# บทที่ 3 การวิเคราะห์และการออกแบบระบบ

การวิเคราะห์และออกแบบระบบนั้นเป็นขั้นตอนที่สำคัญในการพัฒนาโปรแกรมประยุกต์เพราะการวิเคราะห์และออกแบบระบบที่ดีนั้นจะทำให้กำหนดรูปแบบและขอบเขตในการพัฒนา และช่วยให้การปรับแก้ไขระบบสามารถทำได้ง่ายขึ้น ซึ่งในบทนี้จะกล่าวถึงการวิเคราะห์และออกแบบระบบโดยใช้ภาษา Unified Modeling Language (UML) ซึ่งเป็นภาษาที่สื่อด้วยภาพและใช้ในการสร้างและจัดทำเอกสารเพื่อประกอบการพัฒนาระบบ โดยจะใช้แผนภาพต่าง ๆ ดังนี้ Activity Diagram ใช้เพื่อแสดงภาพรวมของระบบ Use Case Diagram ใช้เพื่อแสดงความเกี่ยวข้องระว่างผู้ใช้กับระบบ Sequence Diagram ใช้แสดงขั้นตอนการทำงานของระบบ Class Diagram ใช้แสดงข้อมูลความสัมพันธ์ระหว่างคลาสต่าง ๆ ที่มีในระบบ

1. **ภาพรวมของระบบ**

ภาพรวมของระบบนี้จะนำเสนอโดย Activity Diagram เป็นแผนภาพแสดงกิจกรรมใช้เพื่ออธิบายกิจกรรมที่เกิดขึ้นในลักษณะการไหลของการทำงาน (Workflow) ซึ่งจะมีลักษณะเดียวกับการแสดงขั้นตอนการทำงานของระบบ (Flowchart) โดยขั้นตอนในการทำงานในแต่ละขั้นตอนจะเรียกว่า Activity และ Activity นี้สามารถเป็นการทำงานต่างๆได้ เช่น การคำนวณผลลัพธ์ การเปลี่ยนแปลงสภานะของระบบ การส่งค่ากลับคืน การส่งสัญญาณ การเรียกตัวดำเนินการอื่นๆเพื่อทำงาน การสร้างหรือทำลายวัตถุ

**ตารางที่ 3.1 สัญลักษณ์ที่ใช้ในแผนภาพกิจกรรม**

|  |  |  |
| --- | --- | --- |
| สัญลักษณ์ | ชื่อ | ความหมาย |
|  | initial state | จุดเริ่มต้นของกิจกรรม |
|  | End state | จุดสิ้นสุดของกิจกรรมทั้งหมด |
|  | Activity | กิจกรรม |
|  | Connection | เส้นทางการไหลของกิจกรรม |
|  | Join | จุดรวมของกิจกกรรม |
|  | Fork | การแยกกิจกรรมแล้วทำไปพร้อมๆกัน |
|  | Decision | ทางเลือกของระบบ |

1. **Use Case Diagram**

Use Case Diagram คือแผนภาพที่แสดงการทำงานของผู้ใช้ระบบ และความสัมพันธ์กับระบบย่อย ภายในระบบใหญ่ ในการเขียน Use Case Diagram ผู้ใช้ระบบจะถูกกำหนดให้เป็น Actor และระบบย่อยคือ Use Case จุดประสงค์หลักของการเขียน Use Case Diagram เพื่อเล่าเรื่องราวทั้งหมดของระบบว่ามีการทำงานอะไรบ้าง เป็นการดึง Requirement ของระบบจากผู้ใช้งาน ซึ่งเป็นจุดเริ่มต้นของการวิเคราะห์และออกแบบระบบ

**ตารางที่ 3.2 สัญลักษณ์ที่ใช้ใน Use Case Diagram**

|  |  |  |
| --- | --- | --- |
| สัญลักษณ์ | ชื่อ | ความหมาย |
|  | Use Case | ความสามาระ/หน้าที่ของระบบ |
| https://upload.wikimedia.org/wikipedia/commons/thumb/a/a6/Use_case_diagram_actor_people.svg/2000px-Use_case_diagram_actor_people.svg.png | Actor | ผู้กระทำ/ผู้ใช้งาน Use Case |
|  | Connection | เส้นเชื่อมระหว่าง Actor กับ Use Case |
| <<Include>> | Include Relationship | เป็นความสัมพันธ์ในกรณีที่ Use Case หนึ่งไปเรียกใช้หรือดึงกิจกรรมของอีก Use Case หนึ่ง เพื่อให้กิจกรรมนั้นเกิดจริงในตนเอง |
| <<Extend>> | Extend Relationship | เป็นความสัมพันธ์ที่ Use Case หนึ่งไปมีผลต่อการทำงานปกติของอีก Use Case หนึ่ง |

1. **รายละเอียดของยูสเคส**

Use Case Specification คือการแสดงรายละเอียดของแต่งละยูสเคส โดยที่ 1 Use Case ต่อ 1 Use Case Specification ซึ่งในแต่ละ Use Case Specification จะประกอบไปด้วย ชื่อยูสเคส ID ของยูสเคส ชื่อ Actor ที่เกี่ยวข้องกับยูสเคสนั้นๆ Preconditions คือเงื่อนไขที่ต้องทำก่อนหน้ายูสเคสนี้ Flow of events คือลำดับของการทำงานในยูสเคสนี้ Postconditions คือข้อมูลออกหลังการจบการทำงานของยูสเคส

1. **Sequence Diagram**

การแสดงขั้นตอนการทำงานของระบบนี้จะนำเสนอโดยใช้ Sequence Diagram ซึ่งเป็นการแสดงให้เห็นถึงการติดต่อและการเปลี่ยนแปลงสถานะของ Object เพื่อตอบสนองเหตุการณ์ที่เกิดขึ้นหรือสิ่งที่มากระตุ้นการตอบสนองโดยการเปลี่ยนสถานะตามลำดับเหตุการณ์ที่เกิดขึ้นในเวลาที่กำหนด ซึ่ง Message ที่เกิดขึ้นระหว่างแต่ละ Class จะสามารถนำไปสร้าง Method ใน Class ที่เกี่ยวข้องได้

|  |  |  |
| --- | --- | --- |
| สัญลักษณ์ | ชื่อ | ความหมาย |
|  | Object | Object ที่ตอบสนองกับระบบ |
|  | Message | ข้อความที่ Object หนึ่งส่งให้อีก Object เพื่อติดต่อสื่อสารกัน |
|  | Activation | จุดเริ่มต้น และสิ้นสุดของกิจกรรมระหว่างที่ Object มีชีวิตอยู่ |
|  | Return | การตอบกลับเมื่อสิ้นสุดกิจกรรมของ Object |
|  | Entity Class | คลาสที่ทำหน้าที่เก็บข้อมูล |
|  | Control Class | คลาสที่ทำหน้าที่ติดต่อประสานงานระหว่าง Entity Class และ User Interface Class |
|  | User Interface Class | คลาสที่ทำหน้าที่เป็นส่วนต่อประสานกับผู้ใช้งาน |
| https://upload.wikimedia.org/wikipedia/commons/thumb/a/a6/Use_case_diagram_actor_people.svg/2000px-Use_case_diagram_actor_people.svg.png | Actor | ผู้เกี่ยวข้องกับระบบ |
|  | Lifeline | เส้นแสดงการมีชีวิตของ Object |