



รายงาน
การวิเคราะห์และออกแบบระบบ
การจองตั้งรักษาอ่อนไลน์

เสนอ
รองศาสตราจารย์ สถาوارต์ จงพัฒนาการ

จัดทำโดย
นาย ชีระพงษ์ พยัคฆ์รังสี
รหัสนิสิต 6521600494
หมู่เรียน 700 เลขที่ 10

วิชาการคิดเคราะห์และออกแบบระบบสารสนเทศ
สาขาเทคโนโลยีสารสนเทศเพื่อการสื่อสาร
ภาควิชาวิทยาศาสตร์การคำนวณและเทคโนโลยีดิจิตอล
คณะศิลปศาสตร์และวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
ปีการศึกษา 2568

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา 02739322 การวิเคราะห์และออกแบบระบบสารสนเทศ (Information System Analysis and Design) ซึ่งจัดทำขึ้นโดยมีวัตถุประสงค์เพื่อออกแบบส่วนต่อประสานกับผู้ใช้ในระบบจองตัวรถไฟออนไลน์ และศึกษากระบวนการในการออกแบบเว็บไซต์ เพื่อให้สามารถให้บริการการจองตัวที่สุดวาก รวดเร็ว และปลอดภัย การพัฒนาระบบจองตัวรถไฟออนไลน์ได้รับการออกแบบมาเพื่อให้ผู้ใช้สามารถจองตัวได้อย่างสะดวก ผ่านช่องทางออนไลน์ เพียงแค่มีการเชื่อมต่ออินเทอร์เน็ต ซึ่งช่วยลดความยุ่งยากในการเดินทางไปยังสถานีและสามารถวางแผนการเดินทางได้อย่างสะดวก

ผู้จัดทำหวังว่ารายงานฉบับนี้จะเป็นประโยชน์ต่อผู้อ่าน และเป็นแนวทางในการพัฒนาระบบจองตัวออนไลน์ในอนาคต หากมีข้อผิดพลาดประการใด ทางผู้จัดทำขออภัยไว้ ณ ที่นี่

ธีระพงษ์ พยัคฆ์รังสี

สารบัญ

เรื่อง	หน้า
บทที่ 1 Proposal	
1.1 ชื่อเรื่อง	1
1.2 ความเป็นมาและที่มาของปัญหา	1
1.3 วัตถุประสงค์	1
1.4 ขอบเขตของระบบ	1
1.5 ความต้องการของระบบใหม่	2
1.6 ทีมงานที่รับผิดชอบ	2
1.7 แนวทางการพัฒนาระบบ	2
1.8 แผนการดำเนินงาน	3
1.9 ประโยชน์ที่จะได้รับ	4
1.10 Requirement	5
บทที่ 2 การออกแบบ	
2.1 แผนภาพบริบท (Context diagram)	6
2.2 แผนภาพการไหลของข้อมูล ระดับที่ 0	7
2.3 แผนภาพการไหลของข้อมูล ระดับที่ 1	8
2.4 คำอธิบายกระบวนการ (Process Description)	9
2.5 แผนภาพ E-R	10
2.6 Database	11-13
2.7 แสดงพจนานุกรมข้อมูล	14-17
2.8 ต้นไม้มีการตัดสินใจ (Decision Tree)	17
2.9 ตารางการการตัดสินใจ (Decision Table)	18
บทที่ 3 การวิเคราะห์ และ ออกแบบระบบเชิงวัตถุด้วย UML	
3.1 Use case Diagram	19
3.2 Class Diagram	20
3.3 Sequence Diagram	21
3.4 Activity Diagram	22
3.5 Flowchart	23
3.6 HTA	24

บทที่ 4 การออกแบบส่วนต่อประสานผู้ใช้	
การใช้งานหน้าเว็บ	25-37
บทที่ 5 Code และการเขียนโปรแกรม (Programming)	
การเขียนโค้ดในแต่ละหน้า	38-91
บทที่ 6 แบบฟอร์ม ประเมินผลระบบ	
แบบฟอร์ม	92
ภาคผนวก	
การใช้งานโปรแกรม	93-95
บรรณานุกรม	
อ้างอิง	96

บทที่ 1 Proposal

1.1 ชื่อเรื่อง ระบบจองตั๋วรถไฟออนไลน์

1.2 ความเป็นมาและที่มาของปัญหา

ปัจจุบันการเดินทางด้วยรถไฟเป็นทางเลือกยอดนิยมสำหรับการเดินทางทั้งในเมืองและระหว่างจังหวัดเนื่องจากมีค่าโดยสารที่เข้าถึงได้ง่ายและเป็นมิตรกับสิ่งแวดล้อมอย่างไร้ตามผู้โดยสารยังคงประสบปัญหาหลายประการ เช่น:

- การค้นหาข้อมูลตารางรถไฟที่ต้องการยาก
- การซื้อตั๋วที่สถานีใช้เวลานานเนื่องจากคิวที่ยาว
- ความไม่สะดวกในการเปลี่ยนแปลงการจองหรือยกเลิกตั๋ว
- ข้อผิดพลาดจากการบันทึกข้อมูลด้วยวิธีดังเดิม

ดังนั้น การพัฒนาระบบจองตั๋วรถไฟออนไลน์ขึ้นมา จะช่วยให้ผู้โดยสารสามารถค้นหาเที่ยวรถไฟที่เหมาะสม ซื้อตั๋วได้สะดวกผ่านช่องทางออนไลน์ ลดเวลาในการเดินทางไปยังสถานี และสามารถพิมพ์หรือเก็บตัวในรูปแบบอิเล็กทรอนิกส์ไว้ใช้งานได้ทันทีหลังการชำระเงิน

1.3 วัตถุประสงค์

โครงการ ระบบจองตั๋วรถไฟออนไลน์ มีวัตถุประสงค์ดังนี้:

- พัฒนาระบบจองตั๋วรถไฟที่สะดวก รวดเร็ว และปลอดภัย
- เพิ่มความสะดวกให้ผู้โดยสารในการค้นหาเที่ยวรถไฟ จองตั๋ว และจัดการการเดินทาง
- ลดข้อผิดพลาดที่อาจเกิดขึ้นจากการดำเนินการแบบดั้งเดิม
- รองรับการทำงานบนอุปกรณ์หลากหลาย เช่น คอมพิวเตอร์ โทรศัพท์มือถือ และแท็บเล็ต
- ช่วยเพิ่มประสิทธิภาพการทำงานของเจ้าหน้าที่ และยกระดับการให้บริการผู้โดยสาร

1.4 ขอบเขตของระบบ

ฟังก์ชันการค้นหาเส้นทางรถไฟ

- ค้นหาเส้นทางรถไฟตามต้นทาง ปลายทาง วันที่เดินทาง และจำนวนผู้โดยสาร
- แสดงสถานะที่นั่งว่างแบบเรียลไทม์

ฟังก์ชันการเลือกที่นั่ง

- แสดงรายละเอียดการเดินทาง เช่น ประเภทที่นั่ง เวลาออกเดินทาง ค่าโดยสาร และสถานะที่นั่งว่าง
- รองรับการเลือกที่นั่งตามความต้องการ

ฟังก์ชันการกรอกข้อมูลผู้โดยสาร

- รองรับการกรอกข้อมูลส่วนตัว เช่น ชื่อ เบอร์โทรศัพท์ และอีเมล
- บันทึกข้อมูลการเดินทางสำหรับการใช้งานครั้งถัดไป

ฟังก์ชันการชำระเงิน

- รองรับการชำระเงินผ่านช่องทางหลากหลาย เช่น บัตรเครดิต/เดบิต, QR Code, และ e-Wallet
- ระบบตรวจสอบสถานะการชำระเงินอัตโนมัติ

ฟังก์ชันการยืนยันการจอง

- ส่งอีเมลหรือ SMS ยืนยันการจอง พร้อมหมายเลขการจอง
- ผู้โดยสารสามารถพิมพ์ตัวรถไฟได้เอง หรือแสดง QR Code ที่สถานี

ฟังก์ชันการจัดการการจอง

- ผู้ใช้สามารถเปลี่ยนแปลงหรือยกเลิกการจองได้ตามเงื่อนไข
- แสดงประวัติการจองและสถานะการเดินทาง

ฟังก์ชันแจ้งเตือน

- ส่งการแจ้งเตือนผ่าน SMS หรืออีเมลเกี่ยวกับการเดินทาง เช่น เวลาออกเดินทาง และการเปลี่ยนแปลงของเที่ยวรถไฟ

1.5 ความต้องการของระบบใหม่

อินเทอร์เฟชที่ใช้งานง่าย รองรับการแสดงผลหลายภาษา เช่น ภาษาไทยและภาษาอังกฤษ
รองรับการใช้งานผ่านเว็บไซต์และแอปพลิเคชันบนอุปกรณ์มือถือ
ระบบความปลอดภัยสำหรับข้อมูลผู้ใช้งานและการชำระเงิน
ระบบสามารถปรับขยายได้ในอนาคต เช่น การเพิ่มเส้นทางใหม่ หรือฟีเจอร์พิเศษ
มีระบบแจ้งเตือนที่แม่นยำผ่านช่องทางต่าง ๆ

1.6 ทีมงานที่รับผิดชอบ

นายธีระพงษ์ พยัคฆ์รังสี เป็นผู้รับผิดชอบโครงการ

1.7 แนวทางการพัฒนาระบบ

การเริ่มต้นและวางแผนโครงการ

กำหนดเป้าหมายและขอบเขตของระบบ เช่น การค้นหาเส้นทาง การจอง การชำระเงิน และการแจ้งเตือน
วางแผนการพัฒนาด้วยการจัดทำด้วยการและกำหนดระยะเวลา

การวิเคราะห์ระบบ

วิเคราะห์ความต้องการของผู้ใช้งาน เช่น การค้นหาเส้นทางที่สะดวก และการชำระเงินที่ปลอดภัย
ออกแบบโครงสร้างฐานข้อมูล เช่น ผู้ใช้, เส้นทางรถไฟ, การจอง, การชำระเงิน

การออกแบบระบบ

ออกแบบอินเทอร์เฟชที่ใช้งานง่ายและเข้าถึงได้ทุกกลุ่มผู้โดยสาร
เลือกเทคโนโลยีที่เหมาะสม เช่น React หรือ Flutter สำหรับ Frontend และ Node.js หรือ Django สำหรับ Backend

การพัฒนาและทดสอบระบบ

พัฒนาฟังก์ชันต่าง ๆ เช่น การค้นหาเส้นทาง การจอง และการยืนยันการชำระเงิน
ทดสอบระบบในทุกฟังก์ชันการทำงานเพื่อแก้ไขข้อผิดพลาด

การบำรุงรักษา

อัปเดตระบบอย่างสม่ำเสมอเพื่อรับผู้ใช้ที่เพิ่มขึ้น
แก้ไขปัญหาต่าง ๆ และเพิ่มฟีเจอร์ใหม่ตามความต้องการ

1.8 แผนการดำเนินงาน

รหัส	ชื่องาน	รายละเอียด	ระยะเวลา (สัปดาห์)
1	วิเคราะห์ปัญหาและวางแผน	ศึกษาความต้องการของผู้ใช้งาน กำหนดขอบเขตงาน	2
2	ออกแบบระบบ	ออกแบบฐานข้อมูล, กระบวนการทำงาน และอินเทอร์เฟซ	3
3	พัฒนาระบบ	พัฒนา Frontend, Backend และฐานข้อมูล	6
4	ทดสอบระบบ	ทดสอบระบบแบบ Unit, Integration, System และ UAT	4
5	ติดตั้งระบบและเปิดใช้งาน	ติดตั้งระบบบนเซิร์ฟเวอร์จีรังสี เชื่อมต่อกับ Payment Gateway และเปิดตัวระบบ	2
6	บำรุงรักษาและปรับปรุงระบบ	บำรุงรักษาและเพิ่มฟีเจอร์ใหม่ต่อเนื่อง	ต่อเนื่อง

งบประมาณที่ใช้

รายการ	รายละเอียด	ค่าใช้จ่าย (บาท)
1. ค่าพัฒนาระบบ	พัฒนา Frontend, Backend, ฐานข้อมูล และระบบความปลอดภัย	500,000
2. ค่าออกแบบ UI/UX	ออกแบบอินเทอร์เฟซผู้ใช้งานให้ใช้งานง่าย และรองรับทุกอุปกรณ์	30,000
3. ค่า Cloud และ Hosting	บริการ Cloud เช่น AWS, Google Cloud สำหรับ Hosting ระบบ	20,000/ปี
4. ค่า Payment Gateway Integration	ติดตั้งและเชื่อมต่อระบบกับ Payment Gateway	10,000
5. ค่าทดสอบระบบ	ทดสอบระบบในทุกฟังก์ชัน เช่น Unit Testing, Integration Testing	20,000
6. ค่าอบรมการใช้งาน	จัดอบรมเจ้าหน้าที่สถานีรถไฟฟ้าและพนักงานในการใช้งานระบบ	5,000
7. ค่าโปรแกรมชั้นการปิดตัวระบบ	การประชาสัมพันธ์และโปรแกรมผ่านช่องทางออนไลน์	15,000
8. ค่าสำรองข้อมูล (Backup)	บริการสำรองข้อมูลและระบบรักษาความปลอดภัยเพิ่มเติม	10,000/ปี
9. ค่าวัสดุและบำรุงรักษา	ปรับปรุงระบบตามฟีดแบ็คและเพิ่มฟีเจอร์ใหม่ทุก 6 เดือน	40,000/ปี

รวมค่าใช้จ่ายทั้งหมดเริ่มต้น: 610,000 บาท

ค่าใช้จ่ายรายปี (สำหรับ Hosting และบำรุงรักษา): 70,000 บาท/ปี

1.9 ประโยชน์ที่จะได้รับ

1. เพิ่มความสะดวกสบายให้ผู้โดยสาร

ผู้โดยสารสามารถจองตัวได้ทุกที่ทุกเวลา ผ่านระบบออนไลน์ โดยไม่จำเป็นต้องเดินทางไปที่สถานีรถไฟ ลดเวลาในการรอคิวซื้อตั๋วที่สถานี

2. ลดข้อผิดพลาดในการจองตั๋ว

ระบบช่วยตรวจสอบข้อมูลและสถานะที่นั่งว่างแบบเรียลไทม์

ลดข้อผิดพลาดที่เกิดจากการบันทึกข้อมูลด้วยมือ เช่น การจองที่นั่งซ้ำ

3. เพิ่มประสิทธิภาพการจัดการสำหรับเจ้าหน้าที่

ลดภาระงานของเจ้าหน้าที่สถานี เช่น การจัดการการจอง การตรวจสอบที่นั่ง และการยืนยันการจอง ทำให้เจ้าหน้าที่สามารถมุ่งเน้นการให้บริการลูกค้าในด้านอื่น ๆ

4. รองรับการเติบโตของผู้โดยสาร

ระบบสามารถปรับขยายและรองรับผู้ใช้งานจำนวนมากได้ในอนาคต

ส่งเสริมการเดินทางด้วยรถไฟ ซึ่งเป็นการเดินทางที่ประหยัดและยั่งยืน

5. สร้างประสบการณ์ที่ดีให้กับผู้โดยสาร

การใช้งานระบบที่รวดเร็วและเสถียรช่วยเพิ่มความพึงพอใจของผู้โดยสาร

การแจ้งเตือนแบบเรียลไทม์ เช่น เวลาออกเดินทางและการเปลี่ยนแปลงเที่ยวรถไฟ ช่วยให้ผู้โดยสารมั่นใจและวางแผนการเดินทางได้ดีขึ้น

6. ลดต้นทุนการดำเนินงานขององค์กร

ลดการใช้กระดาษและต้นทุนการพิมพ์ตั๋วแบบดั้งเดิม

ลดต้นทุนในการจ้างงานเจ้าหน้าที่สำหรับงานที่สามารถดำเนินการโดยระบบอัตโนมัติ

7. ข้อมูลเชิงลึกเพื่อการวางแผนและพัฒนา

ระบบสามารถเก็บข้อมูลพฤติกรรมผู้ใช้งาน เช่น เส้นทางยอดนิยมและช่วงเวลาที่มีการจองมากที่สุด

ข้อมูลเหล่านี้สามารถใช้ในการวางแผนปรับปรุงเส้นทางรถไฟและบริการเพิ่มเติมในอนาคต

8. ส่งเสริมการใช้เทคโนโลยีดิจิทัล

ระบบช่วยให้การเดินทางด้วยรถไฟทันสมัยและเข้าถึงได้ง่ายขึ้น

สนับสนุนการเปลี่ยนผ่านสู่ระบบดิจิทัลในภาคการขนส่ง

9. เพิ่มรายได้และโอกาสทางธุรกิจ

การจองตั๋วออนไลน์เปิดโอกาสให้ผู้โดยสารใหม่ที่ไม่สะดวกซื้อตั๋วที่สถานีเข้าถึงบริการได้

ส่งเสริมการจองล่วงหน้าและเพิ่มโอกาสในการขายบริการเสริม เช่น อาหารและเครื่องดื่มนบนรถไฟ

10. ส่งเสริมความยั่งยืนด้านสิ่งแวดล้อม

การลดการใช้กระดาษและการเดินทางไปยังสถานีช่วยลดคาร์บอนฟุตพรินท์

ส่งเสริมการใช้รถไฟ ซึ่งเป็นวิธีการเดินทางที่เป็นมิตรต่อสิ่งแวดล้อม

1.10 Requirement

System :	ระบบจองตั๋วรถไฟฟ้าออนไลน์
Module :	การจำหน่ายตั๋วผ่านทางอินเทอร์เน็ต
Objective :	เพื่อพัฒนาระบบการจองตั๋วที่สะดวกและใช้งานง่าย
Programmer:	ธีระพงษ์ พยัคฆ์รังสี
Requirements:	<p>นอกเหนือจากการให้ลูกค้าได้จองตั๋วรถไฟฟ้าจากสถานีแล้ว ลูกค้าสามารถจองตั๋วรถไฟฟ้าออนไลน์ผ่านเครือข่ายอินเทอร์เน็ตเพื่อความสะดวกและรวดเร็ว โดยระบบมีความสามารถเลือกจองตั๋วสำหรับเส้นทางต่างๆ และเลือกที่นั่งที่ต้องการได้ตามความสะดวก การสืบค้นข้อมูลในระบบจองตั๋วรถไฟฟ้าออนไลน์คือ:</p> <ul style="list-style-type: none"> แสดงรายละเอียดของตัว แสดงข้อมูลเกี่ยวกับตัว เช่น เส้นทางการเดินทาง, ประเภทตัว (ชั้น 1, ชั้น 2), เวลาเดินทาง, ราคา, ที่นั่งที่สามารถเลือกได้ แสดงราคาตัว ระบบจะแสดงราคาของตัวสำหรับเส้นทางต่างๆ และประเภทของที่นั่งที่มีให้เลือก รายงานการชำระเงิน ระบบสามารถแสดงรายงานการชำระเงินและสถานะการชำระเงินจากผู้ใช้ เช่น การชำระเงินที่สำเร็จหรือยังของการชำระเงิน รายงานสถานะการจองตั๋ว รายงานสถานะการจองตั๋ว เช่น การยืนยันการจอง, การยกเลิกการจอง หรือสถานะของการชำระเงิน ซึ่งจะแสดงผลผ่านระบบออนไลน์ <p>ฮาร์ดแวร์ที่ใช้:</p> <ul style="list-style-type: none"> คอมพิวเตอร์ CPU: AMD Ryzen 5 3500 RAM: 16.0 GB ซอฟต์แวร์ที่ใช้: <ul style="list-style-type: none"> React/Angular: สำหรับพัฒนา Frontend ของระบบ Node.js หรือ Python Django: สำหรับพัฒนา Backend Database: MySQL หรือ PostgreSQL

บทที่ 2

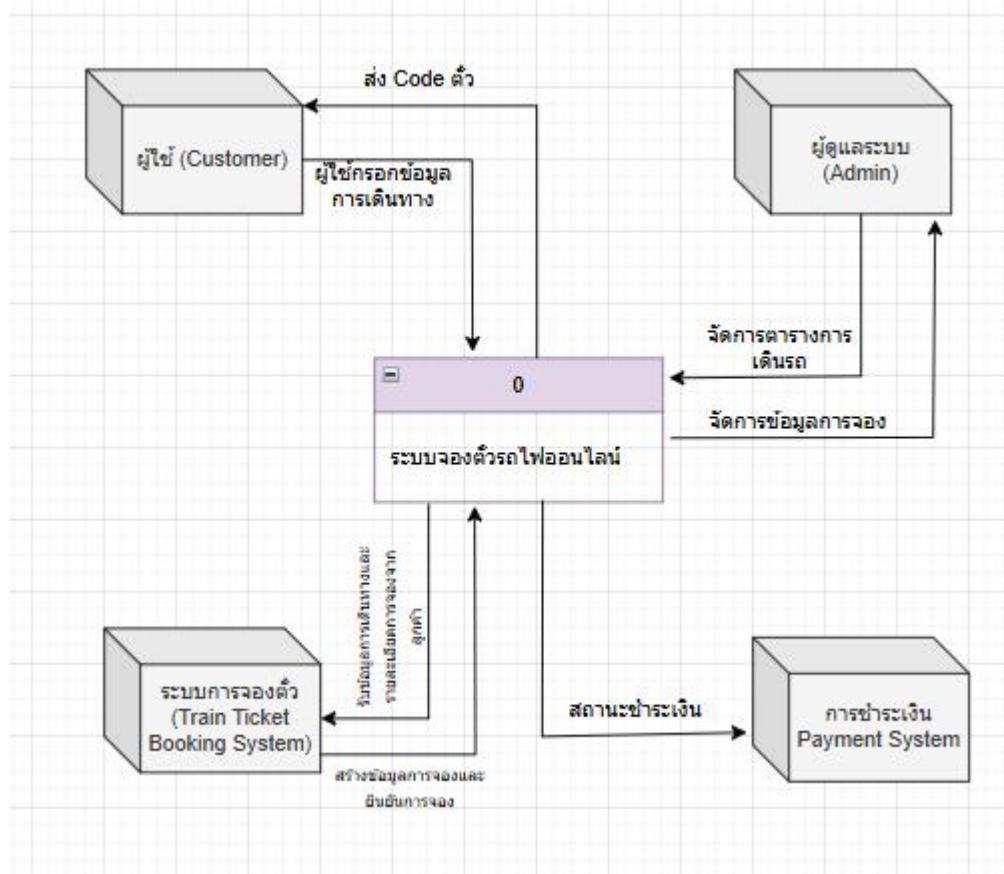
การออกแบบ

2.1 แผนภาพบริบท (Context diagram)

แผนภาพบริบท จะช่วยแสดงภาพรวมของระบบจองตั๋วรถไฟฟ้าออนไลน์ โดยการแสดงถึงกระบวนการหลัก (Process) ของระบบ และเอนทิตี้ (Entity) ที่ติดต่อกับกระบวนการนั้น รวมถึงการไหลของข้อมูลระหว่างกระบวนการและเอนทิตี้ต่างๆ

รายละเอียดของแผนภาพบริบท:

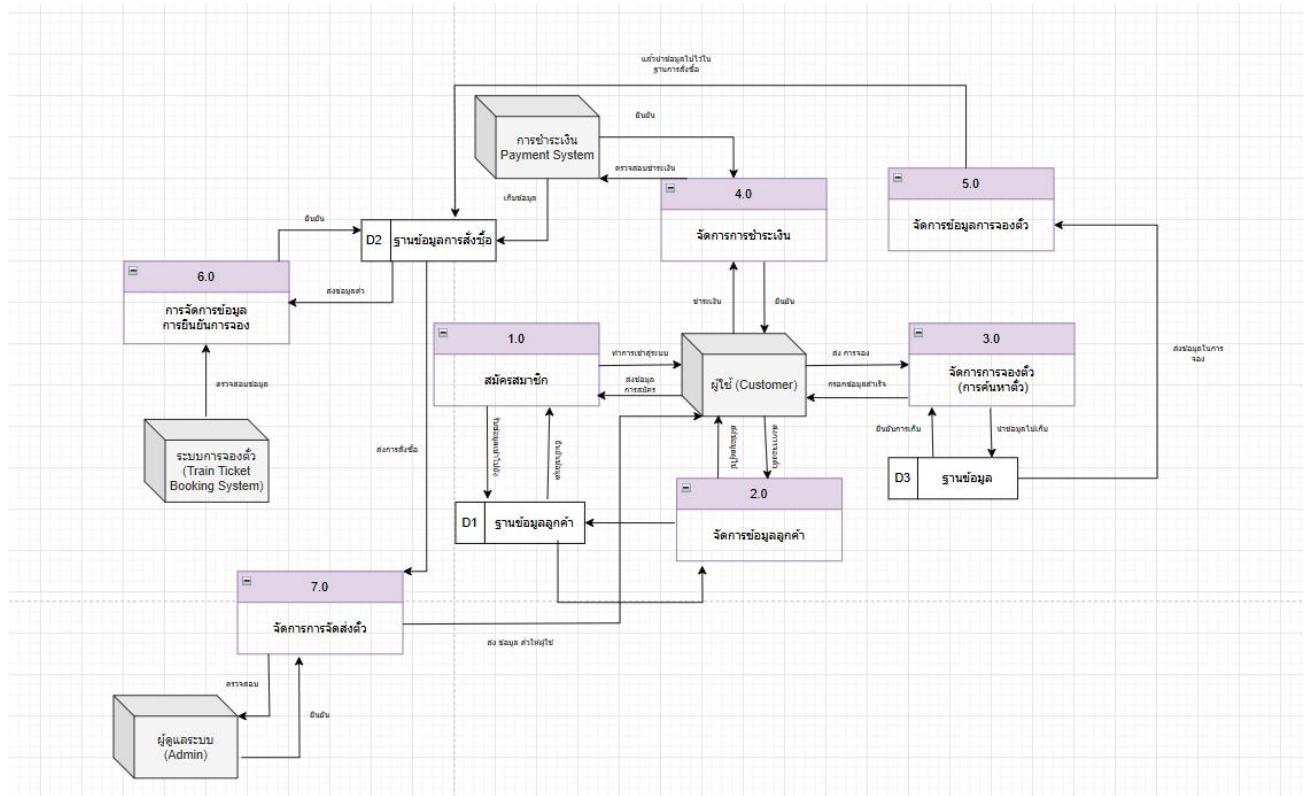
กระบวนการหลัก (Process): ระบบจองตั๋วรถไฟฟ้าออนไลน์ (Ticket Booking System): นี่คือกระบวนการหลักที่รับคำขอจากผู้ใช้และให้บริการ การจองตั๋วรถไฟฟ้า (Entity): ผู้ใช้ (User): ผู้โดยสารที่ต้องการจองตั๋วระบบการชำระเงิน (Payment Gateway): ระบบที่ใช้สำหรับการชำระเงินของผู้ใช้ เช่น การชำระเงินผ่านบัตรเครดิต หรือการโอนเงินออนไลน์ฐานข้อมูล (Database): ใช้ในการจัดเก็บข้อมูลตัวการจอง ข้อมูล ผู้ใช้ ข้อมูลการชำระเงิน และการยืนยันการจองระบบการยืนยัน (Confirmation System): ส่งข้อมูลการยืนยันการจองตัวไปยังผู้ใช้



รูปที่ 2.1.1 แสดง Context Diagram ของ ระบบจองตั๋วรถไฟฟ้าออนไลน์ นี้แสดงถึงการไหลของข้อมูลระหว่างลูกค้า, ระบบจองตั๋ว, ระบบชำระเงิน, และผู้ดูแลระบบ ซึ่งแสดงให้เห็นถึงการเชื่อมโยงระหว่างแต่ละส่วนประกอบของระบบและการติดต่อระหว่างผู้ใช้งานและระบบอย่างชัดเจน

2.2 แผนภาพการไหลของข้อมูล

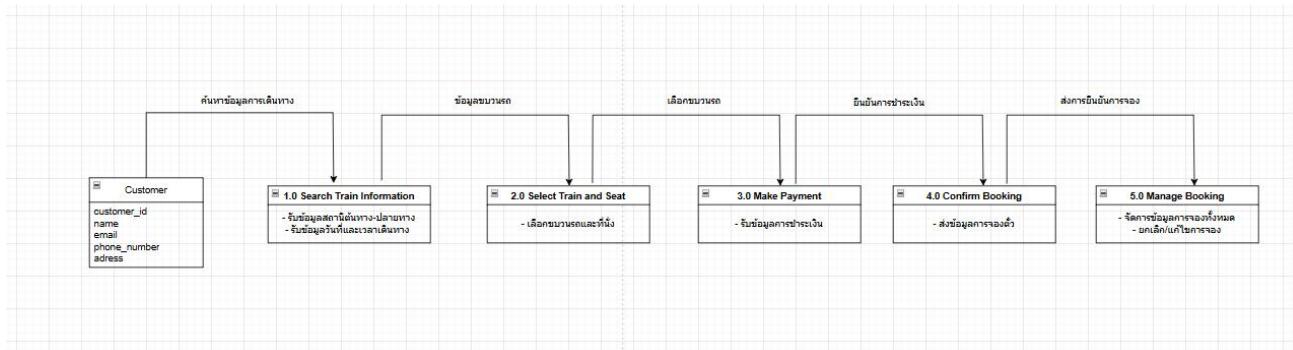
ระดับที่ 0 (Data Flow Diagram Level 0)



รูปที่ 2.2.1 แสดงแผนภาพ DFD Level 0 นี้แสดงให้เห็นภาพรวมของการทำงานของ ระบบจองตั๋วรถไฟฟ้าออนไลน์ โดยการไหลของข้อมูลระหว่าง ผู้ใช้ (Customer), ระบบจองตั๋ว (Train Ticket Booking System), ระบบชำระเงิน (Online Payment System), และ ผู้ดูแลระบบ (Admin) ซึ่งแสดงการโต้ตอบของแต่ละส่วนประกอบในระบบอย่างชัดเจน

2.3 แผนภาพการไหลของข้อมูล ระดับที่ 1 (Data Flow Diagram Level 1)

Data Flow Diagram (DFD) Level 1 สำหรับ ระบบจองตั๋วรถไฟฟ่อนไลน์ เป็นการขยายความจาก DFD Level 0 โดยแสดงถึงกระบวนการที่สำคัญในระบบ เช่น การค้นหาบวนรถ, การเลือกที่นั่ง, การชำระเงิน และการยืนยันการจอง โดยการให้ผลของข้อมูลในแต่ละกระบวนการจะมีความซัดเจนมากขึ้น



รูปที่ 2.3.1 DFD Level 1 นี้ช่วยขยายความจาก DFD Level 0 โดยแสดงรายละเอียดของกระบวนการหลักในระบบ จองตั๋วรถไฟฟ่อนไลน์ เช่น การค้นหาบวนรถ, การเลือกที่นั่ง, การชำระเงิน, การยืนยันการจอง และการจัดการข้อมูล การจอง. ทำให้เห็นภาพการไหลของข้อมูลภายในระบบและการโต้ตอบระหว่างผู้ใช้และระบบได้อย่างชัดเจนยิ่งขึ้น.

2.4 คำอธิบายกระบวนการ (Process Description)

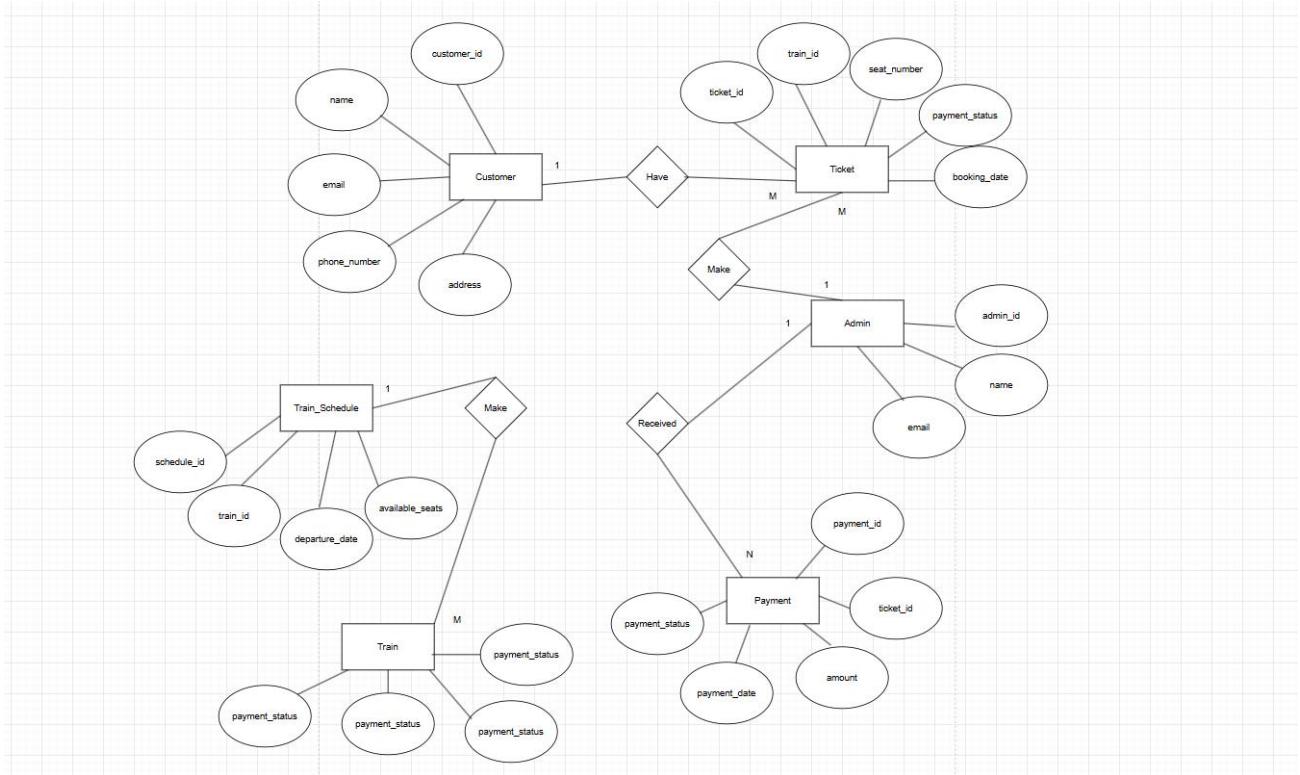
System : ระบบจองตั๋วรถไฟฟ้าออนไลน์
Process : 1. การเข้าสู่ระบบ
Date: 29 มกราคม 2568
Task of Activities:
1.1 สมัครสมาชิก
ลูกค้ากรอกข้อมูลส่วนตัวและสมัครสมาชิกในระบบ
1.2 เข้าสู่ระบบ (Log In)
ลูกค้าเลือกอินเทอร์เฟซที่มีความต้องการใช้งาน
1.3 ค้นหาเส้นทางและขบวนรถ (Search Route and Train)
ลูกค้าเลือกสถานที่เดินทาง, สถานีปลายทาง, และวันที่/เวลาเดินทาง
1.4 เลือกที่นั่ง (Select Seat)
ลูกค้าเลือกที่นั่งในห้องที่ต้องการ
1.5 เลือกวิธีการชำระเงิน (Select Payment Method)
ลูกค้าเลือกวิธีการชำระเงิน
1.6 ชำระเงิน (Make Payment)
ลูกค้ากรอกข้อมูลการชำระเงินและยืนยัน
1.7 ยืนยันการจอง (Confirm Booking)
ระบบแสดงการยืนยันการจองและข้อมูลต่างๆ
• ลูกค้า จะทำการสมัครสมาชิก, เข้าสู่ระบบ, และเลือกเส้นทางการเดินทางที่ต้องการ
• ระบบจะแสดงขบวนรถที่มีพารามิเตอร์ที่นั่งว่างให้ลูกค้าเลือก
• ลูกค้าจะเลือกที่นั่งและทำการชำระเงินผ่านช่องทางที่เลือก
• เมื่อชำระเงินแล้วเสร็จสิ้น ระบบจะยืนยันการจองและส่งทันทีให้แก่ลูกค้า

รูปที่ 2.4.1 คำอธิบายของกระบวนการ Process Description

จากรูป ระบบจองตั๋วรถไฟฟ้าออนไลน์ กระบวนการเริ่มต้นจากการเลือกประเภทบริการและการเข้าสู่ระบบ/การสมัครสมาชิก จากนั้นลูกค้าจะเลือกเส้นทางการเดินทางและขบวนรถที่ต้องการ พร้อมทั้งเลือกที่นั่งและทำการยืนยันการจอง หลังจากนั้นลูกค้าทำการชำระเงินและติดตามสถานะการจัดส่งตั๋วผ่านอีเมล ระบบปั้นช่วยให้ลูกค้าสามารถจัดการการจองตั๋วได้อย่างสะดวกและมีความปลอดภัยในการชำระเงิน

2.5 แผนภาพ E-R (Entity Relationship Diagram หรือ ER Diagram หรือ E-R Diagram)

เป็นแผนภาพที่ใช้สำหรับออกแบบจำลองข้อมูลที่แสดงถึงความสัมพันธ์ของข้อมูลที่ เกิดขึ้นในระบบ โดยแผนภาพประกอบด้วยเอนทิตี้ที่ แสดงความสัมพันธ์ของข้อมูลที่เกิดขึ้นใน ระบบทั้งหมด



รูปที่ 2.5.1 แผนภาพ E-R Diagram

หมายเหตุ จากรูป จะแสดงข้อมูลความสัมพันธ์ของระบบที่เกิดขึ้น

2.6 Database

User ตารางผู้ใช้

User_ID(PK)	First_Name	Last_Name	Email	Phone_number	Password	Address
U001	ธีระพงษ์	พอกมรรจกี	teerapong@gmail.com	0123456789	Password123	ตระง
U002	จริยา	ขอดราช	kaika@gmail.com	0991235676	Arisa@456	ตาก
U003	กิตติ	รัตนวงศ์	sasom@gmail.com	086745335	KittiPass12	เชียงใหม่
U004	สาริน	เจริญสุข	erin@gmail.com	0871562431	Sarin@789	ขอนแก่น
U005	วิภา	ใจดิชช	wipa@gmail.com	03644140	Wipa@321	ภูเก็ต
U006	สมเกียรติ	ช่วยเกิด	feddo@gmail.com	0987456123	NuttaPass99	นครปฐม
U007	พิมพ์	แก้ววนิ	keal@gmail.com	0923416578	Pim@456	ราชบุรี
U008	กานุวัฒน์	เรืองขาวัญ	123grrsz@gmail.com	0947852369	SiriPass78	กรุงเทพ
U009	แก้ววนิ	เกิดสุข	yuqza@gmail.com	0831562489	Song@pass	ศรีราษฎร์ ชานี
U010	พวงพาณ	แสงทอง	poooh@gmail.com	0809563471	Pawina@951	นครราชสีมา

รูปที่ 2.6.1 แสดงพจนานุกรมข้อมูลที่ใช้อธิบายรายละเอียดของแอทริบิวต์ ของเอนทิตี้ User

Routes ตารางเส้นทาง

route_id(PK)	start_station	end_station	distance (km)	duration (hours)
R001	ตัวจั่ง	กรุงเทพฯ	700	12
R002	นครศรีธรรมราช	แม่ริม	450	8
R003	เชียงใหม่	ชัยภูมิ	900	15
R004	ตาก	น่าน	130	2.5
R005	ลำปาง	เลย	650	11
R006	เลย	ลำปาง	120	10
R007	น่าน	ตาก	500	11
R008	ชัยภูมิ	เชียงใหม่	200	10
R009	แม่ริม	นครศรีธรรมราช	300	6
R010	ยะลา	ตัวจั่ง	400	5

รูปที่ 2.6.2 แสดงพจนานุกรมข้อมูลที่ใช้อธิบายรายละเอียดของแอทริบิวต์ ของอนทิตี Trains

Bookings ตารางการจองตั๋ว

booking_id (PK)	user_id (FK)	schedule_id (FK)	booking_date	seat_number	payment_status
B001	U001	S001	2025-02-05	12A	ชำระแล้ว
B002	U002	S002	2025-02-05	15B	รอค่าบินกวาง
B003	U003	S003	2025-02-06	05C	ชำระแล้ว
B004	U004	S004	2025-02-06	22D	ชำระแล้ว
B005	U005	S005	2025-02-07	09E	รอค่าบินกวาง
B006	U006	S006	2025-02-07	03A	ชำระแล้ว
B007	U007	S007	2025-02-08	10B	รอค่าบินกวาง
B008	U008	S008	2025-02-08	06C	ชำระแล้ว
B009	U009	S009	2025-03-21	17D	รอค่าบินกวาง
B010	U010	S010	2025-03-22	21E	ชำระแล้ว

รูปที่ 2.6.3 แสดงพจนานุกรมข้อมูลที่ใช้อธิบายรายละเอียดของแอทริบิวต์ ของอนทิตี Trains_Schedules

Payments ตารางการซื้อขายเงิน

payment_id(PK)	booking_id (FK)	payment_dated	amount	payment_method
P001	B001	2025-02-05	800	2025-02-05 20:00:00
P002	B002	2025-02-06	750	2025-02-05 17:30:00
P003	B003	2025-02-07	950	2025-02-06 22:00:00
P004	B004	2025-02-07	600	2025-02-06 09:00:00
P005	B005	2025-02-08	850	2025-02-07 21:00:00
P006	B006	2025-02-09	1000	2025-02-07 14:30:00
P007	B007	2025-02-09	900	2025-02-08 19:00:00
P008	B008	2025-02-05	700	2025-02-08 18:00:00
P009	B009	2025-02-05	800	2025-02-09 21:30:00
P010	B010	2025-02-08	950	2025-02-09 22:00:00

รูปที่ 2.6.4 แสดงพจนานุกรมข้อมูลที่ใช้อธิบายรายละเอียดของแอทริบิวต์ ของเอนทิตี้ Payments

ตารางที่ 2.7 แสดงพจนานุกรมข้อมูล

2.5 Data Dictionary

1. ตาราง Users (ผู้ใช้)

Field Name	Data Type	Description
user_id	INT	รหัสผู้ใช้ (Primary Key)
username	VARCHAR(50)	ชื่อผู้ใช้ (Unique)
password	VARCHAR(255)	รหัสผ่านผู้ใช้
email	VARCHAR(100)	อีเมลผู้ใช้ (Unique)
phone_number	VARCHAR(15)	หมายเลขโทรศัพท์ผู้ใช้
address	TEXT	ที่อยู่ของผู้ใช้
role	VARCHAR(20)	บทบาทของผู้ใช้ (เช่น admin, customer)

รูปที่ 2.7.1 แสดงพจนานุกรมข้อมูลที่ใช้อธิบายรายละเอียดของแอ็ตทริบิวต์ของเอนทิตี้ User

2. ตาราง Trains (ขบวนรถ)

Field Name	Data Type	Description
train_id	INT	รหัสขบวนรถ (Primary Key)
train_name	VARCHAR(100)	ชื่อขบวนรถ
departure_station	VARCHAR(100)	สถานีต้นทาง
arrival_station	VARCHAR(100)	สถานีปลายทาง
departure_time	DATETIME	เวลาที่ขบวนรถออกจากสถานีต้นทาง
arrival_time	DATETIME	เวลาที่ขบวนรถถึงสถานีปลายทาง
available_seats	INT	จำนวนที่นั่งที่ว่างในขบวนรถ

รูปที่ 2.7.2 แสดงพจนานุกรมข้อมูลที่ใช้อธิบายรายละเอียดของแอ็ตทริบิวต์ของเอนทิตี้ Trains

3. ตาราง Train_Schedules (ตารางการเดินรถ)

Field Name	Data Type	Description
schedule_id	INT	รหัสตารางเดินรถ (Primary Key)
train_id	INT	รหัสขบวนรถ (Foreign Key จากตาราง Trains)
departure_date	DATE	วันที่เดินทาง
available_seats	INT	จำนวนที่นั่งที่ยังว่างในตารางนี้

รูปที่ 2.7.3 แสดงพจนานุกรมข้อมูลที่ใช้อธิบายรายละเอียดของแอตทริบิวต์ของเอนทิตี้ Train_Schedules

4. ตาราง Tickets (ต่อ)

Field Name	Data Type	Description
ticket_id	INT	รหัสตั๋ว (Primary Key)
user_id	INT	รหัสผู้ใช้ที่ทำการจอง (Foreign Key จาก Users)
train_id	INT	รหัสขบวนรถที่ผู้ใช้เลือก (Foreign Key จาก Trains)
schedule_id	INT	รหัสตารางการเดินรถ (Foreign Key จาก Train_Schedules)
seat_number	VARCHAR(10)	หมายเลขที่นั่งที่เลือก
booking_date	DATETIME	วันที่และเวลาที่ทำการจอง
payment_status	VARCHAR(20)	สถานะการชำระเงิน (เช่น 'paid', 'pending')
booking_status	VARCHAR(20)	สถานะการจอง (เช่น 'confirmed', 'cancelled')

รูปที่ 2.7.4 แสดงพจนานุกรมข้อมูลที่ใช้อธิบายรายละเอียดของแอตทริบิวต์ของเอนทิตี้ Tickets

5. ตาราง Payments (การชำระเงิน)

Field Name	Data Type	Description
payment_id	INT	รหัสการชำระเงิน (Primary Key)
ticket_id	INT	รหัสตั๋วที่ชำระ (Foreign Key จาก Tickets)
payment_date	DATETIME	วันที่และเวลาที่ทำการชำระเงิน
amount	DECIMAL(10, 2)	จำนวนเงินที่ชำระ
payment_status	VARCHAR(20)	สถานะการชำระเงิน (เช่น 'successful', 'failed')

รูปที่ 2.7.5 แสดงพจนานุกรมข้อมูลที่ใช้อธิบายรายละเอียดของแอ็ตทริบิวต์ของเอนทิตี้ Payments

6. ตาราง Train_Stations (สถานีรถไฟ)

Field Name	Data Type	Description
station_id	INT	รหัสสถานี (Primary Key)
station_name	VARCHAR(100)	ชื่อสถานี
city	VARCHAR(100)	เมืองที่สถานีตั้งอยู่

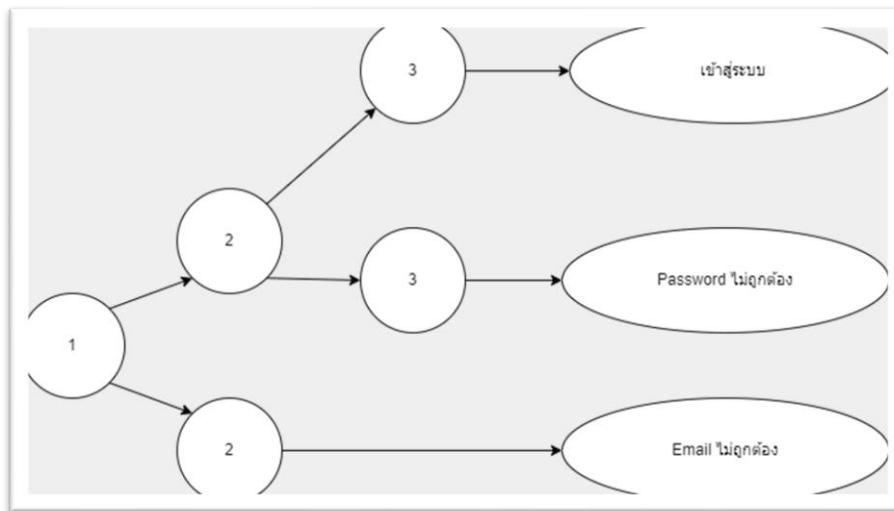
รูปที่ 2.7.6 แสดงพจนานุกรมข้อมูลที่ใช้อธิบายรายละเอียดของแอ็ตทริบิวต์ของเอนทิตี้ Train_Stations

7. ตาราง Admin (ผู้ดูแลระบบ)

Field Name	Data Type	Description
admin_id	INT	รหัสผู้ดูแลระบบ (Primary Key)
admin_name	VARCHAR(100)	ชื่อผู้ดูแลระบบ
email	VARCHAR(100)	อีเมลของผู้ดูแลระบบ
password	VARCHAR(255)	รหัสผ่านของผู้ดูแลระบบ
role	VARCHAR(20)	บทบาทของผู้ดูแลระบบ (เช่น admin)

รูปที่ 2.7.7 แสดงพจนานุกรมข้อมูลที่ใช้อธิบายรายละเอียดของแอ็ตทริบิวต์ของเอนทิตี้ Admin

2.8 ต้นไม้การตัดสินใจ (Decision Tree)



ต้นไม้การตัดสินใจ (Decision Tree)

รูปที่ 2.8.1 แสดงรูปแบบการเขียนต้นการตัดสินใจของกระบวนการ login

จากรูป กรอกข้อมูล Email หรือ Password ผิด ระบบจะไม่อนุญาตให้เข้าสู่ระบบได้ และเงื่อนไขในการเข้าสู่ระบบก็คือ ต้องกรอก ทั้ง Email และ Password ให้ผ่าน จึงจะเข้าสู่ระบบได้

2.9 ตารางการตัดสินใจ (Decision Table)

ตารางที่ 2.9.1 ตารางการตัดสินใจของร้านขายสินค้าสำหรับสัตว์เลี้ยง

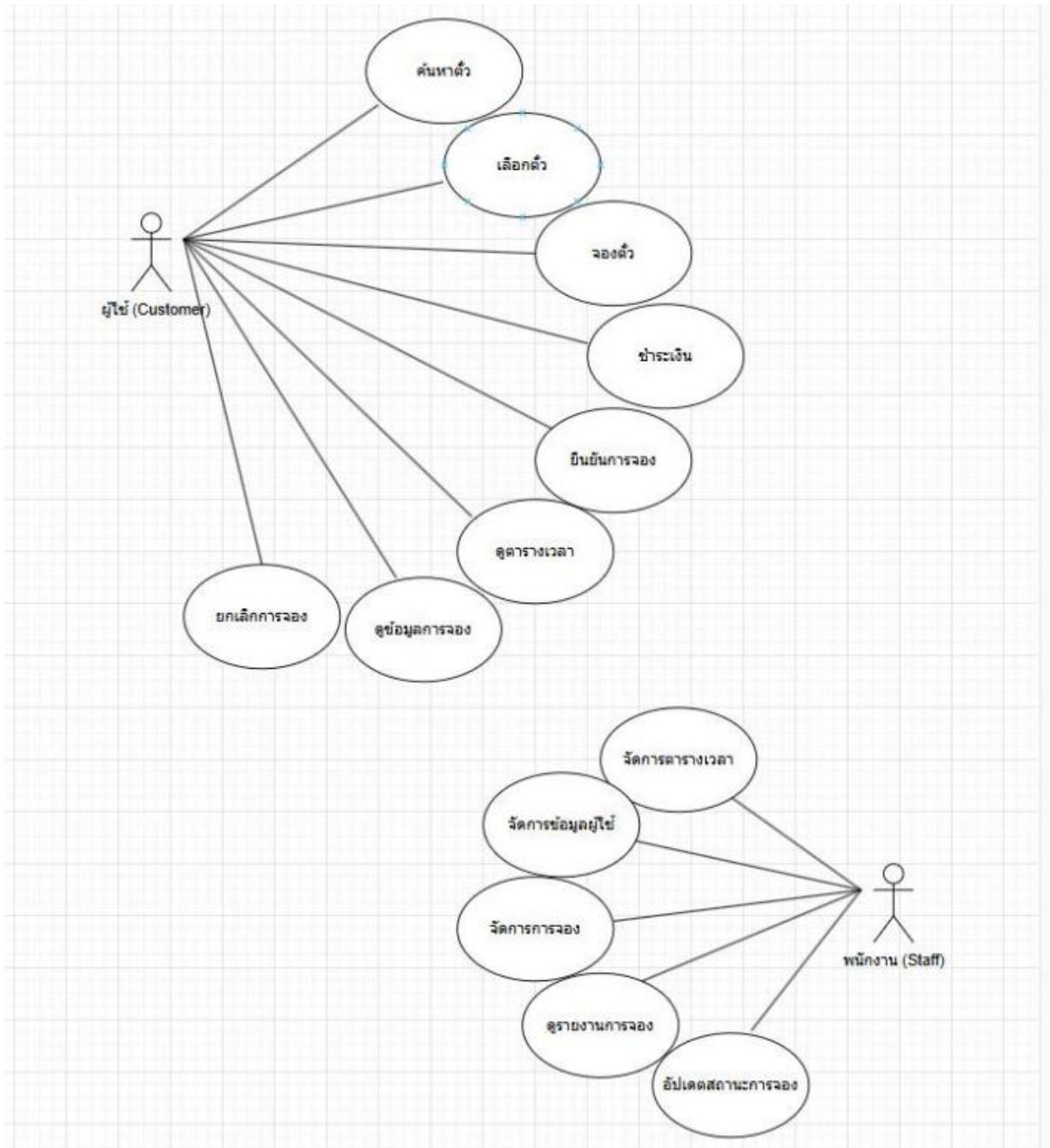
เงื่อนไข	หลักเกณฑ์							
	1	2	3	4	5	6	7	8
Email ถูก	Y	Y	N	N	Y	Y	N	N
Password ถูก	Y	N	Y	N	Y	N	Y	N
เข้าสู่ระบบ	X				X			
Email ผิด			X	X			X	X
Password ผิด		X				X		

ตารางที่ 2.9.2 ตารางการตัดสินใจของร้านขายสินค้าสำหรับสัตว์เลี้ยงที่มีการรวมเงื่อนไข

เงื่อนไข	หลักเกณฑ์		
	1'	2'	3'
Email ถูก	Y	N	Y
Password ถูก	Y	-	Y
เข้าสู่ระบบ	X		
Email ผิด		X	
Password ผิด			X

3. การวิเคราะห์ และ ออกแบบระบบเชิงวัตถุด้วย UML

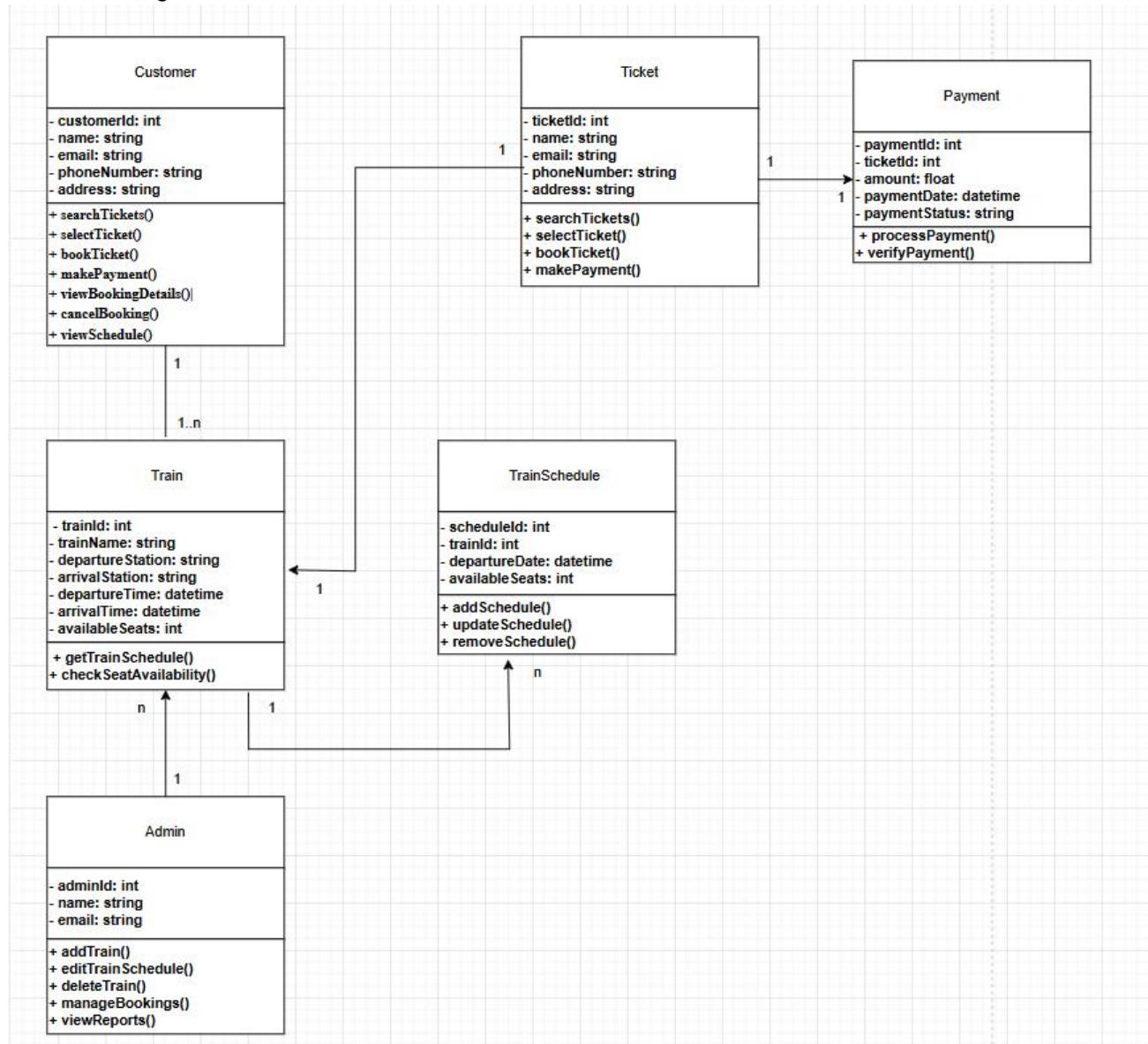
3.1 Use case Diagram



รูปที่ 3.1.1แสดงรูปแบบของความสัมพันธ์ระหว่างลูกค้ากับพนักงาน

จากรูป แผนภาพที่ใช้ในการแสดงฟังก์ชันหรือบทบาทที่ผู้ใช้ (users) สามารถทำได้ในระบบซึ่ง มักจะ ถูกใช้ในขั้นตอนการวางแผนและออกแบบระบบโดยให้เพิกถอนที่ความต้องการและประโยชน์ ของผู้ใช้จริง ซึ่งบทบาทของลูกค้ามีทั้งการสมัครสมาชิกเลือกซื้อสินค้าสั่งซื้อสินค้าและชำระเงิน บทบาทของ พนักงานมีทั้งบันทึกรายการสินค้าจัดส่งสินค้าและบันทึกยอดสินค้ารายการต่างๆ

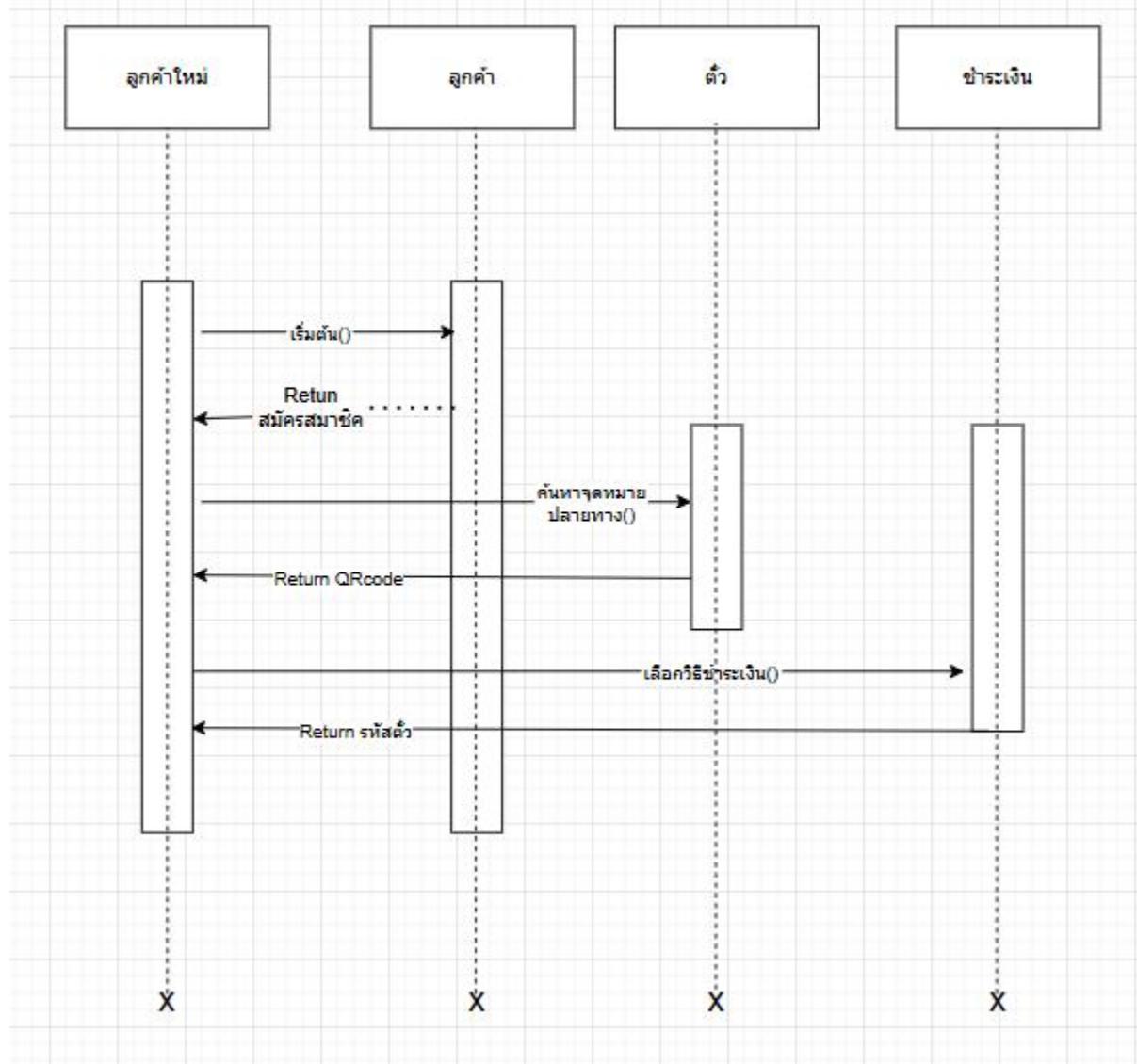
3.2 Class diagram



รูปที่ 3.2.1 แสดง Class Diagram ของการจองตั๋วรถไฟฟ้าออนไลน์

แผนภาพ Class diagram เป็นหนึ่งในแผนภาพที่ใช้ในการแสดงโครงสร้างของคลาสและ ความสัมพันธ์ระหว่างคลาสในระบบซึ่งการเขียนโดยง่ายให้ผู้ที่มีส่วนเกี่ยวข้องในการพัฒนาโปรแกรมหรือระบบสารสนเทศเข้าใจความสัมพันธ์ระหว่างคลาสและ ความสัมพันธ์ที่เกิดขึ้นในระบบโดยง่ายขึ้น

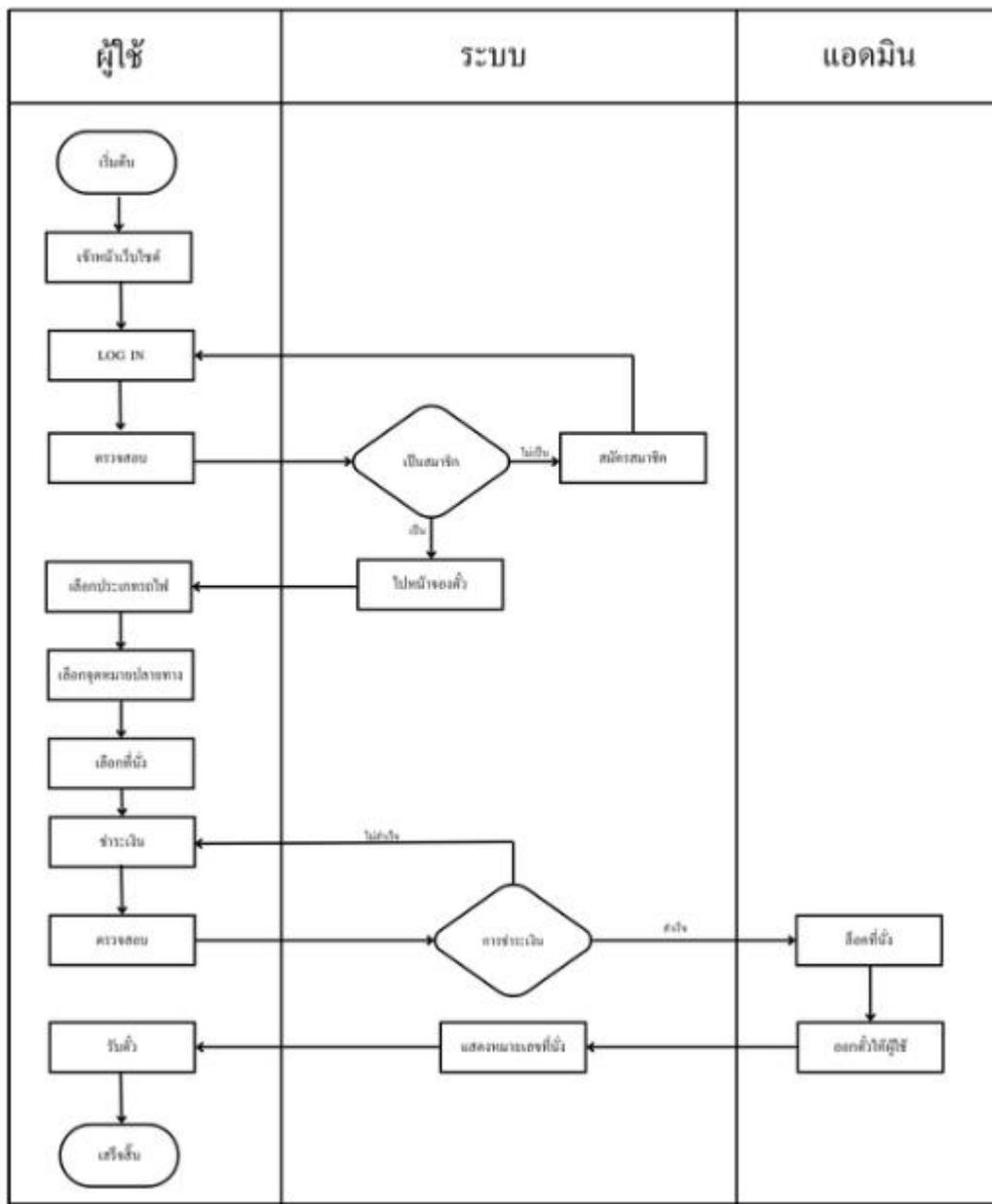
3.3 Sequence Diagram



รูปที่ 3.3.1 แสดง Sequence Diagram ของการจองตั๋วรถไฟออนไลน์

จากรูป Sequence diagram แผนภาพลำดับ (Sequence Diagram) นี้ช่วยให้เห็นขั้นตอนและการไหลของข้อมูลระหว่างผู้ใช้ ระบบการจองตั๋ว ระบบการชำระเงิน และระบบการยืนยันการจอง เพื่อให้เข้าใจถึงการทำงานภายในระบบจองตั๋วรถไฟออนไลน์ตั้งแต่ต้นจนจบ รวมถึงการประมวลผลต่างๆ ที่เกิดขึ้นระหว่างขั้นตอนต่างๆ

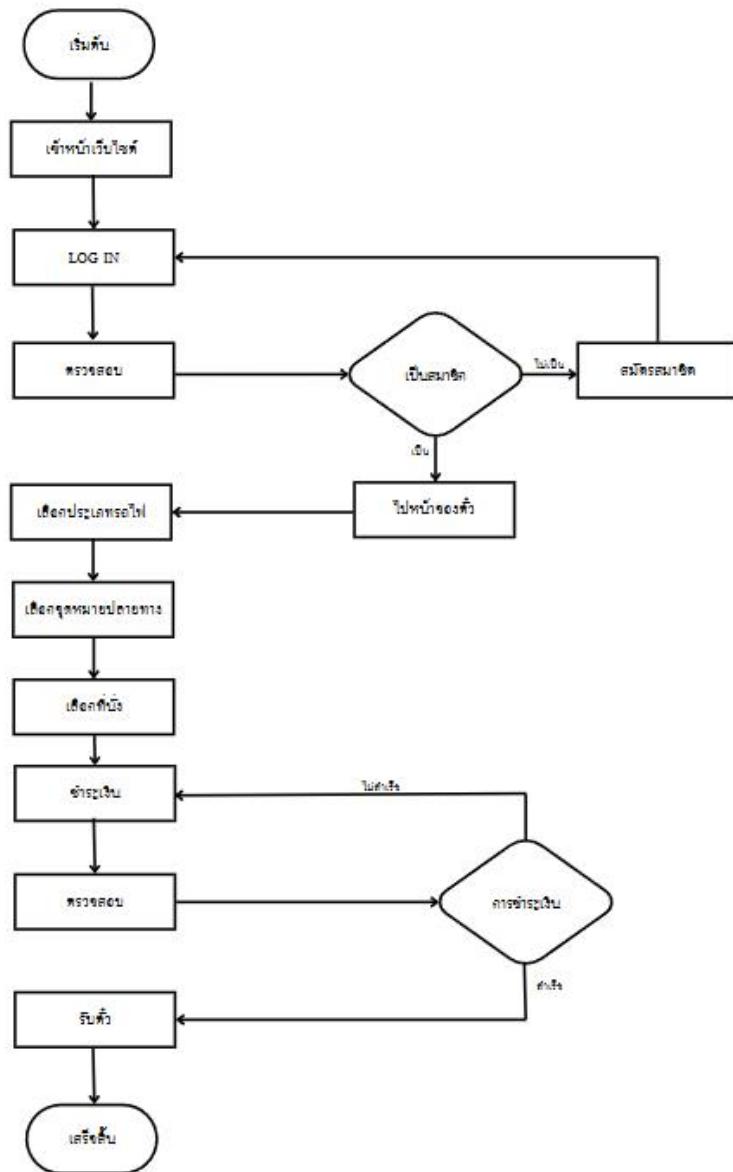
3.4 Activity Diagram



รูปที่ 3.4.1 แสดง Activity Diagram เกี่ยวกับกระบวนการจองตั๋ว

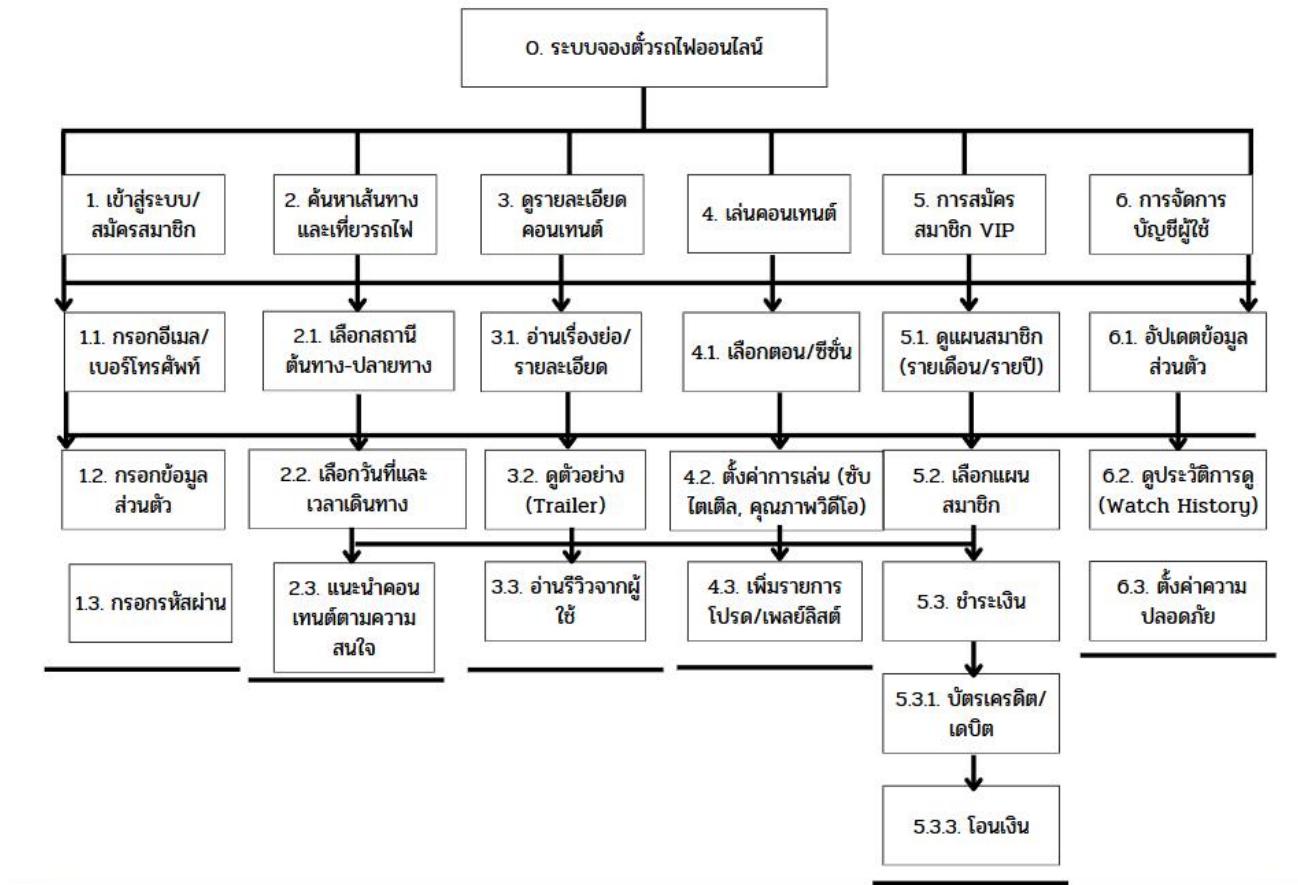
แผนภาพ Activity Diagram สำหรับ ระบบจองตั๋วรถไฟฟ้าออนไลน์ จะแสดงขั้นตอนและกิจกรรมที่ผู้ใช้ต้องดำเนินการจากการเลือกตั๋วจนถึงการได้รับการยืนยันการจองตั๋ว

3.5 Flow Chart ระบบจองตั๋ว



รูปที่ 3.5.1 แสดง Flow Chart ของระบบสั่งสินค้าเกี่ยวกับสัตว์เลี้ยงออนไลน์
จากรูป ในระบบสั่งสินค้า เป็นแผนภาพที่ใช้ในการแสดงกระบวนการหรือลำดับ ขั้นตอนที่ต้องทำ เมื่อมีการซื้อสินค้าใน
ระบบแสดงลำดับการซื้อสินค้าตั้งแต่การ สมัครสมาชิกเพื่อเข้าใช้ระบบ

3.6 HTA

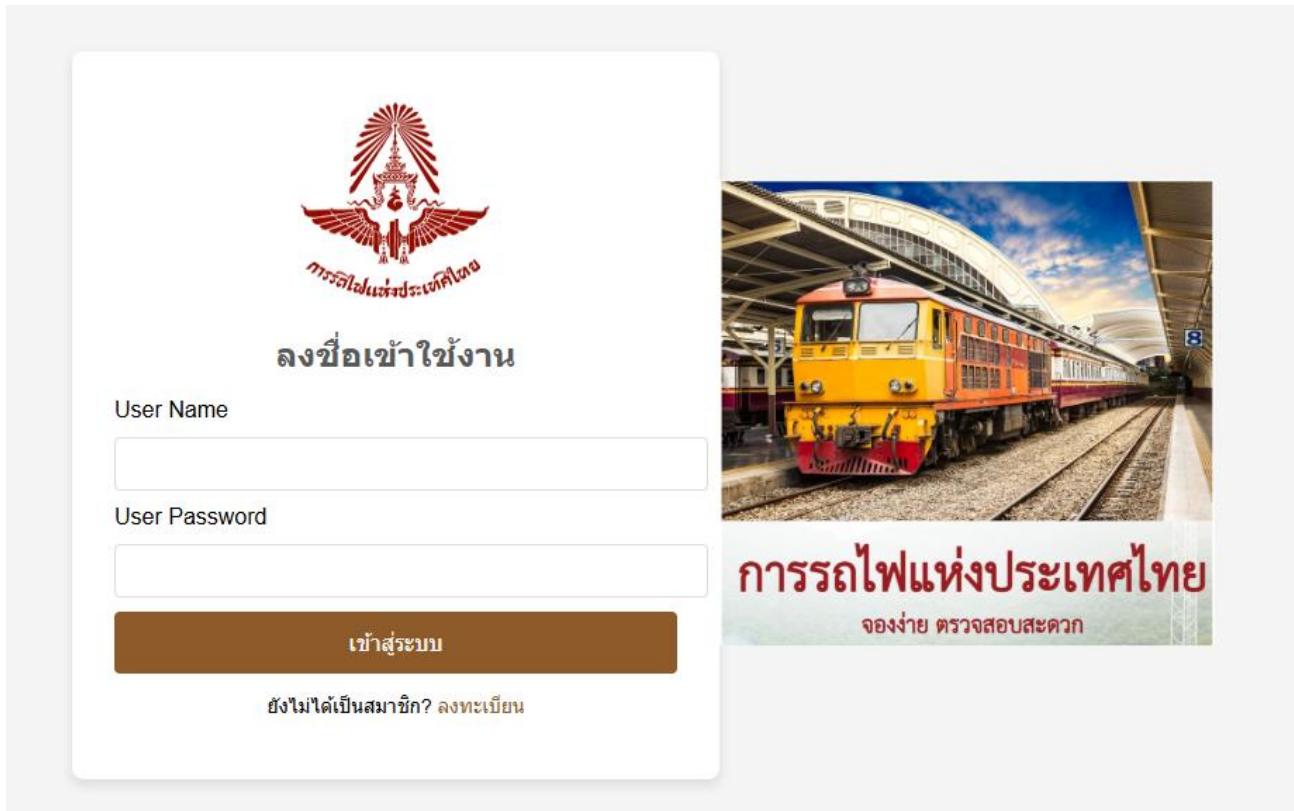


รูปที่ 3.6.1 แสดง HTA ของระบบสั่งสินค้าเกี่ยวกับสัตว์เลี้ยงออนไลน์

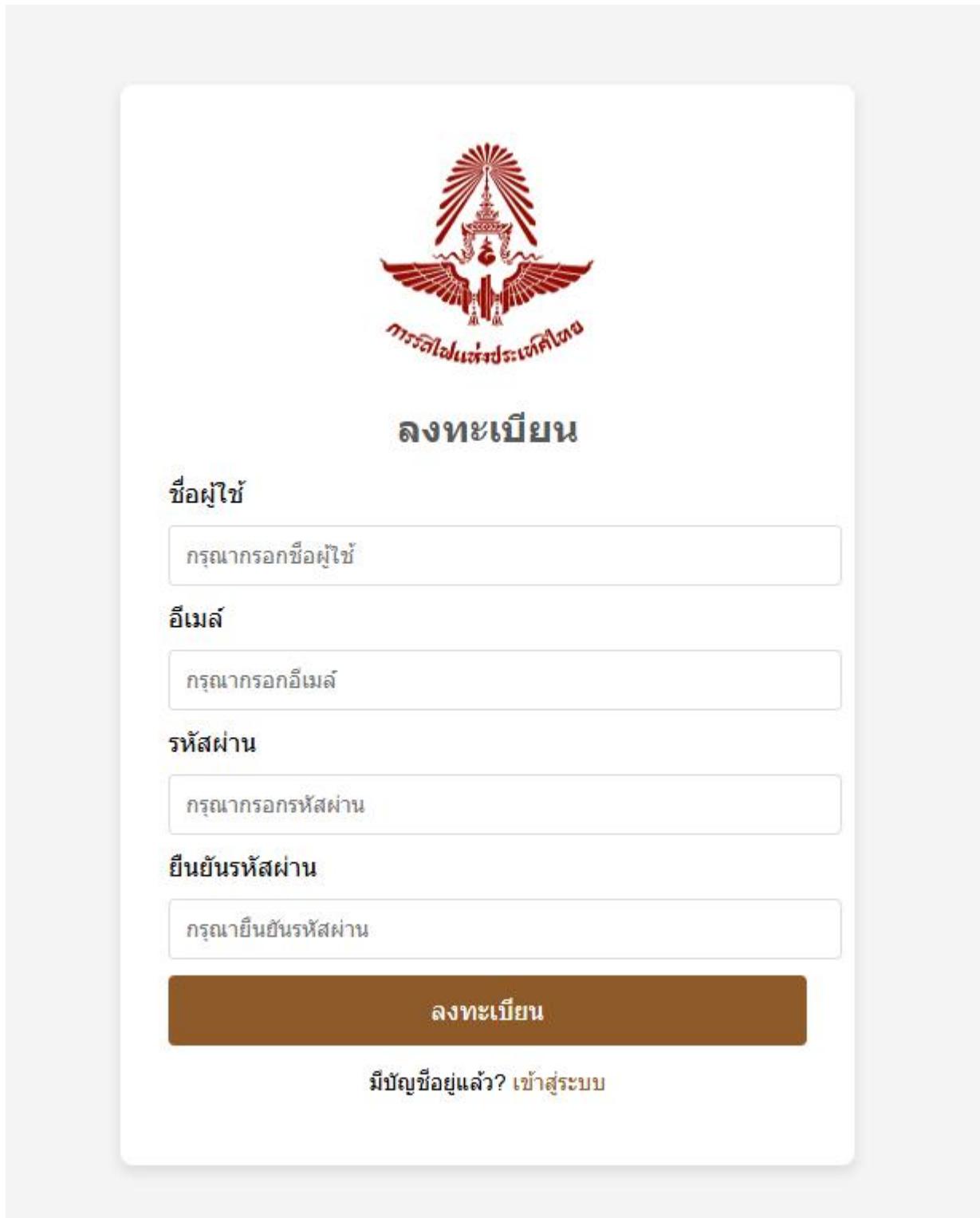
จากรูป แผนภาพ Diagrammatic HTA แสดงถึงขั้นตอนการวิเคราะห์และอธิบายกระบวนการ หรืองานที่ต้องทำอย่างละเอียด โดยแบ่งงานหรือกิจกรรมออกเป็นระดับย่อย จนถึงระดับที่เล็กที่สุด เพื่อทำความเข้าใจในกระบวนการต่างๆ ที่เกิดขึ้นในการดำเนินงาน

บทที่ 4 การออกแบบส่วนต่อประสานผู้ใช้ (Graphical User Interface)

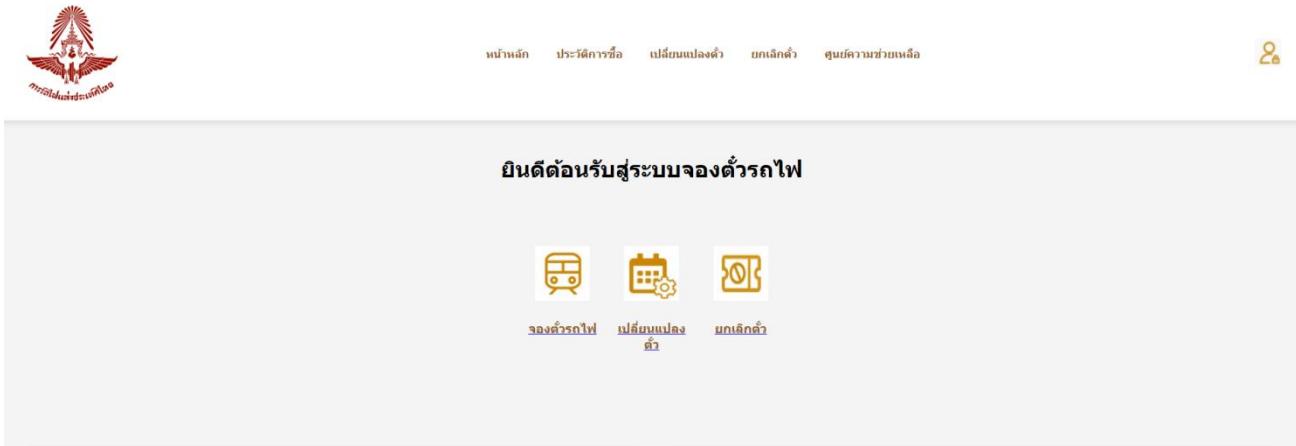
4.1 หน้าแรกใน ที่เปิด เว็บไซต์ขึ้นมา



รูปที่ 4.1.1 ลงชื่อเข้าใช้ or เข้าสู่ระบบ จะไปที่หน้าหน้า



รูปที่ 4.1.2 หน้าที่ หากไม่บัญชีก็จะเข้าสู่หน้าลงทะเบียน



รูปที่ 4.1.3 หน้าหลักเมื่อ ทำการ login เข้ามาแล้ว



รูปที่ 4.1.4 เมื่อเลื่อนมาส์ลลงมาในหน้าแรกก็จะเจอประกาศเทศการณ์ต่างๆ



รูปที่ 4.1.5 เมื่อกดตรงไอคอนหรือข้อความ “จองตั๋วรถไฟ”

จองตัวรถไฟ

สถานีต้นทาง

กรุณากรอกสถานีต้นทาง

สถานีปลายทาง

กรุณากรอกสถานีปลายทาง

วันที่เดินทาง

วว/ดด/ปปปป

□

เวลา

--:--

⌚

จองตัว

ชำระเงินสำหรับการจองตั๋ว

สถานีต้นทาง: กรุงเทพ

สถานีปลายทาง: เชียงใหม่

วันที่เดินทาง: 10 กุมภาพันธ์ 2568

เวลา: 09:00 น.

จำนวนเงินที่ต้องชำระ: 500 บาท



ยืนยันการชำระเงิน

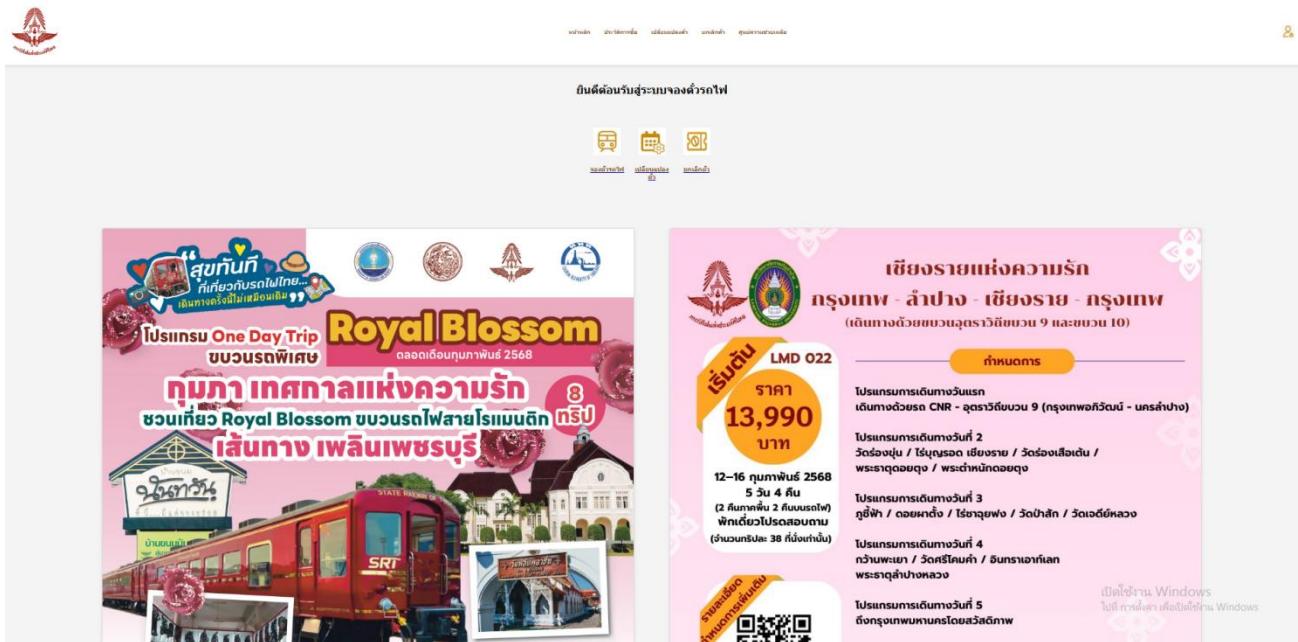
ยกเลิก

รูปที่ 4.1.7 เมื่อข้อมูลเสร็จให้ก็จะขึ้นหน้า ชำระเงินขึ้นมาแล้ว แสดง QRcode

ข่าวสารเว็บแล้ว

ไปยังหน้าหลัก

รูปที่ 4.1.8 เมื่อกดปุ่ม คำะเงิน ก็จะขึ้น หน้าต่างนี้



รูปที่ 4.1.9 เมื่อกดก็จะกลับมาหน้าหลักนั่นเอง

ประวัติการซื้อ				
หมายเลขเอกสาร	ชื่อผู้โดยสาร	สถานที่-ปลายทาง	วันที่และเวลา	สถานะการจัดส่ง
#12345	นายสมชาย วันพีร์สว่าง	กรุงเทพ → เชียงใหม่	10 กุมภาพันธ์ 2568 เวลา 09:00 น.	กำลังจัดส่ง
#12346	นางสาวสมศรี ศรีสวัสดิ์	กรุงเทพ → หัวหิน	10 กุมภาพันธ์ 2568 เวลา 12:00 น.	รอจัดส่ง

รูปที่ 4.1.10 เมื่อกด navbar ที่คำว่า “ประวัติการซื้อ” ก็จะปรากฏหน้าต่างนี้ขึ้นมา



รูปที่ 4.1.10 เมื่อกดที่รูป หรือ ข้อความ เปลี่ยนแปลงตัว

เปลี่ยนแปลงตัวรถไฟ

กรอกข้อมูลใหม่เพื่อรับการเปลี่ยนแปลงตัว

สถานที่เดินทาง	<input type="text" value="กรุงเทพมหานคร/เชียงใหม่"/>	
สถานที่ปลายทาง	<input type="text" value="กรุงเทพมหานคร/เชียงใหม่"/>	
วันที่เดินทาง	<input type="text" value="วัน/เดือน/ปี"/> <input type="button" value=""/>	
เวลา	<input type="text" value="--::--"/> <input type="radio"/>	
<input type="button" value="บันทึกการเปลี่ยนแปลง"/>		

หมายเลขการจด	ชื่อผู้โดยสาร	เดินทาง-ไปยังท่า	วันที่และเวลา	สถานะ	จัดการ
#12345	นางสมชาย จันทร์สว่าง	กรุงเทพ — เชียงใหม่	10 กุมภาพันธ์ 2568 เวลา 09:00 น.	รอการยืนยัน	<input type="button" value="บันทึก"/> <input type="button" value="ยกเลิก"/>
#12346	นางสาวสมรรช์ ศรีสวัสดิ์	กรุงเทพ — หัวหิน	10 กุมภาพันธ์ 2568 เวลา 12:00 น.	รอการยืนยัน	<input type="button" value="บันทึก"/> <input type="button" value="ยกเลิก"/>

ในส่วนของการเดินทางท่องเที่ยว
ไม่ใช่การเดินทางทางการ

รูปที่ 4.1.11 ก็จะขึ้นหน้า การเปลี่ยนแปลงตัว จะมีประวัติตัวที่ได้ซื้อไปก่อนหน้านี้

เปลี่ยนแปลงตัวรถไฟ

กรอกข้อมูลใหม่สำหรับการเปลี่ยนแปลงตัว

สถานีต้นทาง

กรุงเทพ

สถานีปลายทาง

เชียงใหม่

วันที่เดินทาง

๒๖/๐๘/๒๕๖๖



เวลา

--:--



ยืนยันการเปลี่ยนแปลง

รูปที่ 4.1.12 เมื่อกดปุ่มเปลี่ยนแปลงที่ตัวอันได้อนหนึ่งข้อมูลก็จะปรากฏในหน้าต่างนี้ให้ทำการแก้

#12345	นายสมชาย รัตนพรสว่าง	กรุงเทพ → เชียงใหม่	10 กุมภาพันธ์ 2568 เวลา 09:00 น.	เปลี่ยนแปลงดำเนิน	บล็อก ยกเลิก
#12346	นางสาวสมศรี ศรีสวัสดิ์	กรุงเทพ → หัวหิน	10 กุมภาพันธ์ 2568 เวลา 12:00 น.	รายการเป็นปัจจุบัน	บล็อก ยกเลิก

รูปที่ 4.1.13 เมื่อทำการเปลี่ยนข้อมูล ข้อความจะปรากฏว่าทำการเปลี่ยนแปลงแล้ว



รูปที่ 4.1.14 ในการยกเลิกตัว สามารถทำได้ คือ กดตรงไอค่อน หรือข้อความ “ยกเลิกตัว”

ยกเลิกตั๋วรถไฟ					
หมายเลขการจอง	ชื่อผู้โดยสาร	สถานที่เดินทาง	วันที่เดินทาง	สถานะ	จัดการ
#12345	นายสมชาย จันทร์สว่าง	กรุงเทพ → เชียงใหม่	10 กุมภาพันธ์ 2568 เวลา 09:00 น.	รอการยืนยัน	<button>ยืนยันการยกเลิก</button> <button>ยกเลิก</button>
#12346	นางสาวสมศรี ศรีสวัสดิ์	กรุงเทพ → เชียงใหม่	10 กุมภาพันธ์ 2568 เวลา 12:00 น.	รอการยืนยัน	<button>ยืนยันการยกเลิก</button> <button>ยกเลิก</button>

รูปที่ 4.1.15 จะปรากฏหน้าต่าง มาให้สามารถกดยกเลิกตัวที่ทำการเมื่อครั้งก่อน

หมายเลขรายการ	ชื่อผู้โดยสาร	ต้นทาง-ปลายทาง	วันเดือนปี	สถานะ	จัดการ
#12345	นายสมชาย จันทร์สว่าง	กรุงเทพ → เชียงใหม่	10 กุมภาพันธ์ 2568 เวลา 09:00 น.	ยกเลิกสำเร็จ	
#12346	นางสาวสมศรี ศรีสุวรรณ	กรุงเทพ → พัทยา	10 กุมภาพันธ์ 2568 เวลา 12:00 น.	ออกตั๋วบิน	ยกเว้นการยกเลิก ยกเลิก

รูปที่ 4.1.16 เมื่อกดยืนยันการยกเลิก สถานะก็จะแสดง “ยกเลิกสำเร็จ”

ยกเลิกตัวรถไฟ					
หมายเลขการจอง	ชื่อผู้โดยสาร	ต้นทาง-ปลายทาง	วันที่และเวลา	สถานะ	จัดการ
#12346	นางสาวสมศรี ศรีสวัสดิ์	กรุงเทพ → พัทยา	10 กุมภาพันธ์ 2568 เวลา 12:00 น.	รอการยืนยัน	ยืนยันการยกเลิก ยกเลิก

รูปที่ 4.1.17 และรายการก็จะหายไปจากประวัติการจอง

ศูนย์ความช่วยเหลือ

รูปที่ 4.1.18 เมื่อกดที่ปุ่มจะเข้าไปหน้า ศูนย์ให้ความเชื่อเหลือเมื่อลูกค้าติดปัญหาอะไร



หน้าหลัก ประวัติการซื้อ เปลี่ยนแปลงครั้ง ยกเลิกครั้ง ศูนย์ความช่วยเหลือ



รูปที่ 4.1.19 ตรง narbar ก็สามารถกด คำสั่งต่างได้เหมือนกัน



รูปที่ 4.1.19 เป็น icon ในการเปลี่ยนรายละเอียดของ Profile เมื่อกดเข้าไป

รายละเอียดส่วนบุคคล

ข้อมูลส่วนตัว

ชื่อผู้ใช้
สมชาย จันทร์สว่าง

อีเมล
somchai@example.com

เบอร์โทรศัพท์
012-345-6789

รหัสผ่าน
.....

[บันทึกการเปลี่ยนแปลง](#)

[ออกจากระบบ](#)

รูปที่ 4.1.20 เมื่อกดเข้ามา ก็จะเจอกับหน้านี้ และสามารถทำการเปลี่ยนแปลงข้อมูลได้

หน้านี้บอกร้า
ข้อมูลส่วนบุคคลของคุณถูกบันทึกเรียบร้อยแล้ว!



รูปที่ 4.1.21 ก็จะขึ้นข้อความเตือนว่าได้ทำการเปลี่ยนข้อมูลแล้ว



รูปที่ 4.1.22 กดปุ่มออกจากระบบเพื่อออกไปยังหน้า login

บทที่ 5 Code และการเขียนโปรแกรม (Programming)

```

5 login.html > ...
1   <!DOCTYPE html>
2   <html lang="th">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Login Page</title>
7       <link rel="stylesheet" href="styles.css">
8   </head>
9   <body>
10      <div class="login-container">
11          <div class="login-box">
12              <div class="logo">
13                  
14              </div>
15              <h1>ลงชื่อเข้าใช้งาน</h1>
16              <form action="home.html" method="POST">
17                  <label for="username">User Name</label>
18                  <input type="text" id="username" name="username" placeholder="" required>
19                  <label for="password">User Password</label>
20                  <input type="password" id="password" name="password" placeholder="" required>
21                  <button type="submit">เข้าสู่ระบบ</button>
22              </form>
23              <p>ยังไม่ได้เป็นสมาชิก? <a href="register.html">ลงทะเบียน</a></p>
24          </div>
25          <div class="login-image">
26              
27          </div>
28      </body>
29  </html>
30

```

รูปที่ 5.1.1 โค้ดนี้เป็นหน้าเข้าสู่ระบบที่ผู้ใช้สามารถกรอกชื่อผู้ใช้และรหัสผ่านเพื่อเข้าสู่ระบบของเว็บไซต์ ระบบจะส่งข้อมูลที่กรอกไปยังหน้า home.html เมื่อการเข้าสู่ระบบเสร็จสมบูรณ์ โดยมีการแสดงภาพโลโก้และภาพเพิ่มเติมเพื่อทำให้หน้าจอมีความสวยงามและใช้งานได้สะดวก.

```

register.html > ...
1  <!DOCTYPE html>
2  <html lang="th">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>ลงทะเบียน</title>
7      <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10     <div class="login-container">
11         <div class="login-box">
12             <div class="logo">
13                 
14             </div>
15             <h1>ลงทะเบียน</h1>
16             <form action="home.html" method="POST">
17                 <label for="username">ชื่อผู้ใช้</label>
18                 <input type="text" id="username" name="username" placeholder="กรุณากรอกชื่อผู้ใช้" required>
19
20                 <label for="email">อีเมล</label>
21                 <input type="email" id="email" name="email" placeholder="กรุณากรอกอีเมล" required>
22
23                 <label for="password">รหัสผ่าน</label>
24                 <input type="password" id="password" name="password" placeholder="กรุณากรองรหัสผ่าน" required>
25
26                 <label for="confirm-password">ยืนยันรหัสผ่าน</label>
27                 <input type="password" id="confirm-password" name="confirm-password" placeholder="กรุณาเย็บยันรหัสผ่าน" required>
28
29                 <button type="submit">ลงทะเบียน</button>
30             </form>
31             <p>มีปัญหาอย่างไร? <a href="login.html">เข้าสู่ระบบ</a></p>
32         </div>
33     </div>
34  </body>
35  </html>
36

```

รูปที่ 5.1.2 โค้ดนี้เป็นหน้าเมื่อกด การลงทะเบียน (Registration) ซึ่งผู้ใช้สามารถกรอกข้อมูลส่วนตัว เช่น ชื่อผู้ใช้, อีเมล, รหัสผ่าน และยืนยันรหัสผ่าน จากนั้นกดปุ่ม "ลงทะเบียน" เพื่อส่งข้อมูลไปยังหน้า home.html เพื่อเข้าสู่ชั้นตอนถัดไป หากผู้ใช้มีบัญชีแล้ว สามารถเข้าสู่ระบบผ่านลิงก์ที่มีอยู่ที่ด้านล่างของฟอร์ม.

```

5 home.html > html > body > main > div.options > div.option > a
1  <!DOCTYPE html>
2  <html lang="th">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>หน้าหลัก</title>
7      <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10     <header>
11         <div class="logo">
12             
13         </div>
14         <nav>
15             <ul>
16                 <li><a href="home.html">หน้าหลัก</a></li>
17                 <li><a href="purchase-history.html">ประวัติการซื้อ</a></li>
18                 <li><a href="change-ticket.html">เปลี่ยนแปลงตั๋ว</a></li>
19                 <li><a href="cancel-ticket.html">ยกเลิกตั๋ว</a></li>
20                 <li><a href="help-center.html">ศูนย์ความช่วยเหลือ</a></li>
21             </ul>
22         </nav>
23         <div class="user">
24             <a href="personal-details.html"></a>
25         </div>
26     </header>
27
28     <main>
29         <h1>ยินดีต้อนรับสู่ระบบจองตั๋วรถไฟ</h1>
30
31         <div class="options">
32             <div class="option">
33                 <a href="book-ticket.html">
34                     
35                     <h3>จองตั๋วรถไฟ</h3>
36                 </a>

```

รูปที่ 5.1.3 โดยนี้แสดงหน้า หน้าหลัก (Home Page) สำหรับระบบจองตั๋วรถไฟออนไลน์ ซึ่งประกอบด้วย:
โลโก้อของระบบ ที่แสดงที่ด้านบน

เมนูนำทาง ที่ให้ผู้ใช้สามารถเข้าถึงหน้าอื่นๆ เช่น การซื้อประวัติการซื้อตั๋ว, การเปลี่ยนตั๋ว, การยกเลิกตั๋ว, ศูนย์ช่วยเหลือ

รายละเอียดผู้ใช้ โดยการแสดงไอคอนของผู้ใช้ที่สามารถคลิกเพื่อดูรายละเอียด
ตัวเลือกการจองตั๋วรถไฟ ที่ผู้ใช้สามารถคลิกเพื่อทำการจองตั๋ว

หน้าเว็บนี้ได้รับการจัดระเบียบอย่างดี โดยใช้โครงสร้างที่ง่ายและตรงไปตรงมา ทั้งนี้จะมีการตกแต่งผ่านไฟล์ CSS ที่เชื่อมโยงอยู่ใน styles.css

```

37     </div>
38     <div class="option">
39         <a href="change-ticket.html">
40             
41             <h3>เปลี่ยนแปลงตัว</h3>
42         </a>
43     </div>
44     <div class="option">
45         <a href="cancel-ticket.html">
46             
47             <h3>ยกเลิกตัว</h3>
48         </a>
49     </div>
50 </div>
51
52 <div class="news">
53     <div class="news-item">
54         
55         <p>Royal Blossom @เพชรบุรี</p>
56     </div>
57     <div class="news-item">
58         
59         <p>เชียงรายแห่งความรัก</p>
60     </div>
61     <!-- เพิ่มไอเท็มข่าวเพิ่มเติมที่นี่ -->
62 </div>
63 </main>
64 </body>
65 </html>

```

รูปที่ 5.1.4 โค้ดที่เหลือเป็นส่วนของหน้า เว็บสำหรับการจัดการตั๋วรถไฟ ซึ่งรวมถึงตัวเลือกการเปลี่ยนแปลงตัว, การยกเลิกตัว และแสดงข่าวหรือโพสต์อัตโนมัติในหน้าแรก ทั้งหมดนี้ทำให้หน้าเว็บดูมีความเป็นระเบียบและพร้อมสำหรับผู้ใช้ในการจัดการตัวและรับข้อมูลข่าวสารต่างๆ เกี่ยวกับการเดินทาง

```

book-ticket.html > ...
1  <!DOCTYPE html>
2  <html lang="th">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>จองตั๋วรถไฟ</title>
7      <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10     <header>
11         <div class="logo">
12             
13         </div>
14         <nav>
15             <ul>
16                 <li><a href="index.html">หน้าหลัก</a></li>
17                 <li><a href="#">ประวัติการซื้อ</a></li>
18                 <li><a href="#">เปลี่ยนแปลงตั๋ว</a></li>
19                 <li><a href="#">ยกเลิกตั๋ว</a></li>
20                 <li><a href="#">ศูนย์ความช่วยเหลือ</a></li>
21             </ul>
22         </nav>
23     </header>
24
25     <main>
26         <div class="ticket-booking">
27             <h1>จองตั๋วรถไฟ</h1>
28             <form action="payment.html" method="POST">
29                 <label for="departure">สถานีเดินทาง</label>
30                 <input type="text" id="departure" name="departure" placeholder="กรุณากรอกสถานีเดินทาง" required>
31
32                 <label for="destination">สถานีปลายทาง</label>
33                 <input type="text" id="destination" name="destination" placeholder="กรุณากรอกสถานีปลายทาง" required>
34
35                 <label for="date">วันที่เดินทาง</label>
36                 <input type="date" id="date" name="date" required>
37
38                 <label for="time">เวลา</label>
39                 <input type="time" id="time" name="time" required>
40
41                 <button type="submit">จองตั๋ว</button>
42             </form>
43         </div>
44     </main>
45 </body>
46 </html>

```

รูปที่ 5.1.5 โค้ดนี้มีกดไปที่ ไอค่อน รูปรถไฟ และคำว่า “จองตั๋ว” โค้ดนี้เป็นหน้า จองตั๋วรถไฟ โดยมีส่วนที่สำคัญ ประกอบด้วย:

โลโก้ ของระบบที่จะแสดงในส่วนหัวของหน้าเว็บ

เมนูนำทาง ที่ช่วยให้ผู้ใช้สามารถไปยังหน้าอื่นๆ เช่น หน้าหลัก, การเปลี่ยนแปลงตั๋ว, และการยกเลิกตั๋ว ฟอร์มการจองตั๋ว ที่ผู้ใช้สามารถกรอกข้อมูลสถานีเดินทาง, สถานีปลายทาง, วันที่ และเวลาการเดินทาง ปุ่ม “จองตั๋ว” ที่จะนำข้อมูลไปส่งให้กับหน้า payment.html เพื่อดำเนินการชำระเงินต่อไป

```

5 payment.html > ⏺ html > ⏺ body > ⏺ script > ⏺ cancelPayment
1   <!DOCTYPE html>
2   <html lang="th">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>ชำระเงิน</title>
7       <link rel="stylesheet" href="styles.css">
8     <style>
9       /* ตกแต่งปุ่ม */
10    .btn {
11      padding: 10px 20px;
12      border: none;
13      color: black;
14      cursor: pointer;
15      font-size: 16px;
16      border-radius: 5px;
17    }
18
19    .btn-confirm {
20      background-color: green;
21    }
22
23    .btn-cancel {
24      background-color: red;
25    }
26
27    /* ตกแต่งการแสดง popup */
28    .popup {
29      display: none;
30      position: fixed;
31      top: 0;
32      left: 0;
33      width: 100%;
34      height: 100%;
35      background-color: rgba(0, 0, 0, 0.5);
36      justify-content: center;
37      align-items: center;
38    }
39
40    .popup-content {
41      background-color: white;
42      padding: 20px;
43      border-radius: 5px;
44      text-align: center;
45    }

```

รูปที่ 5.1.6 โค้ดนี้คัดนี้เป็นการตกแต่ง ปุ่ม และ ป้อปอัพ สำหรับใช้ในหน้าเว็บของระบบจองตั๋วรถไฟฟ้าออนไลน์:

ปุ่ม: การตกแต่งปุ่มสำหรับยืนยัน (สีเขียว) และยกเลิก (สีแดง)

ป้อปอัพ: การจัดการแสดงป้อปอัพที่มีพื้นหลังสีเทาและเนื้อหาภายในที่มีพื้นหลังขาว พร้อมข้อความกลาง

```

58 </head>
59 <body>
60   <header>
61     <div class="logo">
62       
63     </div>
64     <nav>
65       <ul>
66         <li><a href="home.html">หน้าหลัก</a></li>
67         <li><a href="purchase-history.html">ประวัติการซื้อ</a></li>
68         <li><a href="change-ticket.html">เปลี่ยนแปลงตั๋ว</a></li>
69         <li><a href="cancel-ticket.html">ยกเลิกตั๋ว</a></li>
70         <li><a href="help-center.html">ศูนย์ความช่วยเหลือ</a></li>
71       </ul>
72     </nav>
73   </header>
74
75   <main>
76     <h1>ชำระเงินสำหรับการจองตั๋ว</h1>
77
78     <!-- ข้อมูลการจอง -->
79     <p>สถานีต้นทาง: <span id="departure">กรุงเทพ</span></p>
80     <p>สถานีปลายทาง: <span id="destination">เชียงใหม่</span></p>
81     <p>วันที่เดินทาง: <span id="date">10 กุมภาพันธ์ 2568</span></p>
82     <p>เวลา: <span id="time">09:00 น.</span></p>
83     <p>จำนวนเงินที่ต้องชำระ: <span id="amount">500 บาท</span></p>
84
85     <!-- QR Code -->
86      <!-- แทรกภาพ QR Code -->
87
88     <br><br>
89
90     <!-- ปุ่มยืนยันการชำระเงิน และยกเลิก -->
91     <button class="btn btn-confirm" onclick="confirmPayment()">ยืนยันการชำระเงิน</button>
92     <button class="btn btn-cancel" onclick="cancelPayment()">ยกเลิก</button>
93
94     <!-- Popup สำหรับการยืนยันการชำระเงิน -->
95     <div id="popup" class="popup">
96       <div class="popup-content">
97         <p>ชำระสำเร็จแล้ว</p>
98         <button onclick="goToHome()">ไปยังหน้าหลัก</button>
99       </div>
100     </div>
101   </main>

```

รูปที่ 5.1.7 โค้ดนี้เป็นส่วนของหน้า การชำระเงิน (Payment Page) ในระบบจองตั๋วรถไฟฟออนไลน ซึ่งมี:
ข้อมูลการจอง ที่แสดงรายละเอียดของการเดินทาง เช่น สถานีต้นทาง, สถานีปลายทาง, วันที่เดินทาง, เวลาเดินทาง,
และจำนวนเงินที่ต้องจ่าย

QR Code สำหรับการชำระเงิน

ปุ่มยืนยันการชำระเงิน และ ปุ่มยกเลิก สำหรับการยืนยันการชำระเงินหรือยกเลิกการทำรายการ
Popup ที่จะแสดงขึ้นเมื่อการชำระเงินสำเร็จ ซึ่งมีปุ่มให้ผู้ใช้ไปยังหน้าหลัก

```

<script>
    // พังก์ชันสำหรับการยืนยันการชำระเงิน
    function confirmPayment() {
        // แสดง popup
        document.getElementById("popup").style.display = "flex";
    }

    // พังก์ชันไปยังหน้าหลัก
    function goToHome() {
        window.location.href = "home.html"; // เปลี่ยนเส้นทางไปที่หน้า index.html
    }

    // พังก์ชันสำหรับการยกเลิกการชำระเงิน
    function cancelPayment() {
        window.location.href = "home.html"; // เปลี่ยนเส้นทางไปที่หน้า index.html ทันที
    }
</script>
</body>
</html>

```

รูปที่ 5.1.8 โค้ดนี้เป็นส่วนของหน้า การชำระเงิน (Payment Page) ในระบบจองตัวรถไฟออนไลน์ ซึ่งมี:

ข้อมูลการจอง ที่แสดงรายละเอียดของการเดินทาง เช่น สถานีต้นทาง, สถานีปลายทาง, วันที่เดินทาง, เวลาเดินทาง, และจำนวนเงินที่ต้องจ่าย

QR Code สำหรับการชำระเงิน

บั่มือยืนยันการชำระเงิน และ บั่มือยกเลิก สำหรับการยืนยันการชำระเงินหรือยกเลิกการทำรายการ
Popup ที่จะแสดงขึ้นเมื่อการชำระเงินสำเร็จ ซึ่งมีปุ่มให้ผู้ใช้ไปยังหน้าหลัก

```

5 change-ticket.html > ...
1   <!DOCTYPE html>
2   <html lang="th">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>เปลี่ยนแปลงตัวรถไฟ</title>
7     <link rel="stylesheet" href="styles.css">
8   <style>
9     table {
10       width: 100%;
11       border-collapse: collapse;
12     }
13    th, td {
14      padding: 10px;
15      text-align: center;
16      border: 1px solid #ddd;
17    }
18    th {
19      background-color: #333;
20      color: white;
21    }
22    .btn {
23      padding: 5px 10px;
24      font-size: 14px;
25      color: white;
26      border: none;
27      cursor: pointer;
28    }
29    .confirm-btn {
30      background-color: green;
31    }
32    .cancel-btn {
33      background-color: red;
34    }
35    .form-container {
36      margin-top: 20px;
37    }
38    label {
39      display: block;
40      margin-top: 10px;
41    }
42    input[type="text"], input[type="date"], input[type="time"] {
43      padding: 8px;
44      margin: 5px 0;
45      width: 100%;
46    }

```

รูปที่ 5.1.8 โค้ดนี้ประกอบด้วยการตกแต่ง ตาราง พอร์มและ ปุ่ม สำหรับหน้าการเปลี่ยนแปลงตัวในระบบของตัวรถไฟออนไลน์:

การจัดการสไต์ล์ของ ตาราง (ตารางแสดงข้อมูลการเปลี่ยนแปลงตัว)

การจัดรูปแบบ ปุ่ม สำหรับยืนยันการเปลี่ยนแปลง (ปุ่มสีเขียว) และยกเลิกการเปลี่ยนแปลง (ปุ่มสีแดง)

การตกแต่ง พอร์ม และช่องกรอกข้อมูลให้เหมาะสมกับการใช้งาน

```

5 change-ticket.html > ...
6   <html lang="th">
7     </head>
8   <body>
9     <header>
10       <div class="logo">
11         
12       </div>
13     <nav>
14       <ul>
15         <li><a href="home.html">หน้าหลัก</a></li>
16         <li><a href="purchase-history.html">ประวัติการซื้อ</a></li>
17         <li><a href="change-ticket.html">เปลี่ยนแปลงตั๋ว</a></li>
18         <li><a href="cancel-ticket.html">ยกเลิกตั๋ว</a></li>
19         <li><a href="help-center.html">ศูนย์ความช่วยเหลือ</a></li>
20       </ul>
21     </nav>
22   </header>
23
24 <main>
25   <h1>เปลี่ยนแปลงตั๋วรถไฟ</h1>
26   <!-- ฟอร์มเปลี่ยนแปลงตั๋ว -->
27   <div class="form-container">
28     <h2>กรอกข้อมูลใหม่สำหรับการเปลี่ยนแปลงตั๋ว</h2>
29     <form id="change-ticket-form">
30       <label for="departure">สถานีเดินทาง</label>
31       <input type="text" id="departure" name="departure" placeholder="กรุณากรอกสถานีเดินทาง" required>
32
33       <label for="destination">สถานีปลายทาง</label>
34       <input type="text" id="destination" name="destination" placeholder="กรุณากรอกสถานีปลายทาง" required>
35
36       <label for="date">วันที่เดินทาง</label>
37       <input type="date" id="date" name="date" required>
38
39       <label for="time">เวลา</label>
40       <input type="time" id="time" name="time" required>
41
42       <button type="submit">ยืนยันการเปลี่ยนแปลง</button>
43     </form>
44   </div>
45
46   <!-- ตารางการแสดงข้อมูลตั๋วที่รอค่าเดินทาง -->
47   <table id="ticket-table">
48     <tr>
49       <th>หมายเลขอرجอง</th>
50       <th>ชื่อผู้โดยสาร</th>
51       <th>ต้นทาง-ปลายทาง</th>
52       <th>วันที่และเวลา</th>
53       <th>สถานะ</th>
54     </tr>
55
56     <tbody>
57
58   </tbody>
59 </table>
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104

```

รูปที่ 5.1.9 โค้ดนี้คือหน้า การเปลี่ยนแปลงตั๋ว สำหรับระบบจองตั๋วรถไฟออนไลน์ โดยมีฟอร์มให้ผู้ใช้กรอกข้อมูล การเปลี่ยนแปลงตั๋ว ซึ่งรวมถึงการกรอกสถานีเดินทาง, สถานีปลายทาง, วันที่เดินทาง และเวลาเดินทาง นอกจากนี้ยังมี เมนูนำทาง สำหรับผู้ใช้เพื่อเข้าถึงหน้าอื่นๆ เช่น หน้าหลัก, ประวัติการซื้อ, การเปลี่ยนแปลงตั๋ว และการยกเลิกตั๋ว

```

132 <script>
133     // พังก์ชันสำหรับคลิกเปลี่ยนแปลงตัว
134     function editTicket(ticketId) {
135         var row = document.getElementById("ticket-row-" + ticketId);
136
137         // ดึงข้อมูลจากແກ່ວທີ່ຕ້ອງການປັບປຸງ
138         var departure = row.cells[2].innerText.split(' → ')[0];
139         var destination = row.cells[2].innerText.split(' → ')[1];
140         var date = row.cells[3].innerText.split(' ເລກ ');
141         var dateText = date[0];
142         var timeText = date[1];
143
144         // ຕັ້ງຄາພອມໃຫ້ຕຽບກຳນົດຂໍ້ມູນໃນແກ່ວ
145         document.getElementById('departure').value = departure;
146         document.getElementById('destination').value = destination;
147         document.getElementById('date').value = dateText;
148         document.getElementById('time').value = timeText;
149
150         // ເນື້ອງໃຊ້ຍືນຢັນການປັບປຸງ
151         document.getElementById('change-ticket-form').onsubmit = function(event) {
152             event.preventDefault(); // ປິ່ນຍົງກຳນົດການເພີ້ນທີ່ນີ້
153             row.cells[2].innerText = departure + " → " + destination; // ອັບເດດສອນນີ້
154             row.cells[3].innerText = dateText + " ເລກ " + timeText; // ອັບເດດວັນທີແລະເວລາ
155             row.cells[4].innerText = "ປັບປຸງສໍາເລັດ"; // ອັບເດດສອນນີ້
156
157             // ຮຶບເຂົ້າມີ້ນໍາມາຈຳກັດສົ່ງ
158             document.getElementById('change-ticket-form').reset();
159         };
160     }
161
162     // ພັກໜີ້ມີຟັງກ່ານທີ່ຕ້ອງການປັບປຸງ
163     function cancelTicket(ticketId) {
164         var row = document.getElementById("ticket-row-" + ticketId);
165         row.style.display = "none"; // ຂ່ອນແກ້ວທີ່ເລືອກ
166     }
167 </script>
168 </body>
169 </html>

```

ຮູບທີ່ 5.1.10 ໂດຍມີຟັງກ່ານທີ່ຕ້ອງການປັບປຸງ
 ແກ້ໄຂຂໍ້ມູນຕ້ົວ ໂດຍຄືກປຸ່ມແກ້ໄຂໃນຕາງໆທີ່ເກີຍວ່າຂອງ
 ເນື້ອງໃຊ້ຄືກປຸ່ມ ແກ້ໄຂ, ຂໍ້ມູນຈາກແກ່ວທີ່ເລືອກຈະຖຸກດຶງນາແສດງໃນ ພົມ
 ແລະສາມາດປັບປຸງໄດ້
 ເນື້ອງສ່າງພົມ, ຂໍ້ມູນໃນແກ່ວຈະໄດ້ຮັບການອັບເດດ
 ຍັກເລີກການປັບປຸງແປ່ງຕ້ົວ ໂດຍການໜີ້ມີຟັງກ່ານທີ່ຜູ້ໃຊ້ເລືອກ

```

5 cancel-ticket.html > html > body > script > cancelTicket
1   <!DOCTYPE html>
2   <html lang="th">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>ยกเลิกตั๋วรถไฟ</title>
7       <link rel="stylesheet" href="styles.css">
8     <style>
9       table {
10         width: 100%;
11         border-collapse: collapse;
12       }
13       th, td {
14         padding: 10px;
15         text-align: center;
16         border: 1px solid #ddd;
17       }
18       th {
19         background-color: #333;
20         color: white;
21       }
22       .btn {
23         padding: 5px 10px;
24         font-size: 14px;
25         color: white;
26         border: none;
27         cursor: pointer;
28       }
29       .confirm-btn {
30         background-color: green;
31       }
32       .cancel-btn {
33         background-color: red;
34       }
35     </style>
36   </head>
37   <body>
38     <header>
39       <div class="logo">
40         
41       </div>

```

รูปที่ 5.1.11 โค้ดนี้เป็นหน้า ยกเลิกการจองตั๋ว สำหรับระบบจองตั๋วรถไฟออนไลน์ โดยมีการแสดงข้อมูลการจองตั๋ว ในตาราง และผู้ใช้สามารถดำเนินการ ยกเลิกตั๋ว ได้ด้วยการคลิกปุ่ม "ยืนยันการยกเลิก" หรือ "ยกเลิก" ขึ้นอยู่กับการยืนยันการยกตั๋ว

ตาราง แสดงข้อมูลเกี่ยวกับการจองตั๋ว
ปุ่ม ใช้สำหรับยืนยันการยกเลิกหรือยกเลิกการยกเลิก

```
59 <script>
60     // พังก์ชันยืนยันการยกเลิก
61     function confirmCancel(button) {
62         var row = button.parentElement.parentElement;
63         row.cells[4].innerText = "ยกเลิกสำเร็จ"; // เปลี่ยนสถานะ
64         row.cells[5].innerHTML = ""; // ลบบุม
65     }
66
67     // พังก์ชันยกเลิกตัว
68     function cancelTicket(button) {
69         var row = button.parentElement.parentElement;
70         row.style.display = "none"; // ซ่อนแถวที่เลือก
71     }
72 </script>
73 </body>
74 </html>
```

รูปที่ 5.1.12 โค้ดนี้ประกอบไปด้วยฟังก์ชัน 2 ฟังก์ชันหลัก:

confirmCancel(button): เมื่อคลิกที่ปุ่ม "ยืนยันการยกเลิก" ฟังก์ชันนี้จะทำการเปลี่ยนแปลงสถานะของการจองตัวในตารางเป็น "ยกเลิกสำเร็จ" และจะลบปุ่ม "ยืนยันการยกเลิก" และ "ยกเลิก" ออกจากตาราง

cancelTicket(button): เมื่อคลิกที่ปุ่ม "ยกเลิก" ฟังก์ชันนี้จะทำให้แถวที่เลือก ซ่อน ไปจากหน้าจอ

```

5 personal-details.html > html > body > main > div.form-container > a
2   <html lang="th">
44  <body>
50
51    <main>
52      <h1>รายละเอียดส่วนบุคคล</h1>
53      <div class="form-container">
54        <h2>ข้อมูลส่วนตัว</h2>
55        <form id="personal-details-form">
56          <label for="username">ชื่อผู้ใช้</label>
57          <input type="text" id="username" name="username" value="สมชาย สันทรสว่าง" required>
58
59          <label for="email">อีเมล</label>
60          <input type="email" id="email" name="email" value="somchai@example.com" required>
61
62          <label for="phone">เบอร์โทรศัพท์</label>
63          <input type="tel" id="phone" name="phone" value="012-345-6789" required>
64
65          <label for="password">รหัสผ่าน</label>
66          <input type="password" id="password" name="password" value="password123" required>
67
68          <button type="submit">บันทึกการเปลี่ยนแปลง</button>
69        </form>
70        <a href="logout.html">
71          <button class="btn logout-btn">ออกจากระบบ</button>
72        </a>
73      </div>
74    </main>
75
76
77
78
79
80
81
82
83
84
85

```

รูปที่ 5.1.12 โดยด้านนี้แสดงฟอร์มสำหรับ ข้อมูลส่วนบุคคล ที่ประกอบไปด้วยช่องกรอกข้อมูล เช่น ชื่อผู้ใช้, อีเมล, หมายเลขโทรศัพท์ และรหัสผ่าน และมีปุ่ม "บันทึกการเปลี่ยนแปลง" สำหรับบันทึกข้อมูลที่ผู้ใช้กรอกไว้ ฟอร์มนี้ช่วยให้ผู้ใช้สามารถ อัปเดตข้อมูลส่วนตัว ได้ ปุ่ม "ออกจากระบบ" ช่วยให้ผู้ใช้สามารถออกจากระบบได้โดยการคลิกปุ่ม ซึ่งจะเปลี่ยนเส้นทางไปยังหน้า logout.html

```

<script>
    // เมื่อผู้ใช้ส่งฟอร์ม
    document.getElementById('personal-details-form').addEventListener('submit', function(event) {
        event.preventDefault(); // ป้องกันการรีเฟรชหน้า

        // ดึงค่าจากฟอร์ม
        var username = document.getElementById('username').value;
        var email = document.getElementById('email').value;
        var phone = document.getElementById('phone').value;
        var password = document.getElementById('password').value;

        // แสดงข้อมูลที่เปลี่ยนแปลง (ในกรณีนี้จะแสดงในコンโซล)
        console.log('ข้อมูลที่บันทึก:');
        console.log('ชื่อผู้ใช้: ' + username);
        console.log('อีเมล: ' + email);
        console.log('เบอร์โทรศัพท์: ' + phone);
        console.log('รหัสผ่าน: ' + password);

        alert("ข้อมูลส่วนบุคคลของคุณถูกบันทึกเรียบร้อยแล้ว!");
    });
</script>
</body>
</html>

```

รูปที่ 5.1.13 โค้ดนี้จัดการกับการส่งฟอร์ม ข้อมูลส่วนบุคคล โดยการ:

ป้องกันการรีเฟรชหน้า เมื่อผู้ใช้กด submit ฟอร์ม

ดึงข้อมูลจากช่องกรอก เช่น ชื่อผู้ใช้, อีเมล, หมายเลขโทรศัพท์, และรหัสผ่าน

แสดงข้อมูลที่กรอก ใน Console เพื่อให้ผู้ใช้สามารถตรวจสอบข้อมูล

แสดงข้อความแจ้งเตือน (alert) ว่าข้อมูลส่วนตัวของผู้ใช้ได้ถูกบันทึกเรียบร้อยแล้ว

```

5 purchase-history.html > ...
2   <html lang="th">
3     </head>
4     <body>
5       <header>
6         <div class="logo">
7           
8         </div>
9         <nav>
10          <ul>
11            <li><a href="home.html">หน้าหลัก</a></li>
12            <li><a href="purchase-history.html">ประวัติการซื้อ</a></li>
13            <li><a href="change-ticket.html">เปลี่ยนแปลงตั๋ว</a></li>
14            <li><a href="cancel-ticket.html">ยกเลิกตั๋ว</a></li>
15            <li><a href="help-center.html">ศูนย์ความช่วยเหลือ</a></li>
16          </ul>
17        </nav>
18      </header>
19
20      <main>
21        <h1>ประวัติการซื้อ</h1>
22        <!-- ตารางการแสดงข้อมูลตั๋วที่ได้ทำการซื้อ -->
23        <table id="ticket-table">
24          <tr>
25            <th>หมายเลขการจอง</th>
26            <th>ชื่อผู้โดยสาร</th>
27            <th>ต้นทาง-ปลายทาง</th>
28            <th>วันที่และเวลา</th>
29            <th>สถานะชำระเงิน</th>
30          </tr>
31          <tr>
32            <td>#12345</td>
33            <td>นายสมชาย จันทร์สว่าง</td>
34            <td>กรุงเทพ → เชียงใหม่</td>
35            <td>10 กุมภาพันธ์ 2568 เวลา 09:00 น.</td>
36            <td>ชำระแล้ว</td>
37          </tr>
38          <tr>
39            <td>#12346</td>
40            <td>นางสาวสมศรี ศรีสวัสดิ์</td>
41            <td>กรุงเทพ → หัวหิน</td>
42            <td>10 กุมภาพันธ์ 2568 เวลา 12:00 น.</td>
43            <td>รอชำระเงิน</td>
44          </tr>
45        </table>
46      </main>
47    </body>
48  </html>
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69

```

รูปที่ 5.1.14 โค้ดนี้เป็นหน้า ประวัติการซื้อ ของระบบจองตั๋วรถไฟออนไลน์ โดยแสดงข้อมูลการจองตั๋วที่ผู้ใช้ได้ทำการซื้อแล้ว:

เมนูนำทาง ให้ผู้ใช้สามารถไปยังหน้าต่างๆ เช่น หน้าหลัก, การเปลี่ยนแปลงตั๋ว, การยกเลิกตั๋ว, และศูนย์ช่วยเหลือ ตารางการจองตั๋ว แสดงข้อมูลเกี่ยวกับการจองตั๋ว เช่น หมายเลขการจอง, สถานีต้นทาง, สถานีปลายทาง, วันที่เดินทาง และสถานะการชำระเงิน

```
25   <main>
26     <h1>ศูนย์ความช่วยเหลือ</h1>
27     <p>หากท่านมีคำถามเกี่ยวกับการใช้งานบริการของเรา โปรดเลือกหัวข้อที่ท่านต้องการความช่วยเหลือ:</p>
28     <ul>
29       <li><a href="#">คำถามที่พบบ่อย (FAQ)</a></li>
30       <li><a href="#">ติดต่อเรา</a></li>
31       <li><a href="#">ข้อกำหนดและเงื่อนไข</a></li>
32     </ul>
33   </main>
34 </body>
35 </html>
36
```

รูปที่ 5.1.15 โค้ดนี้แสดงหน้า ศูนย์ช่วยเหลือ สำหรับระบบจองตั๋วรถไฟออนไลน์ โดยมีข้อความแนะนำให้ผู้ใช้เลือก หัวข้อที่ต้องการความช่วยเหลือ:

หัวข้อหลัก "ศูนย์ความช่วยเหลือ"

ข้อความที่อธิบายให้ผู้ใช้เลือกหัวข้อที่ต้องการ

ลิงก์ที่เชื่อมไปยัง คำถามที่พบบ่อย, ติดต่อเรา, และ ข้อกำหนดและเงื่อนไข ซึ่งเป็นหัวข้อที่ผู้ใช้สามารถคลิกเพื่อค้นหา ข้อมูลที่เกี่ยวข้อง

```
5 logout.html > ...
1  <!DOCTYPE html>
2  <html lang="th">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>ออกจากระบบ</title>
7      <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10     <div class="logout-message">
11         <h1>คุณได้ออกจากระบบแล้ว</h1>
12         <p>ขอบคุณที่ใช้บริการ! กำลังพาคุณกลับไปยังหน้าแรก...</p>
13         <script>
14             setTimeout(function() {
15                 window.location.href = "login.html"; // พาผู้ใช้กลับไปที่หน้าแรกหลังจากออกจากระบบ
16             }, 3000); // ใช้เวลา 3 วินาทีในการเปลี่ยนหน้า
17         </script>
18     </div>
19 </body>
20 </html>
21
```

รูปที่ 5.1.16 โค้ดนี้เป็นส่วนของหน้า ออกจากระบบ (Logout) ซึ่งมีฟังก์ชันการทำงานดังนี้:
แสดงข้อความแจ้งเตือนผู้ใช้ว่า ออกจากระบบแล้ว พร้อมข้อความเพิ่มเติมที่บอกว่า กำลังพาผู้ใช้กลับไปยังหน้าผู้ใช้.
หลังจาก 3 วินาที (ด้วย setTimeout()), ระบบจะ เปลี่ยนเส้นทางผู้ใช้ไปยังหน้าเข้าสู่ระบบ (login.html).

```

styles.css > ...
1  /* General Styles */
2 ∵ body {
3     font-family: Arial, sans-serif;
4     margin: 0;
5     padding: 0;
6     background-color: #f4f4f4;
7 }
8
9 ∵ header {
10    display: flex;
11    justify-content: space-between;
12    align-items: center;
13    background-color: #fff;
14    padding: 10px 20px;
15    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
16 }
17
18 ∵ header .logo img {
19    width: 150px;
20    height: 150px;
21 }
22
23 ∵ header nav ul {
24    display: flex;
25    list-style-type: none;
26    margin: 0;
27 }
28
29 ∵ header nav ul li {
30    margin: 0 10px;
31 }
32
33 ∵ header nav ul li a {
34    text-decoration: none;
35    color: #8e5a2a;
36    font-weight: bold;
37 }
38
39 ∵ header .user img {
40    width: 30px;
41    height: 30px;
42 }

```

รูปที่ 5.1.17 โค้ดนี้เป็นแค่ตัวอย่าง ประกอบไปด้วยการจัดรูปแบบส่วนต่างๆ ของหน้าเว็บ เช่น header และ เมนูนำทาง โดยมีการใช้ flexbox เพื่อจัดเรียงองค์ประกอบต่างๆ ให้อยู่ในแนวเดียวกันและเว้นระยะห่างอย่างเหมาะสม:
header: ใช้ flexbox เพื่อจัดตำแหน่งโลโก้และเมนูนำทางให้อยู่ในแนวเดียวกัน
โลโก้: กำหนดขนาดของโลโก้ให้มีขนาด 150px
เมนูนำทาง: ใช้ flexbox และกำหนดระยะห่างระหว่างรายการเมนู
ลิงก์: กำหนดสีต่ำสุดให้ลิงก์ไม่มีชีดเส้นใต้และสีเทาเข้มๆ
ไอคอนผู้ใช้: กำหนดขนาดของไอคอนเป็น 30px x 30px

```

43  /* Main Content */
44  .options {
45    display: flex;
46    justify-content: space-around;
47    margin-top: 50px;
48    padding: 20px;
49  }
50
51
52 .option {
53   text-align: center;
54 }
55
56 .option img {
57   width: 50px;
58   height: 50px;
59 }
60
61 .option h3 {
62   margin-top: 10px;
63   font-size: 16px;
64   color: #8e5a2a;
65 }
66
67 .news {
68   display: flex;
69   justify-content: space-around;
70   flex-wrap: wrap;
71   margin-top: 20px;
72 }
73
74 .news-item {
75   width: 30%;
76   margin: 10px;
77   border: 1px solid #ddd;
78   border-radius: 8px;
79   overflow: hidden;
80   box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
81 }
82
83 .news-item img {
84   width: 100%;
85   height: auto;
86 }

```

รูปที่ 5.1.18 โค้ดนี้มีการจัดการส్ಟైลสำหรับ เนื้อหาหลัก ในหน้าเว็บ โดยเฉพาะส่วนของ ตัวเลือก (options) และ ข่าวสาร (news):

options: จัดการกับการจัดเรียงตัวเลือก (ไอคอนและข้อความ) ให้จัดเรียงในแนวนอนและมีการเว้นระยะห่างที่เหมาะสม

news: จัดการกับการแสดงผลของ ข่าวสาร โดยทำให้ news items อยู่ในรูปแบบตารางที่ยืดหยุ่นและสามารถจัดเรียงได้เมื่อหน้าจอขนาดเล็ก

news-item: เพิ่มลักษณะกรอบและเงาเพื่อให้แต่ละข่าวสารดูน่าสนใจมากขึ้น

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
}

/* Header Styles */
header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    background-color: #ffffff;
    padding: 20px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

header .logo img {
    width: 150px;
    height: 150px;
}

header nav ul {
    display: flex;
    list-style-type: none;
    margin: 0;
}

header nav ul li {
    margin: 0 15px;
}
```

รูปที่ 5.1.19

โค้ดนี้มีการใช้ flexbox เพื่อจัดเรียงองค์ประกอบใน header และ nav ให้ดูเรียบง่ายและตรงกลาง ซึ่งเป็นวิธีการที่นิยมในการจัดレイアウトเว็บเพจที่มีความยืดหยุ่น

```
main {
    padding: 30px;
}

/* Options Section */
.options {
    display: flex;
    justify-content: space-around;
    margin-bottom: 40px;
}

.option {
    text-align: center;
    width: 30%;
}

.option img {
    width: 80px;
    height: 80px;
    margin-bottom: 15px;
    transition: transform 0.3s ease;
}

.option img:hover {
    transform: scale(1.1);
}

.option h3 {
    font-size: 18px;
    color: #8e5a2a;
}
```

รูปที่ 5.1.20

โค้ดนี้ใช้สำหรับจัดรูปแบบในส่วนที่มีตัวเลือกต่างๆ ในเว็บเพจ พร้อมทั้งเพิ่มเอฟเฟกต์ขยายขนาดรูปภาพเมื่อผู้ใช้ชี้มาส์ไปที่รูปนั้น

```
.news-item {  
    width: 48%;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
    border-radius: 8px;  
    overflow: hidden;  
    background-color: white;  
    transition: transform 0.3s ease;  
}  
  
.news-item img {  
    width: 100%;  
    height: auto;  
    border-bottom: 2px solid #ddd;  
}  
  
.news-item p {  
    text-align: center;  
    padding: 15px;  
    font-size: 16px;  
    color: #8e5a2a;  
    background-color: #f8f8f8;  
}  
  
.news-item:hover {  
    transform: scale(1.05);  
}
```

รูปที่ 5.1.21

โค้ดนี้ช่วยให้ข่าวแต่ละชิ้นมีรูปลักษณ์ที่สวยงามและมีการเปลี่ยนแปลงขนาดอย่างนุ่มนวลเมื่อผู้ใช้ต้องกับมัน (เช่น การซื้อมาสีไปที่ข่าว)

```

@media screen and (max-width: 768px) {
    header {
        flex-direction: column;
        align-items: flex-start;
    }

    header nav ul {
        flex-direction: column;
        align-items: flex-start;
    }

    header nav ul li {
        margin: 10px 0;
    }

    .options {
        flex-direction: column;
        align-items: center;
    }

    .option {
        width: 80%;
        margin-bottom: 20px;
    }

    .news {
        flex-direction: column;
        align-items: center;
    }

    .news-item {
        width: 80%;
    }
}

```

รูปที่ 5.1.22

โค้ดนี้จะทำให้เว็บไซต์ตอบสนองได้ดีขึ้นบนอุปกรณ์ที่มีขนาดหน้าจอเล็ก เช่น มือถือหรือแท็บเล็ต โดยจะจัดเรียงองค์ประกอบต่างๆ ในแนวตั้งและจัดให้อยู่กลางหน้าจอ

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
}

/* Header Styles */
header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    background-color: #ffffff;
    padding: 20px 30px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

header .logo img {
    width: 150px;
    height: 150px;
}

header nav ul {
    display: flex;
    list-style-type: none;
    margin: 0;
}

header nav ul li {
    margin: 0 20px;
}
```

รูปที่ 5.1.23

โค้ดนี้ช่วยให้การจัดรูปแบบใน header ของเว็บไซต์ดูสะอาดและมีระเบียบ พร้อมทั้งรองรับการจัดวางแบบ flexbox ซึ่งทำให้การจัดเรียงของโลโก้และเมนูนำทางเป็นไปอย่างเหมาะสม

```
header nav ul li {
    margin: 0 20px;
}

header nav ul li a {
    text-decoration: none;
    color: #8e5a2a;
    font-weight: bold;
    font-size: 16px;
    transition: color 0.3s ease;
}

header nav ul li a:hover {
    color: #5a3c1a;
}

header .user img {
    width: 40px;
    height: 40px;
}

/* Main Content */
main {
    padding: 30px;
    display: flex;
    flex-direction: column;
    align-items: center;
}
```

รูปที่ 5.1.24

โค้ดนี้ใช้เพื่อจัดรูปแบบเมนูนำทางและเนื้อหาหลักของเว็บไซต์ รวมถึงการใช้ flexbox สำหรับการจัดเรียงองค์ประกอบใน main และการเพิ่มเอฟเฟกต์ที่น่าสนใจสำหรับลิงก์ในเมนู

```
/* Main Content */
main {
    padding: 30px;
    display: flex;
    flex-direction: column;
    align-items: center;
}

/* Options Section */
.options {
    display: flex;
    justify-content: space-around;
    margin-bottom: 50px;
    gap: 30px;
}

.option {
    text-align: center;
    width: 30%;
    transition: transform 0.3s ease;
}

.option img {
    width: 80px;
    height: 80px;
    margin-bottom: 15px;
    transition: transform 0.3s ease;
}

.option img:hover {
    transform: scale(1.1);
}

.option h3 {
    font-size: 18px;
    color: #8e5a2a;
    margin-top: 10px;
}
```

รูปที่ 5.1.25

โค้ดนี้จะช่วยในการจัดรูปแบบส่วนตัวเลือกในเว็บไซต์ให้มีการจัดเรียงที่เรียบร้อย และเพิ่มเอฟเฟกต์ให้กับรูปภาพและข้อความเพื่อให้มีความน่าสนใจเมื่อผู้ใช้ต้องกับเว็บไซต์

```

/* News Section */
.news {
    display: flex;
    justify-content: space-between;
    flex-wrap: wrap;
    gap: 20px;
    margin-top: 40px;
}

.news-item {
    width: 48%;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    border-radius: 8px;
    overflow: hidden;
    background-color: white;
    transition: transform 0.3s ease;
}

.news-item img {
    width: 100%;
    height: auto;
    border-bottom: 2px solid #ddd;
}

.news-item p {
    text-align: center;
    padding: 15px;
    font-size: 16px;
    color: #8e5a2a;
    background-color: #f8f8f8;
}

.news-item:hover {
    transform: scale(1.05);
}

```

รูปที่ 5.1.26

โค้ดนี้ใช้เพื่อจัดรูปแบบให้ส่วนของ ข่าว มีการจัดเรียงที่สวยงามและมีเอฟเฟกต์ขยายขนาดในขณะโต้ตอบ ทำให้เว็บไซต์ดูน่าสนใจและมีการเคลื่อนไหวที่เรียบง่าย

```
@media screen and (max-width: 768px) {  
    header {  
        flex-direction: column;  
        align-items: flex-start;  
    }  
  
    header nav ul {  
        flex-direction: column;  
        align-items: flex-start;  
    }  
  
    header nav ul li {  
        margin: 10px 0;  
    }  
  
    .options {  
        flex-direction: column;  
        align-items: center;  
    }  
  
    .option {  
        width: 80%;  
        margin-bottom: 20px;  
    }  
  
    .news {  
        flex-direction: column;  
        align-items: center;  
    }  
  
    .news-item {  
        width: 80%;  
    }  
}
```

รูปที่ 5.1.27

โค้ดนี้ช่วยให้เว็บไซต์แสดงผลได้อย่างเหมาะสมและตอบสนองได้ดีเมื่อแสดงบนอุปกรณ์ที่มีหน้าจอขนาดเล็ก เช่น โทรศัพท์มือถือและแท็บเล็ต

```
/* Ticket Booking Page Styles */
.ticket-booking {
    background-color: white;
    padding: 30px;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    width: 400px;
    margin: 0 auto;
}

.ticket-booking h1 {
    text-align: center;
    font-size: 24px;
    color: #8e5a2a;
    margin-bottom: 20px;
}

.ticket-booking label {
    display: block;
    margin: 10px 0 5px;
    font-size: 14px;
}
```

รูปที่ 5.1.28

โค้ดนี้จะช่วยให้หน้า Ticket Booking ดูเรียบร้อย และช่วยจัดรูปแบบองค์ประกอบต่างๆ ให้มีความสะอาดสวยงามและสวยงามในการใช้งาน

```
.ticket-booking input {  
    width: 100%;  
    padding: 10px;  
    margin: 10px 0;  
    border-radius: 4px;  
    border: 1px solid #ddd;  
    font-size: 14px;  
}  
  
.ticket-booking button {  
    width: 100%;  
    padding: 12px;  
    background-color: #8e5a2a;  
    border: none;  
    border-radius: 4px;  
    color: white;  
    font-size: 16px;  
    cursor: pointer;  
}  
  
.ticket-booking button:hover {  
    background-color: #7b4b22;  
}
```

รูปที่ 5.1.29

โค้ดนี้ช่วยให้การจัดรูปแบบของกรอกข้อมูลและปุ่มในหน้า Ticket Booking เป็นระเบียบและดูดีขึ้น พร้อมทั้งเพิ่มการเปลี่ยนแปลงสีเมื่อผู้ใช้ต้องบันทึกปุ่ม

```
<header>
  <div class="logo">
    
  </div>
  <nav>
    <ul>
      <li><a href="index.html">หน้าหลัก</a></li>
      <li><a href="#">ประวัติการซื้อ</a></li>
      <li><a href="#">เปลี่ยนแปลงตัว</a></li>
      <li><a href="#">ยกเลิกตัว</a></li>
      <li><a href="#">ศูนย์ความช่วยเหลือ</a></li>
    </ul>
  </nav>
</header>
```

รูปที่ 5.1.30

โค้ดนี้เป็นการตั้งค่าเบื้องต้นสำหรับส่วน header ซึ่งจะมีโลโก้และเมนูการนำทางของเว็บไซต์.

```
<script>
    // พังก์ชันยืนยันการยกเลิก
    function confirmCancel(button) {
        var row = button.parentElement.parentElement;
        row.cells[4].innerText = "ยกเลิกสำเร็จ"; // เปลี่ยนสถานะ
        row.cells[5].innerHTML = ""; // ลบปุ่ม
    }

    // พังก์ชันยกเลิกตัว
    function cancelTicket(button) {
        var row = button.parentElement.parentElement;
        row.style.display = "none"; // ซ่อนแถวที่เลือก
    }
</script>
</body>
</html>
```

รูปที่ 5.1.31

พังก์ชันเหล่านี้จะช่วยให้สามารถยืนยันการยกเลิกการจองตัวและซ่อนแถวที่มีการยกเลิกได้ในหน้าเว็บ

```

<script>
    // พังก์ชันสำหรับคลิกเปลี่ยนแปลงตัว
    function editTicket(ticketId) {
        var row = document.getElementById("ticket-row-" + ticketId);

        // ดึงข้อมูลจากแต่ที่ต้องการเปลี่ยนแปลง
        var departure = row.cells[2].innerText.split(' → ')[0];
        var destination = row.cells[2].innerText.split(' → ')[1];
        var dateTime = row.cells[3].innerText.split(' เวลา ');
        var date = dateTime[0];
        var time = dateTime[1];

        // ตั้งค่าฟอร์มให้ตรงกับข้อมูลใหม่ๆ
        document.getElementById('departure').value = departure;
        document.getElementById('destination').value = destination;
        document.getElementById('date').value = date;
        document.getElementById('time').value = time;

        // เมื่อผู้ใช้ยืนยันการเปลี่ยนแปลง
        document.getElementById('change-ticket-form').onsubmit = function(event) {
            event.preventDefault(); // ป้องกันการรีเฟรชหน้า
            row.cells[2].innerText = departure + " → " + destination; // อัปเดตสถานี
            row.cells[3].innerText = date + " เวลา " + time; // อัปเดตวันที่และเวลา
            row.cells[4].innerText = "เปลี่ยนแปลงสำเร็จ"; // อัปเดตสถานะ

            // รีเซ็ตฟอร์มหลังจากส่ง
            document.getElementById('change-ticket-form').reset();
        };
    }

    // พังก์ชันสำหรับยกเลิกการจองตัว
    function cancelTicket(ticketId) {
        var row = document.getElementById("ticket-row-" + ticketId);
        row.style.display = "none"; // ซ่อนแถวที่เลือก
    }
</script>

```

รูปที่ 5.1.31

โค้ดนี้ช่วยให้สามารถแก้ไขข้อมูลการจองตัวได้โดยการแสดงผลข้อมูลในฟอร์มแล้วอัปเดตข้อมูลในตาราง เมื่อผู้ใช้ทำการแก้ไขข้อมูลเสร็จและยืนยัน ข้อมูลใหม่จะถูกอัปเดตและแสดงข้อความยืนยันการเปลี่ยนแปลง. นอกจากนี้ยังสามารถยกเลิกการจองตัวได้โดยการซ่อนแถวที่เกี่ยวข้อง

```
<style>
    table {
        width: 100%;
        border-collapse: collapse;
    }
    th, td {
        padding: 10px;
        text-align: center;
        border: 1px solid #ddd;
    }
    th {
        background-color: #333;
        color: white;
    }
    .btn {
        padding: 5px 10px;
        font-size: 14px;
        color: white;
        border: none;
        cursor: pointer;
    }

```

รูปที่ 5.1.32

โค้ดนี้ช่วยในการจัดรูปแบบของตารางให้ดูเรียบร้อย และเพิ่มสีトイล์ให้กับปุ่มเพื่อให้มีลักษณะที่ดูน่าสนใจเมื่อผู้ใช้ต้องกับมัน

```
.confirm-btn {  
    background-color: green;  
}  
.cancel-btn {  
    background-color: red;  
}  
.form-container {  
    margin-top: 20px;  
}  
label {  
    display: block;  
    margin-top: 10px;  
}
```

รูปที่ 5.1.33

โค้ดนี้ใช้ในการกำหนดลักษณะการแสดงผลของปุ่มยืนยันและยกเลิกให้มีสีพื้นหลังที่แตกต่างกัน และเพิ่มการจัดรูปแบบให้กับฟอร์มและ label เพื่อให้ฟอร์มมีความสวยงามในการใช้งาน

```
input[type="text"], input[type="date"], input[type="time"] {  
    padding: 8px;  
    margin: 5px 0;  
    width: 100%;  
}  
button {  
    padding: 10px;  
    background-color: green;  
    color: white;  
    border: none;  
    cursor: pointer;  
}  
button:hover {  
    background-color: darkgreen;  
}
```

รูปที่ 5.1.34

โค้ดนี้ใช้ในการจัดรูปแบบฟอร์ม input และปุ่ม โดยกำหนดลักษณะของฟิล์ด input ให้อยู่เต็มความกว้าง และทำให้ปุ่ม มีสีที่สวยงามพร้อมกับการเปลี่ยนแปลงสีเมื่อมีการตีตوبจากผู้ใช้

```
<main>
  <h1>ศูนย์ความช่วยเหลือ</h1>
  <p>หากท่านมีคำถามเกี่ยวกับการใช้บริการของเรา โปรดเลือกหัวข้อที่ท่านต้องการความช่วยเหลือ:</p>
  <ul>
    <li><a href="#">คำถามที่พบบ่อย (FAQ)</a></li>
    <li><a href="#">ติดต่อเรา</a></li>
    <li><a href="#">ข้อกำหนดและเงื่อนไข</a></li>
  </ul>
</main>
```

รูปที่ 5.1.35

โค้ดนี้เป็นการจัดรูปแบบหน้าเว็บสำหรับศูนย์ความช่วยเหลือ โดยใช้หัวข้อ `<h1>` และข้อความ `<p>` เพื่อแนะนำให้ผู้ใช้เลือกหัวข้อที่เกี่ยวข้องจากการลิงก์ต่างๆ เพื่อขอความช่วยเหลือ

```
<div class="news">
    <div class="news-item">
        
        <p>Royal Blossom @เพชรบุรี</p>
    </div>
    <div class="news-item">
        
        <p>เชียงรายแห่งความรัก</p>
    </div>
    <div class="news-item">
        
        <p>เชียงใหม่ ท่องเที่ยวไทย</p>
    </div>
```

รูปที่ 5.1.36

โค้ดนี้แสดงข่าวในรูปแบบโพสต์ที่ประกอบด้วยภาพและข้อความที่เกี่ยวข้อง โดยใช้ `<div>` เพื่อจัดกลุ่มแต่ละข่าวให้เป็นระเบียบ

```
<p>ยังไม่ได้เป็นสมาชิก? <a href="register.html">ลงทะเบียน</a></p>
</div>
<div class="login-image">
    <img src="" alt="Train Image">
</div>
```

รูปที่ 5.1.37

โค้ดนี้ใช้ในการแสดงข้อความเชิญชวนให้ผู้ใช้งานทะเบียนในเว็บไซต์ หากผู้ใช้ยังไม่ได้ลงทะเบียน พร้อมทั้งแสดงภาพในส่วนของการเข้าสู่ระบบ (Login)

```
<div class="logout-message">
    <h1>คุณได้ออกจากระบบแล้ว</h1>
    <p>ขอบคุณที่ใช้บริการ! กำลังพาคุณกลับไปยังหน้าแรก...</p>
    <script>
        setTimeout(function() {
            window.location.href = "login.html"; // พาผู้ใช้กลับไปที่หน้าแรกหลังจากออกจากระบบ
        }, 3000); // ใช้เวลา 3 วินาทีในการเปลี่ยนหน้า
    </script>
```

รูปที่ 5.1.38

โค้ดนี้จะทำการแสดงข้อความหลังจากที่ผู้ใช้ทำการออกจากระบบ และหลังจากนั้นจะทำการเปลี่ยนหน้าไปที่หน้าเข้าสู่ระบบ (login.html) หลังจากหน่วงเวลา 3 วินาที

```
<!DOCTYPE html>
<html lang="th">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ออกจากระบบ</title>
    <link rel="stylesheet" href="styles.css">
</head>
```

รูปที่ 5.1.39

โค้ดนี้เป็นส่วนของ **head** ของเอกสาร HTML ซึ่งกำหนดข้อมูลพื้นฐาน เช่น การเข้ารหัสภาษา, การตั้งค่า **viewport** สำหรับการแสดงผลบนอุปกรณ์มือถือ, ชื่อเรื่องของหน้าเว็บ และการเชื่อมโยงกับไฟล์ CSS สำหรับการจัดรูปแบบเว็บไซต์.

```
.form-container {  
    margin-top: 20px;  
    padding: 20px;  
    background-color: #f4f4f4;  
    border-radius: 8px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
    max-width: 500px;  
    margin: 0 auto;  
}  
label {  
    display: block;  
    margin-top: 10px;  
}
```

รูปที่ 5.1.40

โค้ดนี้ใช้ในการจัดรูปแบบฟอร์มให้มีความสวยงามและสะดวกสบาย โดยการเพิ่มเงา, การทำมุมโค้ง, การตั้งค่าสีพื้นหลัง, และการจัดตำแหน่งฟอร์มให้อยู่ตรงกลางหน้าเว็บไซต์.

```
input[type="text"], input[type="email"], input[type="tel"], input[type="password"] {  
    width: 100%;  
    padding: 10px;  
    margin: 5px 0;  
    border-radius: 4px;  
    border: 1px solid #ddd;  
}  
button {  
    padding: 10px;  
    background-color: green;  
    color: white;  
    border: none;  
    cursor: pointer;  
    font-size: 16px;  
    width: 100%;  
    margin-top: 20px;  
}  
button:hover {  
    background-color: darkgreen;
```

รูปที่ 5.1.41

โค้ดนี้ใช้ในการจัดรูปแบบฟิล์ต `input` และ `button` ในฟอร์มให้มีลักษณะที่ดูเรียบร้อยและสวยงาม พร้อมทั้งเพิ่มการเปลี่ยนแปลงสีพื้นหลังของปุ่มเมื่อผู้ใช้ได้ตอบกับปุ่ม

```
<div class="logo">
    
</div>
```

รูปที่ 5.1.42

โค้ดนี้ใช้ในการแสดงโลโก้ในรูปแบบของภาพ โดยกำหนดขนาดและข้อความทางเลือกเมื่อรูปภาพไม่สามารถโหลดได้

```
<main>
  <h1>ขอบคุณสำหรับการชำระเงิน</h1>
  <p>คุณได้ชำระเงินเรียบร้อยแล้ว. ขอบคุณที่ใช้บริการ!
  <p>หมายเลขการจองของคุณ: <strong>#12345</strong></p>
</main>
```

รูปที่ 5.1.43

โค้ดนี้จะใช้แสดงข้อความยืนยันให้กับผู้ใช้หลังจากการทำการชำระเงินเสร็จสิ้น โดยจะมีการแสดงข้อความขอบคุณและหมายเลขการจองในรูปแบบที่ได้เด่น

```
<style>
    table {
        width: 100%;
        border-collapse: collapse;
    }
    th, td {
        padding: 10px;
        text-align: center;
        border: 1px solid #ddd;
    }
    th {
        background-color: #333;
        color: white;
    }
</style>
```

รูปที่ 5.1.44

โค้ดนี้ใช้ในการจัดรูปแบบตารางให้มีขนาดเต็มความกว้างของพื้นที่ที่มีอยู่ และทำให้เซลล์ในตารางดูเรียบร้อย โดยกำหนดให้ข้อความอยู่ตรงกลางและเพิ่มขอบให้กับเซลล์. นอกจากนี้ยังมีการกำหนดพื้นหลังและสีของข้อความในหัวตาราง (th).

```
<!DOCTYPE html>
<html lang="th">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ออกจากระบบ</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="logout-message">
        <h1>คุณได้ออกจากระบบแล้ว</h1>
        <p>ขอบคุณที่ใช้บริการ! กำลังพาคุณกลับไปยังหน้าแรก...</p>
        <script>
            setTimeout(function() {
                window.location.href = "login.html"; // หากใช้ลิงก์ไปที่หน้าแรกหลังจากออกจากระบบ
            }, 3000); // ให้เวลา 3 วินาทีในการเปลี่ยนหน้า
        </script>
    </div>
</body>
</html>
```

รูปที่ 5.1.45

โค้ดนี้แสดงข้อความหลังจากที่ผู้ใช้ทำการออกจากระบบ และหลังจากนั้นจะทำการเปลี่ยนหน้าไปที่หน้าเข้าสู่ระบบ (login.html) หลังจากหน่วงเวลา 3 วินาที

บทที่ 6
แบบฟอร์ม ประเมินผลระบบ

แบบประเมิน

1. ความสะดวกในการใช้งาน	5	4	3	2	1
1.1 ผู้ใช้สามารถเข้าสู่หน้าเว็บไซต์ได้ง่าย					
1.2 ผู้ใช้สามารถค้นหาข้อมูลตารางการเดินรถได้สะดวก					
1.3 การจองตัวสามารถทำได้ง่ายและรวดเร็ว					
1.4 ระบบการชำระเงินสามารถทำได้สะดวกและรวดเร็ว	5	4	3	2	1
2. ความชัดเจนของระบบ					
2.1 ระบบมีคำแนะนำการใช้งานที่ชัดเจน					
2.2 ข้อมูลการจองตัวแสดงให้ผู้ใช้เข้าใจง่าย					
2.3 ระบบแสดงสถานะการจองที่ถูกต้อง					
3. ความปลอดภัยในการใช้งาน	5	4	3	2	1
3.1 ข้อมูลส่วนบุคคลของผู้ใช้ได้รับการปกป้องเป็นอย่างดี					
3.2 ระบบมีการตรวจสอบความถูกต้องของข้อมูลการจอง					
3.3 ระบบมีการรักษาความปลอดภัยในการชำระเงิน					
4. ความเสียรของระบบ	5	4	3	2	1
4.1 ระบบไม่ล่มระหว่างการใช้งาน					
4.2 การประมวลผลข้อมูลทำได้เร็วและมีประสิทธิภาพ					
5. ความสามารถในการแสดงผลข้อมูล					
5.1 ระบบแสดงข้อมูลตารางการเดินรถได้ครบถ้วนและถูกต้อง					
5.2 ระบบแสดงข้อมูลสถานะการจองตัวได้อย่างถูกต้อง					

ภาคผนวก

วิธีการใช้งานโปรแกรม

1. ทำการดาวน์โหลดไฟล์ zip นี้



2. ทำการแตกไฟล์ โดยกด คลิกขวา และ extract file

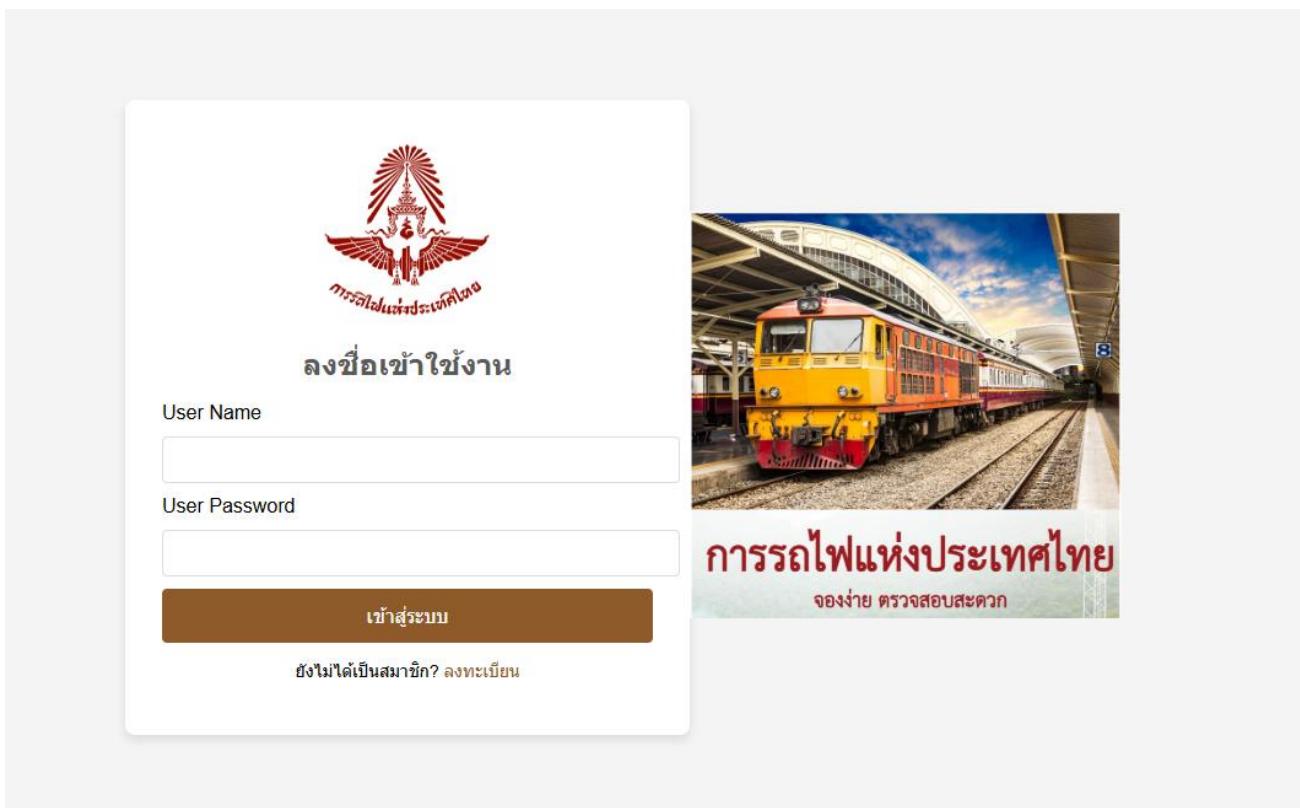
3. หลังจากแตกไฟล์จะได้หน้าต่างนี้

	bank	✓	18/2/2568 0:29	แฟ้ม JPG 33 KB
	book-ticket	✓	18/2/2568 0:25	Chrome HTML Do... 2 KB
	cancel	✓	5/2/2568 20:47	แฟ้ม PNG 7 KB
	cancel-ticket	✓	17/2/2568 21:57	Chrome HTML Do... 5 KB
	change	✓	5/2/2568 20:46	แฟ้ม PNG 8 KB
	change-ticket	✓	17/2/2568 22:05	Chrome HTML Do... 8 KB
	help-center	✓	17/2/2568 21:46	Chrome HTML Do... 2 KB
	home	✓	17/2/2568 22:30	Chrome HTML Do... 3 KB
	jong	✓	5/2/2568 20:46	แฟ้ม PNG 9 KB
	login	✓	17/2/2568 21:38	Chrome HTML Do... 2 KB
	logout	✓	17/2/2568 22:29	Chrome HTML Do... 1 KB
	payment	✓	18/2/2568 0:30	Chrome HTML Do... 5 KB
	personal-details	✓	17/2/2568 22:28	Chrome HTML Do... 5 KB
	poster	✓	5/2/2568 20:57	แฟ้ม JPG 1,613 KB
	poster1	✓	5/2/2568 20:59	แฟ้ม JPG 444 KB
	purchase-history	✓	17/2/2568 22:16	Chrome HTML Do... 3 KB
	register	✓	17/2/2568 21:41	Chrome HTML Do... 2 KB
	styles	✓	5/2/2568 21:12	CSS Source File 9 KB
	thank-you	✓	18/2/2568 0:17	Chrome HTML Do... 2 KB
	train	✓	5/2/2568 20:21	แฟ้ม PNG 619 KB
	train2	✓	5/2/2568 20:23	แฟ้ม PNG 252 KB
	user	✓	5/2/2568 20:48	แฟ้ม PNG 3 KB

4. จากนั้นให้กดตรงไฟล์ login.html



5. หลังจากกดก็จะได้หน้าตาของหน้าเว็บมานั้นเอง



บรรณานุกรม

สกาวรัตน์ จงพัฒนาการ. (2551). การวิเคราะห์และออกแบบระบบสารสนเทศ (พิมพ์ครั้งที่ 2). กรุงเทพฯ

การรถไฟแห่งประเทศไทย. (n.d.). หน้าแรก - ระบบจองตั๋วออนไลน์. การรถไฟแห่งประเทศไทย.
สืบค้นจาก

<https://www.dticket.railway.co.th/DTicketPublicWeb/home/Home>