บทที่ 3
วิธีดำเนินการวิจัย

ในการวิจัยครั้งนี้เป็นการวิจัยประยุกต์ (Applied Research) โดยใช้ระยะเวลาการศึกษาและวิจัยเป็นเวลา 9 เดือน ตั้งแต่เดือนพฤษภาคม 2554 ถึง กุมภาพันธ์ 2555 กรรมวิธีในการพัฒนาระบบจะใช้หลักการวงจรการพัฒนาระบบ (System Development Life Cycle) แบบน้ำตก (Waterfall Model) ของ Winston W. Royce ในส่วนการวิเคราะห์และออกแบบระบบเชิงวัตถุ (Object-Oriented Analysis and Design) โดยใช้แผนภาพ UML (Unified Modeling Language) ในการเขียนโปรแกรมใช้หลักการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) ในการออกแบบระบบฐานข้อมูลใช้แผนภาพความสัมพันธ์ระหว่างเอนทิตี (Entity-Relationship Diagram) ส่วนซอฟต์แวร์ที่ใช้คือ PHP, MySQL, JavaScript, XML, NetBean และ Apache web server โดยมีขั้นตอนการพัฒนาดังนี้ คือ

3.1 ขั้นตอนการวิเคราะห์ระบบ

3.1.1 แผนภาพสถิตย์ (Static Object-oriented Diagram)

เป็นแผนภาพที่แสดงโดเมนของปัญหา (Problem Domain) ในแบบสถิตย์ กล่าวคือจะแสดงการมีอยู่ของคลาสต่างๆ และความสัมพันธ์ของคลาสเหล่านั้นในระบบ แต่จะไม่แสดงกิจกรรมที่เกิดขึ้น ได้แก่

3.1.1.1 แผนภาพส่วนประกอบและความสัมพันธ์ (Use Case Diagram)

เป็นแผนภาพที่แสดงส่วนประกอบต่างๆ ของโดเมนปัญหาและความสัมพันธ์ของส่วนประกอบต่างๆ เหล่านั้น เราเรียกส่วนประกอบเหล่านั้นว่ายูสเคส (Use case) ซึ่งเปรียบเสมือนคลาสหนึ่งคลาส

3.1.1.2 แผนภาพคลาส (Class Diagram)

เป็นแผนภาพที่แสดงคลาสทั้งหมดที่อยู่ในโดเมนของปัญหาหรือยูสเคสหนึ่งๆ โดยแต่ละคลาสจะมีความสัมพันธ์เชิงแอบสแตรคชัน (Abstraction)

3.1.2 แผนภาพพลวัต (Dynamic Object-oriented Diagram)

เป็นแผนภาพแสดงกิจกรรมในโดเมนของปัญหา ซึ่งแบ่งได้ 2 แผนภาพย่อย ได้แก่

3.1.2.1 แผนภาพลำดับ (Sequence Diagram)

เป็นแผนภาพแสดงกิจกรรมรวมของระบบที่เกิดจากการเรียกใช้ฟังก์ชันที่มีอยู่ในคลาสต่างๆ

3.1.2.2 แผนภาพสถานะ (State Diagram)

เป็นแผนภาพที่เจาะจงลงไปที่ฟังก์ชันต่างๆ ของคลาสแต่ละตัวว่าแต่ละฟังกชันของคลาสจะทำให้คลาสมีสถานะใดบ้างและจะเปลี่ยนสถานะได้เมื่อใด

3.2 ขั้นตอนการออกแบบระบบ

3.2.1 ปรับปรุงแผนภาพ (Refinement)

เป็นการปรับปรุงแผนภาพต่างๆ ที่ได้จากขั้นตอนการวิเคราะห์ให้มีความละเอียดมากขึ้นซึ่งทำให้สื่อความหมายได้ดีขึ้นและสามารถนำไปเป็นต้นแบบในการพัฒนาระบบงานให้เกิดขึ้นจริงๆ ได้

3.2.2 การออกแบบสถาปัตยกรรมของแอพพลิเคชัน (Application Architecture Design)

เป็นหลักการออกแบบซอฟต์แวร์หรือแอพพลิเคชันที่จะสร้างขึ้นในคอมพิวเตอร์ โดยแผนภาพที่จะใช้เพื่อการนี้เรียกว่าแผนภาพคอมโพเนนท์ (Component Diagram) ซึ่งมีข้อดีคือจะทำให้สามารถแบ่งระบบงานทั้งหมดออกเป็นคอมโพเนนท์หรือโมดูลย่อยๆ ทำให้สามารถพัฒนางานในแต่ละคอมโพเนนท์ได้โดยอิสระ และสามารถแบ่งงานให้ทีมงานพัฒนาระบบได้ง่าย การแบ่งย่อยระบบทำได้หลายรูปแบบหนึ่งในวิธีที่นิยมกันทั่วไปคือวิธีการดริลดาวน์เอนจิเนียริง (Drill Down Engineering) โดยแบ่งระบบใหญ่ออกเป็น 3 ส่วนหลักๆ ได้แก่

3.2.2.1 ระบบย่อยตรรกในการนำเสนอ (Presentation Logic Subsystem)

ในส่วนนี้จะเกี่ยวข้องกับระบบย่อยที่ติดต่อกับผู้ใช้งานระบบโดยตรง ได้แก่ ส่วนต่อประสานผู้ใช้ (User Interface) ส่วนการแสดงผลลัพธ์ (Output) และส่วนการนำเข้า (Input)

3.2.2.2 ระบบย่อยตรรกในการทำงาน (Working Logic Subsystem) หรือ ตรรกทางธุรกิจ (Business Logic Subsystem)

เป็นส่วนของระบบที่เกี่ยวข้องกับการทำงานที่เกิดขึ้นจริงๆ ในหน่วยประมวลผลกลางของเครื่องคอมพิวเตอร์

3.2.2.3 ระบบย่อยตรรกของฐานข้อมูล (Database Logic Subsystem)

จะจำลองภาพของหน่วยย่อยข้อมูลที่ถูกจัดเก็บอยู่ในสื่อบันทึกข้อมูลทั้งในรูปแบบฐานข้อมูลหรือไฟล์ข้อมูล

3.2.3 การออกแบบสถาปัตยกรรมของระบบ (System Architecture Design)

เป็นการออกแบบฮาร์ดแวร์ของระบบโดยใช้แผนภาพดีพลอยเมนท์ (Deployment Diagram) ซึ่งมีประโยชน์ในการระบุประเภทฮาร์ดแวร์ หลักการสื่อสารข้อมูล รูปแบบแพลตฟอร์ม โดยรวม ตลอดจนระบบจัดการฐานข้อมูลที่เฉพาะเจาะจงอย่างชัดเจน อย่างไรก็ตามสถาปัตยกรรมของระบบต้องสอดคล้องกับสถาปัตยกรรมของแอพพลิเคชันด้วย

3.2.4 การออกแบบฐานข้อมูล (Persistent Data Design)

เป็นการออกแบบฐานข้อมูลเพื่อเก็บข้อมูลต่างๆ ซึ่งจะใช้แผนภาพความสัมพันธ์เอนทิตี (Entity Relationship Diagram) หรือย่อๆ ว่า E-R Diagram เพื่อออกแบบฐานข้อมูลเชิงสัมพันธ์ *(Relational Database)*

3.3 ขั้นตอนการสร้างโปรแกรม

ก่อนการเขียนโปรแกรมจะมีการจัดเตรียมติดตั้งเครื่องมือในการพัฒนาระบบที่จำเป็นพื้นฐานเสียก่อน ได้แก่

1) Apache Web Server สำหรับเว็บเซิร์ฟเวอร์
2) PHP และ JavaScript สำหรับภาษาโปรแกรม
3) MySQL Database System สำหรับระบบฐานข้อมูล
4) NetBean IDE สำหรับเขียนโปรแกรม

ในการเขียนโปรแกรมจะนำแผนภาพที่ได้จากขั้นตอนการวิเคราะห์ระบบมาแปลงให้เป็นโปรแกรมทีละโมดูลตามที่ได้ออกแบบไว้ และหลังจากเขียนโปรแกรมเสร็จจะมีการทดสอบในทุกขั้นตอน ได้แก่

1) การทดสอบโมดูลย่อย (Module Testing) เพื่อทดสอบโมดูลย่อย
2) การทดสอบรวม (Integrated Testing) เพื่อทดสอบโมดูลย่อยเมื่อมาทำงานร่วมกัน
3) การทดสอบระบบ (System Testing) เพื่อทดสอบระบบโดยรวม
4) การทดสอบเพื่อการยอมรับ (Acceptance Testing) เพื่อหาข้อยุติในความสมบูรณ์ของโปรแกรม

3.4 ขั้นตอนการทดลองใช้ระบบ

การทดลองระบบที่พัฒนาสำเร็จแล้ว ผู้วิจัยมีขั้นตอนการทดลองใช้ระบบดังต่อไปนี้

1) การเลือกกลุ่มตัวอย่างของนักศึกษาที่จะสอบผ่านระบบออนไลน์ เพื่อทดลองการใช้ระบบ

2) การเลือกรูปแบบข้อสอบ เพื่อให้เหมาะสมกับการทดลอง

3) การจัดการห้องสอบ เพื่อให้เหมาะสมกับขนาดของกลุ่มตัวอย่างของนักศึกษาที่จะเข้าสอบ

4) การดำเนินการสอบแบบออนไลน์ เพื่อจัดการการสอบให้มีประสิทธิภาพในสภาพแวดล้อมจริง

3.5 ขั้นตอนการประเมินความพึงพอใจการใช้ระบบ

การประเมินความพึงพอใจของผู้ใช้ระบบเพื่อนำผลที่ได้รับไปปรับปรุงระบบในอนาคต