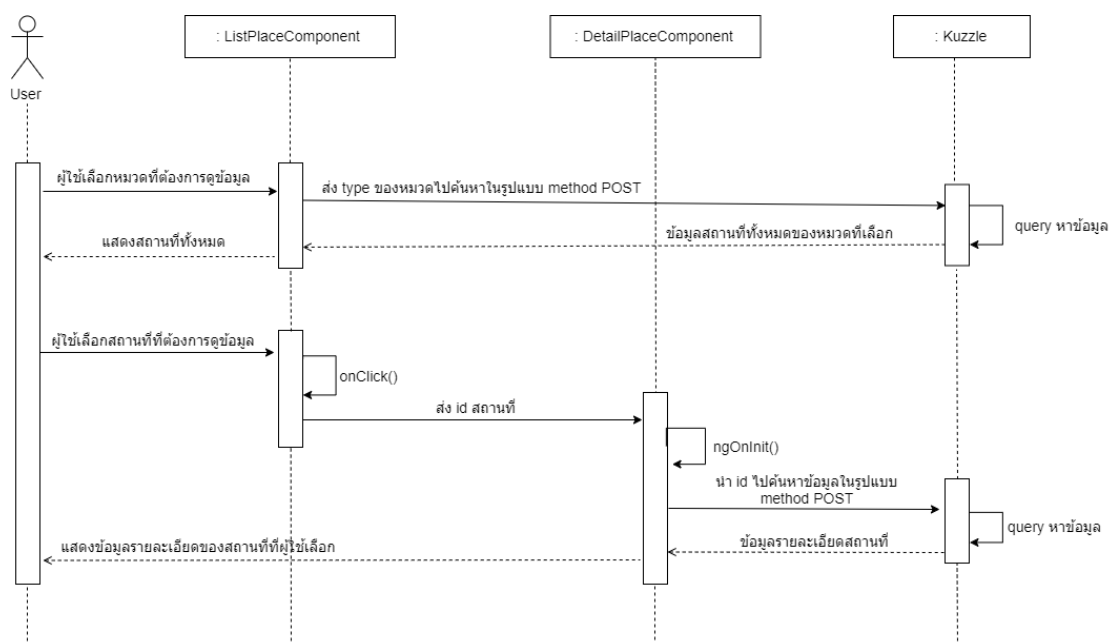
3.6.2 Sequence Diagram ของการเข้าสู่ระบบ



รูปที่ 3.23: Sequence Diagram ของการเข้าสู่ระบบ

จากรูปที่ 3.23 สามารถอธิบายแผนภาพ Sequence Diagram ของการเข้าสู่ระบบ ได้ดังต่อไปนี้ เมื่อผู้ใช้งานกดปุ่มเข้าสู่ระบบ จากนั้นระบบจะเรียกใช้ฟังก์ชัน `onClickLogin()` โดยฟังก์ชัน `onClickLogin()` จะไปเรียกใช้ฟังก์ชัน `ngOnInit()` ของคลาส `DialogLoginComponent` มาแสดงหน้าเข้าสู่ระบบต่อมาผู้ใช้กรอก username และ password เสร็จกดเข้าสู่ระบบทำให้ระบบไปเรียกใช้ฟังก์ชัน `Login()` โดยจะส่งข้อมูล username และ password ไปตรวจสอบในส่วน security และ Kuzzle ทำการค้นหาและส่ง status การล็อกอินกลับไปให้คลาส `DialogLoginComponent` ซึ่งในคลาสนี้จะทำการเช็คสถานะการเข้าสู่ระบบและจะแสดงข้อความแจ้งผู้ใช้ให้ทราบว่าเข้าสู่ระบบสำเร็จ หรือ ไม่สำเร็จ

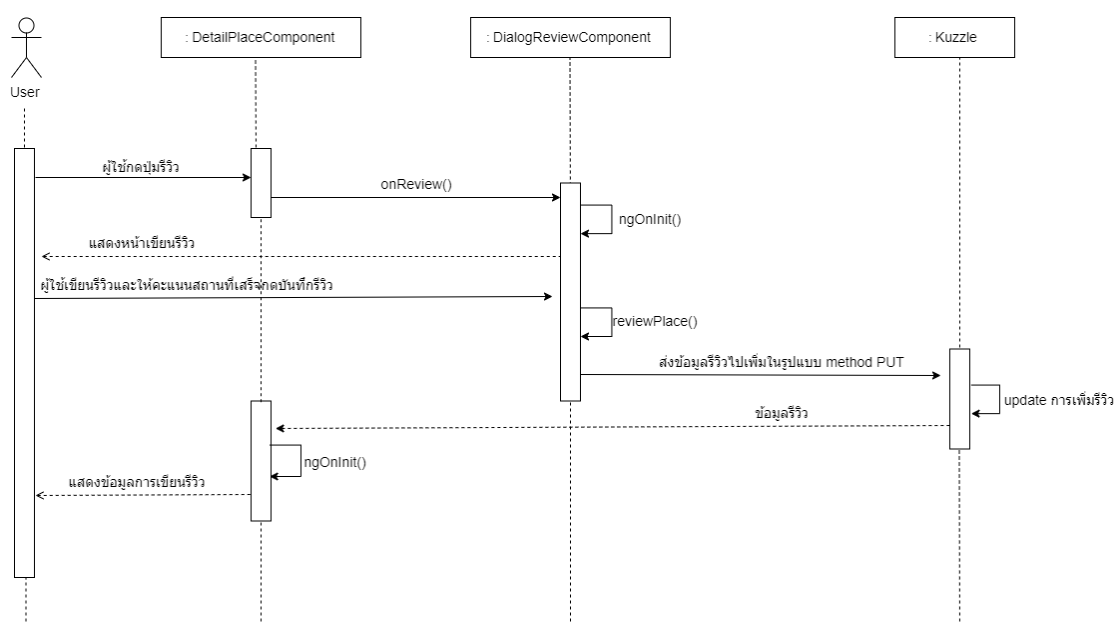
3.6.3 Sequence Diagram ของการดูข้อมูลรายละเอียดสถานที่



รูปที่ 3.24: Sequence Diagram ของการดูข้อมูลรายละเอียดสถานที่

จากรูปที่ 3.24 สามารถอธิบายแผนภาพ Sequence Diagram ของการดูข้อมูลรายละเอียดสถานที่ ได้ดังต่อไปนี้ เมื่อผู้ใช้กดเลือกหมวดสถานที่ที่ต้องการดูข้อมูล ระบบจะส่ง type ของหมวดสถานที่และส่ง id มาให้คลาส ListPlaceComponent และจะส่งข้อมูลที่ได้รับมาในตัวแปร typeGroup เพื่อไปค้นหาข้อมูลสถานที่ตามหมวดที่ผู้ใช้เลือกในฐานข้อมูล Kuzzle เมื่อได้ข้อมูลมาแล้วระบบจะส่งข้อมูลสถานที่ทั้งหมดของหมวดที่เลือกไปให้คลาส ListPlaceComponent ในคลาส ListPlaceComponent จะแสดงข้อมูลสถานที่ให้ผู้ใช้ได้ดู เมื่อผู้ใช้กดเลือกสถานที่ที่ต้องการดูข้อมูลระบบจะเรียกใช้ฟังก์ชัน onClick() ที่ ListPlaceComponent หลังทำงานในฟังก์ชัน onClick() สำเร็จระบบจะส่งข้อมูล id สถานที่ไปให้ DetailPlaceComponent และจะไปเรียกใช้ฟังก์ชัน ngOnInit() เพื่อที่จะทำการนำ id ที่ได้มาไป query หาข้อมูลของสถานที่ที่มีชื่อ id ตรงกันใน Kuzzle แล้วส่งข้อมูลรายละเอียดสถานที่มาเก็บไว้ที่ DetailPlaceComponent และแสดงหน้าข้อมูลรายละเอียดสถานที่ให้กับผู้ใช้

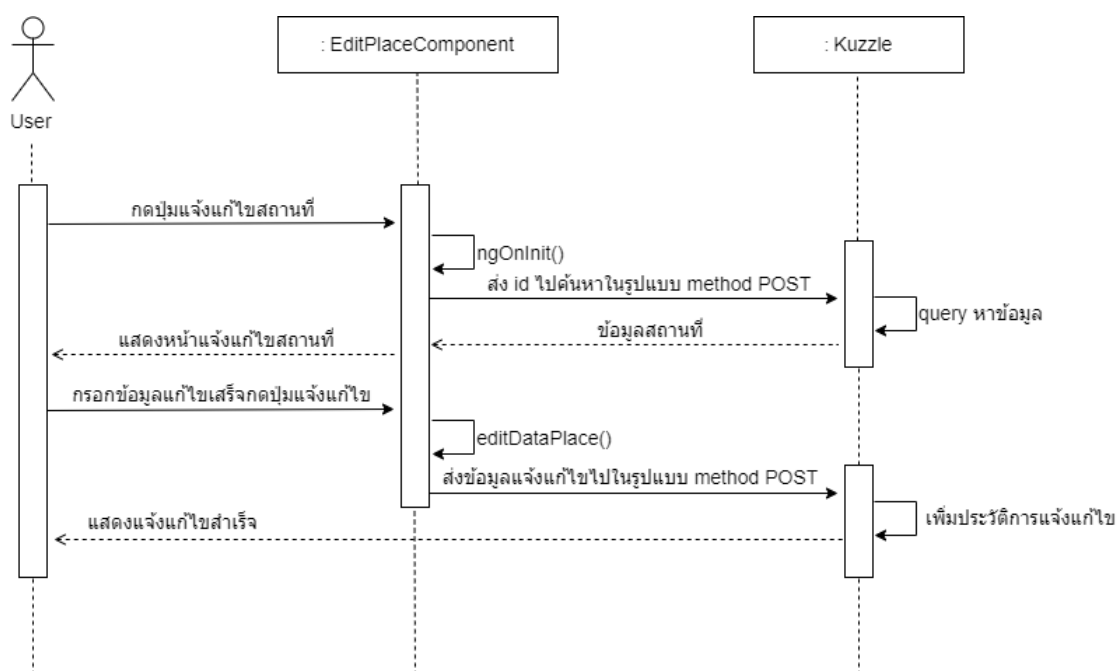
3.6.4 Sequence Diagram ของการเขียนรีวิวให้คะแนนสถานที่



รูปที่ 3.25: Sequence Diagram ของการเขียนรีวิวให้คะแนนสถานที่

จากรูปที่ 3.25 สามารถอธิบายแผนภาพ Sequence Diagram ของการเขียนรีวิวให้คะแนนสถานที่ ได้ดังต่อไปนี้ เมื่อผู้ใช้งานเข้าสู่หน้าแสดงรายละเอียดสถานที่ ผู้ใช้กดปุ่มรีวิวจากนั้นระบบจะเรียกใช้ฟังก์ชัน `onReview()` ของคลาส `DetailPlaceComponent` โดยส่ง id สถานที่และ type ของสถานที่ไปเรียกใช้ฟังก์ชัน `ngOnInit()` ของคลาส `DialogReviewComponent` มาแสดงหน้าเขียนรีวิว ต่อมาผู้ใช้เขียนรีวิวและให้คะแนนสถานที่เสร็จจากบันทึกรีวิว ซึ่งทำให้ระบบเรียกใช้ฟังก์ชัน `reviewPlace()` และเก็บข้อมูลการเขียนรีวิวไว้ในตัวแปร `dataReview` และทำการส่งข้อมูลการเขียนรีวิวเข้าไปบันทึกลงในฐานข้อมูล `Kuzzle` หลังจากทำการอัปเดตเพิ่มรีวิวเสร็จระบบจะส่งข้อมูลรีวิวไปให้คลาส `DetailPlaceComponent` และเรียกใช้ฟังก์ชัน `ngOnInit()` เพื่อทำการแสดงข้อมูลการเขียนรีวิวให้กับผู้ใช้ได้เห็น

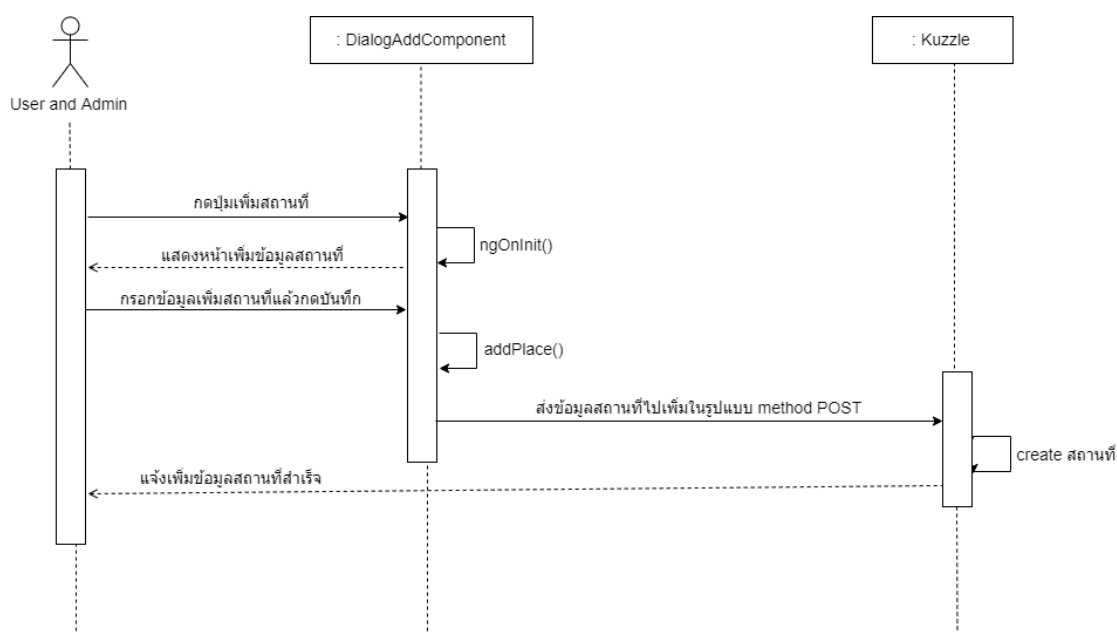
3.6.5 Sequence Diagram ของการแก้ไขข้อมูลสถานที่



รูปที่ 3.26: Sequence Diagram ของการแก้ไขข้อมูลสถานที่

จากรูปที่ 3.26 สามารถอธิบายแผนภาพ Sequence Diagram ของการแก้ไขข้อมูลสถานที่ได้ดังต่อไปนี้ เมื่อผู้ใช้กดปุ่มแก้ไขจากนั้นระบบจะเรียกใช้ฟังก์ชัน ngOnInit() ของคลาส EditPlaceComponent โดยส่ง id สถานที่ไปค้นหาในฐานข้อมูล Kuzzle ทำการ query หาข้อมูลและส่งข้อมูลสถานที่มาเก็บไว้ที่คลาส PageEditComponent และจะแสดงหน้าแก้ไขข้อมูลสถานที่ เมื่อผู้ใช้กรอกข้อมูลแก้ไขเสร็จกดปุ่มแก้ไขไปเรียกใช้ฟังก์ชัน editDataPlace() หลังจากนั้นจะส่งข้อมูลการแก้ไขไปเพิ่มลงในฐานข้อมูล Kuzzle ฐานข้อมูลทำการเพิ่มประวัติการแก้ไขและแสดงข้อความแก้ไขข้อมูลสำเร็จ

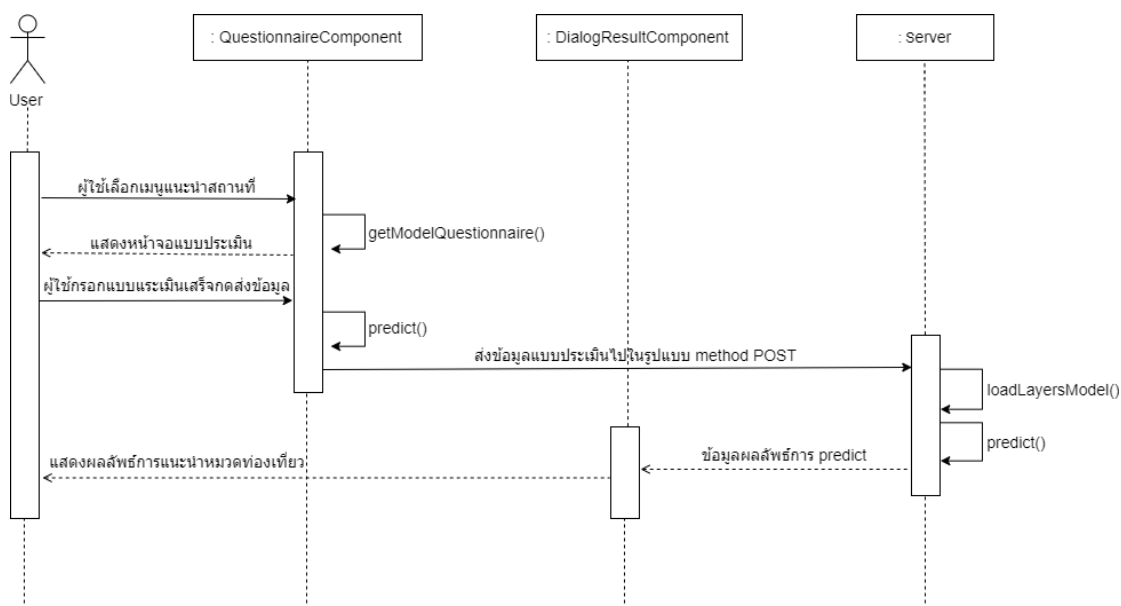
3.6.6 Sequence Diagram ของการเพิ่มข้อมูลสถานที่



รูปที่ 3.27: Sequence Diagram ของการเพิ่มข้อมูลสถานที่

จากรูปที่ 3.27 สามารถอธิบายแผนภาพ Sequence Diagram ของการเพิ่มข้อมูลสถานที่ ได้ดังต่อไปนี้ เมื่อผู้ใช้งานและผู้ดูแลระบบกดปุ่มเพิ่มสถานที่จากนั้นระบบจะเรียกใช้ฟังก์ชัน `ngOnInit()` ของคลาส `DialogAddComponent` โดยทำการแสดงหน้าเพิ่มสถานที่ให้กับผู้ใช้และผู้ดูแลระบบได้กรอกข้อมูลเมื่อกรอกเสร็จกดบันทึกที่ระบบจะไปเรียกใช้ฟังก์ชัน `addPlace()` ของคลาส `DialogAddComponent` ทำการเก็บข้อมูลทั้งหมดที่ผู้ใช้และผู้ดูแลระบบกรอกเข้ามาไว้ในรูปแบบของ object ในตัวแปร `data` และทำการส่งข้อมูลการเพิ่มสถานที่ไปเพิ่มลงในฐานข้อมูล `Kuzzle` หลังจากเพิ่มข้อมูลลงฐานข้อมูล `Kuzzle` สำเร็จระบบจะปิดหน้าเพิ่มสถานที่และแสดงแจ้งเพิ่มข้อมูลสถานที่สำเร็จ

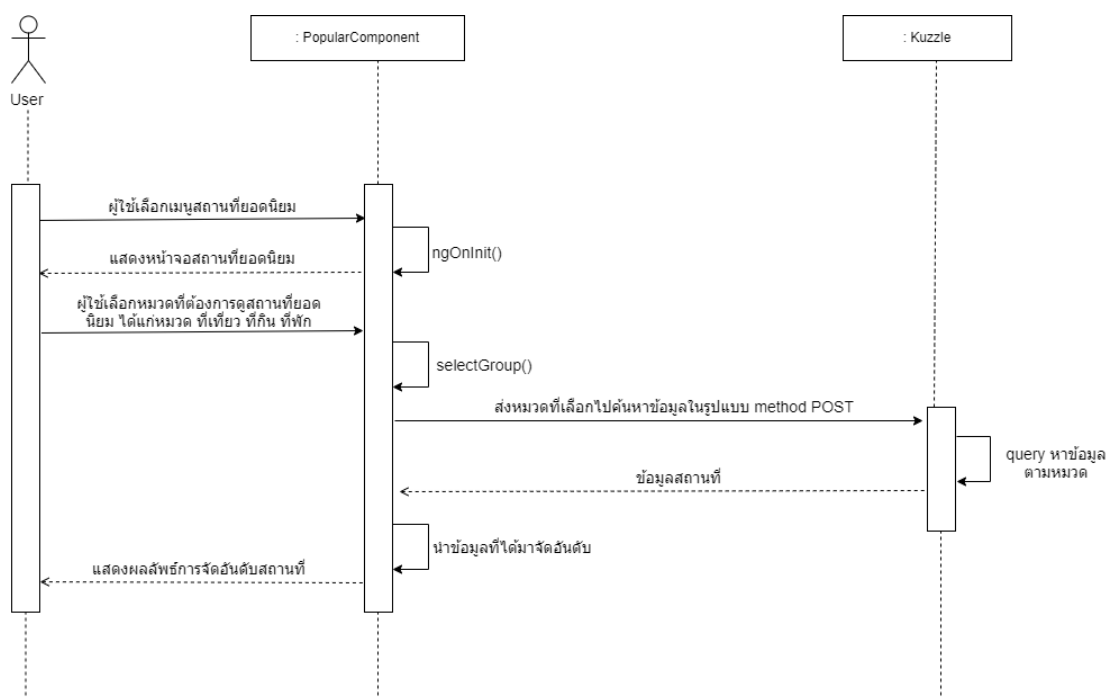
3.6.7 Sequence Diagram ของการกรอกแบบประเมิน



รูปที่ 3.28: Sequence Diagram ของการกรอกแบบประเมิน

จากรูปที่ 3.28 สามารถอธิบายแผนภาพ Sequence Diagram ของการกรอกแบบประเมินได้ดังต่อไปนี้ เมื่อผู้ใช้งานเข้าสู่หน้าแนะนำสถานที่หรือคลาส QuestionnaireComponent จากนั้นระบบจะเรียกใช้ getModelQuestionnaire() เพื่อแสดงหน้าจอแบบประเมินเมื่อผู้ใช้งานกรอกข้อมูลแบบประเมินเสร็จกดส่งข้อมูล จะไปเรียกใช้ฟังก์ชัน predict() โดยทำการส่งข้อมูลการกรอกแบบประเมินของผู้ใช้ในรูปแบบอาเรย์ในตัวแปร resultModel ไปให้เซิร์ฟเวอร์และเซิร์ฟเวอร์ทำการโหลดโมเดลนำโมเดลที่ได้ทำนายผลลัพธ์มาเก็บไว้ที่คลาส DialogResultComponent และคลาส DialogResultComponent จะแสดงผลลัพธ์การแนะนำหมวดท่องเที่ยวให้กับผู้ใช้ได้ดู

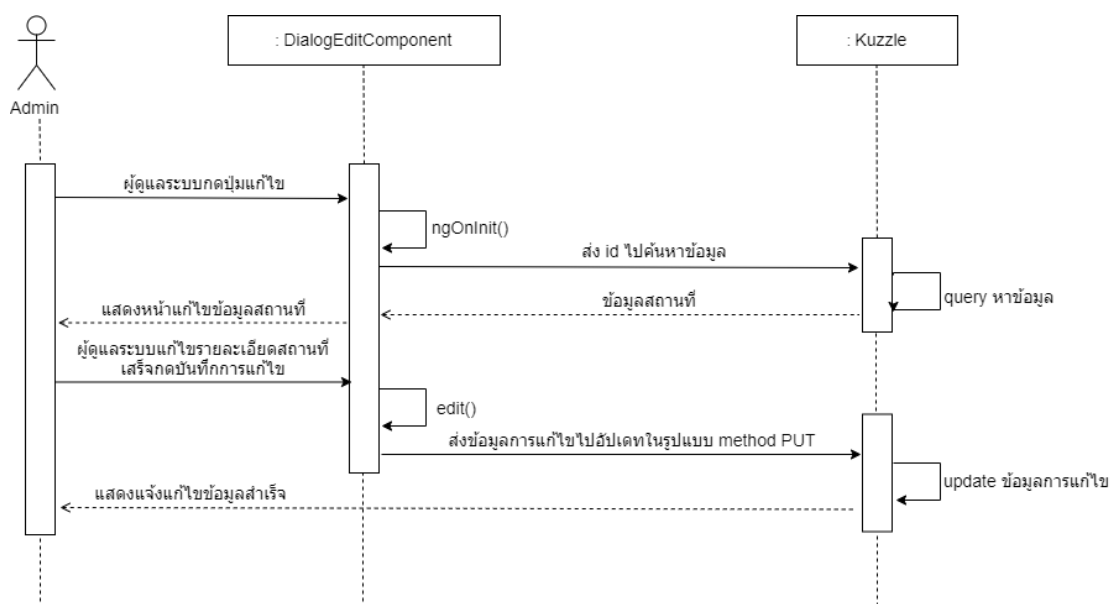
3.6.8 Sequence Diagram ของการดูผลลัพธ์การจัดอันดับ



รูปที่ 3.29: Sequence Diagram ของการดูผลลัพธ์การจัดอันดับ

จากรูปที่ 3.29 สามารถอธิบายแผนภาพ Sequence Diagram ของการดูผลลัพธ์การจัดอันดับ ได้ดังต่อไปนี้ เมื่อผู้ใช้งานเข้าสู่หน้าสถานที่ยอดนิยมหรือคลาส PopularComponent จากนั้นระบบจะเรียกฟังก์ชัน `ngOnInit()` ของคลาส PopularComponent โดยทำการนำข้อมูลจากฐานข้อมูลมาแสดงในหน้าสถานที่ยอดนิยม ต่อมาผู้ใช้งานเลือกหมวดที่ต้องการดูสถานที่ยอดนิยม ได้แก่หมวดสถานที่ท่องเที่ยว ที่พักและหมวดร้านอาหาร ซึ่งทำให้ระบบเรียกใช้งานฟังก์ชัน `selectGroup` และนำข้อมูลหมวดที่เลือกไปเก็บไว้ในตัวแปร `select` และส่งไปค้นหาข้อมูลจากฐานข้อมูล Kuzzle หลังจากได้ข้อมูลจะส่งข้อมูลสถานที่ทั้งหมดของหมวดที่ผู้ใช้เลือกมาเก็บไว้ในคลาส PopularComponent จะนำข้อมูลที่ได้มาจัดอันดับและแสดงผลการจัดอันดับ

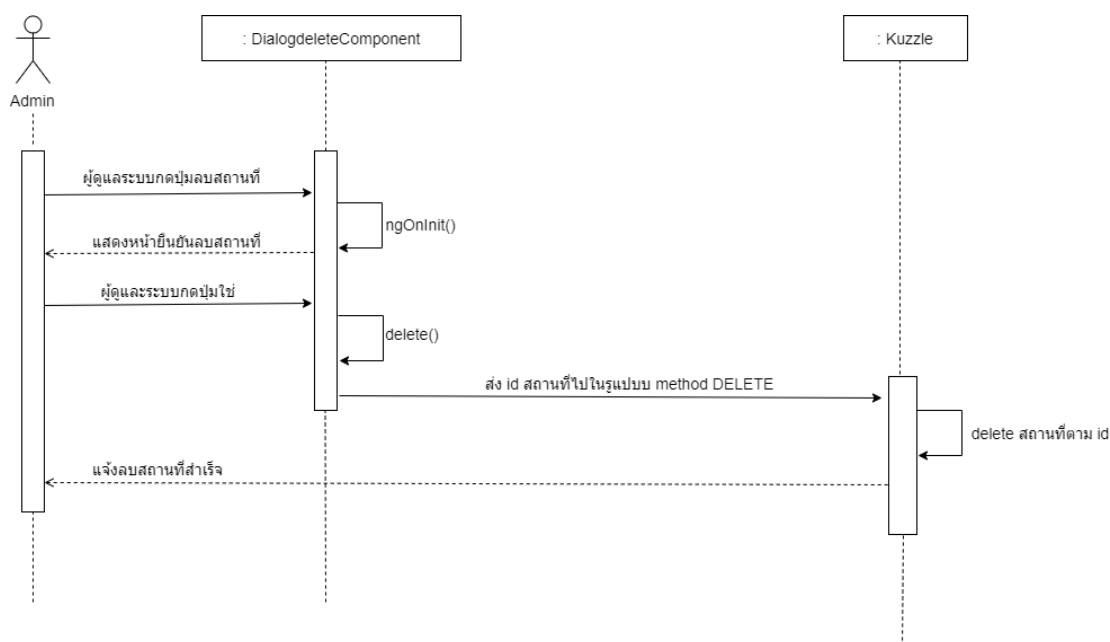
3.6.9 Sequence Diagram ของการจัดการแก้ไขข้อมูลสถานที่



รูปที่ 3.30: Sequence Diagram ของการจัดการแก้ไขข้อมูลสถานที่

จากรูปที่ 3.30 สามารถอธิบายแผนภาพ Sequence Diagram ของการจัดการแก้ไขข้อมูลสถานที่ ได้ดังต่อไปนี้ เมื่อผู้ดูแลระบบกดปุ่มแก้ไข จากนั้นระบบเรียกใช้ฟังก์ชัน ngOnInit() ของคลาส DialogeditComponent และส่ง id ไปค้นหาข้อมูลในฐานข้อมูล Kuzzle ทำการ query หาข้อมูลส่งข้อมูลสถานที่มาเก็บไว้ที่คลาส DialogeditComponent เพื่อแสดงหน้าแก้ไขข้อมูลสถานที่ ต่อมาผู้ดูแลระบบแก้ไขรายละเอียดสถานที่เสร็จกดบันทึกการแก้ไขระบบจะไปเรียกใช้งานฟังก์ชัน edit() ของคลาส DialogeditComponent โดยส่งข้อมูลรายละเอียดการแก้ไขไปเก็บไว้ในตัวแปร data หลังจากนั้นนำข้อมูลที่ส่งมาไปให้ฐานข้อมูล Kuzzle เพื่อทำการอัปเดตข้อมูลการแก้ไข เมื่อบันทึกข้อมูลการแก้ไขสำเร็จระบบจะปิดหน้าแก้ไขข้อมูลสถานที่ลง

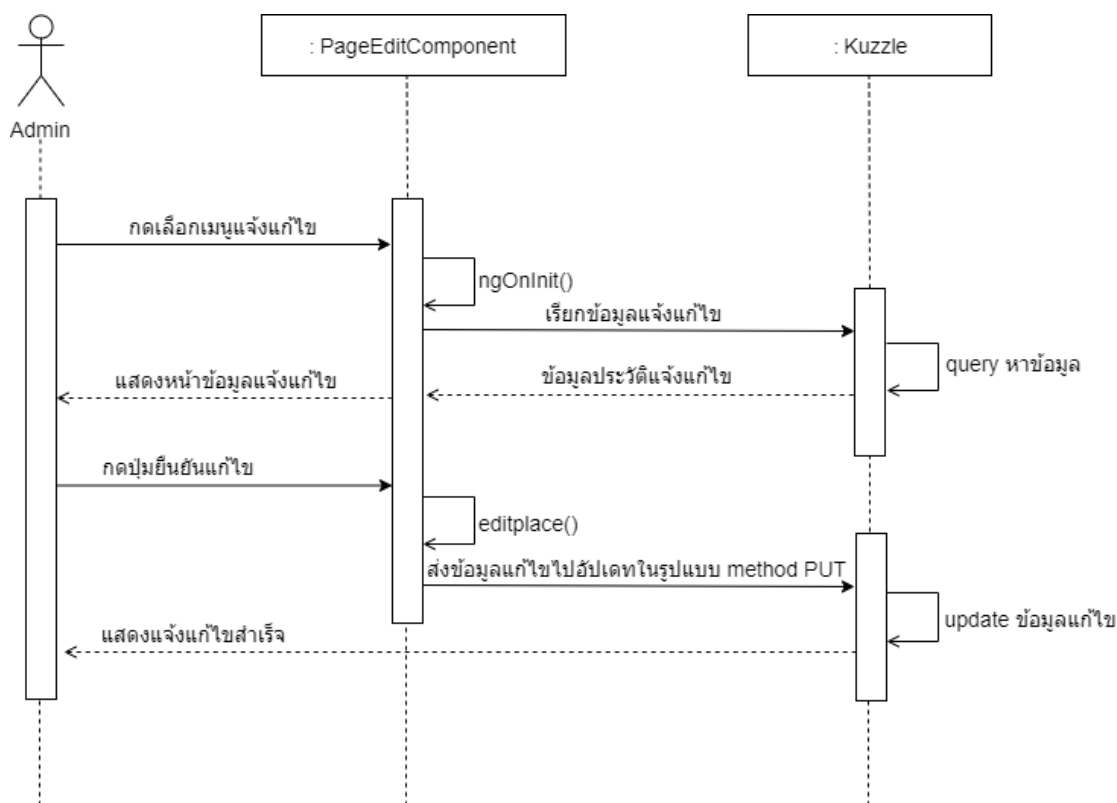
3.6.10 Sequence Diagram ของการจัดการลบข้อมูลสถานที่



รูปที่ 3.31: Sequence Diagram ของการจัดการลบข้อมูลสถานที่

จากรูปที่ 3.31 สามารถอธิบายแผนภาพ Sequence Diagram ของการจัดการลบข้อมูลสถานที่ ได้ดังต่อไปนี้ เมื่อผู้ดูแลระบบกดปุ่มลบ จากนั้นระบบจะเรียกใช้ฟังก์ชัน ngOnInit() ของคลาส DialogdeleteComponent เพื่อแสดงหน้าข้อความถามเพื่อยืนยันการลบข้อมูลสถานที่ โดยจะมีปุ่มให้เลือกอยู่ 2 ปุ่มคือ ใช่ และ ไม่ใช่ ต่อมาผู้ดูแลระบบกดปุ่มใช่ระบบเรียกใช้งานฟังก์ชัน delete() ของคลาส DialogdeleteComponent โดยส่ง id สถานที่ที่ต้องการลบและนำ id สถานที่ไปค้นหาในฐานข้อมูล Kuzzle เพื่อทำการลบข้อมูลสถานที่ออกจากฐานข้อมูล เมื่อลบข้อมูลสถานที่สำเร็จระบบจะปิดหน้ายืนยันการลบข้อมูลสถานที่ลงและแสดงหน้าจัดการข้อมูลสถานที่

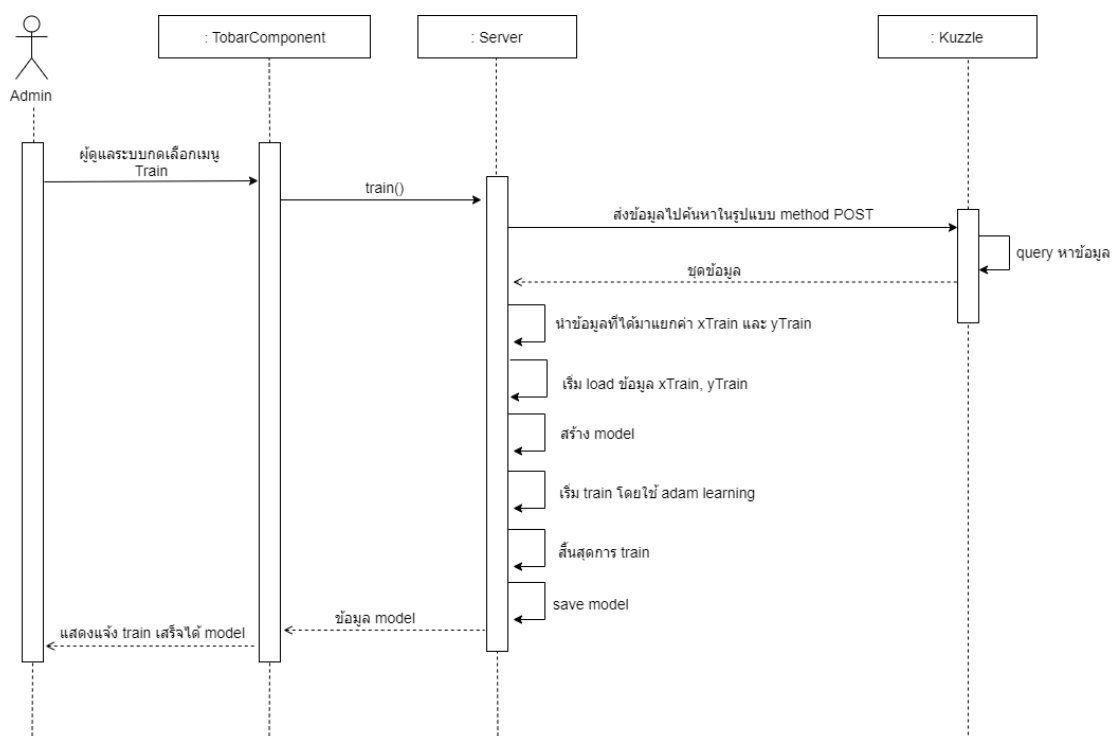
3.6.11 Sequence Diagram ของการจัดการแจ้งแก้ไขข้อมูลสถานที่



รูปที่ 3.32: Sequence Diagram ของการจัดการแจ้งแก้ไขข้อมูลสถานที่

จากรูปที่ 3.32 สามารถอธิบายแผนภาพ Sequence Diagram ของการจัดการแจ้งแก้ไขข้อมูลสถานที่ ได้ดังต่อไปนี้ เมื่อผู้ดูแลระบบกดเมนูแจ้งแก้ไข จากนั้นระบบจะเรียกใช้ฟังก์ชัน ngOnInit() ของคลาส PageEditComponent และจะเรียกใช้ข้อมูลแจ้งแก้ไขในฐานข้อมูล Kuzzle ทำการค้นหาข้อมูลและส่งข้อมูลประวัติแจ้งแก้ไขมาเก็บไว้ที่คลาส PageEditComponent เพื่อแสดงหน้าข้อมูลแจ้งแก้ไข ต่อมาผู้ดูแลระบบกดปุ่มยืนยันการแก้ไขระบบจะไปเรียกใช้งานฟังก์ชัน editplace() ของคลาส PageEditComponent โดยส่งข้อมูลการแก้ไขไปอัปเดตในฐานข้อมูล Kuzzle และฐานข้อมูลทำการอัปเดตข้อมูลและแสดงแจ้งแก้ไขข้อมูลสำเร็จ

3.6.12 Sequence Diagram ของการจัดการ train model









รูปที่ 3.33: Sequence Diagram ของการจัดการ train model

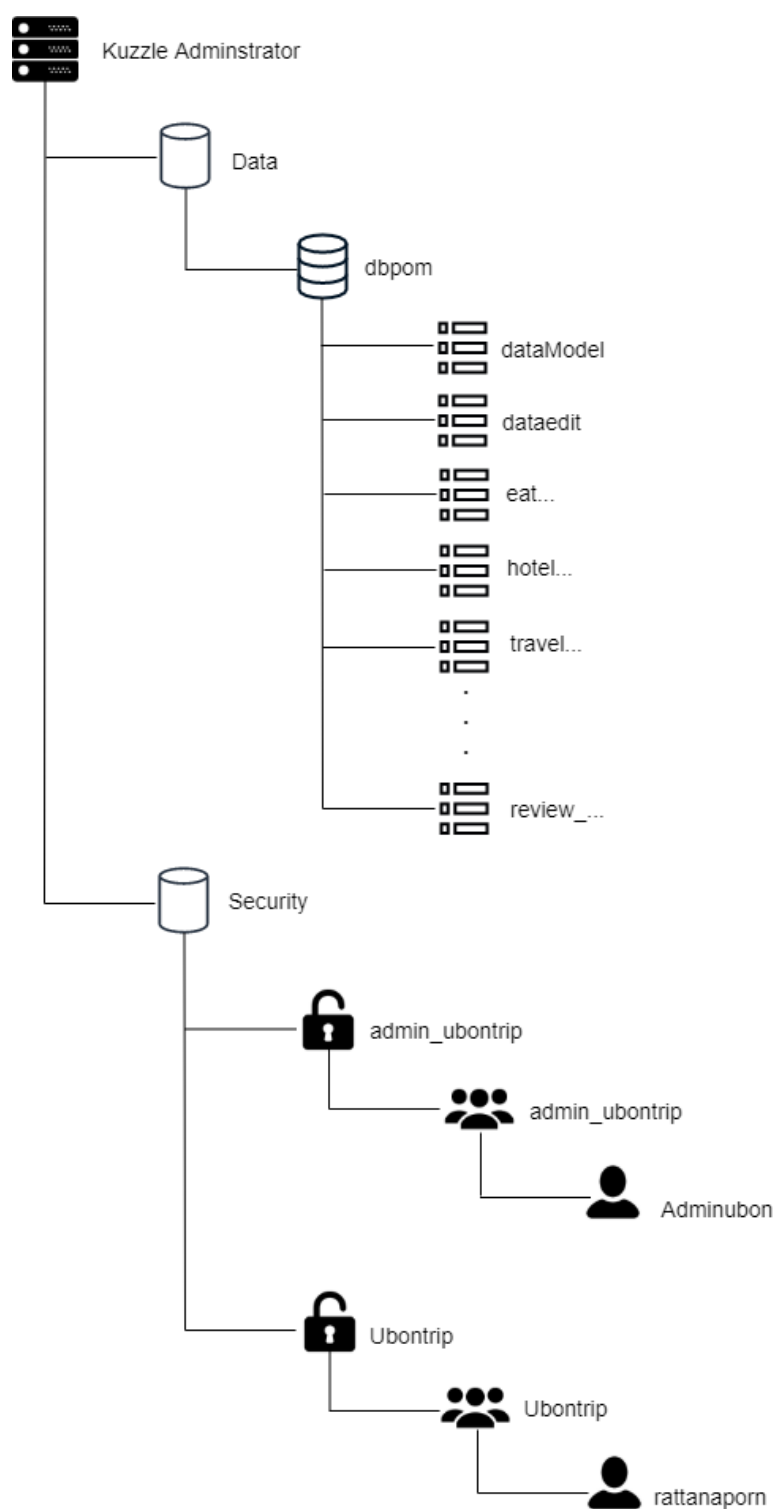
จากรูปที่ 3.33 สามารถอธิบายแผนภาพ Sequence Diagram ของการจัดการ train model ได้ดังต่อไปนี้ เมื่อผู้ดูแลระบบกดเลือกเมนู train ของคลาส TopbarComponent จากนั้นระบบจะเรียกใช้ฟังก์ชัน train() และจะไปเรียกใช้งาน server โดยส่ง url ที่อยู่ของข้อมูลไปค้นหาในฐานข้อมูล Kuzzle เมื่อค้นหาและได้ข้อมูลมาจะส่งชุดข้อมูลไปใช้งานต่อที่ server หลังจากที่ได้ชุดข้อมูลนำชุดข้อมูลที่ได้อีกมาแยกค่า xTrain yTrain ต่อมาก็เริ่มทำการ load ข้อมูล xTrain yTrain เมื่อโหลดข้อมูลสำเร็จจะทำการสร้างโมเดล ด้วยคำสั่ง tf.sequential() หลังจากสร้างโมเดลสำเร็จระบบจะเริ่ม train โดยใช้ adam learning เมื่อ train สำเร็จจะทำการ savemodel เมื่อ savemode เสร็จจะแจ้งให้ผู้ดูแลระบบได้ทราบ

3.7 โครงสร้างฐานข้อมูลคาสเซิล(Kuzzle Database Structure)

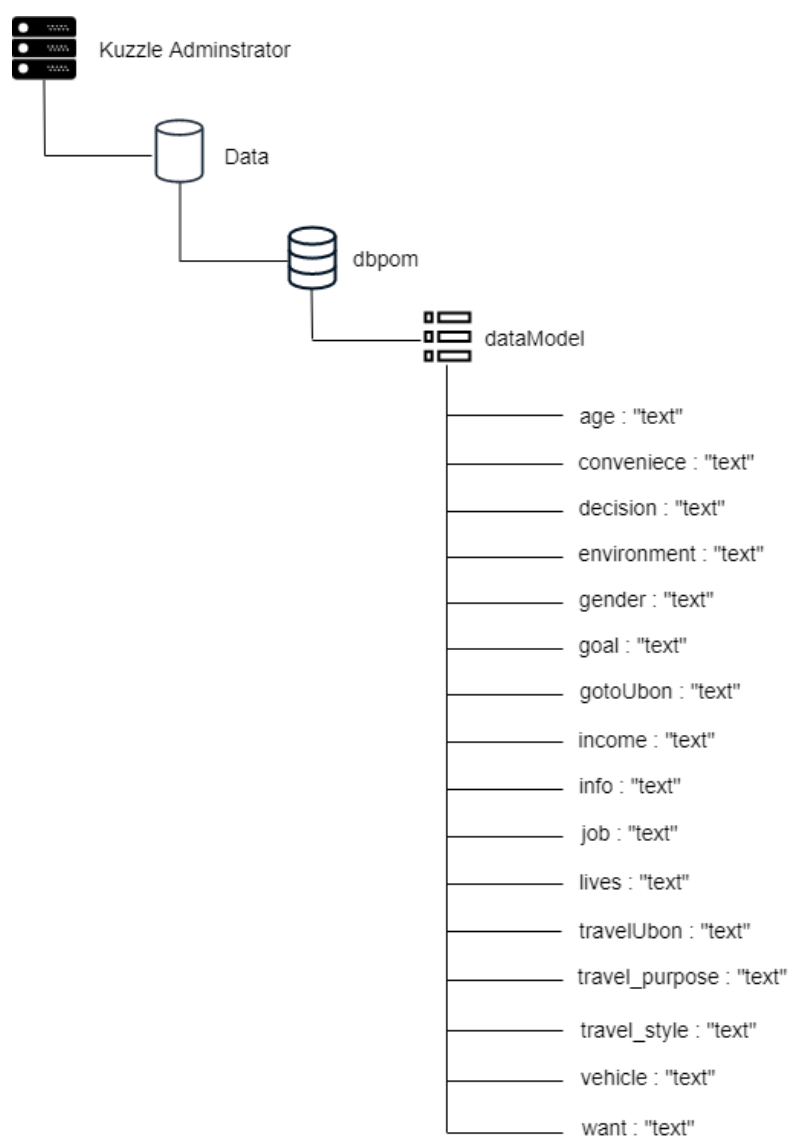
Kuzzle Database นั้นเป็น Database แบบ NoSQL และเป็น JSON database ที่มีโครงสร้างที่เป็น Key และ Value จัดเก็บข้อมูลในลักษณะโหนด หากต้องการเรียกงานจะเรียกใช้โดยการท่องไปยังโหนดที่ต้องการ ส่วนประกอบสัญลักษณ์ที่ใช้ในการเขียนโครงสร้างฐานข้อมูลแบบ Kuzzle แสดงดังรูปภาพที่ 3.34

สัญลักษณ์	คำอธิบาย
	Database เป็นการเรียกชื่อแทนโหนด(Node)บนสุดที่ใช้ในการเก็บข้อมูล
	Collection เป็นการเรียกชื่อแทนของการเก็บหลาย ๆ เอกสารไว้ด้วยกัน
	Document เป็นการเรียกชื่อแทนหน่วยการเก็บของข้อมูลใน Kuzzle ภายในจะประกอบไปด้วย ชื่อของ Document ชื่อของคีย์ (key) และ ค่าข้อมูล (value) โดยชื่อของ Document ห้ามซ้ำกัน ซึ่งใน Kuzzle สามารถระบุประเภทของข้อมูลได้ 12 ประเภทได้แก่ text, date, short, integer, long, float, double, boolean, object, nested, geo_point และ geo_shape
	Roles เป็นตัวกำหนดสิทธิ์ในการเข้าถึงการจัดการข้อมูล
	Profiles เป็นตัวกำหนดประเภทของผู้ใช้ ซึ่งจะมีสิทธิ์การจัดการข้อมูลตาม Roles ที่ได้เลือกไว้
	Users เป็นผู้ใช้งานที่อยู่ภายใต้ Profiles ซึ่งสิทธิ์การเข้าถึงจะอิงตาม Profiles

รูปที่ 3.34: สัญลักษณ์ของโครงสร้างฐานข้อมูลแบบ Kuzzle



รูปที่ 3.35: โครงสร้างฐานข้อมูลภาพรวมใน Kuzzle



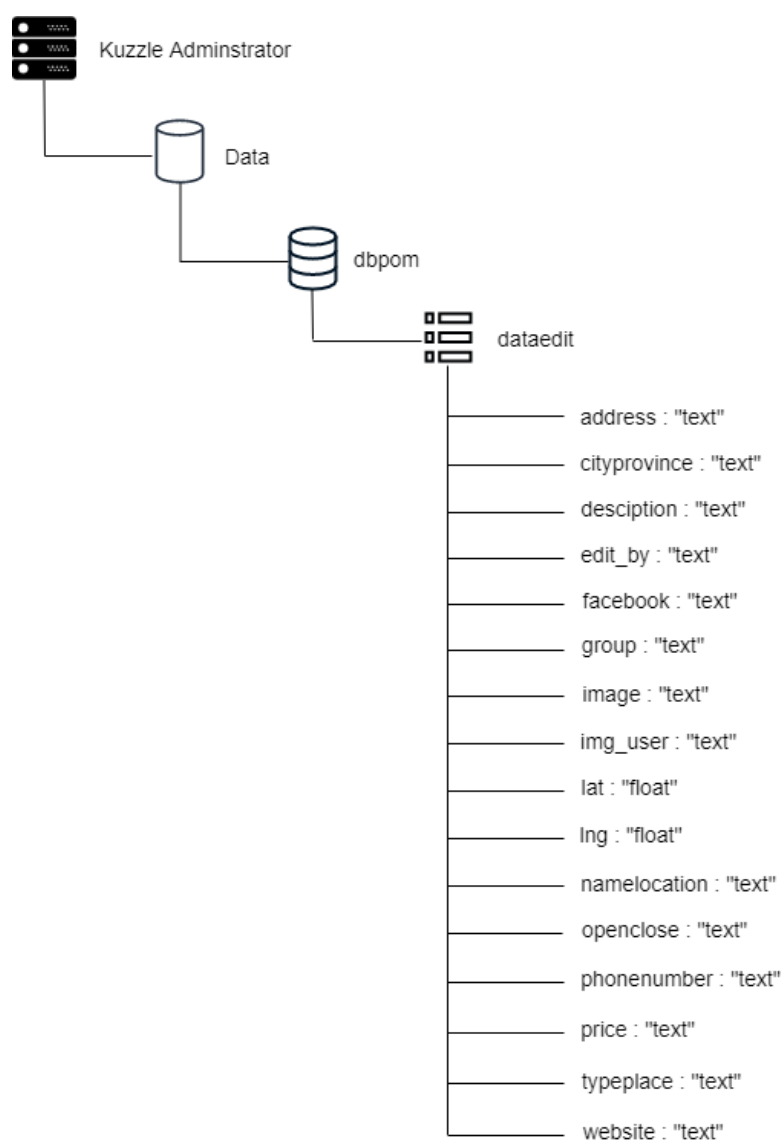
รูปที่ 3.36: โครงสร้างฐานข้อมูล datamodel

ตารางที่ 3.23: อธิบายโครงสร้างฐานข้อมูลของ datamodel จากรูปที่ 3.36

Key	คำอธิบาย
age	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”อายุ”
convenience	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”ความสะดวกในการเข้าถึงแหล่งท่องเที่ยว”
decision	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”สิ่งที่มีผลต่อการตัดสินใจเลือกมาเที่ยวในครั้งนี้”
environment	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”สภาพแวดล้อมของสถานที่ท่องเที่ยวที่ท่านต้องการ”
gender	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”เพศ”
goal	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”จุดประสงค์ที่ท่านเดินทางมาเที่ยวครั้งนี้”
gotoUbon	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”ท่านเคยเดินทางมาเที่ยวที่จังหวัดอุบลราชธานีหรือไม่”
income	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”รายได้ต่อเดือน”
info	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”ท่านทราบข้อมูลท่องเที่ยวจากแหล่งใด”
job	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”อาชีพ”
lives	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”คุณเป็นคนจังหวัดอุบลราชธานีหรือไม่”
travelUbon	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”ความถี่ในการท่องเที่ยวของท่าน”
travel purpose	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”วัตถุประสงค์ในการเดินทางมาอุบลราชธานีในครั้งนี้”

ตารางที่ 3.24: อธิบายโครงสร้างฐานข้อมูลของ datamodel จากรูปที่ 3.36 (ต่อ)

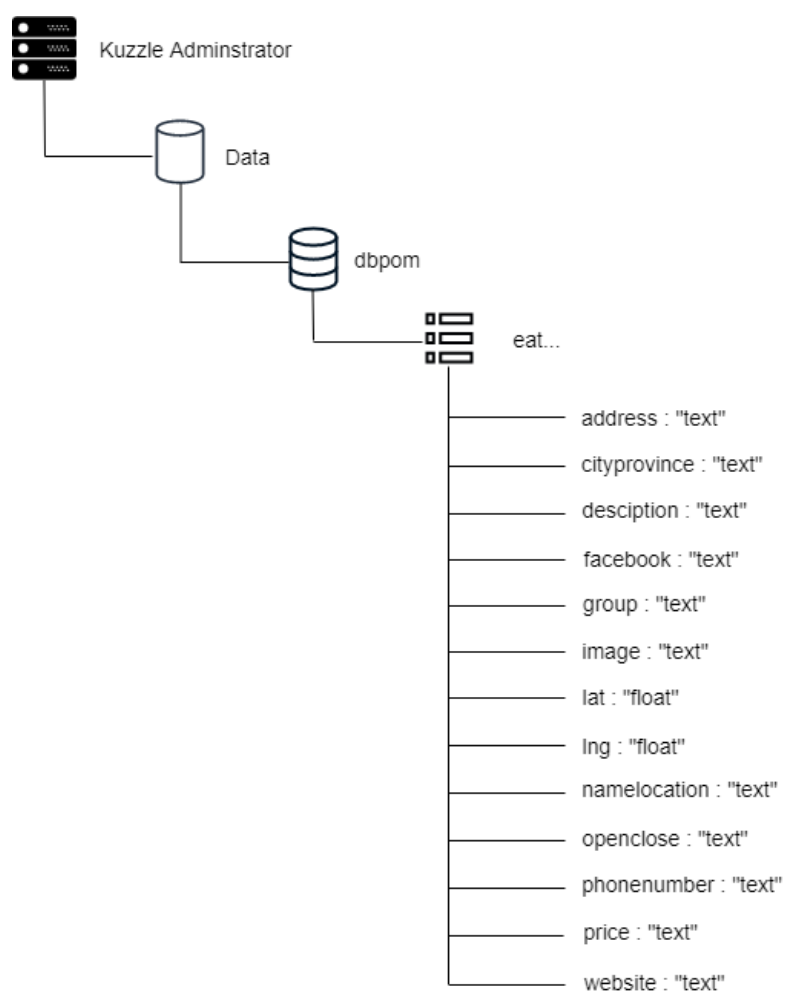
Key	คำอธิบาย
travel style	สำหรับเก็บข้อมูลข้อสงสัยตัวเลือกของคำถาม”ลักษณะการเดินทาง”
vehicle	สำหรับเก็บ ข้อมูลข้อสงสัยตัวเลือกของคำถาม”ยานพาหนะในการเดินทาง”
want	สำหรับเก็บ ข้อมูลข้อสงสัยตัวเลือกของคำถาม”ท่านต้องการได้รับอะไรจากสถานที่ท่องเที่ยว”



รูปที่ 3.37: โครงสร้างฐานข้อมูล dataedit

ตารางที่ 3.25: อธิบายโครงสร้างฐานข้อมูลของ dataedit จากรูปที่ 3.37

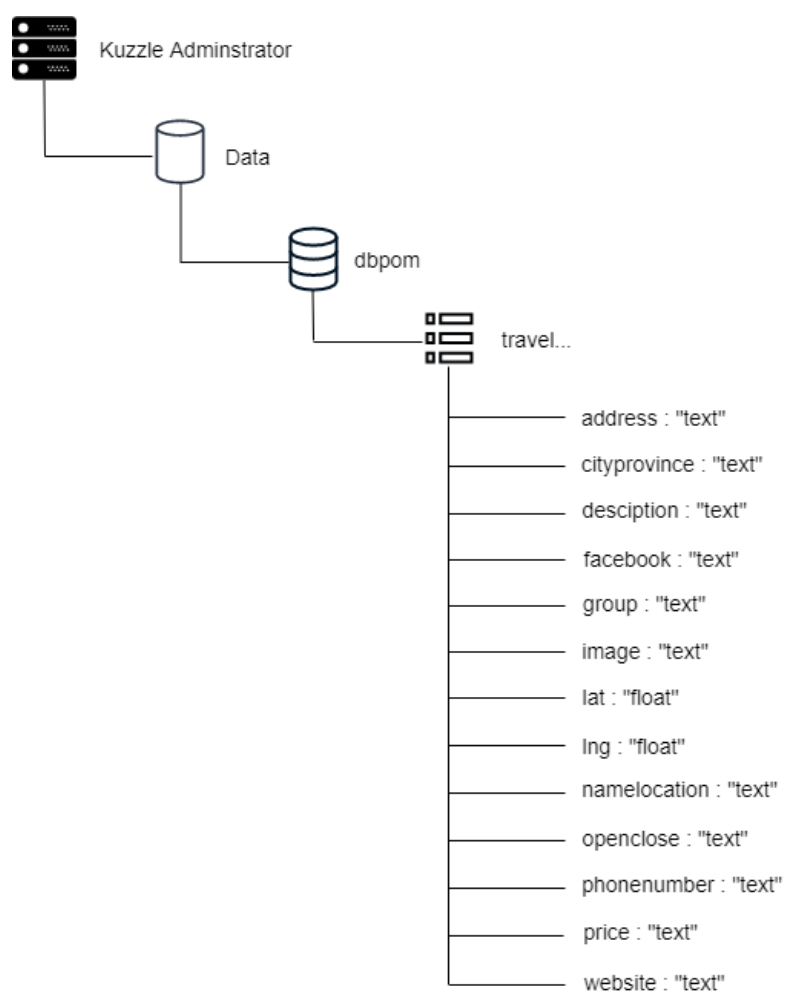
Key	คำอธิบาย
address	สำหรับเก็บข้อมูลที่อยู่ของสถานที่
cityprovince	สำหรับเก็บข้อมูลอำเภอและจังหวัดของสถานที่
description	สำหรับเก็บข้อมูลรายละเอียดของสถานที่
edit by	สำหรับเก็บข้อมูลชื่อผู้แจ้งแก้ไข
facebook	สำหรับเก็บข้อมูลชื่อเฟซบุ๊กของสถานที่
group	สำหรับเก็บข้อมูลประเภทของสถานที่
image	สำหรับเก็บข้อมูลรูปภาพสถานที่ โดยเก็บเป็น base64
img user	สำหรับเก็บข้อมูลรูปภาพผู้ใช้ โดยเก็บเป็น base64
lat	สำหรับเก็บข้อมูลค่าละติจูดของสถานที่
lng	สำหรับเก็บข้อมูลค่าลองจิจูดของสถานที่
namelocation	สำหรับเก็บข้อมูลชื่อสถานที่
openclose	สำหรับเก็บข้อมูลวันเวลาเปิด-ปิดของสถานที่
phonenumner	สำหรับเก็บข้อมูลเบอร์โทรศัพท์ติดต่อสถานที่
price	สำหรับเก็บข้อมูลราคาอาหาร ราคาที่พัก และราคาเข้าชมสถานที่
website	สำหรับเก็บข้อมูล url เว็บไซต์ของสถานที่



รูปที่ 3.38: โครงสร้างฐานข้อมูล eat

ตารางที่ 3.26: อธิบายโครงสร้างฐานข้อมูลที่ใช้เก็บข้อมูลสถานที่กิน จากรูปที่ 3.38

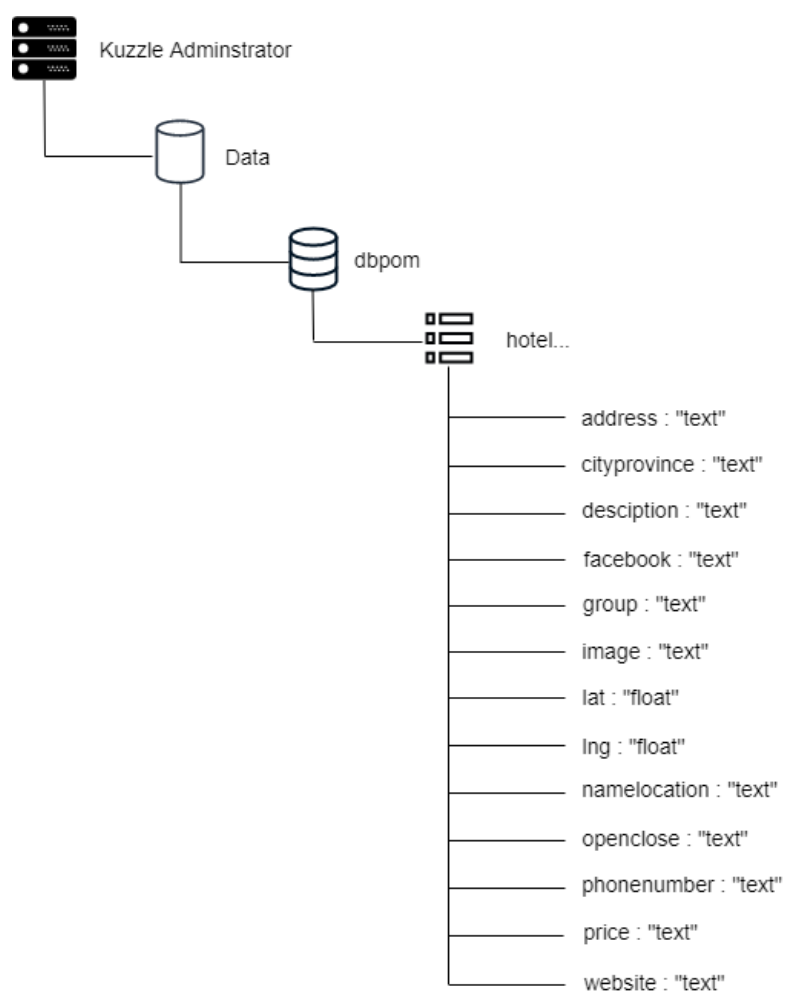
Key	คำอธิบาย
address	สำหรับเก็บข้อมูลที่อยู่ของสถานที่
cityprovince	สำหรับเก็บข้อมูลอำเภอและจังหวัดของสถานที่
description	สำหรับเก็บข้อมูลรายละเอียดของสถานที่
facebook	สำหรับเก็บข้อมูลชื่อเฟซบุ๊กของสถานที่
group	สำหรับเก็บข้อมูลประเภทของสถานที่
image	สำหรับเก็บข้อมูลรูปภาพสถานที่ โดยเก็บเป็น base64
lat	สำหรับเก็บข้อมูลค่าละติจูดของสถานที่
lng	สำหรับเก็บข้อมูลค่าลองจิจูดของสถานที่
namelocation	สำหรับเก็บข้อมูลชื่อของสถานที่กิน
openclose	สำหรับเก็บข้อมูลวันเวลาเปิด-ปิดของสถานที่
phonenummer	สำหรับเก็บข้อมูลเบอร์โทรศัพท์ติดต่อของสถานที่
price	สำหรับเก็บข้อมูลราคาอาหาร ราคาที่พัก และราคาเข้าชมสถานที่
website	สำหรับเก็บข้อมูล url เว็บไซต์ของสถานที่



รูปที่ 3.39: โครงสร้างฐานข้อมูล travel

ตารางที่ 3.27: อธิบายโครงสร้างฐานข้อมูลที่ใช้เก็บข้อมูลสถานที่เที่ยว จากรูปที่ 3.39

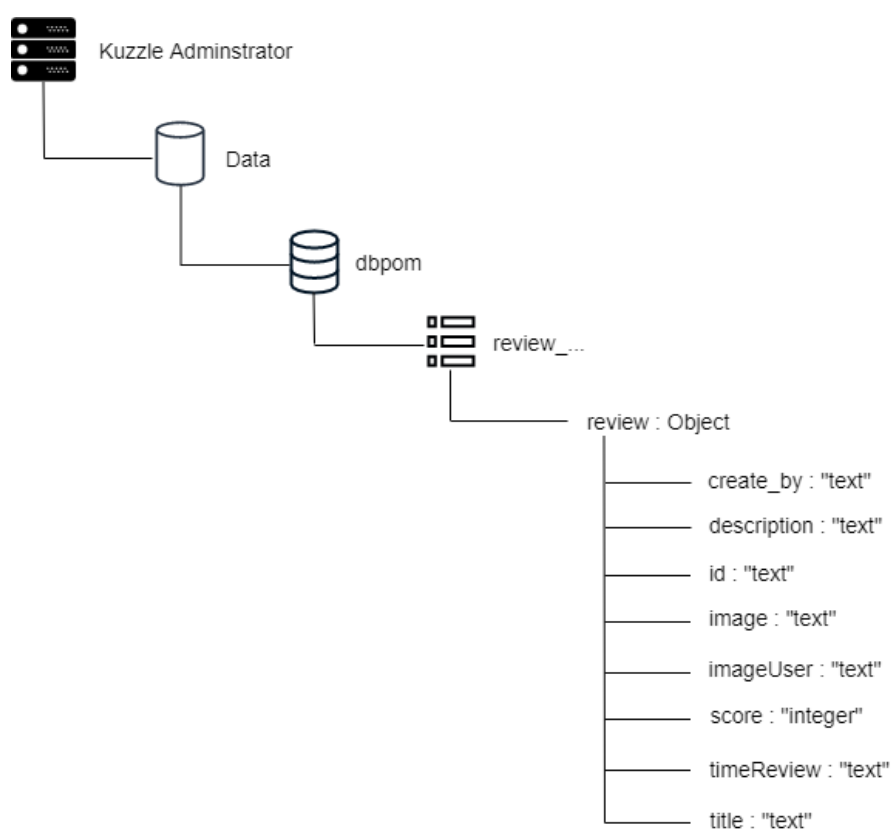
Key	คำอธิบาย
address	สำหรับเก็บข้อมูลที่อยู่ของสถานที่
cityprovince	สำหรับเก็บข้อมูลอำเภอและจังหวัดของสถานที่
description	สำหรับเก็บข้อมูลรายละเอียดของสถานที่
facebook	สำหรับเก็บข้อมูลชื่อเฟสบุ๊คของสถานที่
group	สำหรับเก็บข้อมูลประเภทของสถานที่
image	สำหรับเก็บข้อมูลรูปภาพสถานที่ โดยเก็บเป็น base64
lat	สำหรับเก็บข้อมูลค่าละติจูดของสถานที่
lng	สำหรับเก็บข้อมูลค่าลองจิจูดของสถานที่
namelocation	สำหรับเก็บข้อมูลชื่อของสถานที่เที่ยว
openclose	สำหรับเก็บข้อมูลวันเวลาเปิด-ปิดของสถานที่
phonenummer	สำหรับเก็บข้อมูลเบอร์โทรศัพท์ติดต่อของสถานที่
price	สำหรับเก็บข้อมูลราคาอาหาร ราคาที่พัก และราคาเข้าชมสถานที่
website	สำหรับเก็บข้อมูล url เว็บไซต์ของสถานที่



รูปที่ 3.40: โครงสร้างฐานข้อมูล hotel

ตารางที่ 3.28: อธิบายโครงสร้างฐานข้อมูลที่ใช้เก็บข้อมูลสถานที่พัก จากรูปที่ 3.40

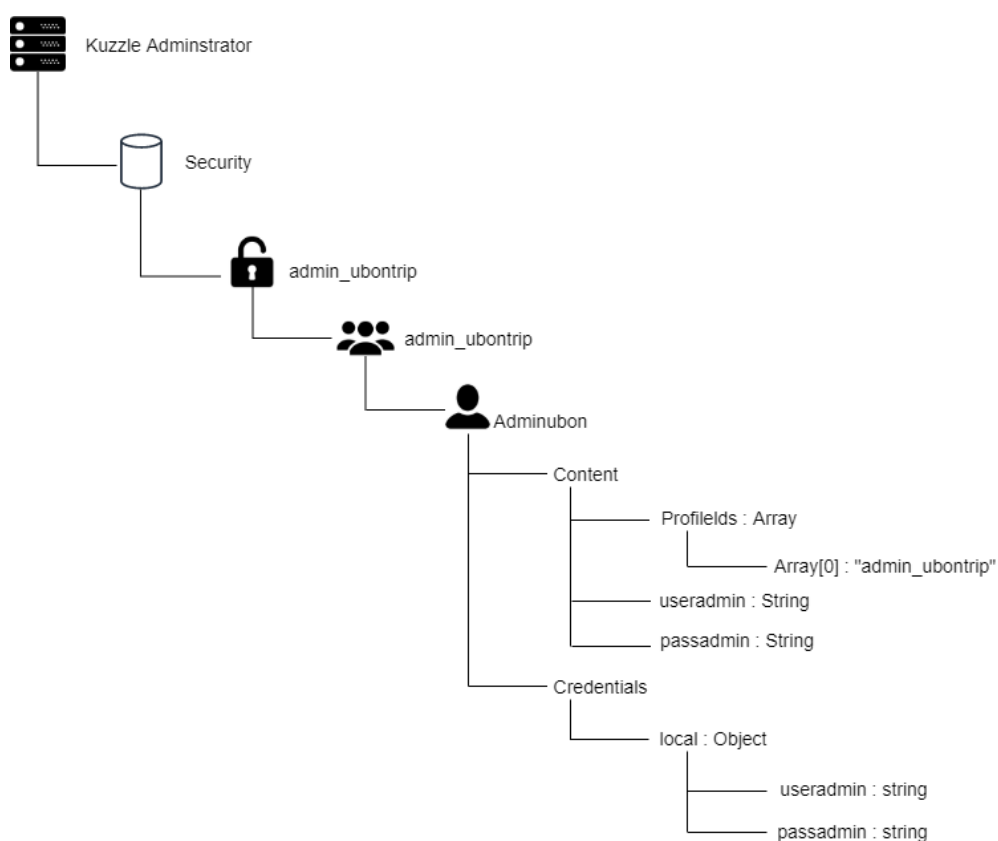
Key	คำอธิบาย
address	สำหรับเก็บข้อมูลที่อยู่ของสถานที่
cityprovince	สำหรับเก็บข้อมูลอำเภอและจังหวัดของสถานที่
description	สำหรับเก็บข้อมูลรายละเอียดของสถานที่
facebook	สำหรับเก็บข้อมูลชื่อเฟสบุ๊คของสถานที่
group	สำหรับเก็บข้อมูลประเภทของสถานที่
image	สำหรับเก็บข้อมูลรูปภาพสถานที่ โดยเก็บเป็น base64
lat	สำหรับเก็บข้อมูลค่าละติจูดของสถานที่
lng	สำหรับเก็บข้อมูลค่าลองจิจูดของสถานที่
namelocation	สำหรับเก็บข้อมูลชื่อของสถานที่พัก
openclose	สำหรับเก็บข้อมูลวันเวลาเปิด-ปิดของสถานที่
phonenummer	สำหรับเก็บข้อมูลเบอร์โทรศัพท์ติดต่อของสถานที่
price	สำหรับเก็บข้อมูลราคาอาหาร ราคาที่พัก และราคาเข้าชมสถานที่
website	สำหรับเก็บข้อมูล url เว็บไซต์ของสถานที่



รูปที่ 3.41: โครงสร้างฐานข้อมูล review

ตารางที่ 3.29: อธิบายโครงสร้างฐานข้อมูลที่ใช้เก็บข้อมูลการเขียนรีวิวสถานที่ จากรูปที่ 3.41

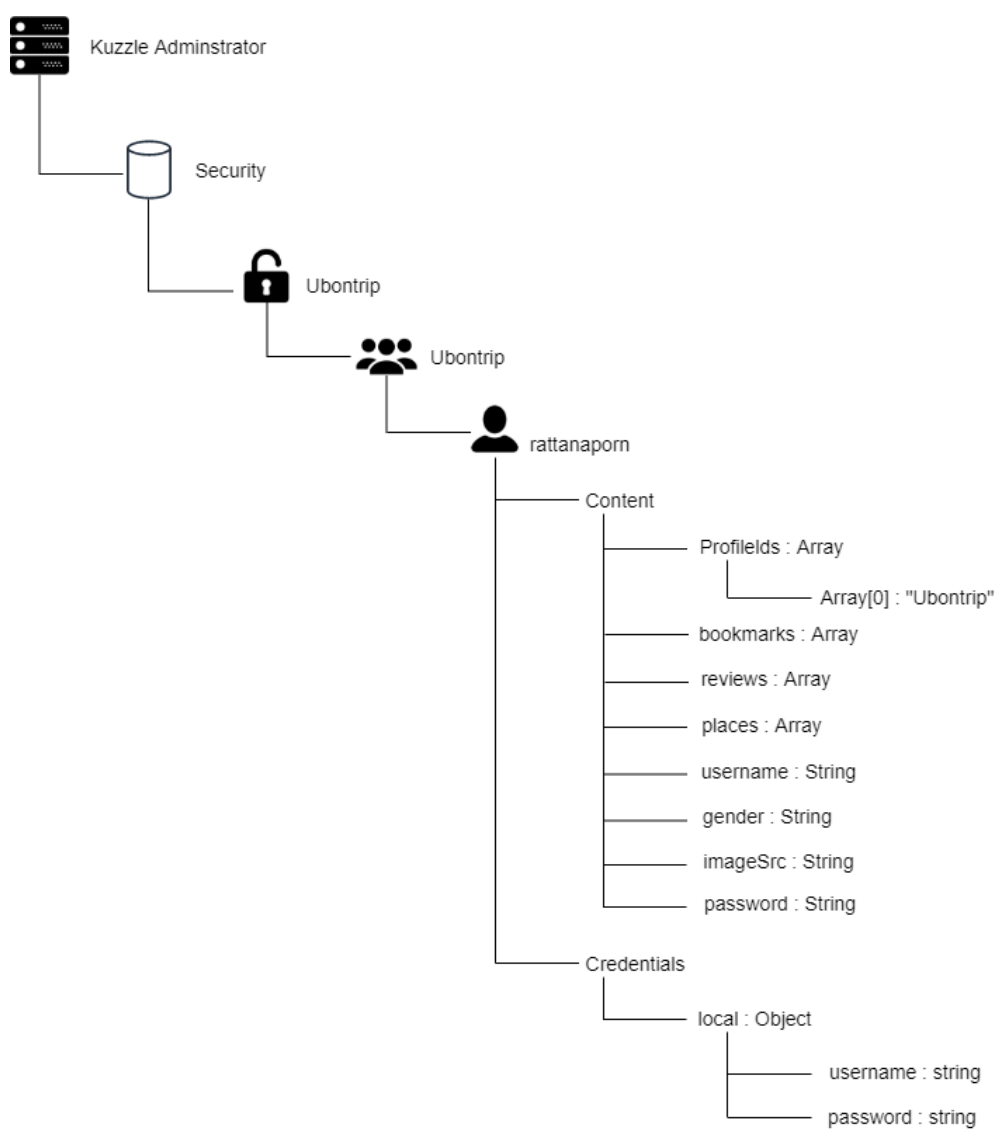
Key	คำอธิบาย
review	เป็น array ที่เก็บเป็น object รายละเอียดข้อมูลต่างๆ ของการเขียนรีวิวโดยใน review จะเก็บข้อมูลการเขียนรีวิวทั้งหมด 8 อย่าง ชื่อผู้เขียนรีวิว รูปภาพผู้เขียนรีวิว หัวข้อรีวิว รายละเอียดรีวิว รูปภาพของรีวิว คะแนนรีวิว วันเวลาในการเขียนรีวิวและไอดี(id)ของสถานที่
create _{by}	สำหรับเก็บข้อมูลชื่อผู้ใช้ที่เขียนรีวิว
description	สำหรับเก็บข้อมูลรายละเอียดของรีวิว
id	สำหรับเก็บข้อมูลไอดี(id) ของสถานที่
image	สำหรับเก็บข้อมูลรูปภาพรีวิว โดยเก็บเป็น base64
imageUser	สำหรับเก็บข้อมูลรูปภาพผู้ใช้ที่เขียนรีวิว โดยเก็บเป็น base64
score	สำหรับเก็บข้อมูลการให้คะแนนรีวิวสถานที่
timeReview	สำหรับเก็บข้อมูลวันเวลาในการเขียนรีวิว
title	สำหรับเก็บข้อมูลหัวข้อของรีวิว



รูปที่ 3.42: โครงสร้างฐานข้อมูล Admin

ตารางที่ 3.30: อธิบายโครงสร้างฐานข้อมูลของ admin ในส่วน security จากรูปที่ 3.42

Key	คำอธิบาย
Content	สำหรับเก็บข้อมูลทั่วไปที่เราต้องการบันทึก
Profiles	เก็บข้อมูลว่าเรามีสิทธิ์ในการจัดการข้อมูลในระดับไหน
username	สำหรับเก็บข้อมูลชื่อผู้ดูแลระบบ
password	สำหรับเก็บข้อมูลรหัสผ่านของผู้ดูแลระบบ
Credentials	สำหรับเก็บข้อมูลสิ่งที่สำคัญเพื่อใช้ระบุตัวตน
local	สำหรับเก็บข้อมูล username ของผู้ดูแลระบบและเก็บ id หรือ Token ของผู้ดูแลระบบไว้ใน kuid



รูปที่ 3.43: โครงสร้างฐานข้อมูล User

ตารางที่ 3.31: อธิบายโครงสร้างฐานข้อมูลของ user ในส่วน security จากรูปที่ 3.43

Key	คำอธิบาย
Content	สำหรับเก็บข้อมูลทั่วไปที่เราต้องการบันทึก
Profiles	เก็บข้อมูลว่าเรามีสิทธิในการจัดการข้อมูลในระดับไหน
bookmarks	สำหรับเก็บข้อมูลสถานที่ทั้งหมดที่ผู้ใช้กดบันทึกสถานที่
reviews	สำหรับเก็บข้อมูลประวัติการเขียนรีวิวสถานที่ต่างๆของผู้ใช้
places	สำหรับเก็บข้อมูลการเพิ่มสถานที่ของผู้ใช้
username	สำหรับเก็บข้อมูลชื่อผู้ใช้ในระบบ
gender	สำหรับเก็บข้อมูลเพศของผู้ใช้ในระบบ
imageSrc	สำหรับเก็บข้อมูลรูปภาพของผู้ใช้ในระบบ โดยเก็บเก็บ base64
password	สำหรับเก็บข้อมูลรหัสผ่านของผู้ใช้ในระบบ
Credentials	สำหรับเก็บข้อมูลสิ่งที่สำคัญเพื่อใช้ระบุตัวตน
local	สำหรับเก็บข้อมูล username ของผู้ใช้และเก็บ id หรือ Token ของผู้ใช้ไว้ใน kuid

บทที่ 4

การพัฒนาระบบ

ในบทนี้จะกล่าวถึงการสร้างระบบงานของระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี โดยนำผลที่ได้จากการวิเคราะห์และออกแบบระบบมาสร้างเป็นระบบงานซึ่งจะอธิบายถึงตัวอย่างการเขียน โปรแกรมการทำงานของระบบในส่วนต่างๆดังต่อไปนี้

4.1 การพัฒนาในส่วนสมัครสมาชิก

เมื่อผู้ใช้กรอกข้อมูลการสมัครเสร็จแล้วทำการกดปุ่มสมัครสมาชิก ระบบจะมีความทำงานแสดงดังรูปที่ 4.1 - 4.2

รูปที่ 4.1: การทำงานของระบบเมื่อกดปุ่มสมัครสมาชิก

จากภาพที่ 4.1 โครงสร้างของไฟล์ register.component.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 ฟังก์ชัน saveRegister() เป็นฟังก์ชันที่ใช้ควบคุมการทำงานในการสมัครสมาชิก
- บรรทัดที่ 2 - 7 เป็นการกำหนดค่าข้อมูลเพศเมื่อผู้ใช้ออกเข้ามาจะเช็คว่า value ที่ผู้ใช้เลือกเป็น Male หรือ Female
- บรรทัดที่ 8 เก็บข้อมูลไว้ที่ data โดยรูปแบบที่เก็บจะต้องตรงกับรูปแบบที่อยู่ในฐานข้อมูล
- บรรทัดที่ 9 content จะใช้เก็บข้อมูลทั่วไปที่เราต้องการจะบันทึก
- บรรทัดที่ 10 เป็นการกำหนด roles ให้กับผู้ใช้
- บรรทัดที่ 11 - 17 เป็นข้อมูลพื้นฐานที่จะใช้เก็บประวัติของผู้ใช้ bookmarks เก็บข้อมูลการบันทึกสถานที่ reviews เก็บข้อมูลการเขียนรีวิว places เก็บข้อมูลการเพิ่มสถานที่
- บรรทัดที่ 19 - 24 เป็นการกำหนดให้ใช้ข้อมูล 2 อย่างนี้ในการเข้าสู่ระบบ คือ username และ password

รูปที่ 4.2: การทำงานของระบบเมื่อกดปุ่มสมัครสมาชิก (ต่อ)

จากภาพที่ 4.2 โครงสร้างของไฟล์ register.component.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 - 3 เป็นการเช็การกรอกข้อมูลรหัสผ่านและรหัสยืนยันว่าตรงกันไหมถ้าตรงกันระบบจะทำการสร้าง user เข้าไปในระบบฐานข้อมูล

- บรรทัดที่ 4 - 5 เมื่อสร้าง user สำเร็จจะแสดงข้อความแจ้งผู้ใช้ว่า "ลงทะเบียนสำเร็จ" และทำการ router ไปยังหน้า popular
- บรรทัดที่ 6 - 8 เป็นการเช็คเงื่อนไขกรณีที่รหัสผ่านไม่ตรงกัน จะแสดงข้อความแจ้งผู้ใช้ว่า "รหัสผ่านและยืนยันรหัสผ่านไม่ตรงกัน กรุณากรอกรหัสผ่านใหม่อีกครั้ง"

4.2 การพัฒนาในส่วนการเข้าสู่ระบบ

เมื่อผู้ใช้กรอกข้อมูลทั้งหมดเสร็จทำการกดปุ่มเข้าสู่ระบบ ระบบจะมีความทำงาน แสดงดังรูป

ที่ 4.3

รูปที่ 4.3: การทำงานของระบบเมื่อกดเข้าสู่ระบบ

จากภาพที่ 4.3 โครงสร้างของไฟล์ dialoglogin.component.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เมื่อผู้ใช้กดเข้าสู่ระบบจะเข้ามาทำงานในฟังก์ชัน Login()
- บรรทัดที่ 2- 5 เป็นการยิงข้อมูล username และ password ที่ผู้ใช้กรอกเข้ามาเพื่อนำไปเชีครกับข้อมูลที่อยู่ในฐานข้อมูล
- บรรทัดที่ 7 เป็นการสร้างตัวแปร JWT ขึ้นมาเก็บค่าของ token ไว้ใน localStorage
- บรรทัดที่ 10 เป็นการอ่านค่าตัวแปร JWT
- บรรทัดที่ 13 เป็นการเชีคเงื่อนไขสแตตัสในการเข้าสู่ระบบ
- บรรทัดที่ 14 ถ้าสแตตัสที่ลือคอินเข้าสู่ระบบเป็น 200 จะสร้าง UserInfo ขึ้นมาเก็บค่าข้อมูลไว้ใน localStorage
- บรรทัดที่ 15 - 18 เรียกใช้ service isAuth เพื่อเก็บค่าข้อมูลผู้ใช้ โดยข้อมูลจะถูกเก็บไว้ในตัวแปร data
- บรรทัดที่ 19 เป็นการปิดหน้าการเข้าสู่ระบบลง

4.3 การพัฒนาในส่วนการจัดอันดับสถานที่

เมื่อผู้ใช้กดเลือกหมวดเพื่อดูการจัดอันดับ ระบบจะมีความทำงาน แสดงดังรูปที่ 4.4 - 4.5

รูปที่ 4.4: การทำงานของระบบเมื่อผู้ใช้กดเลือกหมวด

จากภาพที่ 4.4 โครงสร้างของไฟล์ popular.component.ts อธิบายการทำงานได้ดังนี้ โดยผู้พัฒนาจะขอยกตัวอย่างการทำงานแค่ของหมวดสถานที่ท่องเที่ยว(case:travel) เนื่องจากว่าทั้ง 3 case มีการทำงานเหมือนกันได้แก่ case: "travel" case:"eat" case: "hotel"

- บรรทัดที่ 1 เมื่อผู้ใช้กดเลือกหมวดสถานที่ที่จะเข้ามาทำงานในฟังก์ชัน selectGroup() โดยรับค่าพารามิเตอร์มา 1 ตัว
- บรรทัดที่ 2 เป็นการเซตค่าที่ผู้ใช้กดเลือกมาเก็บไว้ในตัวแปร select
- บรรทัดที่ 3 เป็นการเขียน switch case เซตค่าที่ผู้ใช้เลือก โดยจะมีด้วยกันทั้งหมด 3 case ได้แก่ case:"travel" , case:"eat", case:"hotel"
- บรรทัดที่ 4 ถ้าผู้ใช้เลือกหมวดท่องเที่ยวก็จะเข้ามาทำงานใน case:"travel" แต่ถ้าผู้ใช้ไม่ได้เลือกหมวดท่องเที่ยวก็จะไปทำงานใน case อื่นที่ไม่ใช่ travel
- บรรทัดที่ 6 - 25 เป็นการประกาศค่าเพื่อเก็บข้อมูลในรูปแบบ Object โดย type เก็บประเภทของสถานที่ และ review เก็บประเภทของรีวิว
- บรรทัดที่ 26 นำข้อมูลที่ถูกเก็บไว้ใน listallTravel มาทำการ map เพื่อดูข้อมูลทั้งหมด
- บรรทัดที่ 27 - 30 เป็นการกำหนดจำนวนของข้อมูลที่ต้องการค้นหาออกมาแสดง
- บรรทัดที่ 31 เป็นการ query หาข้อมูลในฐานข้อมูลตามประเภทสถานที่ที่ผู้ใช้เลือก โดยใช้รูปแบบ method POST ในการค้นหา

รูปที่ 4.5: การทำงานของระบบเมื่อผู้ใช้กดเลือกหมวด (ต่อ)

จากภาพที่ 4.5 โครงสร้างของไฟล์ popular.component.ts อธิบายการทำงานได้ดังนี้(ต่อ)

- บรรทัดที่ 1 เป็นการ subscribe เพื่อดูข้อมูลข้างในที่ได้จากการ query ข้อมูล
- บรรทัดที่ 2 เป็นการจิ้มข้อมูลเข้าไปชั้นข้างในเพื่อให้ได้ข้อมูลที่ต้องการและทำการวนฟอดดูข้อมูลข้างใน
- บรรทัดที่ 3 เป็นการประกาศตัวแปรtotalPlace ใช้เก็บค่าผลรวม ส่วน avg ใช้เก็บค่าเฉลี่ยของการให้คะแนนสถานที่
- บรรทัดที่ 4 - 7 เป็นการ ค้นหาข้อมูลจากฐานข้อมูล เมื่อได้ข้อมูลมาจะใช้ .subscribe ในการ เข้าดูข้อมูลโดยจะประกาศ res ขึ้นมาเก็บค่าข้อมูลที่ได้ทำการค้นหาและทำการจิ้มข้อมูลใน res เข้าไปถึงส่วนของข้อมูลที่เรากำลังต้องการ
- บรรทัดที่ 8 เป็นการนำค่าที่ได้ในการวนฟอดในแต่ละรอบมาเก็บไว้ที่ totalplace
- บรรทัดที่ 10 เป็นการคำนวณคะแนนเฉลี่ยในการรีวิวของแต่ละสถานที่และค่าที่จะนำมา เก็บไว้ในตัวแปร avg
- บรรทัดที่ 11 - 13 เป็นการ push ค่าที่ได้ไปเก็บไว้ที่ resultspopulartravel
- บรรทัดที่ 15 - 17 เป็นการนำค่าที่ถูกเก็บไว้มาทำการจัดอันดับโดยจะเช็คค่า ค่าเฉลี่ยของ ตัวที่ 1 มีค่ามากกว่าตัวที่ 2 ถ้าใช่ก็ให้มีค่าเป็น 1 ถ้าค่าเฉลี่ยตัวที่ 2 มากกว่าตัวที่ 1 ก็ให้มี ค่าเป็น -1 และทำการ reverse() ข้อมูลโดยจะเช็คไปแบบนี้จนครบถึงจะหยุดทำงาน
- บรรทัดที่ 18 - 21 ถ้าข้อมูลสถานที่นั้นยังไม่มีรีวิว ก็จะทำการ push ข้อมูลสถานที่โดยจะ กำหนดให้ค่าของ avg = 0
- บรรทัดที่ 23 - 25 เป็นการนำค่าที่ถูกเก็บไว้มาทำการจัดอันดับโดยจะเช็คค่า ค่าเฉลี่ยของ ตัวที่ 1 มีค่ามากกว่าตัวที่ 2 ถ้าใช่ก็ให้มีค่าเป็น 1 ถ้าค่าเฉลี่ยตัวที่ 2 มากกว่าตัวที่ 1 ก็ให้มี ค่าเป็น -1 และทำการ reverse() ข้อมูลโดยจะเช็คไปแบบนี้จนครบถึงจะหยุดทำงาน

4.4 การพัฒนาในส่วนของการเขียนรีวิว

เมื่อผู้ใช้กดปุ่มบันทึกรีวิว ระบบจะทำการงาน แสดงดังรูปที่ 4.6 - 4.8

รูปที่ 4.6: การทำงานของระบบเมื่อผู้ใช้กดปุ่มเขียนรีวิว

จากภาพที่ 4.6 โครงสร้างของไฟล์ dialog-review.component.ts อธิบายการทำงานได้ดังนี้(ต่อ)

- บรรทัดที่ 1 - 3 ฟังก์ชัน onClick() เป็นฟังก์ชันของการให้คะแนนสถานที่ โดยจะเก็บคะแนนที่ผู้ใช้ให้ไว้ในต rating
- บรรทัดที่ 4 ฟังก์ชัน reviewPlace() จะทำงานเมื่อผู้ใช้กดปุ่ม บันทึกรีวิว โดยจะทำการบันทึกข้อมูลรีวิว
- บรรทัดที่ 5 เป็นการประกาศ now ขึ้นมาเก็บค่าของวันที่ในการเขียนรีวิว โดยจะมี format วันที่เป็น วัน/เดือน/ปี
- บรรทัดที่ 6 เป็นเงื่อนไขในการเช็คค่า id ของรีวิวสถานที่นี้ว่า มีค่าเท่ากับ undefined หรือไม่ถ้าเท่าจะเข้ามาทำงานในเงื่อนไข
- บรรทัดที่ 7 - 8 เป็นการประกาศตัวแปรชื่อ dataReview มาเก็บข้อมูลการเขียนรีวิวโดยใน dataReview จะมีการเก็บเป็นอาเรย์ในตัวแปร review
- บรรทัดที่ 10 เป็นการเก็บ id ของสถานที่รีวิวไว้ในตัวแปรชื่อ id
- บรรทัดที่ 11 เป็นการเก็บชื่อผู้ใช้ที่เขียนรีวิวโดยจะไปอ่านค่ามาจาก localStorage ใน filed ชื่อ username และเก็บไว้ในตัวแปรชื่อ createby
- บรรทัดที่ 12 เป็นการเก็บรูปภาพของผู้ใช้ที่เขียนรีวิวโดยจะไปอ่านค่ามาจาก localStorage ใน filed ชื่อ imageSrc และเก็บไว้ในตัวแปรชื่อ imageUser
- บรรทัดที่ 13 เป็นการเก็บหัวข้อของรีวิวไว้ในตัวแปรชื่อ title
- บรรทัดที่ 14 เป็นการเก็บรายละเอียดของการเขียนรีวิวไว้ในตัวแปรชื่อ description
- บรรทัดที่ 15 เป็นการเก็บรูปภาพรีวิวไว้ในตัวแปรชื่อ image
- บรรทัดที่ 16 เป็นการเก็บค่าของคะแนนรีวิวไว้ในตัวแปร score
- บรรทัดที่ 17 เป็นการเก็บเวลาของการเขียนรีวิวไว้ในตัวแปรชื่อ timeReview
- บรรทัดที่ 19 - 21 เป็นการทำงานโดยจะเอาข้อมูลที่ได้จากการเขียนรีวิวในตัวแปรชื่อ dataReview ไปทำการสร้าง document เพื่อเก็บข้อมูลรีวิวของสถานที่นั้นโดยใช้ method post ในการสร้าง หลังจากสร้างเสร็จจะ log ค่าออกมาดูเพื่อดูผลลัพธ์

- บรรทัดที่ 23 - 27 เป็นการประกาศตัวแปรชื่อ reviews มาเก็บประวัติของผู้ใช้ว่าเคยเขียนรีวิวสถานที่ไหนบ้าง โดยจะเก็บ id คือเก็บชื่อสถานที่ type คือประเภทสถานที่และ review คือประเภทของสถานที่ที่เขียนรีวิว
- บรรทัดที่ 28 - 29 เป็นการนำข้อมูลที่อยู่ในก้อน reviews ไปทำการอัปเดตข้อมูลผู้ใช้ในส่วนของการเขียนรีวิว

รูปที่ 4.7: การทำงานของระบบเมื่อผู้ใช้กดปุ่มเขียนรีวิว(ต่อ)

จากภาพที่ 4.7 โครงสร้างของไฟล์ dialog-review.component.ts อธิบายการทำงานได้ดังนี้(ต่อ)

- บรรทัดที่ 1 เป็นเงื่อนไขเช็คค่าถ้า id ของสถานที่ไม่เท่ากับ undefined ก็จะเข้ามาทำงานในส่วนนี้
- บรรทัดที่ 2 - 3 เป็นการประกาศตัวแปรชื่อ dataReview มาเก็บข้อมูลการเขียนรีวิวโดยใน dataReview จะมีการเก็บเป็นอาเรย์ในตัวแปร review
- บรรทัดที่ 4 เป็นนำค่ารีวิวเดิมเดิมที่มีอยู่ มาต่อกับรีวิวตัวใหม่
- บรรทัดที่ 6 เป็นการเก็บ id ของสถานที่ที่รีวิวไว้ในตัวแปรชื่อ id
- บรรทัดที่ 7 เป็นการเก็บชื่อผู้ใช้ที่เขียนรีวิวโดยจะไปอ่านค่ามาจาก localStorage ใน filed ชื่อ username และเก็บไว้ในตัวแปรชื่อ createby
- บรรทัดที่ 8 เป็นการเก็บรูปภาพของผู้ใช้ที่เขียนรีวิวโดยจะไปอ่านค่ามาจาก localStorage ใน filed ชื่อ imageSrc และเก็บไว้ในตัวแปรชื่อ imageUser
- บรรทัดที่ 9 เป็นการเก็บหัวข้อของรีวิวไว้ในตัวแปรชื่อ title
- บรรทัดที่ 10 เป็นการเก็บรายละเอียดของการเขียนรีวิวไว้ในตัวแปรชื่อ description
- บรรทัดที่ 11 เป็นการเก็บรูปภาพรีวิวไว้ในตัวแปรชื่อ image
- บรรทัดที่ 12 เป็นการเก็บค่าของคะแนนรีวิวไว้ในตัวแปร score
- บรรทัดที่ 13 เป็นการเก็บเวลาของการเขียนรีวิวไว้ในตัวแปรชื่อ timeReview
- บรรทัดที่ 17 - 19 เป็นการทำงานโดยเอาข้อมูลที่ได้จาก dataReview ไปทำการอัปเดตข้อมูลการเขียนรีวิวของสถานที่นั้นในฐานข้อมูลโดยทำผ่าน method put และเมื่อทำสำเร็จจะแสดงข้อความบอกว่า เพิ่มคอมเม้นสำเร็จ ใน log
- บรรทัดที่ 21 - 22 ถ้าทำงานล้มเหลวจะให้แสดง error

รูปที่ 4.8: การทำงานของระบบเมื่อผู้ใช้กดปุ่มเขียนรีวิว(ต่อ)

จากภาพที่ 4.8 โครงสร้างของไฟล์ dialog-review.component.ts อธิบายการทำงานได้ดังนี้(ต่อ)

- บรรทัดที่ 1 เป็นการประกาศ check ขึ้นมาเพื่อค้นหาข้อมูลการเขียนรีวิวทั้งหมดที่มีอยู่โดยใช้ item เป็นรับข้อมูลเข้าไปค้นหาต่อ
- บรรทัดที่ 2 - 3 เป็นการเช็คเงื่อนไขว่าถ้าเช็ค เมื่อได้ข้อมูล reviews ก็จะทำการวนพ้อเพื่อดูข้อมูล
- บรรทัดที่ 4 - 5 เมื่อมันผ่านเงื่อนไขก็จะเข้ามาเช็คดูอีกครั้งว่าค่าของ id มีค่าเท่ากับ idplace ใหม่ ถ้าตรงกันก็จะให้รีวิวที่มีอยู่ออกไปแต่ถ้าไม่ใช่ก็จะให้ค่าตัวนั้นมีค่าเป็น null
- บรรทัดที่ 6 - 11 เป็นการประกาศตัวแปรชื่อ review ขึ้นมาเพื่อเก็บประวัติข้อมูลการเขียนรีวิวของผู้ใช้
- บรรทัดที่ 12 - 14 นำข้อมูลที่ได้จาก reviews ไปทำการอัปเดตข้อมูลของผู้ใช้
- บรรทัดที่ 15 - 19 เป็นการ log ดูค่าเมื่ออัปเดตสำเร็จ และเมื่ออัปเดตไม่สำเร็จจะแสดง error
- บรรทัดที่ 21 - 23 ฟังก์ชัน onNoClick เมื่อผู้ใช้ไม่ต้องที่จะเขียนรีวิว ระบบก็จะปิดหน้าต่างเขียนรีวิวลง

4.5 การพัฒนาในส่วนของการเพิ่มสถานที่

เมื่อผู้ใช้กดปุ่มเพิ่มสถานที่ ระบบจะมีความทำงาน แสดงดังรูปที่ 4.9 - 4.11

รูปที่ 4.9: การทำงานของระบบเมื่อผู้ใช้กดปุ่มเพิ่มสถานที่

จากภาพที่ 4.9 โครงสร้างของไฟล์ dialog-add.component.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 - 15 เป็นฟังก์ชันการอัปโหลดรูปภาพแนวนำรูปภาพที่อัปโหลดมาแปลงเป็น base64 เพื่อที่จะไปเก็บไว้ในฐานข้อมูล

รูปที่ 4.10: การทำงานของระบบเมื่อผู้ใช้กดปุ่มเพิ่มสถานที่(ต่อ)

จากภาพที่ 4.10 โครงสร้างของไฟล์ `dialog-add.component.ts` อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 ฟังก์ชัน `addPlace()` เป็นฟังก์ชันที่ใช้ทำการเพิ่มข้อมูลสถานที่ที่จะทำงานก็ต่อเมื่อผู้ใช้กดปุ่มบันทึก
- บรรทัดที่ 2 เป็นการประกาศตัวแปร `selectplace` ที่มี type เป็น `string` เพื่อที่จะใช้เก็บค่าประเภทของสถานที่ที่ผู้ใช้เลือก
- บรรทัดที่ 3 - 38 เป็นการเช็คเงื่อนไขเมื่อผู้ใช้เลือกประเภทของสถานที่เพื่อที่จะกำหนดค่าให้กับ `selectplace`

รูปที่ 4.11: การทำงานของระบบเมื่อผู้ใช้กดปุ่มเพิ่มสถานที่(ต่อ)

จากภาพที่ 4.11 โครงสร้างของไฟล์ dialog-add.component.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เป็นการประกาศตัวแปรชื่อ data โดยมี type เป็น object เพื่อที่จะมาเก็บค่าของรายละเอียดข้อมูลสถานที่ที่ผู้ใช้กรอกเข้ามา
- บรรทัดที่ 2 เป็นการประกาศ image ขึ้นมาเก็บรูปภาพของสถานที่โดยมีรูปแบบการเก็บเป็น base64
- บรรทัดที่ 3 เป็นการประกาศ namelocation ขึ้นมาเพื่อที่จะเก็บชื่อสถานที่
- บรรทัดที่ 4 เป็นการเก็บประเภทของสถานที่ที่ได้จากการเลือกของผู้ใช้ไว้ใน group
- บรรทัดที่ 5 เป็นการเก็บที่อยู่ของสถานที่ที่ได้จากการกรอกเข้ามาจากผู้ใช้ไว้ใน address
- บรรทัดที่ 6 เป็นการเก็บชื่ออำเภอและจังหวัดที่ได้จากการกรอกเข้ามาจากผู้ใช้ไว้ใน cityprovince
- บรรทัดที่ 7 เป็นการเก็บค่าละติจูดที่ได้จากการกรอกข้อมูลเข้ามาจากผู้ใช้ไว้ใน lat
- บรรทัดที่ 8 เป็นการเก็บค่าลองจิจูดที่ได้จากการกรอกข้อมูลเข้ามาจากผู้ใช้ไว้ใน lng
- บรรทัดที่ 9 เป็นการเก็บข้อมูลเวลาเปิด ปิด ของสถานที่ที่ได้จากการกรอกเข้ามาจากผู้ใช้ไว้ใน openclose
- บรรทัดที่ 10 เป็นการเก็บข้อมูลเบอร์โทรของสถานที่ที่ได้จากการกรอกเข้ามาจากผู้ใช้ไว้ใน phonenumber
- บรรทัดที่ 11 เป็นการเก็บข้อมูลราคาของอาหาร ราคาเข้าชมสถานที่ ที่ได้จากการกรอกเข้ามาจากผู้ใช้ไว้ใน price
- บรรทัดที่ 12 เป็นการเก็บข้อมูลรายละเอียดของสถานที่ที่ได้จากการกรอกเข้ามาจากผู้ใช้ไว้ใน description
- บรรทัดที่ 13 เป็นการเก็บชื่อ facebook ของสถานที่ที่ได้จากการกรอกเข้ามาจากผู้ใช้ไว้ใน facebook
- บรรทัดที่ 14 เป็นการเก็บชื่อ website ของสถานที่ที่ได้จากการกรอกเข้ามาจากผู้ใช้ไว้ใน website
- บรรทัดที่ 16 เป็น การนำข้อมูลสถานที่ที่ได้จาก data ไปทำการเพิ่มข้อมูลลงในฐานข้อมูลตามเป็นเกณฑ์ที่ผู้ใช้เลือก

- บรรทัดที่ 18 เป็นการประกาศตัวแปรชื่อ idname เพื่อมาเก็บค่าของ id สถานที่
- บรรทัดที่ 20 เป็นการเรียกใช้ places เพื่อที่จะนำมาเก็บค่าข้อมูล
- บรรทัดที่ 21 เป็นการเก็บชื่อ id สถานที่ไว้ใน id
- บรรทัดที่ 22 เป็นการเก็บประเภทของสถานที่ไว้ใน type
- บรรทัดที่ 23 เป็นการเก็บประเภทของรีวิวสถานที่ไว้ใน review
- บรรทัดที่ 25 - 27 เป็นการนำข้อมูลที่ได้จาก places มาทำการอัปเดตข้อมูลในส่วนของผู้ใช้เพื่อที่จะเก็บประวัติการเพิ่มสถานที่ของผู้ใช้ไว้และนำข้อมูลที่ได้ไปแสดงในหน้าโปรไฟล์ผู้ใช้

4.6 การพัฒนาในส่วนของการลบบรีว

เมื่อผู้ใช้กดปุ่มลบบรีว ระบบจะมีการทำงาน แสดงดังรูปที่ 4.12

รูปที่ 4.12: การทำงานของระบบเมื่อผู้ใช้กดปุ่มลบบรีว

จากภาพที่ 4.9 โครงสร้างของไฟล์ delete-review.component.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 ฟังก์ชัน deleteReview()g เป็นฟังก์ชันที่ใช้ในการลบรีวิวจะถูกเรียกใช้เมื่อผู้ใช้งานกดปุ่มลบรีวิว
- บรรทัดที่ 2 - 5 เป็นการลบข้อมูลรีวิวออกจากฐานข้อมูลโดยจะลบตามประเภทของสถานที่และ id ของสถานที่ที่ โดยใช้ method delete
- บรรทัดที่ 6 เป็นการประกาศ check ขึ้นมาเพื่อค้นหาข้อมูลการเขียนรีวิวทั้งหมดที่มีอยู่ในฐานข้อมูลของผู้ใช้โดยใช้ item เป็นรับข้อมูลเข้าไปค้นหาต่อ
- บรรทัดที่ 7 - 10 เป็นการเช็คเงื่อนไขเมื่อเข้าเงื่อนไขก็จะเข้ามาทำงานในส่วนนี้โดยใช้ข้อมูลที่เก็บไว้ใน reviews ทำการวนพ้อเพื่อดูข้อมูล และก็เข้าเงื่อนไขก็จะเช็คดูอีกครั้งว่าค่าของ id มีค่าเท่ากับ idplace หใหม่ ถ้าตรงกันก็จะให้รีวิวลบรีวิวที่มีชื่อ id ตรงกับตัวที่ค้นหาออก
- บรรทัดที่ 11 - 15 เป็นการอัปเดตข้อมูลในส่วนของผู้ใช้อีกครั้ง โดยใช้ method put เพื่อที่จะทำการอัปเดตข้อมูลการเขียนรีวิว
- บรรทัดที่ 16 - 18 เป็นการ log ดูค่า error ในกรณีที่อัปเดตข้อมูลไม่สำเร็จ

- 4.7 การพัฒนาในส่วนของการบันทึกสถานที่
เมื่อผู้ใช้กดปุ่มบันทึก ระบบจะการทำงาน แสดงดังรูปที่ 4.13

รูปที่ 4.13: การทำงานของระบบเมื่อผู้ใช้กดปุ่มบันทึกสถานที่

จากภาพที่ 4.9 โครงสร้างของไฟล์ detail-place.component.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 ฟังก์ชัน changeBookMark() ฟังก์ชันนี้จะถูกเรียกใช้ก็ต่อเมื่อผู้ใช้กดปุ่มบันทึก
- บรรทัดที่ 2 - 5 เป็นการเช็คสถานะการล็อกอิน ถ้าสถานะการเข้าสู่ระบบเป็น false จะมีข้อความแจ้งผู้ใช้ให้เข้าสู่ระบบก่อนกดปุ่มบันทึกสถานที่
- บรรทัดที่ 6 - 11 เป็นการเช็คสถานะการล็อกอิน ในกรณีเข้าสู่ระบบสำเร็จแล้วผู้ใช้สามารถกดปุ่มบันทึกสถานที่ได้ โดยจะเก็บค่าของ id สถานที่ type สถานที่ type ของรีวิว ไว้ในตัวแปรชื่อ dataplace
- บรรทัดที่ 12 เป็นการประกาศ check ขึ้นมาเพื่อเก็บค่าผลลัพธ์การค้นหาข้อมูลการบันทึกสถานที่ทั้งหมดที่มีอยู่ในฐานข้อมูลของผู้ใช้โดยใช้ item เป็นรับข้อมูลในการค้นหา
- บรรทัดที่ 13 - 16 เป็นการเช็คเงื่อนไขเมื่อเข้าเงื่อนไขก็จะเข้ามาทำงานในส่วนนี้โดยใช้ข้อมูลที่เก็บไว้ใน bookmarks มาทำการวนพเพื่อดูข้อมูล และก็เข้าเงื่อนไขจะเช็คดูอีกครั้งว่าค่าของ item.id มีค่าเท่ากับ place.id ไหม ถ้าตรงกันก็จะให้ลบชื่อสถานที่ตัวนั้นทิ้ง
- บรรทัดที่ 17 - 22 เป็นการเช็คเงื่อนไขในกรณีที่ข้อมูลการบันทึกสถานที่นั้นยังไม่เคยกดบันทึกก็จะทำการเพิ่มข้อมูลสถานที่นั้นลงไป ใน bookmarks
- บรรทัดที่ 24 - 28 เป็นการอัปเดตข้อมูลในส่วนของ filed ที่ชื่อ bookmarks ในฐานข้อมูลของผู้ใช้ โดยอัปเดตผ่าน method put
- บรรทัดที่ 29 เป็นการรีเซ็ตค่าตรงปุ่มกดบันทึกให้เปลี่ยนเป็นบันทึก

4.8 การพัฒนาในส่วนของการแจ้งแก้ไขข้อมูล

เมื่อผู้ใช้กดปุ่มแจ้งแก้ไข ระบบจะมีความทำงาน แสดงดังรูปที่ 4.14 - 4.15

รูปที่ 4.14: การทำงานของระบบเมื่อผู้ใช้กดปุ่มแจ้งแก้ไขสถานที่

จากภาพที่ 4.14 โครงสร้างของไฟล์ detail-place.component.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 ฟังก์ชัน editDataPlace() ฟังก์ชันนี้จะทำงานก็ต่อเมื่อผู้ใช้กดปุ่มแจ้งแก้ไข
- บรรทัดที่ 2 - 20 เป็นการเช็คเงื่อนไขดูว่าสถานที่ที่จะแจ้งแก้ไขเคยมีประวัติการแจ้งแก้ไขไหม ถ้า `this.idplace != undefined` ก็จะเข้ามาทำงานในส่วนนี้โดยจะทำการเก็บข้อมูลรายละเอียดการแก้ไขไว้ในตัวแปรชื่อ `dataplace`

- บรรทัดที่ 21 - 24 เป็นการส่งข้อมูลที่ได้จาก dataplace ไปทำการอัปเดตข้อมูลการแจ้งแก้ไขเพื่อที่จะส่งไปให้แอดมินเป็นผู้อนุมัติแก้ไขข้อมูล

รูปที่ 4.15: การทำงานของระบบเมื่อผู้ใช้กดปุ่มแจ้งแก้ไขสถานที่

จากภาพที่ 4.15 โครงสร้างของไฟล์ detail-place.component.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 - 19 เป็นเช็คต่อจากภาพที่ 4.14 โดยในเงื่อนไขนี้คือถ้าของ idสถานที่ที่แจ้งแก้ไขมาเป็น undefined หมายความว่าสถานที่นี้ยังไม่มีประวัติการแจ้งแก้ไขโดยจะทำการเก็บข้อมูลการแจ้งแก้ไขที่ได้จากผู้ใช้มาเก็บไว้ใน dataplace
- บรรทัดที่ 20 - 24 เป็นการเรียกใช้ dataplace ที่เก็บข้อมูลการแจ้งแก้ไขไว้มาทำการเพิ่มลงไปในฐานะข้อมูลของการแจ้งแก้ไข

4.9 การพัฒนาในส่วนของการค้นหาสถานที่

เมื่อผู้ใช้กดปุ่มค้นหา ระบบจะมีความทำงาน แสดงดังรูปที่ 4.16 - 4.17

รูปที่ 4.16: การทำงานของระบบเมื่อผู้ใช้กดปุ่มเลือกประเภท

จากภาพที่ 4.16 สามารถอธิบายการทำงานของฟังก์ชันการค้นหาสถานที่ได้ดังนี้

- บรรทัดที่ 1 ฟังก์ชัน dataPlace() จะทำงานก็ต่อเมื่อผู้ใช้กดปุ่มเลือกประเภทสถานที่
- บรรทัดที่ 2 เป็นการเช็คเงื่อนไขดูว่า ประเภทที่ผู้ใช้เลือกนั้นเป็นประเภทไหน ถ้าผู้ใช้เลือกทั้งหมด ก็จะมาเข้าเงื่อนไขนี้
- บรรทัดที่ 4 - 5 เป็นการประกาศ list ขึ้นมาเก็บค่าของประเภท โดยมี type เป็น array
- บรรทัดที่ 6 - 10 เป็นการนำข้อมูลที่อยู่ใน list มาทำการวนพ้อเพื่อนำไปค้นหาข้อมูลในฐานข้อมูลตามประเภททั้งหมดที่มีอยู่ใน list
- บรรทัดที่ 13 เป็นการทำงานหลังจากที่ค้นหาข้อมูลเสร็จจะนำข้อมูลที่ได้มาเก็บไว้ใน resultsPlace
- บรรทัดที่ 17 - 24 เป็นการเช็คเงื่อนไขในกรณีที่ประเภทที่ผู้ใช้เลือกมาไม่ใช่ทั้งหมด ก็จะทำให้การค้นหาข้อมูลในฐานข้อมูลตามประเภทที่ผู้ใช้เลือก เมื่อค้นหาเสร็จก็จะนำข้อมูลที่ได้มาเก็บไว้ใน resultsPlace

รูปที่ 4.17: การทำงานของระบบเมื่อผู้ใช้กดปุ่มค้นหา(ต่อ)

จากภาพที่ 4.17 สามารถอธิบายการทำงานของฟังก์ชันการค้นหาสถานที่ได้ดังนี้

- บรรทัดที่ 1 ฟังก์ชัน onClickplace() จะทำงานก็ต่อเมื่อผู้ใช้กดปุ่มค้นหา
- บรรทัดที่ 2 - 3 เป็นการเช็คเงื่อนไขว่าช่องให้กรอกชื่อสถานที่จะต้องไม่เป็น !null ถ้าเป็น null จะเข้าเงื่อนไขนี้ โดยจะมีความแจ้งผู้ใช้งานว่ากรุณากรอกข้อมูลก่อนกดปุ่มค้นหา
- บรรทัดที่ 4 - 12 เป็นการเช็คเงื่อนไขในกรณีที่ผู้ใช้กรอกชื่อสถานที่เรียบร้อยแล้วก่อนกดปุ่มค้นหา ก็จะเข้ามาทำงานในส่วนนี้โดยจะประกาศตัวแปรชื่อ data ขึ้นมาเพื่อเก็บค่าของ id สถานที่ type สถานที่ type ของรีวิว โดยนำข้อมูลที่ได้ไปเขียนลง localStorage ที่มีชื่อว่า dataPlace หลังจากนั้นก็ router ไปหน้า list-place

4.10 การพัฒนาในส่วนของการทำแบบประเมิน

เมื่อผู้ทำการกรอกแบบประเมิน ระบบจะมีความทำงาน แสดงดังรูปที่ 4.18 - 4.20

รูปที่ 4.18: การทำงานของระบบเมื่อผู้กรอกแบบประเมิน

จากภาพที่ 4.18 โครงสร้างไฟล์ questionnaire.component.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 - 18 เป็นฟอร์มที่ใช้เก็บค่าจากผู้ใช้เมื่อผู้ใช้เลือกข้อสงสัยของแต่ละคำถามซึ่งจะมีด้วยกันทั้งหมด 16 ข้อ

รูปที่ 4.19: การทำงานของระบบเมื่อผู้ใช้กรอกแบบประเมินเสร็จกดปุ่มส่งข้อมูล(ต่อ)

จากภาพที่ 4.19 โครงสร้างไฟล์ questionnaire.component.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 ฟังก์ชัน predict() ฟังก์ชันนี้จะทำงานก็ต่อเมื่อผู้ใช้กดปุ่มส่งข้อมูลโดยจะส่งข้อมูลที่ได้จากการกรอกแบบประเมินไปให้ฝั่ง server
- บรรทัดที่ 2 - 5 เป็นการเช็คว่าฟอร์มที่ผู้ใช้กรอกมาตอบครบทุกข้อหรือไม่ ถ้ายังไม่ครบทุกข้อก็จะไม่สามารถกดส่งข้อมูลได้โดยระบบจะแจ้งข้อที่ผู้ใช้ข้าม
- บรรทัดที่ 6 ในกรณีที่ผู้ใช้ตอบครบทุกข้อก็จะสามารถกดปุ่มส่งข้อมูลได้
- บรรทัดที่ 7 - 22 เป็นการเก็บค่าข้อมูลคำตอบที่ผู้ใช้เลือกจาก userForm ไว้ใน user.q1 - user.q16 โดยเรียงตามข้อเริ่มจากข้อ 1 ถึง 16
- บรรทัดที่ 24 - 30 เป็นการเรียกใช้ resultModel เพื่อเก็บค่าผลลัพธ์ของคำตอบที่ผู้ใช้เลือก ซึ่งมีการเก็บเป็นแบบ array

รูปที่ 4.20: การทำงานของระบบเมื่อผู้ใช้กรอกแบบประเมินเสร็จกดปุ่มส่งข้อมูล(ต่อ)

จากภาพที่ 4.20 โครงสร้างไฟล์ questionnaire.component.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 - 2 เป็นการนำข้อมูลที่ได้จาก resultModel ส่งค่าไปให้ server ที่ port 8000 ทำงานต่อในส่วนของการ predict ข้อมูล

- บรรทัดที่ 4 - 8 เป็นการนำเรียกข้อมูลได้จากจากฝั่ง server มาเก็บไว้ใน resultmodels โดยจะเก็บผลลัพธ์ของการกรอกแบบสอบถามไว้ใน result เก็บ ข้อมูลผลลัพธ์การ predict ไว้ที่ number
- บรรทัดที่ 9 เป็นการเรียกใช้ service ของ sendpredict เพื่อที่จะเอาค่าข้อมูลที่เก็บไว้ใน resultmodels ไปบันทึกไว้ที่ sendpredict
- บรรทัดที่ 10 เป็นการเรียกใช้ dialog.open เพื่อเปิดหน้าต่างของ DialogResultComponent มาแสดงผลการแนะนำ

4.11 การพัฒนาในส่วนของ Server ที่ใช้ในการ train model

เมื่อผู้ดูแลระบบกดปุ่ม train จะมีกระบวนการทำงาน แสดงดังรูปที่ 4.21 - 4.23

รูปที่ 4.21: ขั้นตอนกระบวนการ train model

จากภาพที่ 4.21 โครงสร้างของไฟล์ app.js อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เมื่อผู้ดูแลระบบกดปุ่ม train ก็จะมาเรียกใช้งานที่ path:"api/train"
- บรรทัดที่ 2 เป็นการกำหนดเวลาเริ่มต้นการเทรน
- บรรทัดที่ 3 เป็นการค้นหาข้อมูลจากฐานข้อมูลของ model เพื่อนำข้อมูลที่ใช้กรอกแบบประเมินมาทำการ train
- บรรทัดที่ 4 - 25 เป็นการใช้ .map ในการวนพอค้นหาข้อมูลทั้งหมดที่มีอยู่ในฐานข้อมูลและทำการแยกค่าของ xTrain yTrain โดย xTrain จะเก็บค่าของ q1 - q16 ส่วน yTrain จะเก็บค่าของ q17

รูปที่ 4.22: ขั้นตอนกระบวนการ train model(ต่อ)

จากภาพที่ 4.22 โครงสร้างของไฟล์ app.js อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 - 9 เริ่ม load ข้อมูล xTrain , yTrain โดยค่าของ xTrain จะถูกเก็บไว้ที่ XTrain และค่าของ yTrain จะถูกเก็บไว้ที่ YTrain ซึ่งค่าของ yTrain จะมี 0,1,2,3,4,5
- บรรทัดที่ 10 เป็นการสร้าง model จาก tf.sequential()
- บรรทัดที่ 11 - 24 เป็นการเพิ่ม layer sigmoid และ layer softmax
- บรรทัดที่ 25 เป็นการแสดง summary ของ model

รูปที่ 4.23: ขั้นตอนกระบวนการ train model(ต่อ)

จากภาพที่ 4.23 โครงสร้างของไฟล์ app.js สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 - 7 เป็นการเริ่ม train model โดยใช้ adam ซึ่งค่า learningRate ที่ใช้ 0.01
- บรรทัดที่ 8 เป็นการประกาศ dataAcc มาเก็บค่าของ accuracy
- บรรทัดที่ 9 เป็นการเพิ่มรอบในการเทรน โดยจะเทรนทั้งหมด 1000 รอบ
- บรรทัดที่ 10 เป็นการสิ้นสุดการเทรนได้โมเดล
- บรรทัดที่ 11 เป็นการเก็บเวลาสิ้นสุดในการเทรน
- บรรทัดที่ 12 เป็นการเก็บค่า accuracy มาไว้ใน array dataAcc
- บรรทัดที่ 13 เป็นการหาค่าสูงสุดของ accuracy
- บรรทัดที่ 14 - 16 เป็นการนำโมเดลที่ได้มาบันทึกเป็นไฟล์ json ไว้ที่โฟลเดอร์ models
- บรรทัดที่ 17 - 18 เป็นการแสดงเออเรอในกรณีที่บันทึกไฟล์โมเดลไม่สำเร็จ

4.12 การพัฒนาในส่วนของ Server ที่ใช้ในการ predict ข้อมูล
เมื่อผู้ใช้กดปุ่มส่งข้อมูล จะมีกระบวนการทำงาน แสดงดังรูปที่ 4.24

รูปที่ 4.24: ขั้นตอนกระบวนการ predict ข้อมูล

จากภาพที่ 4.24 โครงสร้างของไฟล์ app.js สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เมื่อผู้ใช้กดปุ่มส่งข้อมูล ก็จะมาเรียกใช้งานที่ path:"api/predict" เพื่อนำข้อมูลที่ส่งให้ฝั่ง server ทำการประมวลผลให้
- บรรทัดที่ 2 เป็นการประกาศ result ขึ้นมาเพื่อมาเก็บข้อมูลแบบประเมินที่ผู้ใช้กรอกเข้ามาในฝั่งของ Client ซึ่งค่าที่ส่งมาเป็น string เราจะต้องมาแปลงให้เป็นตัวเลขก่อนโดยใช้คำสั่ง parseInt()
- บรรทัดที่ 3 เป็นการนำเอา result มาทำให้เป็นข้อมูล 2 มิติ และเก็บไว้ที่ xTest
- บรรทัดที่ 4 - 6 เป็นการโหลดไฟล์โมเดลมาใช้งานในการทำนายผลลัพธ์
- บรรทัดที่ 7 เป็นการนำเอาตัวแปร xTest มาทำการ predict ข้อมูลและเก็บไว้ที่ predictOut
- บรรทัดที่ 8 - 9 เป็นการเรียกใช้ predictOut มาทำการวิเคราะห์ผลลัพธ์การแนะนำโดยค่าที่ได้จะออกมาเป็นตัวเลขทศนิยมซึ่งจะต้องนำค่ามาคิดเป็นเปอร์เซ็นต์และเก็บไว้ที่ตัวแปร category
- บรรทัดที่ 10 - 15 เป็นการนำค่าของ category มาค้นหาและไป push เก็บไว้ในรูปแบบ label และ value ซึ่ง label จะเก็บชื่อของหมวดและค่าเปอร์เซ็นต์ ส่วน value จะเก็บตำแหน่งของชื่อหมวด และเก็บไว้ใน dataShow
- บรรทัดที่ 16 - 18 เป็นการนำเอาค่าข้อมูลที่เก็บไว้ใน dataShow มาทำการจัดอันดับซึ่งเรียงจากค่าที่มีเปอร์เซ็นต์มากที่สุดไปน้อยสุด
- บรรทัดที่ 19 - 25 เป็นการส่งค่าผลลัพธ์ที่ได้กลับไปให้ฝั่ง Client เพื่อทำการแสดงผลข้อมูลการแนะนำให้กับผู้ใช้ได้ทราบ

บทที่ 5

การทดสอบระบบ

หลังจากที่ได้ผ่านขั้นตอนสำหรับการพัฒนาระบบทั้งหมดแล้ว ขั้นตอนต่อไปคือการทดสอบระบบเพื่อตรวจสอบการทำงานของแต่ละฟังก์ชันกันรวมถึงตรวจสอบความถูกต้องของการทำงานของระบบเพื่อให้การพัฒนาโปรแกรมเป็นไปอย่างมีประสิทธิภาพ ทดสอบระบบเพื่อหาข้อผิดพลาดและแก้ไขข้อบกพร่องนั้นก่อนการใช้งานจริง โดยในการทดสอบระบบจะแบ่งเป็น 2 ส่วน ได้แก่ การทดสอบในส่วนฟังก์ชันของระบบ และการทดสอบประสิทธิภาพของโมเดล

5.1 การทดสอบในส่วนฟังก์ชันของระบบ

5.1.1 ผลการทดสอบการสมัครสมาชิก

ตารางที่ 5.1: ผลการทดสอบการสมัครสมาชิก

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบการสมัครสมาชิก	ผู้ใช้งานเข้ามาในหน้าสมัครสมาชิก	ระบบแสดงหน้าสมัครสมาชิก
	ผู้ใช้งานกดปุ่มเลือกรูปภาพ	ระบบ แสดง หน้า ให้ เลือก รูป-ภาพ
	ผู้ใช้ กด ปุ่ม สมัคร สมาชิก โดยกรอกชื่อซ้ำกับชื่อที่มีอยู่ในระบบ	มี ข้อความ แสดง บอก ผู้ใช้ ว่า "username นี้มีผู้ใช้ไปแล้ว"
	ผู้ใช้ กด ปุ่ม สมัคร สมาชิก โดยกรอกชื่อไม่ซ้ำกับชื่อที่มีอยู่ในระบบ	มี ข้อความ แสดง บอก ผู้ใช้ ว่า "สมัครสมาชิกสำเร็จ"

5.1.2 ผลการทดสอบการเข้าสู่ระบบ

ตารางที่ 5.2: ผลการทดสอบการเข้าสู่ระบบ

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบการเข้าสู่ระบบ	ผู้ใช้เข้ามาในหน้าเข้าสู่ระบบ	ระบบแสดงหน้าเข้าสู่ระบบ
	ผู้ใช้ กด ปุ่ม เข้าสู่ ระบบ โดย ไม่ กรอก username และ password	ระบบแสดงข้อความบอกผู้ใช้ ว่า ”กรุณากรอก username และ password ก่อนกดปุ่ม เข้าสู่ระบบ”
	ผู้ใช้กดเข้าสู่ระบบโดยกรอก username ไม่ถูกต้อง	ระบบแสดงข้อความบอกผู้ใช้ ว่า ”เข้าสู่ระบบไม่สำเร็จ ชื่อ ผู้ใช้ หรือ รหัสผ่านไม่ตรงกัน กรุณาเข้าสู่ระบบใหม่อีกครั้ง”
	ผู้ใช้กดเข้าสู่ระบบโดยกรอก username ถูกต้องแต่กรอก password ไม่ถูกต้อง	ระบบแสดงข้อความบอกผู้ใช้ ว่า ”เข้าสู่ระบบไม่สำเร็จ ชื่อ ผู้ใช้ หรือ รหัสผ่านไม่ตรงกัน กรุณาเข้าสู่ระบบใหม่อีกครั้ง”
	ผู้ใช้กดเข้าสู่ระบบโดยกรอก username และ password ถูกต้อง	ระบบแสดงข้อความบอกผู้ใช้ ว่า ”เข้าสู่ระบบสำเร็จ” และ จะแสดงชื่อผู้ใช้รูปโปรไฟล์ที่ แลบบนด้านขวา

5.1.3 ผลการทดสอบหน้าจัดอันดับสถานที่ยอดนิยม

ตารางที่ 5.3: ผลการทดสอบหน้าจัดอันดับสถานที่ยอดนิยม

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบหน้าจัดอันดับสถานที่ยอดนิยม	ผู้ใช้กดเลือกเมนูสถานที่ยอดนิยม	ระบบ แสดงหน้า สถานที่ยอดนิยม โดย จะมี หมวด ให้ผู้ใช้เลือกอยู่ 3 หมวด ได้แก่ สถานที่ท่องเที่ยว ร้านอาหารและที่พัก
	ผู้ใช้กดเลือกหมวด	ระบบ แสดงผล การ จัด อันดับของหมวดที่ผู้ใช้เลือก
	ผู้ใช้คลิกที่ชื่อสถานที่	ระบบ เปลี่ยนหน้าเว็บ ไป หน้าแสดงรายละเอียดข้อมูลสถานที่

5.1.4 ผลการทดสอบหน้าหมวดที่เที่ยว

ตารางที่ 5.4: ผลการทดสอบหน้าหมวดที่เที่ยว

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบหน้าหมวดที่เที่ยว	ผู้ใช้กดเลือกเมนูที่เที่ยว	ระบบแสดงหน้าหมวดที่เที่ยว ซึ่งประกอบด้วย 6 หมวดได้แก่ หมวดนำตก หมวดสวนน้ำและสวนสัตว์ หมวดสถานที่บันเทิง หมวดอุทยาน และธรรมชาติ หมวดถนนคนเดิน และสุดท้าย หมวดสถานบันเทิง
	ผู้ใช้ กด เลือก ประเภท แต่ไม่กรอกชื่อสถานที่ แล้ว กด ปุ่ม ค้นหา	ระบบแสดงข้อความแจ้งผู้ใช้ว่า ”กรุณากรอกชื่อสถานที่ ก่อนกดปุ่มค้นหา”
	ผู้ใช้ กด เลือก ประเภท และกรอกชื่อสถานที่ แล้ว กด ปุ่ม ค้นหา	ระบบแสดงหน้าข้อมูลสถานที่ที่ผู้ใช้ทำการค้นหา
	ผู้ใช้กดเลือกหมวด	ระบบเปลี่ยนหน้าเว็บไปหน้าแสดง ราย ชื่อ สถานที่ ตามหมวดหมู่ที่ผู้ใช้เลือก

5.1.5 ผลการทดสอบหน้าหมวดที่กิน

ตารางที่ 5.5: ผลการทดสอบหน้าหมวดที่กิน

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบหน้าหมวดที่กิน	ผู้ใช้กดเลือกเมนูที่กิน	ระบบแสดงหน้าหมวดที่กินซึ่งประกอบด้วย 6 หมวดได้แก่ หมวดบุฟเฟต์ปิ้งย่าง หมวดคาเฟ่ หมวดเส้น หมวดอาหารท้องถิ่น หมวดอาหารญี่ปุ่น และสุดท้ายหมวดของเด็ก
	ผู้ใช้ กด เลือก ประเภท แต่ไม่กรอก ชื่อ สถานที่ แล้ว กด ปุ่ม ค้นหา	ระบบแสดงข้อความแจ้งผู้ใช้งานว่า "กรุณากรอกชื่อสถานที่ก่อนกดปุ่มค้นหา"
	ผู้ใช้ กด เลือก ประเภท และกรอก ชื่อ สถานที่ แล้ว กด ปุ่ม ค้นหา	ระบบแสดงหน้าข้อมูลสถานที่ที่ผู้ใช้ทำการค้นหา
	ผู้ใช้กดเลือกหมวด	ระบบเปลี่ยนหน้าเว็บไปหน้าแสดง ราย ชื่อ สถานที่ ที่ ตามหมวดที่ผู้ใช้เลือก

5.1.6 ผลการทดสอบหน้าหมวดที่พัก

ตารางที่ 5.6: ผลการทดสอบหน้าหมวดที่พัก

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบหน้าหมวดที่พัก	ผู้ใช้งานเลือกเมนูที่พัก	ระบบ แสดง หน้า หมวด ที่พัก ซึ่งประกอบด้วย 6 หมวดได้แก่ หมวดที่พักวิลล่า หมวดที่พักโรงแรม หมวดที่พักรีสอร์ท หมวดที่พักกางเต้นท์ หมวดที่พักบังกะโล/เกสท์เฮาส์ และสุดท้ายหมวดที่พักโฮมสเตย์
	ผู้ใช้งาน กด เลือก ประเภท แต่ไม่กรอก ชื่อ สถานที่ แล้ว กด ปุ่ม ค้นหา	ระบบ แสดง ข้อความ แจ้ง ผู้ใช้ ว่า ”กรุณา กรอก ชื่อ สถานที่ ก่อนกดปุ่มค้นหา”
	ผู้ใช้งาน กด เลือก ประเภท และกรอก ชื่อ สถานที่ แล้ว กด ปุ่ม ค้นหา	ระบบแสดงหน้าข้อมูลสถานที่ที่ผู้ใช้งานทำการค้นหา
	ผู้ใช้งานเลือกหมวด	ระบบเปลี่ยนหน้าเว็บไปหน้าแสดง ราย ชื่อ สถานที่ ตามหมวดหมู่ที่ผู้ใช้งานเลือก

5.1.7 ผลการทดสอบหน้าแสดงสถานที่

ตารางที่ 5.7: ผลการทดสอบหน้าแสดงสถานที่

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบหน้าแสดงสถานที่	รับข้อมูลที่ส่งจากหน้าหมวดที่ เที่ยว ที่กิน ที่พัก	ระบบแสดงหน้าสถานที่
	ผู้ใช้กดปุ่มเพิ่มสถานที่ ในกรณี ที่ไม่ได้เข้าสู่ระบบ	ระบบ แสดงข้อความ แจ้งผู้ใช้ ว่า ”กรุณาล็อกอินเข้าสู่ระบบ ก่อนกดปุ่มเพิ่มสถานที่”
	ผู้ใช้กดปุ่มเพิ่มสถานที่ ในกรณี เข้าสู่ระบบสำเร็จ	ระบบ แสดง กลุ่ม ข้อความ หน้าเพิ่มสถานที่
	ผู้ใช้กดเลือกสถานที่ที่สนใจ	ระบบเปลี่ยนหน้าเว็บไปแสดง หน้ารายละเอียดข้อมูล สถานที่ ที่

5.1.8 ผลการทดสอบหน้าแสดงรายละเอียดสถานที่

ตารางที่ 5.8: ผลการทดสอบหน้าแสดงรายละเอียดสถานที่

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบ หน้า แสดง รายละเอียดสถานที่	รับ ข้อมูล ที่ ส่ง มา จาก หน้า แสดงสถานที่	ระบบ แสดง หน้า รายละเอียดสถานที่
	ผู้ใช้งานพิมพ์เขียนรีวิว ในกรณีที่ ไม่ได้เข้าสู่ระบบ	ระบบ จะ แสดง ข้อความ แจ้ง ผู้ใช้ ว่า "กรุณา ล็อก อิน เข้าสู่ ระบบก่อนกดปุ่มรีวิว"
	ผู้ใช้งานพิมพ์เขียนรีวิว ในกรณีที่ เข้าสู่ระบบสำเร็จ	ระบบ จะ แสดง ป๊อปอัพ หน้า เขียนรีวิวขึ้นมาให้ผู้ใช้เขียนรีวิว
	ผู้ใช้งานกดปุ่มบันทึก ในกรณีที่ ไม่ได้เข้าสู่ระบบ	ระบบ จะ แสดง ข้อความ แจ้ง ผู้ใช้ ว่า "กรุณา ล็อก อิน เข้าสู่ ระบบก่อนกดปุ่มบันทึกสถานที่"
	ผู้ใช้งานกดปุ่มบันทึก ในกรณีที่ เข้าสู่ระบบสำเร็จ	ผู้ใช้สามารถกดบันทึกสถานที่ ได้และระบบจะนำข้อมูลการ บันทึกไปแสดงที่หน้าโปรไฟล์ ของผู้ใช้
	ผู้ใช้งานกดปุ่มแจ้งแก้ไข ในกรณีที่ ไม่ได้เข้าสู่ระบบ	ระบบ จะ แสดง ข้อความ แจ้ง ผู้ใช้ ว่า "กรุณา ล็อก อิน เข้าสู่ ระบบก่อนกดปุ่มแจ้งแก้ไข"
	ผู้ใช้งานกดปุ่มแจ้งแก้ไข ในกรณีที่ เข้าสู่ระบบสำเร็จ	ระบบ จะ แสดง ป๊อปอัพ หน้า แจ้งแก้ไขข้อมูลสถานที่ขึ้นมา ให้ ผู้ใช้ กรอก ข้อมูล เพื่อ แจ้งแก้ไข

5.1.9 ผลการทดสอบหน้าเขียนรีวิว

ตารางที่ 5.9: ผลการทดสอบหน้าเขียนรีวิว

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบ การ เขียน รีวิว และให้ คะแนนสถานที่	ผู้ใช้ กด ปุ่ม เขียน รีวิว ที่ หน้า แสดงรายละเอียดสถานที่ ใน กรณีที่ไม่ได้เข้าสู่ระบบ	ระบบจะแสดงข้อความแจ้งผู้ ใช้ ว่า ”กรุณา ล็อก อิน เข้า สู่ ระบบก่อนกดปุ่มรีวิว”
	ผู้ใช้ กด ปุ่ม เขียน รีวิว ที่ หน้า แสดงรายละเอียดสถานที่ ใน กรณีที่เข้าสู่ระบบสำเร็จ	ระบบจะแสดงกล่องข้อความ เขียนรีวิวขึ้นมาให้ผู้ใช้เขียนรี- วิว
	ผู้ใช้ให้คะแนน กรอกหัวข้อรี- วิว กรอกรายละเอียดแต่ไม่ได้ เลือกรูปภาพ และ กด ปุ่ม บั- นทึก รีวิว	ระบบบันทึกข้อมูลที่รีวิวลงใน ฐานข้อมูลของสถานที่
	ผู้ใช้ให้คะแนน กรอกหัวข้อรี- วิว กรอกรายละเอียด เลือก รูปภาพและกดปุ่มบันทึก รีวิว	ระบบบันทึกข้อมูลที่รีวิวลงใน ฐานข้อมูลของสถานที่
	ผู้ใช้กดปุ่มยกเลิก	ระบบจะปิดกล่องข้อความ

5.1.10 ผลการทดสอบหน้ากรอกแบบประเมิน

ตารางที่ 5.10: ผลการทดสอบหน้ากรอกแบบประเมิน

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบ หน้า กรอก แบบ ประ- เมิน	ผู้ใช้เลือกเมนูแนะนำสถานที่	ระบบ แสดง หน้า กรอก แบบ ประเมิน
	ผู้ใช้ ทำ แบบ ประเมิน ไม่ ครบ ทุกข้อ และกดปุ่มส่งข้อมูล	ระบบ จะ แสดง ข้อความ ตัว หนังสือสีแดงตรงข้อที่ผู้ใช้ข้าม และ บังคับ ให้ กรอก ข้อมูล ให้ ครบ
	ผู้ใช้ ทำ แบบ ประเมิน ครบ ทุก ข้อ และกดปุ่มส่งข้อมูล	ผู้ใช้สามารถกดปุ่มส่งข้อมูลได้

5.1.11 ผลการทดสอบหน้าแสดงผลการแนะนำหมวดท่องเที่ยว

ตารางที่ 5.11: ผลการทดสอบหน้าแสดงผลการแนะนำหมวดท่องเที่ยว

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบ การ ทำ แบบ ประเมิน เพื่อ ดู ผลลัพธ์ การ แนะนำ หมวดท่องเที่ยว	ผู้ใช้ ทำ แบบ ประเมิน ไม่ ครบ ทุกข้อ และกดปุ่มส่งข้อมูล	ระบบ จะ แสดง ข้อความ ตัวหนังสือสีแดงตรงข้อที่ผู้ใช้ข้าม และ บังคับ ให้ กรอก ข้อมูล ให้ ครบ
	ผู้ใช้ ทำ แบบ ประเมิน ครบ ทุก ข้อ และกดปุ่มส่งข้อมูล	ผู้ใช้ สามารถ กด ปุ่ม ส่ง ข้อมูล ได้และระบบจะแสดงกล่องข้อความ ผลลัพธ์ การ แนะนำ หมวดท่องเที่ยว
	ผู้ใช้งานเลือกหมวดท่องเที่ยวที่ชื่นชอบและกดปุ่มใช้	ระบบ จะบันทึก ข้อมูล การ ทำ แบบ ประเมิน ของ ผู้ใช้ ลง ใน ฐาน ข้อมูล และ ทำการ ปิด กล่อง แสดง ผลลัพธ์ การ แนะนำลง
	ผู้ใช้งานกดปุ่มยกเลิก	ระบบ จะ ปิด กล่อง แสดง ผล-ลัพธ์การแนะนำลง

5.1.12 ผลการทดสอบหน้าเพิ่มสถานที่

ตารางที่ 5.12: ผลการทดสอบหน้าเพิ่มสถานที่

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบการเพิ่มข้อมูลสถานที่ในส่วนของผู้ใช้	ผู้ใช้งานกดปุ่มเพิ่มสถานที่ ในกรณีที่ไม่ได้เข้าสู่ระบบ	ระบบ แสดงข้อความแจ้งผู้ใช้ว่า ”กรุณาล็อกอินเข้าสู่ระบบก่อนกดปุ่มเพิ่มสถานที่”
	ผู้ใช้งานกดปุ่มเพิ่มสถานที่ ในกรณีที่ได้เข้าสู่ระบบสำเร็จ	ระบบ แสดง กล่อง ข้อความ หน้าเพิ่มสถานที่
	ใน กรณี ที่ ผู้ ใช้ ไม่ ได้ เลือกประเภท ของ สถานที่ และ ผู้ใช้กรอกข้อมูลชื่อสถานที่ ที่ อยู่ รายละเอียดสถานที่ เบอร์โทรศัพท์ เว็บไซต์ (website) เฟส บุ๊ค(facebook) ละติจูด(latitude) ลองจิจูด(longitude) และ เลือก รูปภาพ แล้วกดปุ่มบันทึก	การ เพิ่ม ข้อมูล สถานที่ ที่ ล้มเหลว เนื่องจาก ระบบ ไม่สามารถ ทราบ ได้ ว่า ข้อมูลที่ ผู้ใช้เพิ่ม เข้ามา เป็น ข้อมูลประเภทไหน
	ใน กรณี ที่ ผู้ใช้ เลือก ประเภท ของสถานที่ และ ผู้ใช้ กรอกข้อมูล ชื่อ สถานที่ ที่ อยู่ รายละเอียด สถานที่ ที่ เบอร์โทรศัพท์ เว็บไซต์ (website) เฟส บุ๊ค(facebook) ละติจูด(latitude) ลองจิจูด(longitude) และ เลือก รูปภาพ แล้วกดปุ่มบันทึก	ระบบจะบันทึกข้อมูลการเพิ่มสถานที่ลง ในฐานข้อมูล ตามประเภทที่ผู้ใช้เลือก
	ผู้ใช้งานกดปุ่มยกเลิก	ระบบ จะ ปิด กล่อง ข้อความ หน้าเพิ่มสถานที่ลง

5.1.13 ผลการทดสอบการแสดงผลหน้าโปรไฟล์

ตารางที่ 5.13: ผลการทดสอบการแสดงผลหน้าโปรไฟล์

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบ การ แสดง ข้อมูล โปร- ไฟล์	ผู้ใช้กดเลือกเมนูโปรไฟล์	ระบบ แสดง หน้า ข้อมูล โปร- ไฟล์ของผู้ใช้
	ผู้ใช้กดปุ่มแก้ไขโปรไฟล์	ระบบ จะ แสดง กล่องข้อความ หน้าแก้ไขโปรไฟล์ซึ่งข้อมูลที่ แสดง ใน หน้า นั้น จะ มี รูป ประจำตัวของผู้ใช้ ชื่อผู้ใช้นี้ปุ่ม เปลี่ยนรหัสผ่าน ปุ่มเปลี่ยนรูป ประจำตัว และปุ่มบันทึก
	ผู้ ใช้ กด เลือก ที่ บันทึก ไว้ ใน กรณีที่ผู้ใช้ยังไม่มีข้อมูลสถานที่ที่บันทึกไว้	ระบบ จะ แสดง ข้อความ บน หน้าเว็บว่า ”ไม่มีกิจกรรมการ บันทึก”
	ผู้ ใช้ กด เลือก ที่ บันทึก ไว้ ใน กรณีที่ผู้ใช้มีข้อมูลสถานที่ที่ บันทึกไว้	ระบบจะแสดงข้อมูลสถานที่ที่ ผู้ใช้ได้บันทึกไว้
	ผู้ใช้กดเลือกกริวิ ในกรณีที่ผู้ใช้ ยังไม่มีข้อมูลการกริวิสถานที่	ระบบ จะ แสดง ข้อความ บน หน้าเว็บว่า ”ไม่มีกิจกรรมกริ- วิ”
	ผู้ใช้กดเลือกกริวิ ในกรณีที่ผู้ใช้ ยังมีข้อมูลการกริวิสถานที่	ระบบจะแสดงข้อมูลสถานที่ที่ ผู้ใช้เคยเขียนกริวิ
	ผู้ ใช้ กด เลือก สถานที่เพิ่ม ใน กรณีที่ผู้ใช้ยังไม่เคยเพิ่มข้อมูล สถานที่	ระบบ จะ แสดง ข้อความ บน หน้าเว็บว่า ”ไม่มีกิจกรรมเพิ่ม สถานที่”
	ผู้ ใช้ กด เลือก สถานที่เพิ่ม ใน กรณี ที่ ผู้ ใช้ เคย เพิ่ม ข้อมูล สถานที่	ระบบจะแสดงข้อมูลสถานที่ที่ ผู้ใช้ได้เพิ่ม

5.1.14 ผลการทดสอบการแก้ไขรหัสผ่าน

ตารางที่ 5.14: ผลการทดสอบการแก้ไขรหัสผ่าน

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบการแก้ไขรหัสผ่าน	ผู้ใช้กดปุ่มเปลี่ยนรหัสผ่าน	ระบบ แสดง หน้า เก ล่อง ข้อ- ความหน้าแก้ไขรหัสผ่าน
	ผู้ใช้ กรอก รหัส ผ่าน เก่า และ กรอก รหัส ผ่าน ใหม่ แล้ว กด ปุ่มบันทึกการเปลี่ยนแปลงใน กรณีที่ผู้ใช้กรอกรหัสผ่านเก่า ไม่ถูกต้อง	ระบบจะแสดงข้อความแจ้งผู้ ใช้ว่า ”เปลี่ยนรหัสผ่านไม่สำ- เร็จ เนื่องจากกรอกรหัสผ่าน เก่าไม่ถูกต้องและรหัสยืนยัน ไม่ตรงกันกับรหัสผ่านใหม่”
	ผู้ใช้ กรอก รหัส ผ่าน เก่า และ กรอก รหัส ผ่าน ใหม่ แล้ว กด ปุ่มบันทึกการเปลี่ยนแปลงใน กรณีที่ผู้ใช้กรอกรหัสผ่านเก่า ถูกต้อง	ระบบจะแสดงข้อความแจ้งผู้ ใช้ว่า ”เปลี่ยนรหัสผ่านสำเร็จ รหัสผ่านใหม่คือ : ”
	ในกรณีที่ผู้ใช้ กรอกรหัสผ่าน ใหม่และกรอกรหัสผ่านยืนยัน ไม่ตรงกัน แล้ว กดปุ่มบันทึก การเปลี่ยนแปลง	ระบบจะแสดงข้อความแจ้งผู้ ใช้ว่า ”เปลี่ยนรหัสผ่านไม่สำ- เร็จ เนื่องจากกรอกรหัสยืนยัน ไม่ตรงกันกับรหัสผ่านใหม่”
	ในกรณีที่ผู้ใช้ กรอกรหัสผ่าน ใหม่และกรอกรหัสผ่านยืนยัน ตรงกัน แล้วกดปุ่มบันทึกการ เปลี่ยนแปลง	ระบบจะแสดงข้อความแจ้งผู้ ใช้ว่า ”เปลี่ยนรหัสผ่านสำเร็จ รหัสผ่านใหม่คือ : ”

5.1.15 ผลการทดสอบลบสถานที่

ตารางที่ 5.15: ผลการทดสอบลบสถานที่

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบลบสถานที่ในกรณีที่ผู้ใช้เป็นคนเพิ่มสถานที่เอง	ผู้ใช้เข้าไปที่หน้าโปรไฟล์แล้ว กดเลือกสถานที่เพิ่ม หลังจากนั้นผู้ใช้คลิกสถานที่แล้วเลือกเมนูลบ	ระบบ แสดง กล่อง ข้อความถามผู้ใช้ว่า "คุณแน่ใจไหมว่าจะลบข้อมูลสถานที่นี้ทิ้ง" และมีปุ่มให้ผู้ใช้เลือกกดอยู่ 2 อย่างคือ แนใจ และ ไม่แนใจ
	ผู้ใช้กดปุ่มแนใจ	ระบบ ทำการ ลบ ข้อมูล สถานที่ที่ออกจากฐานข้อมูลและลบข้อมูลสถานที่ออกจากข้อมูลของผู้ใช้
	ผู้ใช้กดปุ่มไม่แนใจ	ระบบจะปิดกล่องข้อความลง

5.1.16 ผลการทดสอบลบรีวิว

ตารางที่ 5.16: ผลการทดสอบลบรีวิว

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบลบรีวิว ในกรณีเป็นรีวิวที่ตัวเองเป็นคนเขียน	ผู้ใช้เข้าไปที่หน้าโปรไฟล์แล้ว กดเลือกรีวิว หลังจากนั้นผู้ใช้คลิกสถานที่แล้วเลือกเมนูลบรีวิว	ระบบ แสดง กล่อง ข้อความถามผู้ใช้ว่า "คุณแน่ใจไหมว่าจะลบ ความคิดเห็น ของ คุณทิ้ง" และจะมีปุ่มให้ผู้ใช้เลือกกดอยู่ 2 อย่างคือ แนใจ และ ไม่แนใจ
	ผู้ใช้กดปุ่มแนใจ	ระบบ ทำการ ลบ รีวิว ของ ผู้ใช้คนนั้นออกจากข้อมูลสถานที่ในฐานข้อมูลและลบข้อมูลรีวิวนั้นออกจากข้อมูลของผู้ใช้
	ผู้ใช้กดปุ่มไม่แนใจ	ระบบจะปิดกล่องข้อความลงและข้อมูลรีวิวไม่ถูกลบ

5.1.17 ผลการทดสอบการจัดการแก้ไขข้อมูลสถานที่

ตารางที่ 5.17: ผลการทดสอบการจัดการแก้ไขข้อมูลสถานที่

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบการแก้ไขข้อมูลกรณีที่ผู้ดูแลระบบเข้าสู่ระบบสำเร็จ	ผู้ดูแลระบบกดปุ่มแก้ไข	ระบบแสดงหน้าแก้ไขข้อมูล
	ผู้ดูแลระบบกรอกข้อมูลแก้ไขสถานที่แล้วกดปุ่มบันทึกการแก้ไข	ระบบ บันทึก ข้อมูล การ แก้ไข และ อัปเดตข้อมูลลงฐานข้อมูล
	ผู้ดูแลระบบกดปุ่มยกเลิก	ระบบ ปิด หน้า แก้ไข ข้อมูล ลง และข้อมูลไม่ได้รับการแก้ไข

5.1.18 ผลการทดสอบการจัดการลบข้อมูลสถานที่

ตารางที่ 5.18: ผลการทดสอบการจัดการลบข้อมูลสถานที่

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบการลบข้อมูลสถานที่กรณีที่ผู้ดูแลระบบเข้าสู่ระบบสำเร็จ	ผู้ดูแลระบบกดปุ่มลบ	ระบบ แสดง ข้อความ ถาม ผู้ดูแลระบบ ว่า “คุณ ต้องการลบสถานที่นี้หรือไม่” และมีปุ่มตัวเลือกให้ผู้เลือกคือ ใช่ และไม่ใช่
	ผู้ดูแลระบบกดปุ่มใช่	ระบบ ลบ ข้อมูล สถานที่ ที่ ผู้ดูแลระบบกดลบออกจากฐานข้อมูล
	ผู้ดูแลระบบกดปุ่มไม่ใช่	ข้อมูล สถานที่ ที่ ไม่ ถูก ลบ ออกจากฐานข้อมูล

5.1.19 ผลการทดสอบการจัดการเพิ่มสถานที่

ตารางที่ 5.19: ผลการทดสอบการจัดการเพิ่มสถานที่

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบการเพิ่มสถานที่ กรณีที่ผู้ดูแลระบบเข้าสู่ระบบสำเร็จ	ผู้ดูแลระบบกดปุ่มเพิ่มสถานที่	ระบบ เปิด หน้า เพิ่ม ข้อมูลสถานที่
	ผู้ดูแลระบบ กรอกข้อมูล รายละเอียด สถานที่ ครบ และ กดปุ่มเพิ่มสถานที่	ระบบจะบันทึกข้อมูลรายละเอียดสถานที่ลงในฐานข้อมูลตามประเภทของสถานที่
	ผู้ดูแลระบบกดปุ่มยกเลิก	ระบบ จะ ปิด หน้า เพิ่ม ข้อมูลสถานที่ลงและข้อมูลจะไม่ถูกเพิ่มลงในระบบ

5.1.20 ผลการทดสอบจัดการข้อมูลแจ้งแก้ไข

ตารางที่ 5.20: ผลการทดสอบจัดการข้อมูลแจ้งแก้ไข

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบ การ แก้ไข ข้อมูล จากข้อมูลที่ใช้แจ้งแก้ไขมา กรณีที่ผู้ดูแลระบบเข้าสู่ระบบสำเร็จ	ผู้ดูแลระบบ เลือก เมนู แจ้งแก้ไข	ระบบ แสดง หน้า ข้อมูล แจ้งแก้ไข โดย จะ แสดง ชื่อ ผู้แจ้งแก้ไข ชื่อสถานที่ และปุ่มแก้ไข
	ผู้ดูแลระบบกดแก้ไข	ระบบ แสดง หน้า แก้ไข ข้อมูลสถานที่โดยจะมีข้อมูลที่ใช้กรอกเข้ามาเพื่อทำการแก้ไข
	ผู้ดูแลระบบกดปุ่มบันทึกการแก้ไข	ระบบทำการบันทึกข้อมูลการแก้ไข และ อัปเดต ข้อมูล การแก้ไขสถานที่ลงในฐานข้อมูล
	ผู้ดูแลระบบกดปุ่มยกเลิก	ข้อมูลจะไม่ถูกทำการแก้ไข

5.1.21 ผลการทดสอบการ Train Model

ตารางที่ 5.21: ผลการทดสอบการ Train Model

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบการ Train Model กรณีที่ ผู้ดูแล ระบบ เข้าสู่ ระบบสำเร็จ	ผู้ดูแล ระบบ กด เลือก เมนู Train	ระบบ ทำการ เรียก ใช้ เซิร์ฟเวอร์ในการ train model เมื่อ train model สำเร็จ ระบบ แจ้งข้อความบอกผู้ดูแลระบบ ว่า "train model สำเร็จ ได้ โมเดลแล้ว"

5.2 การทดสอบประสิทธิภาพของโมเดล

5.2.1 ผลการทดสอบความถูกต้องของโมเดล

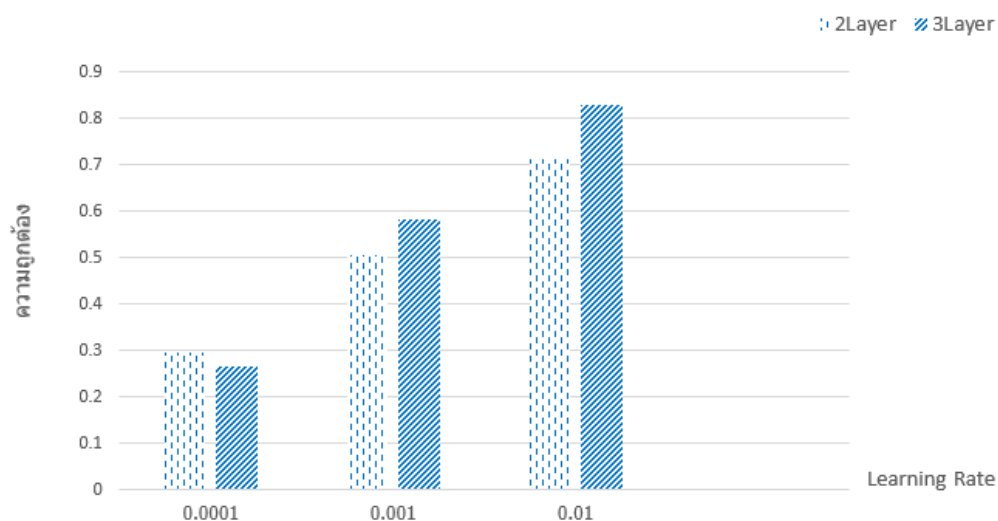
ในการทดสอบความถูกต้องของโมเดลเป็นการทดสอบหาค่าความถูกต้องซึ่งชุดข้อมูลที่ใช้ในการเทรนมีทั้งหมด 460 แถว จะทดสอบเทรนทั้งหมด 5 ครั้ง ต่อหนึ่งค่า Learning Rate ซึ่งในการทดสอบเทรนแต่ละครั้งจะเทรนทั้งหมด 1000 รอบ โดยจะปรับค่า Learning Rate เป็น 0.0001, 0.001, 0.01 และจะทดสอบปรับเลเยอร์ (Layer) เป็น 2 ชั้นเปรียบเทียบกับเลเยอร์ (Layer) 3 ชั้น เพื่อดูผลลัพธ์ค่าความถูกต้องและค่าความผิดพลาดของโมเดลที่ได้ สาเหตุที่ต้องเทรน 5 ครั้งต่อหนึ่งค่า Learning Rate เพราะข้อมูลที่ได้มาเป็นข้อมูลสุ่มทำให้ค่าความถูกต้องที่ได้มีค่าไม่แน่นอน เมื่อเทรนครบ 5 ครั้ง จะนำค่าผลลัพธ์ที่ได้มาหาค่าเฉลี่ยความถูกต้องของโมเดล

ตารางที่ 5.22: ผลลัพธ์การทดสอบเทรนโมเดล 5 ครั้งต่อหนึ่งค่า Learning Rate

จำนวน Layer	Learning Rate	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ค่าเฉลี่ย
2	0.0001	0.294	0.303	0.286	0.301	0.303	0.2974
	0.001	0.485	0.517	0.498	0.513	0.513	0.5052
	0.01	0.699	0.719	0.712	0.706	0.732	0.7136
3	0.0001	0.255	0.271	0.275	0.260	0.264	0.265
	0.001	0.613	0.593	0.623	0.535	0.558	0.5844
	0.01	0.812	0.827	0.825	0.831	0.855	0.83

ตารางที่ 5.23: ผลการทดสอบค่าความถูกต้องและค่าความผิดพลาดของโมเดล

จำนวน Layer	Learning Rate	ความถูกต้อง	ความผิดพลาด
2	0.0001	0.2974	0.7026
	0.001	0.5052	0.4948
	0.01	0.7136	0.2864
3	0.0001	0.265	0.735
	0.001	0.5844	0.4156
	<u>0.01</u>	<u>0.83</u>	<u>0.17</u>



รูปที่ 5.1: กราฟเปรียบเทียบค่า LearningRate

สรุปผลจากการทดสอบปรับค่า Learning Rate เป็น 0.0001, 0.001, 0.01 และได้ปรับเลเยอร์เป็น 2 ชั้นเปรียบเทียบกับเลเยอร์ 3 ชั้น ผลลัพธ์ที่ได้จากการทดสอบปรับค่า Learning Rate และปรับเลเยอร์ ซึ่งค่า Learning Rate ที่ให้ความถูกต้องดีที่สุดคือ 0.01 โดยใช้เลเยอร์ 3 ชั้น ค่าความถูกต้องที่ได้คือ 0.83 และมีค่าความผิดพลาดคือ 0.17

บทที่ 6

สรุปและข้อเสนอแนะ

การดำเนินโครงการเพื่อพัฒนาระบบจองคิวร้านเสริมสวยนี้ พบว่าระบบสามารถทำงานได้ตามที่วิเคราะห์และออกแบบไว้ แต่ก็พบปัญหาและอุปสรรคระหว่างการพัฒนา ในบทนี้ผู้พัฒนาจึงขอสรุปความสามารถของระบบชี้แจงปัญหาและอุปสรรค พร้อมเสนอแนวทางในการพัฒนาระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานีต่อ ตามลำดับ

6.1 สรุปความสามารถของระบบ

โดยแบ่งความสามารถของระบบตามประเภทของผู้ใช้งานดังนี้

1. เจ้าของร้าน

- สามารถลงทะเบียนเข้าสู่ระบบด้วย Email ได้
- สามารถดูคิวที่ผู้ใช้บริการได้ทำการจองคิวไว้
- สามารถจัดการคิวได้
- สามารถ post ภาพผลงานทั้งหมดของร้านได้
- สามารถเพิ่ม แก้ไข และลบรายการให้บริการประจำร้านได้
- สามารถเพิ่ม แก้ไข และลบ ข้อมูลร้านได้
- สามารถเพิ่ม แก้ไข และลบข้อมูลตำแหน่งร้านได้

2. ช่างประจำร้าน

- ลงทะเบียนใช้ web ด้วย Email ได้
- สามารถดูตารางการทำงานของตนเองได้
- สามารถ post ภาพผลงานของตัวเองได้
- สามารถแก้ไขข้อมูลส่วนตัวได้

3. ผู้ใช้บริการ

- สามารถลงทะเบียนเข้าสู่ระบบด้วย Email ได้
- สามารถค้นหาร้านเสริมสวยได้

- สามารถจองคิวของร้านเสริมสวยได้
- สามารถดูคิวว่างของร้านเสริมสวยได้
- สามารถดูข้อมูลต่างๆของร้านเสริมสวยได้
- สามารถดูตำแหน่งของทางร้านได้
- สามารถดูผลงานของร้านได้
- สามารถเขียนรีวิว ดีชม ได้

6.2 ปัญหาและอุปสรรคในการพัฒนา

1. ข้อมูลชุดฝึกหัดและชุดทดสอบมีจำนวนที่น้อย ทำให้ความถูกต้องของการทำนายมีความแม่นยำน้อย แนวทางการแก้ไข : เพิ่มชุดข้อมูลที่มีความสัมพันธ์กันเพื่อให้ผลการทำนายมีความแม่นยำมากยิ่งขึ้น
2. ผู้พัฒนาได้เขียน server สำหรับทำการ train ข้อมูล และเป็นที่สำหรับเก็บข้อมูลแม่แบบ (model) ที่ได้จากการเทรนข้อมูลออกมาเป็นไฟล์นามสกุล json ให้ฝั่ง client เรียกใช้งานต่อได้ แต่ปัญหาที่พบคือ Angular จะบล็อกการร้องขอเรียกใช้งานจากที่อื่น (cross origin request blocked) จึงทำให้ไม่สามารถเรียกข้อมูลมาใช้งานทางฝั่ง Client ได้
แนวทางการแก้ไข : เขียน server ขึ้นมาเพื่อให้ทำการ train ข้อมูลและทำนาย (predict) ข้อมูลให้เสร็จสิ้นทางฝั่ง server แล้วฝั่ง client มีหน้าที่ในการรอรับข้อมูลอย่างเดียว
3. การบันทึกรูปภาพจำนวนมากที่มีขนาดใหญ่เข้าไปในระบบพร้อมกันระบบจะไม่สามารถบันทึกข้อมูลได้
แนวทางการแก้ไข : ขยายการรับข้อมูลของKuzzleหรือลดขนาดไฟล์ก่อนจะส่งไปKuzzle

6.3 แนวทางการพัฒนาต่อ

1. เพิ่มส่วนติดต่อกับสังคมออนไลน์ เช่น Facebook Twitter และ Instagram
2. เพิ่มในส่วนของแผนที่การนำทางไปยังสถานที่ที่ต้องการ
3. เพิ่มในส่วนของผู้ใช้งานให้สามารถติดตาม (follow) ผู้ใช้งานคนอื่นได้
4. เพิ่มชุดข้อมูลและความหลากหลายของชุดข้อมูล
5. เพิ่มประสิทธิภาพการทำงานของระบบให้ดีขึ้น
6. เพิ่มในส่วนของฟังก์ชันแจ้งเตือนให้ผู้ดูแลระบบสามารถแก้ไขข้อมูลสถานที่ได้สะดวกขึ้น

บรรณานุกรม

- [1] Angular คืออะไร ทำความรู้จัก และวิธีใช้งาน [ออนไลน์]. สืบค้นเมื่อ เมษายน 2562. จาก <http://www.helloho.me/getting-started-with-angular/> .
- [2] การใช้งาน angular cli พื้นฐาน (ตอนที่ 1) [ออนไลน์]. สืบค้นเมื่อ เมษายน 2562. จาก <https://medium.com/open-source-technology/การใช้งาน-angular-cli-พื้นฐาน-ตอน-1-168af8ad10b0> .
- [3] Node.js คืออะไร ? + เริ่มต้นใช้งาน node.js [ออนไลน์]. สืบค้นเมื่อ เมษายน 2562. จาก <https://1th.me/24MG> .
- [4] Typescript คืออะไร? เรียนรู้ชนิดข้อมูลพื้นฐานของ typescript [ออนไลน์]. สืบค้นเมื่อ เมษายน 2562. จาก <https://www.babelcoder.com/blog/posts/typescript-data-types> .
- [5] รู้จักกับ visual studio code (วิซวล สตูดิโอ โค้ด) โปรแกรมฟรีจากค่ายไมโครซอฟท์ [ออนไลน์]. สืบค้นเมื่อ เมษายน 2562. จาก <https://www.mindphp.com/บทความ/microsoft/4829-visual-studio-code.html> .
- [6] ตอนที่ 1 : Google maps api (javascript) เบื้องต้นกับการใช้งานแผนที่บนของกูเกิลแมพ (step by step) [ออนไลน์]. สืบค้นเมื่อ เมษายน 2562. จาก <https://www.thaicreate.com/tutorial/google-maps-javascript-api.html> .
- [7] Tensorflow.js [ออนไลน์]. สืบค้นเมื่อ เมษายน 2562. จาก <https://www.tensorflow.org/js> .
- [8] Machine learning คืออะไร? [ออนไลน์]. สืบค้นเมื่อ เมษายน 2562. จาก <http://www.bossup.co.th/site/innovation/machine-learning-คืออะไร/> .
- [9] สรุปแนวคิด neural network แบบไม่มี math [ออนไลน์]. สืบค้นเมื่อ เมษายน 2562. จาก

[https:// coladev.com/ machine-learning/ neural-network/ 2017/02/22/ neural-network-basic](https://coladev.com/machine-learning/neural-network/2017/02/22/neural-network-basic) .

- [10] Deep learning คือ อะไร [ออนไลน์]. สืบค้นเมื่อ เมษายน 2562. จาก <https://medium.com/@athivvat/deep-learning-คืออะไร-785e16d01773> .
- [11] Wongnai [ออนไลน์]. สืบค้นเมื่อ เมษายน 2562. จาก <https://www.wongnai.com/about> .
- [12] ความหมายและความสำคัญของ system architecture [ออนไลน์]. สืบค้นเมื่อ เมษายน 2562. จาก <https://goo.gl/6ZhGQo> .

ภาคผนวก

ภาคผนวก ก

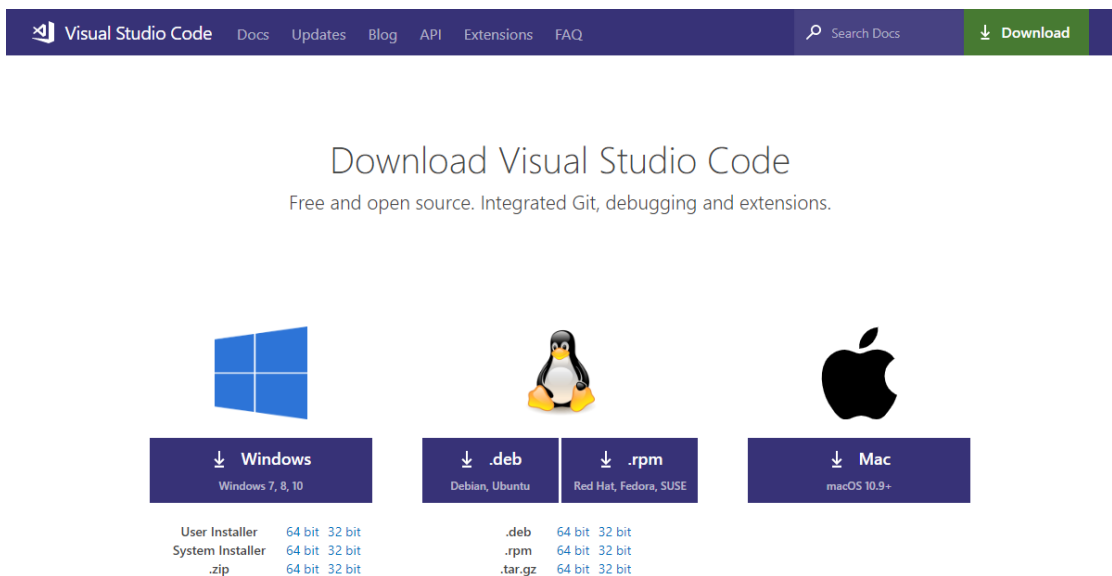
การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม

การติดตั้งเครื่องมือที่ใช้ในการพัฒนาเว็บแอปพลิเคชันระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานีมีโปรแกรมที่จำเป็นในการพัฒนาระบบดังต่อไปนี้

- การติดตั้ง Visual Studio Code
- การติดตั้ง Node.js
- การติดตั้ง Angular Fronted Framwork

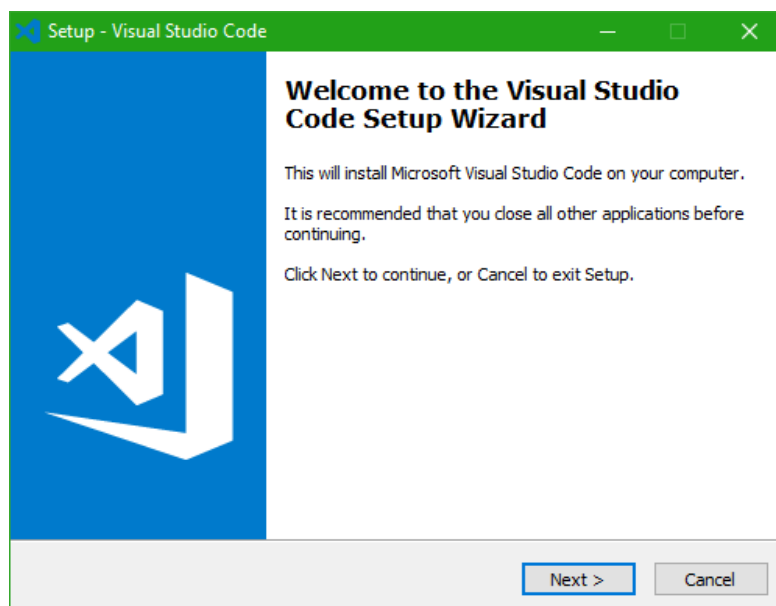
ก.1 การติดตั้ง Visual Studio Code

1. สามารถดาวน์โหลด Visual Studio Code ได้ที่ <https://code.visualstudio.com/download> ดังแสดงในรูปที่ ก.1



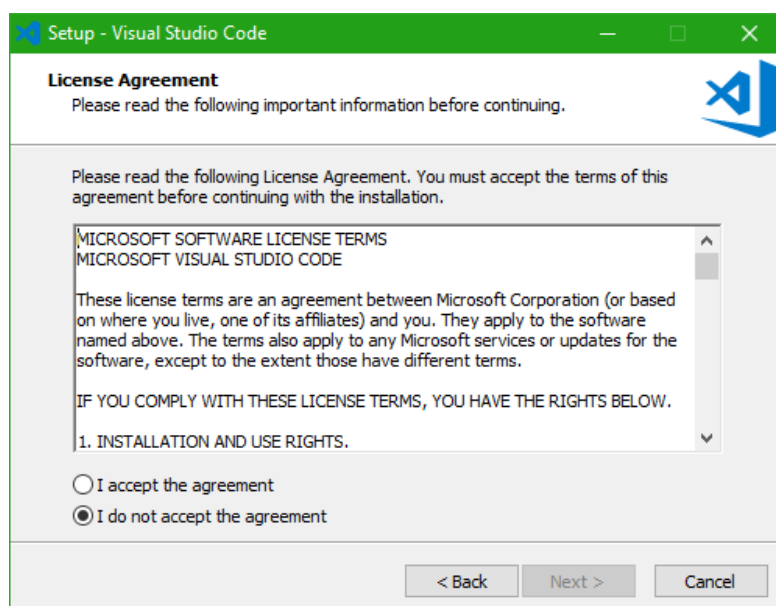
รูปที่ ก.1: หน้าเว็บดาวน์โหลด Visual Studio Code

2. เมื่อเปิดตัวติดตั้งขึ้นมาแล้ว จะแสดงหน้าจอ Welcome to the Visual Studio Code Setup Wizard ให้กดปุ่ม Next เพื่อเริ่มกระบวนการติดตั้ง ดังแสดงในรูปที่ ก.2



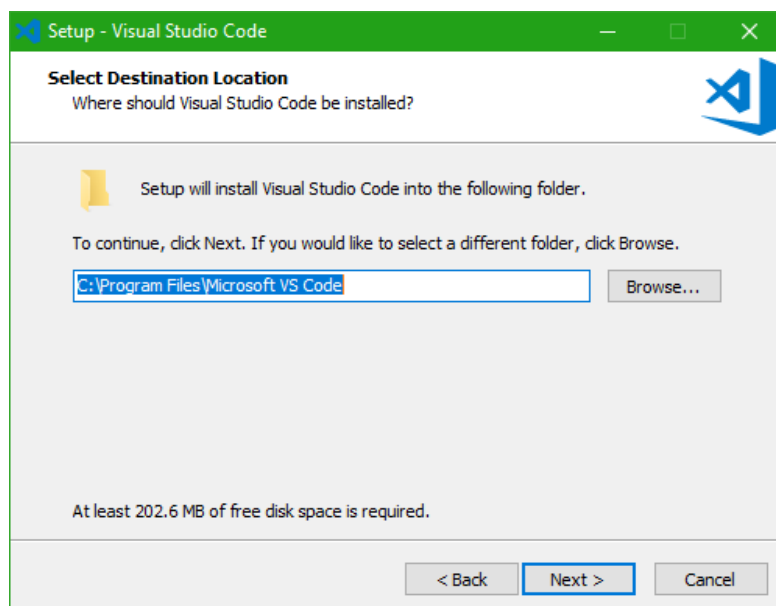
รูปที่ ก.2: หน้าต่างต้อนรับของ Visual Studio Code

3. หลังจากนั้นจะแสดงหน้าต่างข้อตกลงการใช้งาน Visual Studio Code ทำการติ๊กที่ I accept the areement แล้วกด Next ดังแสดงในรูปที่ ก.3



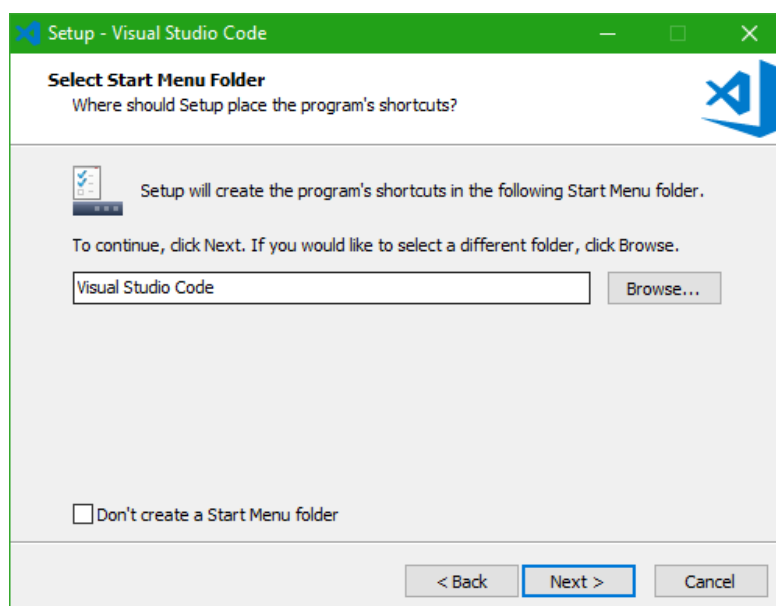
รูปที่ ก.3: หน้าต่างข้อตกลงการใช้งาน Visual Studio Code

4. จากนั้นจะแสดงหน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Visual Studio Code ทำการกด Next ดังแสดงในรูปที่ ก.4



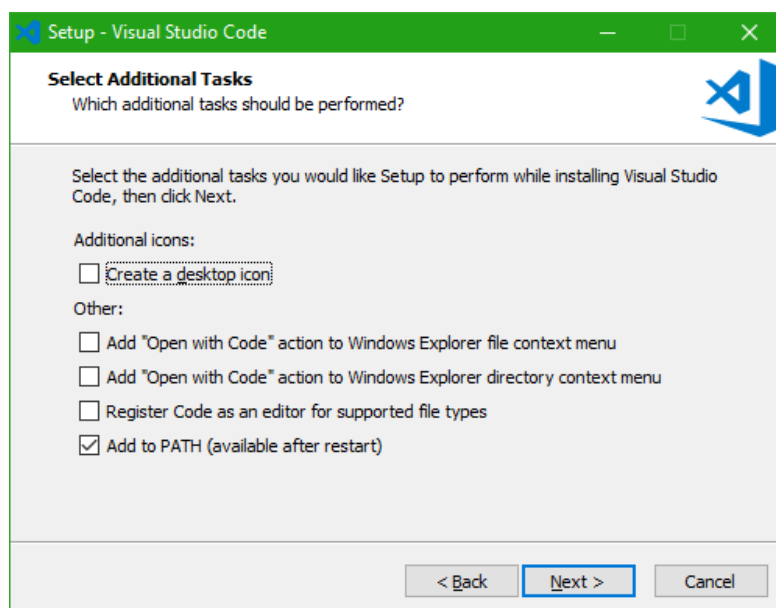
รูปที่ ก.4: หน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Visual Studio Code

5. จากนั้นจะแสดงหน้าต่างการจัดการซอร์คัทของ Visual Studio Code ทำการกด Next ดังแสดงในรูปที่ ก.5



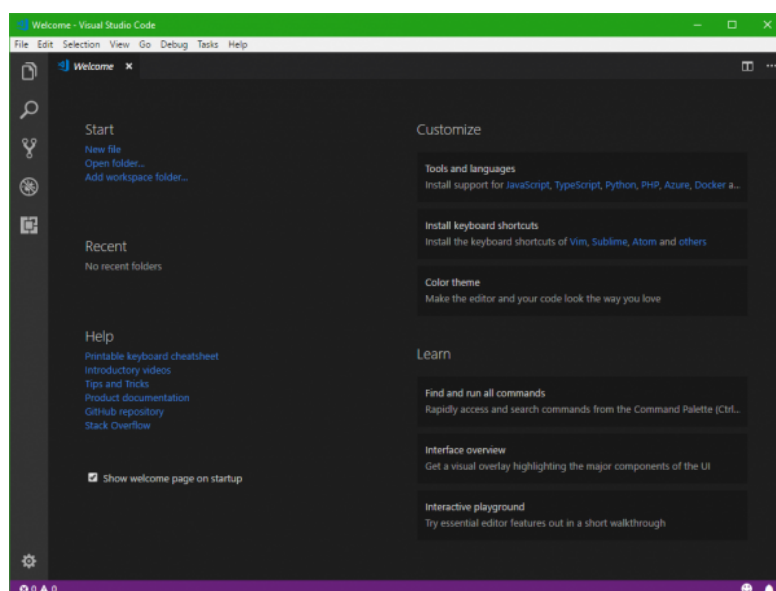
รูปที่ ก.5: หน้าต่างการจัดการซอร์คัท ของ Visual Studio Code

6. จากนั้นแสดงหน้าต่างเริ่มทำการติดตั้งทำการกด Next ดังแสดงในรูปที่ ก.6



รูปที่ ก.6: หน้าต่างเริ่มทำการติดตั้งทำการกด ของ Visual Studio Code

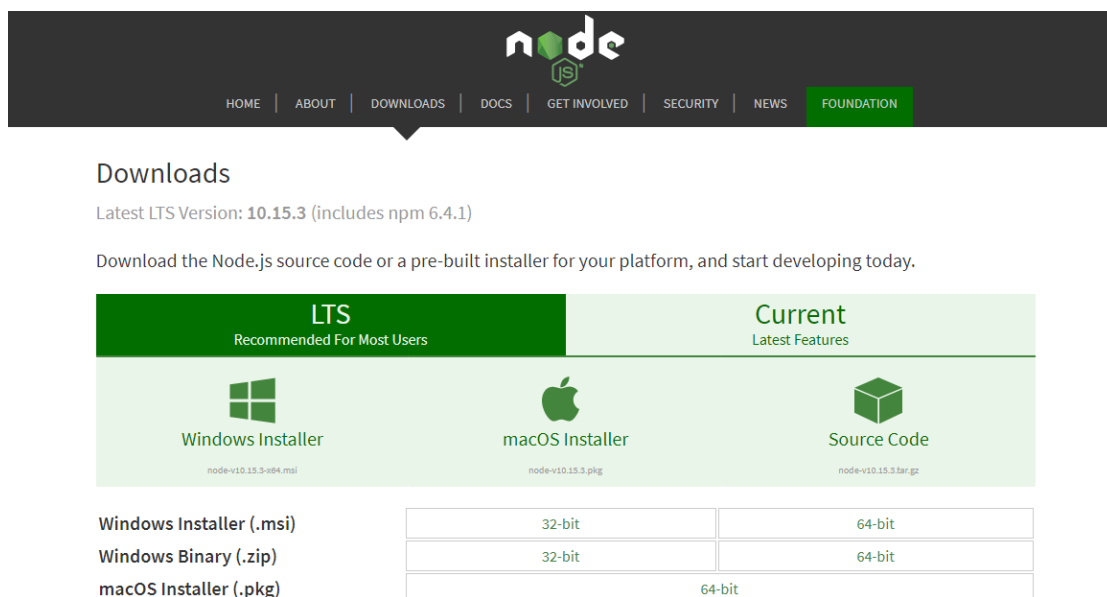
7. จากนั้นจะแสดงหน้าต่างเมื่อเข้าโปรแกรมหลังติดตั้งเสร็จ ดังแสดงในรูปที่ ก.7



รูปที่ ก.7: หน้าต่างเมื่อเข้าโปรแกรมหลังติดตั้งเสร็จ ของ Visual Studio Code

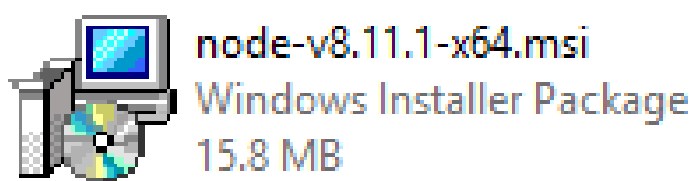
ก.2 การติดตั้ง Node.js

1. สามารถดาวน์โหลด Node.js installer package ได้ที่ <https://nodejs.org/en/download/> ดังแสดงในรูปที่ ก.8



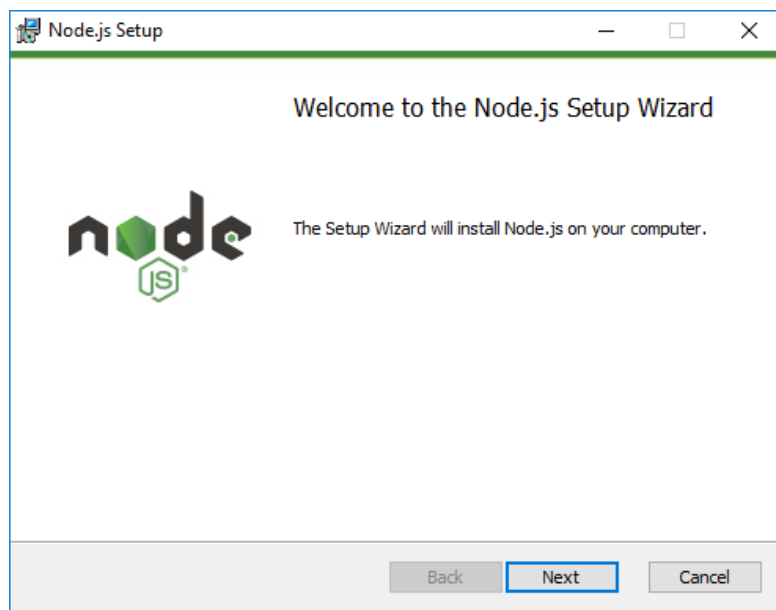
รูปที่ ก.8: หน้าเว็บดาวน์โหลด Node.js

2. เปิดไฟล์ติดตั้ง ชื่อ node-vx.xx.x-x64.msi เพื่อทำการติดตั้ง ดังแสดงในรูปที่ ก.9



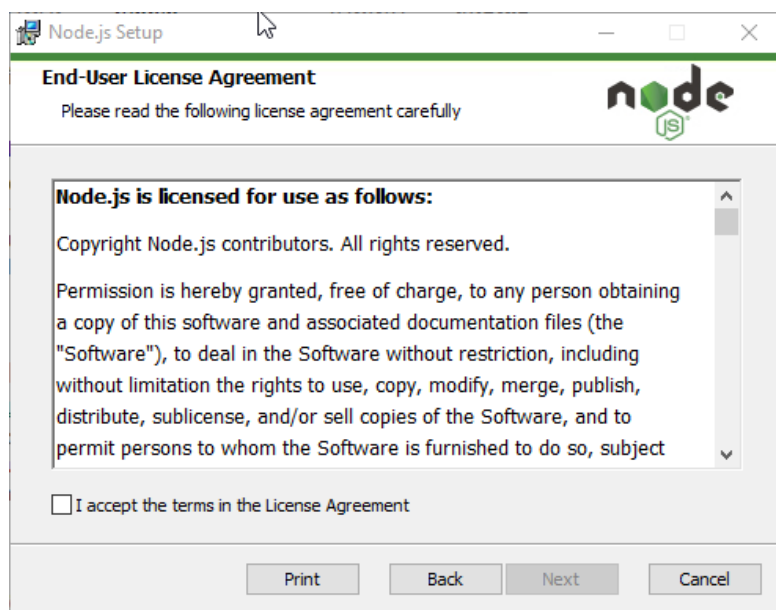
รูปที่ ก.9: ไฟล์ติดตั้งสำหรับติดตั้ง Node.js

3. แสดงหน้าต่างต้อนรับของ Node.js ทำการกด Next เพื่อเริ่มกระบวนการติดตั้ง ดังแสดงในรูปที่ ก.10



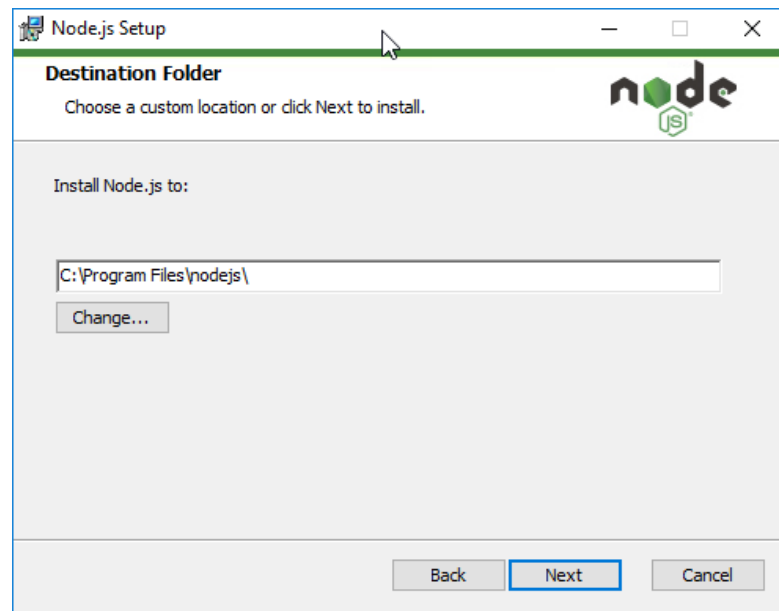
รูปที่ ก.10: หน้าต่างต้อนรับของ Node.js

4. แสดงหน้าต่างข้อตกลงในการใช้ Node.js ให้เลือกช่อง I accept the terms in the License Agreement และกด Next ดังแสดงในรูปที่ ก.11



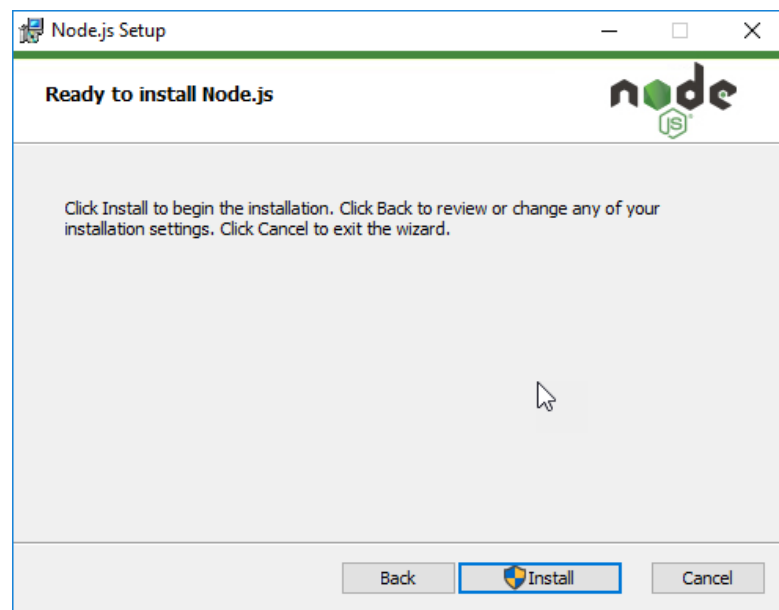
รูปที่ ก.11: หน้าต่างข้อตกลงในการใช้ Node.js

5. แสดงหน้าต่างเลือกโฟลเดอร์ที่จะทำการติดตั้ง ดังแสดงในรูปที่ ก.12



รูปที่ ก.12: หน้าต่างเลือกโฟลเดอร์ที่จะทำการติดตั้ง Node.js

6. แสดงหน้าต่างสำหรับติดตั้ง Node.js ทำการกด Install เพื่อทำการติดตั้ง ดังแสดงในรูปที่ ก.13

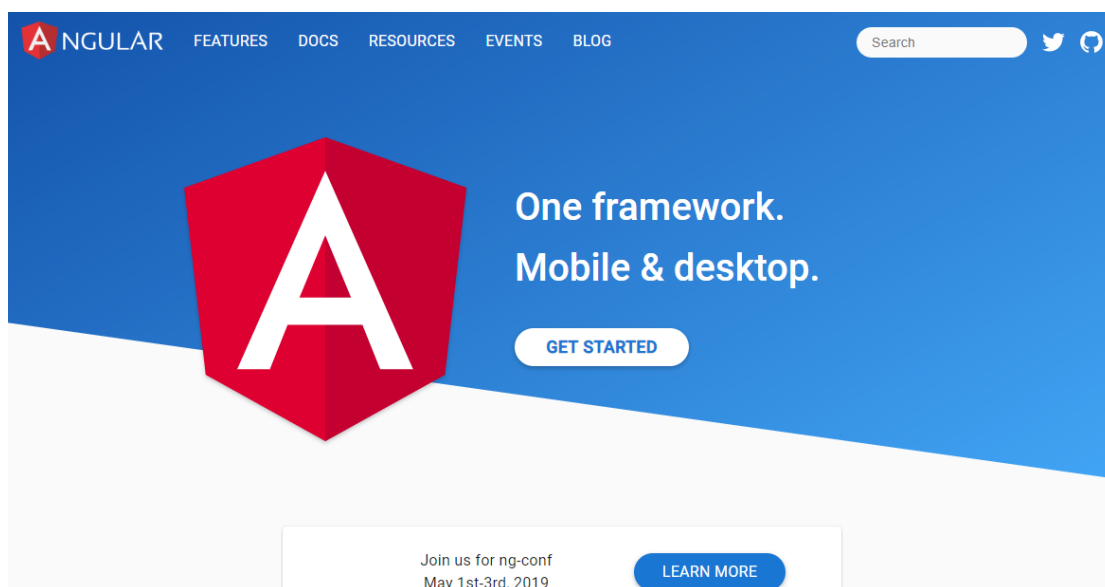


รูปที่ ก.13: หน้าต่างติดตั้ง Node.js

ก.3 การติดตั้ง Angular

1. สามารถติดตั้ง Angular ได้ที่ <https://angular.io/guide/quickstart> ดังแสดงในรูปที่ ก.

14



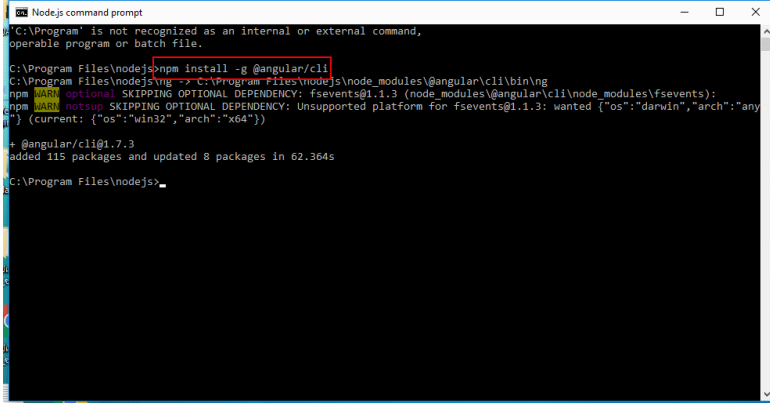
รูปที่ ก.14: หน้าเว็บ Angular

2. เปิด Node.js command prompt ขึ้นมาแล้วใช้คำสั่ง `npm install -g @angular/cli` ดังแสดงในรูปที่ ก.15



รูปที่ ก.15: ใช้คำสั่งติดตั้ง Angular

3. Node.js จะทำการดาวน์โหลดและติดตั้ง Angular 6 ดังแสดงในรูปที่ ก.16



```
Node.js command prompt
C:\Program' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\nodejs>npm install -g @angular/cli
C:\Program Files\nodejs>npm install -g @angular/cli
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.3 (node_modules\@angular\cli\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ @angular/cli@1.7.3
added 115 packages and updated 8 packages in 62.364s

C:\Program Files\nodejs>
```

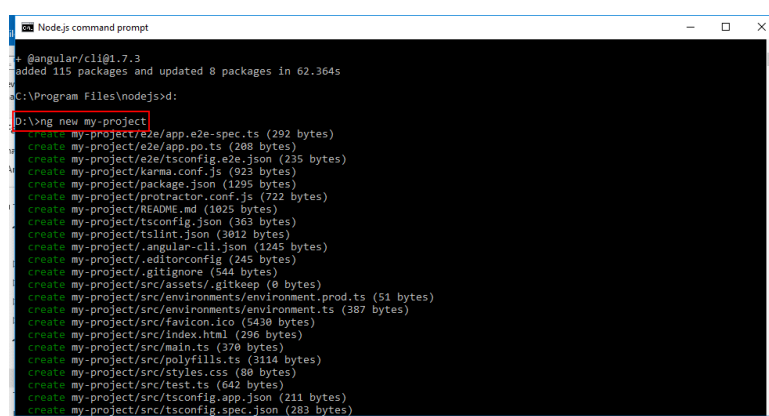
รูปที่ ก.16: ใช้คำสั่งติดตั้ง Angular

4. เข้าสู่ไดเรกทอรีของ project แล้วพิมพ์คำสั่ง `ng new my-project` ดังแสดงในรูปที่ ก.17



รูปที่ ก.17: ใช้คำสั่งสร้าง project

5. `ng` คือ prefix ของคำสั่ง Angular ในที่นี้เราสร้างโปรเจกต์ใหม่ชื่อ `my-project` รอนดาเวนโหลดเทมเพลตของ project เสร็จเรียบร้อย ดังแสดงในรูปที่ ก.18



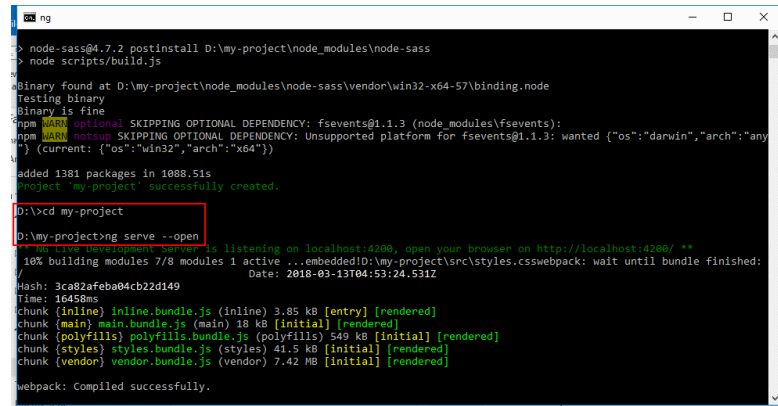
รูปที่ ก.18: ใช้คำสั่งสร้าง project

6. เข้าสู่ไดเรกทอรีของ project แล้วพิมพ์คำสั่ง `cd my-project` และ `ng serve --open` ดังแสดงในรูปที่ ก.19



รูปที่ ก.19: ใช้คำสั่งสร้าง project

7. คำสั่ง `serve` จะไปเรียกใช้งาน `webpack` เพื่อเปิด `localhost` ที่พอร์ต 4200 ดังแสดงในรูปที่ ก.20



```

> node-sass@4.7.2 postinstall D:\my-project\node_modules\node-sass
> node scripts/build.js

Binary found at D:\my-project\node_modules\node-sass\vendor\win32-x64-57\binding.node
Testing binary
Binary is fine
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.3 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

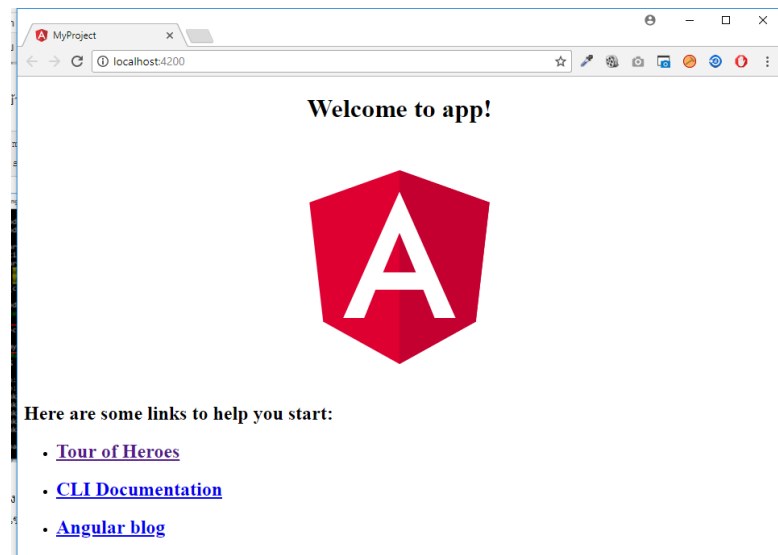
added 1381 packages in 1088.51s
project my-project successfully created.

D:\>cd my-project
D:\my-project>ng serve --open

webpack is listening on localhost:4200, open your browser on http://localhost:4200/ **
10% building modules 7/8 modules 1 active ...embeddedID:my-project\src\styles.csswebpack: wait until bundle finished:
Date: 2018-03-13T04:53:24.531Z
Hash: 3ca82afeba04cb22d149
Time: 16458ms
chunk (inline) inline.bundle.js (inline) 3.85 kB [entry] [rendered]
chunk (main) main.bundle.js (main) 18 kB [initial] [rendered]
chunk (polyfills) polyfills.bundle.js (polyfills) 549 kB [initial] [rendered]
chunk (styles) styles.bundle.js (styles) 41.5 kB [initial] [rendered]
chunk (vendor) vendor.bundle.js (vendor) 7.42 MB [initial] [rendered]
webpack: Compiled successfully.
  
```

รูปที่ ก.20: การใช้คำสั่งรัน project

8. หน้าแรกของเว็บไซต์แสดงดังรูปที่ ก.21



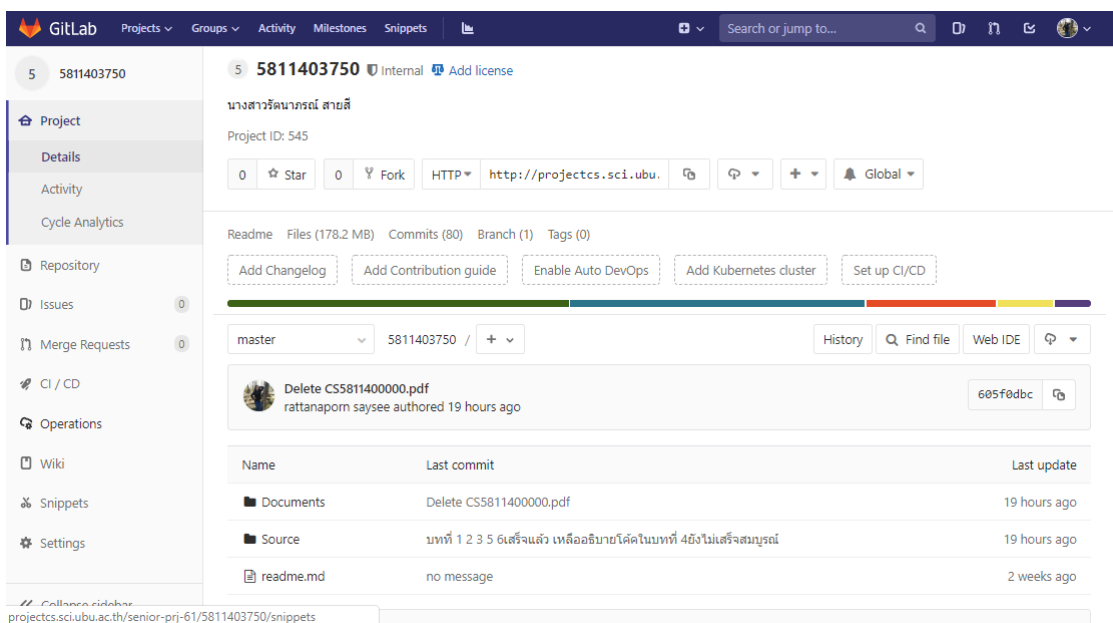
รูปที่ ก.21: หน้าแรกของเว็บไซต์

ภาคผนวก ข

คู่มือการติดตั้งระบบ

ในการติดตั้งเพื่อใช้งานเว็บแอปพลิเคชันระบบจองคิวร้านเสริมสวย สามารถทำได้โดยมีขั้นตอนดังนี้

1. สามารถ Clone Project ได้ที่ <http://projectcs.sci.ubu.ac.th/senior-prj-62/59110440259.git> ดังแสดงในรูปที่ ข.1



รูปที่ ข.1: หน้าเว็บ Repository ของโปรเจ็ค

2. เข้าไปในโฟลเดอร์ Source/Web เพื่อติดตั้ง Package โดยเปิด command prompt แล้วใช้คำสั่งดังรูปที่ ข.2

```
1 npm install
```

รูปที่ ข.2: คำสั่งติดตั้ง Package

3. ทำการพิมพ์คำสั่ง npm run start เพื่อเปิด localhost ที่พอร์ต 3000 ดังแสดงในรูปที่ ข.


```
1 npm run start
```

รูปที่ ข.3: คำสั่งเริ่มใช้งานโปรเจ็ค

ภาคผนวก ค

คู่มือการใช้งานระบบ

คู่มือการใช้งานทั้งหมดของระบบ สามารถแบ่งออกเป็น 2 ส่วน ดังนี้

- ส่วนผู้ใช้งาน
- ส่วนผู้ดูแลระบบ

ค.1 ส่วนผู้ใช้งาน

(a) ส่วนของหน้าสมัครสมาชิก

- เมื่อผู้ใช้กดปุ่มสมัครสมาชิกระบบจะแสดงหน้าต่างกรอกข้อมูลการสมัครสมาชิก ดังแสดงในรูปที่ ค.1

UBON Trip

สถานที่เยี่ยมชม ที่เที่ยว ที่กิน ที่พัก แนะนำสถานที่

1 เข้าสู่ระบบ

สมัครสมาชิก

* ชื่อผู้ใช้: 2

* ชื่อ-นามสกุล: 3

* รหัสผ่าน: รหัสผ่าน 6 ตัวขึ้นไป 4

* ยืนยันรหัสผ่าน: รหัสผ่าน 6 ตัวขึ้นไป 5

* เพศ: เลือกเพศ 6

เลือกรูปภาพโปรไฟล์ 7

9 สมัครสมาชิก 8 ยกเลิก

รูปที่ ค.1: หน้าจอการสมัครสมาชิก

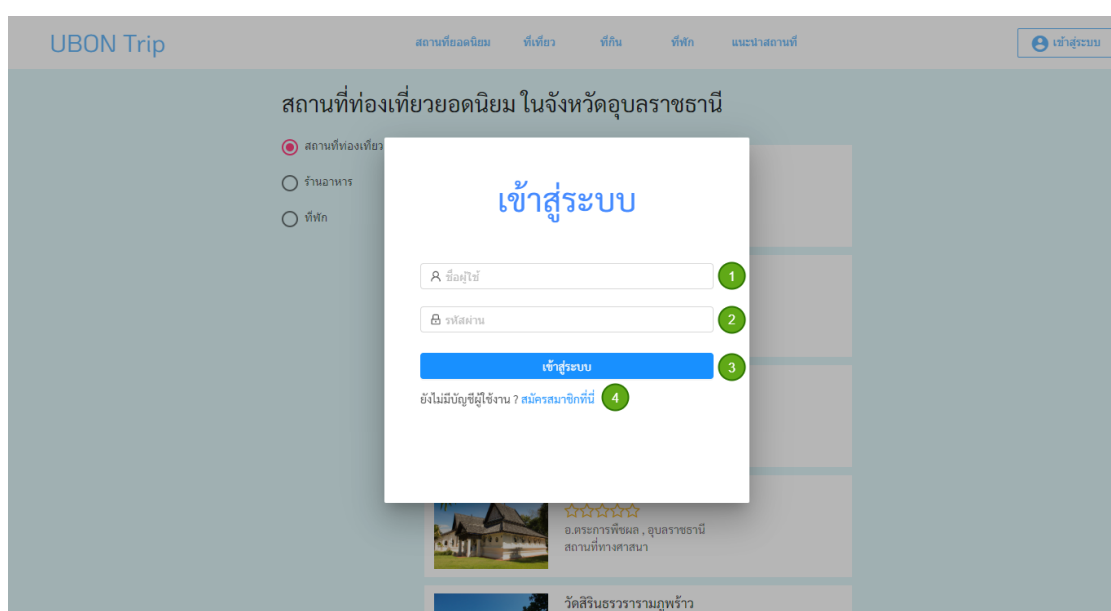
จากรูปที่ ค.1 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ปุ่มเข้าสู่ระบบเพื่อใช้ในการเข้าใช้งานระบบ
- หมายเลข 2 คือ ช่องกรอกชื่อผู้ใช้
- หมายเลข 3 คือ ช่องกรอกชื่อ-นามสกุล
- หมายเลข 4 คือ ช่องกรอกรหัสผ่าน
- หมายเลข 5 คือ ช่องกรอกรหัสผ่านยืนยัน

- หมายเลข 6 คือ เลือกเพศ
- หมายเลข 7 คือ ปุ่มเลือกรูปภาพโปรไฟล์
- หมายเลข 8 คือ ปุ่มสมัครสมาชิกเพื่อยืนยันการสมัครสมาชิก
- หมายเลข 9 คือ ปุ่มยกเลิกเมื่อไม่ต้องการสมัครสมาชิก

(b) ส่วนของหน้าเข้าสู่ระบบ

- เมื่อผู้ใช้สมัครสมาชิกสำเร็จผู้ใช้สามารถเข้าสู่ระบบได้ด้วยการกดปุ่มเข้าสู่ระบบ
ดังแสดงในรูปที่ ค.2



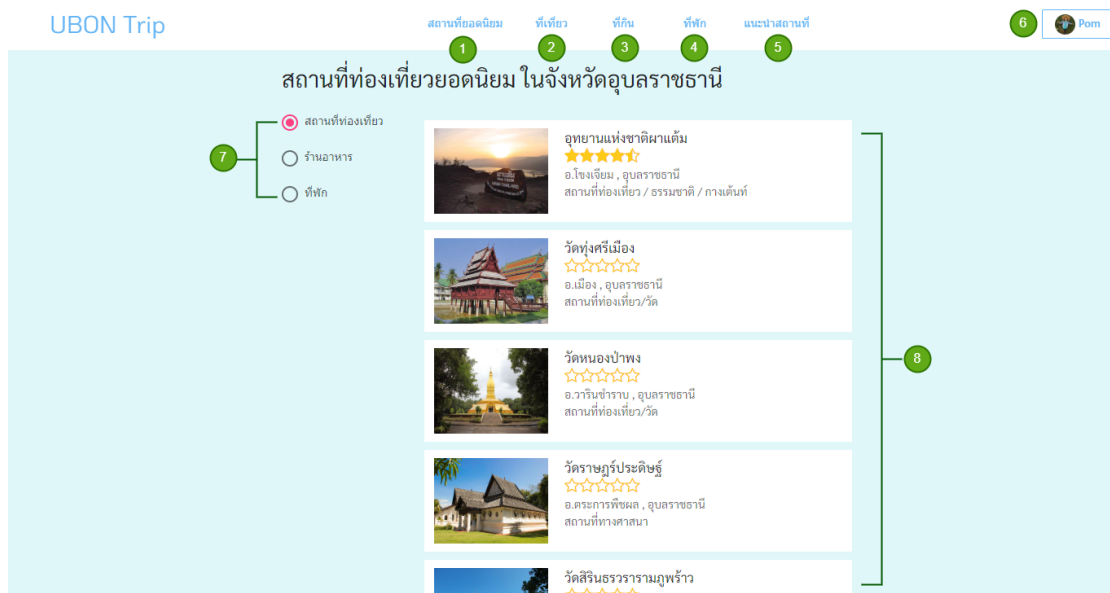
รูปที่ ค.2: หน้าจอการเข้าสู่ระบบ

จากรูปที่ ค.2 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ช่องกรอกชื่อผู้ใช้
- หมายเลข 2 คือ ช่องกรอกรหัสผ่าน
- หมายเลข 3 คือ ปุ่มกดเข้าสู่ระบบ
- หมายเลข 4 คือ เมื่อผู้ใช้คลิกสมัครสมาชิกระบบจะแสดงหน้าสมัครสมาชิก

(c) ส่วนของหน้าจัดอันดับสถานที่ยอดนิยม

- เมื่อผู้ใช้กดเลือกเมนูสถานที่ยอดนิยมระบบจะแสดงหน้าจัดอันดับสถานที่ยอดนิยมให้แก่ผู้ใช้ได้ดู ดังแสดงในรูปที่ ค.3



รูปที่ ค.3: หน้าจอการจัดอันดับสถานที่ยอดนิยม

จากรูปที่ ค.3 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ เมนูสถานที่ยอดนิยม จะแสดงข้อมูลสถานที่ยอดนิยมโดยจะให้ผู้ใช้งานสามารถเลือกกลุ่มที่จะดูสถานที่ยอดนิยมได้
- หมายเลข 2 คือ เมนูที่เที่ยว จะแสดงหน้าให้ผู้ใช้งานเลือกหมวดหมู่ที่เที่ยว
- หมายเลข 3 คือ เมนูที่กิน จะแสดงหน้าให้ผู้ใช้งานเลือกหมวดหมู่ที่กิน
- หมายเลข 4 คือ เมนูที่พัก จะแสดงหน้าให้ผู้ใช้งานเลือกหมวดหมู่ที่พัก
- หมายเลข 5 คือ เมนูแนะนำสถานที่ จะแสดงหน้าข้อมูลให้ผู้ใช้งานกรอกแบบประเมิน
- หมายเลข 6 คือ ปุ่มโปรไฟล์ โดยผู้ใช้งานสามารถดูข้อมูลโปรไฟล์ของตัวเองได้
- หมายเลข 7 คือ เป็นส่วนของการให้ผู้ใช้งานเลือกประเภทที่ต้องการดูข้อมูลสถานที่ยอดนิยม
- หมายเลข 8 คือ แสดงข้อมูลรายละเอียดสถานที่ยอดนิยมโดยเรียงจากอันดับ

ที่หนึ่งไปหาอันกับสุดท้าย

(d) ส่วนของหน้าหมวดสถานที่กิน

- เมื่อผู้ใช้กดเลือกเมนูสถานที่กินระบบจะแสดงหน้าประเภทของร้านอาหารให้แก่ผู้ใช้ได้ดู ดังแสดงในรูปที่ ค.4



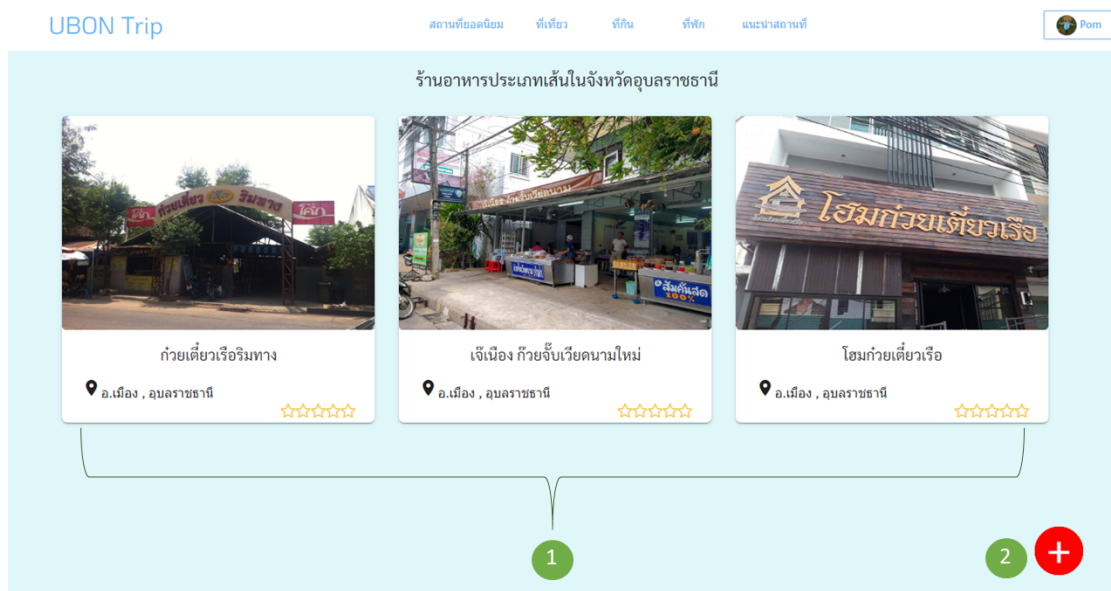
รูปที่ ค.4: หน้าจอหมวดที่กิน

จากรูปที่ ค.4 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ปุ่มเลือกประเภทข้อมูลที่ต้องการที่จะค้นหา
- หมายเลข 2 คือ ช่องกรอกชื่อสถานที่ที่ต้องการค้นหา
- หมายเลข 3 คือ ปุ่มค้นหา เมื่อกรอกชื่อสถานที่เสร็จสามารถกดปุ่มค้นหาได้
- หมายเลข 4 คือ แสดงข้อมูลหมวดต่างๆเพื่อให้ผู้ใช้ได้เลือกดูข้อมูลตามหมวด

(e) ส่วนของข้อมูลสถานที่

- เมื่อผู้ใช้งานเลือกสถานที่ที่ระบบจะแสดงหน้ารายละเอียดสถานที่ให้แก่ผู้
ใช้ได้ดู ดังแสดงในรูปที่ ค.5



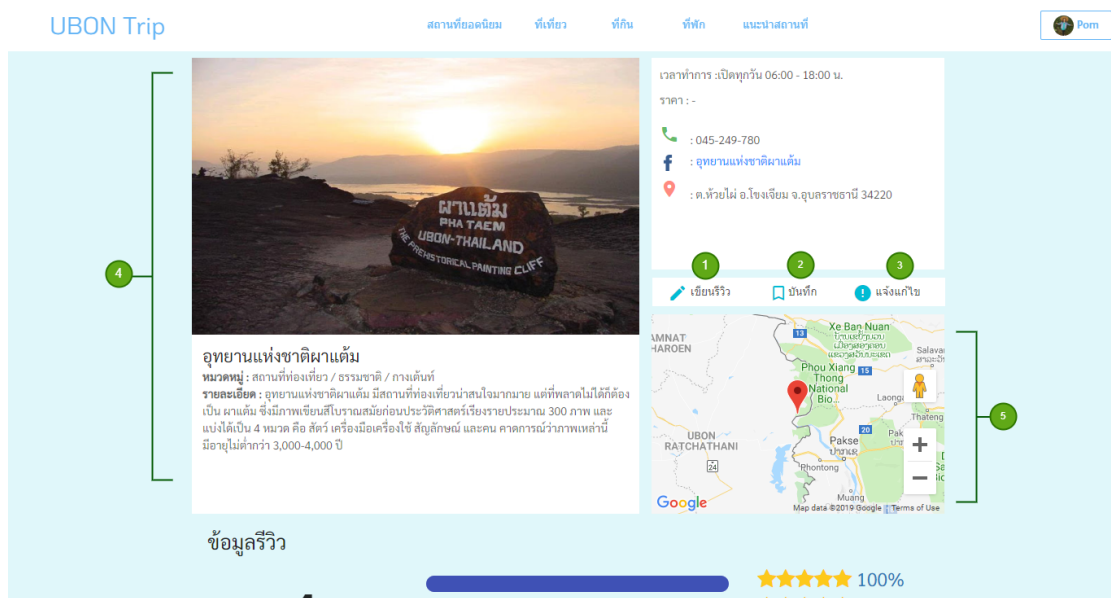
รูปที่ ค.5: หน้าจอแสดงข้อมูลสถานที่

จากรูปที่ ค.5 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ส่วนของการแสดงข้อมูลสถานที่โดยจะประกอบด้วย ชื่อสถานที่ ที่อยู่ คะแนนเฉลี่ยของสถานที่ และรูปภาพสถานที่
- หมายเลข 2 คือ ปุ่มเพิ่มข้อมูล เมื่อผู้ใช้งานต้องการที่จะเพิ่มข้อมูลสถานที่ใหม่เข้าไปในระบบเพื่อให้คนอื่นได้รู้จัก

(f) ส่วนของหน้ารายละเอียดสถานที่

- เมื่อผู้ใช้งานเลือกสถานที่ที่ระบบจะแสดงหน้ารายละเอียดสถานที่ให้แก่ผู้
ใช้ได้ดู ดังแสดงในรูปที่ ค.6



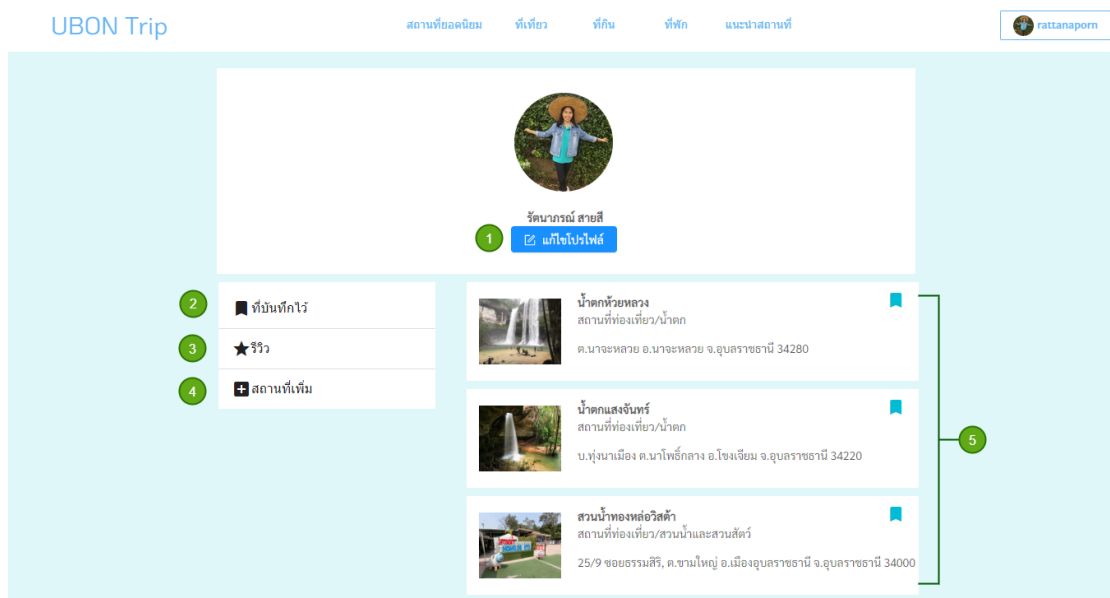
รูปที่ ค.6: หน้าจอแสดงรายละเอียดสถานที่

จากรูปที่ ค.6 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ปุ่มเขียนรีวิว ใช้ในการแสดงหน้าเขียนรีวิวเมื่อผู้ใช้งานต้องการที่จะเขียนรีวิว
- หมายเลข 2 คือ ปุ่มบันทึก ใช้ในการบันทึกสถานที่ไว้เพื่อเข้าไปอ่านข้อมูลรายละเอียดที่หลัง
- หมายเลข 3 คือ ปุ่มแจ้งเตือน ใช้ในการแสดงหน้าแจ้งเตือนเมื่อผู้ใช้งานเห็นว่าข้อมูลในส่วนนี้ไม่ถูกต้อง
- หมายเลข 4 คือ แสดงรายละเอียดข้อมูลของสถานที่
- หมายเลข 5 คือ ส่วนที่ใช้ในการแสดงแผนที่

(g) ส่วนของหน้าแสดงโปรไฟล์ผู้ใช้

- เมื่อผู้ใช้กดเลือกโปรไฟล์ระบบจะแสดงหน้าข้อมูลรายละเอียดของผู้ใช้ ดังแสดงในรูปที่ ค.7



รูปที่ ค.7: หน้าจอแสดงข้อมูลโปรไฟล์ของผู้ใช้

จากรูปที่ ค.7 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ปุ่มแก้ไขโปรไฟล์ โดยจะแสดงหน้าต่างซึ่งจะสามารถแก้ไขรูปภาพ ชื่อผู้ใช้ เปลี่ยนรหัสผ่านได้
- หมายเลข 2 คือ เมนูที่บันทึกไว้ ถ้าผู้ใช้กดเมนูนี้จะแสดงข้อมูลสถานที่ที่ผู้ใช้ได้ทำการกดบันทึกไว้
- หมายเลข 3 คือ เมนูรีวิว ถ้าผู้ใช้กดเมนูนี้จะแสดงข้อมูลสถานที่ที่ผู้ใช้ได้ทำการเขียนรีวิวไว้
- หมายเลข 4 คือ เมนูสถานที่เพิ่ม ถ้าผู้ใช้กดเมนูนี้จะแสดงข้อมูลสถานที่ที่ผู้ใช้ได้เพิ่มลงในระบบ
- หมายเลข 5 คือ ส่วนที่ใช้แสดงข้อมูลสถานที่ต่างๆตามเมนูที่ผู้ใช้เลือก

(h) ส่วนของหน้าเขียนรีวิวและให้คะแนน

- เมื่อผู้ใช้กดปุ่มเขียนรีวิวระบบจะแสดงหน้าเขียนรีวิว ดังแสดงในรูปที่ ค.8

The screenshot displays the 'UBON Trip' website interface. A modal window titled 'ให้คะแนนและเขียนรีวิว' (Rate and Write Review) is centered on the screen. The modal contains a star rating system with five stars, a text input field for the review title, a text area for the review content, a button to add photos, and buttons to submit the review or cancel. The background shows a travel page for 'อุทยานแห่งชาติผาแต้ม' (Pha Thom National Park) with a map and travel details.

รูปที่ ค.8: หน้าจอแสดงเขียนรีวิว

จากรูปที่ ค.8 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ส่วนของการให้คะแนนสถานที่
- หมายเลข 2 คือ ช่องกรอกหัวข้อรีวิว
- หมายเลข 3 คือ ช่องกรอกรายละเอียดรีวิว
- หมายเลข 4 คือ ปุ่มเพิ่มรูปภาพของรีวิว
- หมายเลข 5 คือ ปุ่มบันทึกรีวิว
- หมายเลข 6 คือ ปุ่มยกเลิกการเขียนรีวิว

ค.2 ส่วนผู้ดูแลระบบ

(a) ส่วนของการจัดการข้อมูล

- เมื่อผู้ดูแลระบบเข้าสู่ระบบสำเร็จ จะเข้ามาหน้าจัดการข้อมูลสถานที่ ดังแสดงในรูปที่ ค.9



รูปที่ ค.9: หน้าจอจัดการข้อมูลสถานที่เที่ยว

จากรูปที่ ค.9 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ เมนูแจ้งแก้ไข จะแสดงข้อมูลการแจ้งแก้ไขมาจากฝั่งของผู้ใช้
- หมายเลข 2 คือ เมนูข้อมูลเที่ยว จะแสดงข้อมูลสถานที่เที่ยวทั้งหมดเพื่อให้ผู้ดูแลระบบสามารถจัดการข้อมูลได้
- หมายเลข 3 คือ เมนูข้อมูลที่พัก จะแสดงข้อมูลสถานที่พักทั้งหมดเพื่อให้ผู้ดูแลระบบสามารถจัดการข้อมูลได้
- หมายเลข 4 คือ เมนูข้อมูลที่พัก จะแสดงข้อมูลสถานที่พักทั้งหมดเพื่อให้ผู้ดูแลระบบสามารถจัดการข้อมูลได้
- หมายเลข 5 คือ เมนู train ผู้ดูแลระบบสามารถ train model ใหม่ทุกครั้งได้
- หมายเลข 6 คือ ปุ่มออกจากระบบ
- หมายเลข 7 คือ ปุ่มเลือกประเภทเพื่อดูข้อมูลแค่ประเภทนั้นๆ

- หมายเลข 8 คือ แสดงรายละเอียดข้อมูล
- หมายเลข 9 คือ ปุ่มเพิ่มข้อมูลสถานที่
- หมายเลข 10 คือ ปุ่มแก้ไขข้อมูลสถานที่
- หมายเลข 11 คือ ปุ่มลบข้อมูลสถานที่

(b) ส่วนของการจัดการข้อมูลเพิ่มข้อมูลสถานที่

- เมื่อผู้ดูแลระบบเข้าสู่ระบบและต้องการเพิ่มข้อมูลสามารถกดปุ่มเพิ่มข้อมูลระบบจะแสดงหน้าต่างเพิ่มให้กรอกข้อมูลสถานที่ ดังแสดงในรูปที่ ค.10

รูปที่ ค.10: หน้าจอการเพิ่มสถานที่

จากรูปที่ ค.10 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ส่วนที่แสดงให้ผู้ดูแลระบบกรอกข้อมูลสถานที่ที่ต้องการเพิ่มข้อมูล
- หมายเลข 2 คือ ปุ่มบันทึก เพื่อเป็นการยืนยันการเพิ่มสถานที่
- หมายเลข 3 คือ ปุ่มยกเลิก เพื่อเป็นการยืนยันยกเลิกการเพิ่มสถานที่

(c) ส่วนของการจัดการข้อมูลแก้ไขสถานที่

- เมื่อผู้ดูแลระบบเข้าสู่ระบบและต้องการแก้ไขข้อมูลสามารถกดปุ่มแก้ไข และระบบจะแสดงหน้าแก้ไข ดังแสดงในรูปที่ ค.11

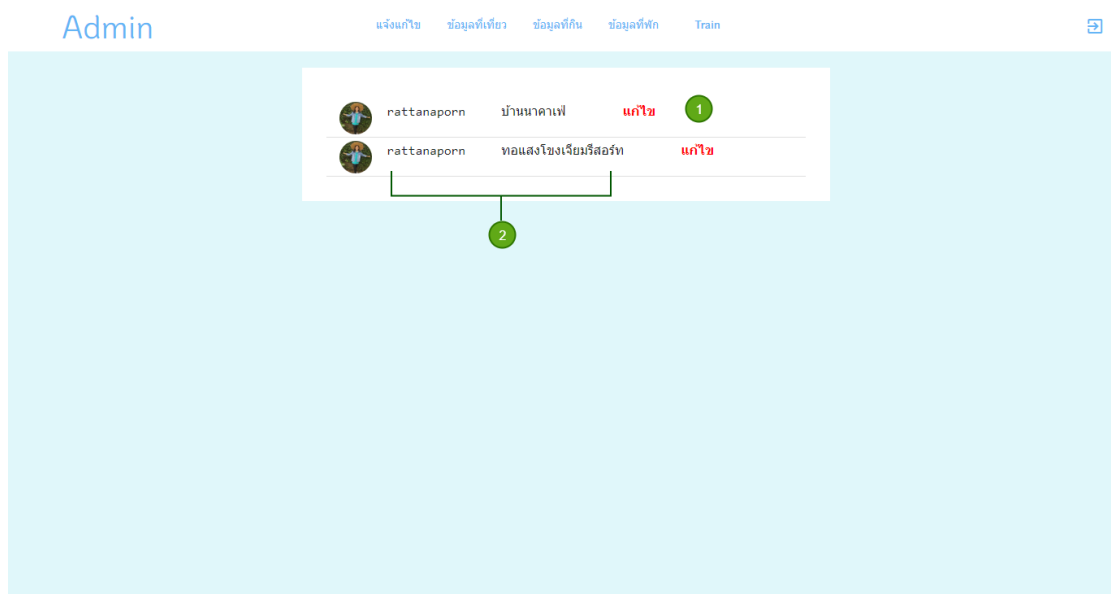
รูปที่ ค.11: หน้าจอการแก้ไขข้อมูลสถานที่

จากรูปที่ ค.11 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ส่วนแสดงข้อมูลรายละเอียดของสถานที่
- หมายเลข 2 คือ ปุ่มบันทึกการแก้ไข เมื่อผู้ดูแลระบบต้องการที่จะบันทึกข้อมูลที่แก้ไข
- หมายเลข 3 คือ ปุ่มยกเลิกการแก้ไข

(d) ส่วนของหน้าจัดการข้อมูลแจ้งแก้ไข

- เมื่อผู้ดูแลระบบเข้าสู่ระบบและต้องการจัดการข้อมูลแจ้งแก้ไขสามารถกดที่เมนูแจ้งแก้ไข ระบบจะแสดงหน้าแจ้งแก้ไข ดังแสดงในรูปที่ ค.12



รูปที่ ค.12: หน้าจอการแจ้งแก้ไข

จากรูปที่ ค.12 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ปุ่มแก้ไข เมื่อผู้ดูแลระบบต้องการแก้ไขข้อมูลตัวที่ผู้ใช้แจ้งเข้ามา
- หมายเลข 2 คือ รายละเอียดข้อมูลผู้แจ้งแก้ไขและชื่อสถานที่ที่แจ้งแก้ไข

ประวัติผู้เขียน

ชื่อ-สกุล: นางสาว ปิยพร อามรศรี

รหัสประจำตัวนักศึกษา: 59110440259

วันเกิด: 12 06 2540

ที่อยู่ที่สามารถติดต่อได้: 7 หมู่ 1 ต.หนองบก อ.เหล่าเสือโก้ก จ.อุบลราชธานี 34000

เบอร์โทรศัพท์: (+66) 64 718 3784

อีเมล: piyaphorn.ar.59@ubu.ac.th

ระดับมัธยมต้น: โรงเรียนหกลีปรรชาวิทยาคม อุบลราชธานี

ระดับมัธยมปลาย: โรงเรียนหกลีปรรชาวิทยาคม อุบลราชธานี

ระดับอุดมศึกษา: ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ สาขาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี