

บทที่ 4

การวิเคราะห์และการออกแบบระบบ

การพัฒนาโครงการ เรื่อง ระบบจองวิทยากร มีการวิเคราะห์ออกแบบระบบซึ่งเป็นการศึกษางานเพื่อออกแบบระบบการทำงาน โดยมีขั้นตอนการวิเคราะห์ระบบ ดังนี้

4.1 การวิเคราะห์ระบบ

ออกแบบระบบและฟังก์ชันการใช้งานตามความต้องการของผู้ใช้งานโดยใช้ข้อมูลที่ได้จากการศึกษาการทำงานของแอปพลิเคชันและ requirement ของผู้ใช้งานและทำ Wireframe และ Design ตามลำดับ เนื่องจากลูกค้าของบริษัท มีผู้ที่มีความรู้ความสามารถที่รู้จักเป็นจำนวนมากที่เกษียณอายุแล้วแต่มีความรู้ความสามารถ จึงรู้สึกเสียดายที่ถ้าหากความรู้นั้นจะไม่ได้รับการแบ่งปันถ่ายทอดต่อไปสู่ผู้อื่น จึงต้องการสร้างพื้นที่เพื่อให้บุคคลเหล่านั้นได้ถ่ายทอดวิชาและต่อยอดความรู้ความสามารถของตน จึงได้ให้ทางบริษัทเปเปอร์เลสจำกัด พัฒนาระบบจองวิทยากรนี้ขึ้นมา ซึ่งฟังก์ชันและรายละเอียดของการจัดทำเป็นไปตามความต้องการของลูกค้าบริษัท

4.1.1 ผู้ดูแลระบบ (Admin) มีหน้าที่ดังนี้

4.1.1.1 จัดการคอร์ส

- สามารถเพิ่มคอร์สได้
- สามารถลบคอร์สได้
- สามารถแก้ไขคอร์สได้

4.1.1.2 จัดการเติมเงิน/เพิ่มคอยน์

- ตรวจสอบยอดโอนเพื่อเติมคอยน์สำหรับผู้ใช้งานและวิทยากร
- ถอนคอยน์เปลี่ยนเป็นเงินเพื่อโอนให้กับวิทยากรและผู้ใช้งานทั่วไป
- เพิ่มคอยน์ให้ผู้ใช้งานทั่วไปและวิทยากร
- ลบคอยน์ให้ผู้ใช้งานทั่วไปและวิทยากร

4.1.1.3 จัดการผู้ใช้งานทั่วไป

- เพิ่มผู้ใช้ทั่วไป
- ลบผู้ใช้ทั่วไป
- แก้ไขผู้ใช้ทั่วไป

4.1.1.4 จัดการผู้ใช้งานวิทยากร

- เพิ่มวิทยากร
- ลบวิทยากร
- แก้ไขวิทยากร

4.1.1.5 ตั้งค่าแบนเนอร์ (Banner)

- เพิ่มแบนเนอร์
- ลบแบนเนอร์
- แก้ไขแบนเนอร์

4.1.1.6 ตั้งค่าแท็ก (Tages)

- เพิ่มแท็ก
- ลบแท็ก
- แก้ไขแท็ก

4.1.2 ผู้ใช้งาน (User) มีหน้าที่ดังนี้

4.1.2.1 สมัครสมาชิก และล็อกอินผ่านไลน์แอปพลิเคชัน

4.1.2.2 แสดงรายชื่อวิทยากรที่สามารถจองได้พร้อมกับข้อมูลเพิ่มเติม เช่น ชื่อ, ตำแหน่ง, ความเชี่ยวชาญ เป็นต้น ผู้ใช้สามารถค้นหาวิทยากรตามเกณฑ์ที่ต้องการได้ เช่น ชื่อ, ความเชี่ยวชาญ

4.1.2.3 ผู้ใช้งานที่มีความรู้ความสามารถ อยากรับวิทยากร สามารถส่งคำขอเพื่อเปลี่ยนสถานะจากผู้ใช้งานปกติเป็นวิทยากรได้

4.1.2.4 แสดงประวัติการจองทั้งหมดของผู้ใช้ รวมถึงวันที่จอง, รายละเอียดวิทยากรที่ถูกจอง, และสถานะการจอง (ยืนยัน, รอการยืนยัน, ยกเลิก) เพื่อให้ผู้ใช้สามารถติดตามและดูประวัติการจองได้

4.1.2.5 มีการระบบชำระค่าวิทยากร ผ่านคอยน์ โดยการโอนเงินเพื่อเปลี่ยนเป็นคอยน์ผ่านระบบแอดมิน เมื่อใช้ไม่หมด ก็สามารถถอนคอยน์ออกมาได้

4.1.3 วิทยากร (User) มีหน้าที่ดังนี้

- 4.1.3.1 สมัครเป็นวิทยากรผ่านแอดมินและล็อกอินผ่านไลน์แอปพลิเคชัน
- 4.1.3.2 แสดงปฏิทินที่แสดงวันที่ว่างและวันที่ที่วิทยากรมีการจองไว้ โดยวิทยากรสามารถตั้งค่าปฏิทินกำหนดวันทำงานของตัวเองได้
- 4.1.3.3 แสดงคำขอจากผู้ใช้งานที่ทำการจองเข้ามา ซึ่งวิทยากรสามารถยืนยันการรับจองและปฏิเสธได้ (โดยมีช่องให้แจ้งเหตุผลในการปฏิเสธ) เพื่อให้ผู้ใช้สามารถติดตามและจัดการการจองได้
- 4.1.3.4 วิทยากรสามารถแก้ไขข้อมูลตัวเองได้ สามารถเพิ่มข้อมูลแนะนำความชำนาญของตัวเองเพิ่มเติมได้
- 4.1.3.5 วิทยากรสามารถแนะนำและเสนอการเพิ่มคอร์สได้ผ่านแอดมิน
- 4.1.3.6 มีรายการคอร์สที่วิทยากรสามารถเข้าร่วมได้ตามความชำนาญของตัวเอง และกำหนดราคาจ้างงานบรรยายเองได้

4.1.4 เพิ่มเติม

- 4.1.4.1 คอยน์ (Coin) เหรียญที่ใช้แทนเงินสด ในการจ่ายค่าตอบแทนวิทยากร (ใช้ได้เฉพาะในระบบการจองวิทยากรนี้เท่านั้น)

4.2 ขั้นตอนการวิเคราะห์ระบบ

4.2.1 ซีนา리오 (Scenario)

- 4.2.1.1 ส่วนของผู้ดูแลระบบ (Admin) แบ่งได้ดังนี้
 - จัดการประเภทผู้ใช้งาน เมื่อผู้ดูแล (**ADMIN**) ต้องการเพิ่ม, แก้ไข, ลบ ที่อยู่ในระบบที่ทำการลงทะเบียนในระบบจองวิทยากรเข้ามา
 - จัดการข้อมูลของแบนเนอร์ที่มีอยู่ในระบบ เมื่อผู้ดูแล (**Admin**) ต้องการเพิ่ม, แก้ไข, ลบ ที่อยู่ในระบบที่ทำการลงทะเบียนในระบบจองวิทยากรเข้ามา

- ค้นหาข้อมูลรายชื่อผู้ใช้งานในระบบ เมื่อผู้ดูแล (ADMIN) ต้องการค้นหาข้อมูลรายชื่อของผู้ที่อยู่ในระบบ สามารถค้นหาได้จากการกรองข้อมูลต่างๆ เช่น ไอดีผู้ใช้ ชื่อ และอื่นๆ

4.2.1.2 ส่วนของผู้ใช้งาน (User) แบ่งได้ดังนี้

- จัดการข้อมูลคอร์สในระบบ เมื่อผู้ใช้งาน (User) ต้องการเพิ่ม แก้ไข และลบ ข้อมูลของคอร์ส วันที่จองคอร์ส สถานที่ วิทยากร และอื่นๆ
- การค้นหาคอร์ส เมื่อผู้ใช้งาน (User) ต้องการหาคอร์สได้จากการกรองข้อมูลต่างๆ ตามแท็ก เช่น การสอนภาษา อาหาร ไอที และอื่นๆ

4.2.2 ยูสเคสไดอะแกรม (Use Case Diagram)

ยูสเคสไดอะแกรม แผนภาพที่ใช้แสดงการทำงานของผู้ใช้งานระบบ (User) และความสัมพันธ์กับระบบย่อย (Sub systems) ภายในระบบใหญ่ในการเขียน Usecase diagram ผู้ใช้งาน จะถูกกำหนดให้ว่าเป็น Actor และระบบย่อย คือ Use case จุดประสงค์หลักของการเขียน Use case diagram ก็เพื่อเล่าเรื่องราวทั้งหมดของระบบว่ามีการทำงานอะไรบ้างเป็นการดึง Requirement หรือเรื่องราวต่างๆของระบบจากผู้ใช้งานซึ่งถือว่าเป็นจุดเริ่มต้นในการวิเคราะห์และออกแบบระบบ

4.2.3 ยูสเคสนาธาธิฟ (Use Case Narrative)

ยูสเคสนาธาธิฟ (Use Case Narrative) เป็นรูปแบบหนึ่งในการเขียนเรื่องราวเพื่ออธิบายกระบวนการทำงานของระบบหรือแอปพลิเคชัน โดยจะอธิบายการทำงานของระบบในมุมมองของผู้ใช้งานหรือลูกค้า ต่อไปนี้คือตัวอย่างของยูสเคสนาธาธิฟ

Actor : ลูกค้า

Goal : ทำการสั่งซื้อสินค้าผ่านระบบออนไลน์

Preconditions : ลูกค้าเข้าสู่ระบบและมีบัญชีผู้ใช้งาน

Main Flow : ลูกค้าเข้าสู่ระบบออนไลน์ของเว็บไซต์ e-commerce.

ลูกค้าเรียกดูสินค้าที่ต้องการและเลือกเพิ่มเข้าตะกร้าสินค้า

ลูกค้าเข้าสู่หน้าตะกร้าสินค้าเพื่อตรวจสอบรายการสินค้าและปริมาณ

ลูกค้ากดปุ่ม "ดำเนินการชำระเงิน" เพื่อทำการสั่งซื้อ

- ระบบแสดงหน้าจอการชำระเงิน ลูกค้าเลือกวิธีการชำระเงิน (บัตรเครดิต, เงินสด, ธนาคาร)
- ลูกค้ากรอกข้อมูลการชำระเงินและที่อยู่ในการจัดส่ง
- ระบบทำการตรวจสอบและยืนยันข้อมูลการสั่งซื้อ
- ระบบดำเนินการสั่งซื้อ และทำการหักเงินจากบัญชีของลูกค้า
- ระบบส่งอีเมลยืนยันการสั่งซื้อพร้อมรายละเอียดการจัดส่ง

Postconditions: ลูกค้าทำการสั่งซื้อสินค้าเสร็จสิ้นและได้รับการยืนยัน

Alternative Flows: ข้อผิดพลาดในการชำระเงิน หากการชำระเงินไม่สำเร็จ

- ระบบจะแสดง ข้อความข้อผิดพลาดและแจ้งให้ลูกค้าทำการชำระเงินใหม่ ไม่มีสินค้าในตะกร้า: หากลูกค้ากดปุ่ม "ดำเนินการชำระเงิน" แต่ไม่มีสินค้าในตะกร้า
- ระบบจะแจ้งเตือนให้ลูกค้าเพิ่มสินค้าลงในตะกร้าก่อนไม่สามารถยืนยันการสั่งซื้อได้หากข้อมูลการจัดส่งไม่ถูกต้องหรือข้อมูลการชำระเงินไม่ถูกต้อง –
- ระบบจะแจ้งเตือนให้ลูกค้าแก้ไขข้อมูลก่อนยืนยันการสั่งซื้อ

Exception Flows: ระบบล่ม หากระบบมีปัญหาที่ทำให้ไม่สามารถดำเนินการสั่งซื้อได้ ระบบจะแสดงข้อความของข้อผิดพลาดและแจ้งให้ลูกค้าลองใหม่ในภายหลัง