

## Recipe Organizer

เสนอ

ผศ.ดร. ศุภชัย ไทยเจริญ

โดย

นายปริญญา	บุญพามา	65102010119
นางสาวพินธูชา	ฤกษ์พินธุ์วัฒน์	65102010421

โครงการนี้เป็นส่วนหนึ่งของการศึกษารายวิชา CP352/353  
System Analysis and Design การวิเคราะห์และออกแบบระบบ  
มหาวิทยาลัยศรีนครินทรวิโรฒ  
ภาคการศึกษาที่ 1 ประจำปีการศึกษา 2567

## สารบัญ

<b>หัวข้อและบทคัดย่อ.....</b>	<b>4</b>
หัวข้อ.....	4
บทคัดย่อ.....	4
<b>การวางแผน (Planning).....</b>	<b>5</b>
คำร้องขอระบบ (System Request).....	5
แผนงาน (Workplan).....	6
Simple Work Breakdown Structure.....	6
Workplan: Gantt Chart.....	7
<b>การวิเคราะห์ (Analysis).....</b>	<b>7</b>
Requirement Definitions.....	7
Functional Requirements.....	7
Non-Functional Requirements.....	8
Use Case Diagrams.....	8
Major Use Cases.....	8
Activity Diagram.....	9
Use Case Description.....	10
Analysis Class Diagram.....	11
Sequence Diagram.....	11
Communication Diagram.....	12
Behavioral State Machine.....	12
<b>การออกแบบ (Design).....</b>	<b>13</b>
Design Class Diagram.....	13
Object Constraint Language (OCL).....	13
Method Contract.....	14
Method Specification.....	15
Data Base Design.....	16
Mapping Class Diagram to Relational Database.....	16

User Interface — Windows Navigation Diagram (WND).....	17
Windows Navigation Diagram สำหรับ Recipe Organizer.....	17
<b>Implementation.....</b>	<b>18</b>
Screenshot Application.....	19
การทำงานของ Application.....	21
<b>บทสรุป.....</b>	<b>22</b>

ชื่อโครงการ : Recipe Organizer

ชื่อผู้ทำโครงการ :

1. นายปริญญา บุญพามา 65102010119

2. นางสาวพินรุชา ฤกษ์พินธุ์ 65102010421

ครูที่ปรึกษา ผศ.ดร.ศุภชัย ไทยเจริญ

สถานศึกษา มหาวิทยาลัยศรีนครินทรวิโรฒ

ปีการศึกษา 2567

## หัวข้อและบทคัดย่อ

### หัวข้อ

แอปพลิเคชัน Recipe Organizer เป็นแอปจัดการสูตรอาหารส่วนตัว

### บทคัดย่อ

แอปพลิเคชัน Recipe Organizer เป็นระบบจัดการสูตรอาหารส่วนตัว ที่ช่วยให้ผู้ใช้สามารถบันทึกและจัดการสูตรอาหารได้อย่างสะดวกสบาย โดยผู้ใช้สามารถเพิ่มข้อมูลส่วนผสม วิธีทำ แบนภาพประกอบ และบันทึกส่วนตัวได้ พร้อมทั้งมีฟีเจอร์ค้นหาสูตรอาหารตามประเภทและวัตถุดิบ แอปพลิเคชันนี้มุ่งเน้นไปที่กลุ่มผู้ใช้งานที่สนใจทำอาหาร เช่น นักเรียน นักศึกษา และแม่บ้าน โดยมีวัตถุประสงค์หลักเพื่อช่วยให้การบันทึกสูตรอาหารอยู่ในรูปแบบดิจิทัล ลดความเสี่ยงในการสูญหายของสูตรอาหารที่เป็นกระดาษ และสนับสนุนการแชร์สูตรผ่านโซเชียลมีเดีย แอปสามารถใช้งานได้บนระบบ Desktop มีการพัฒนาให้ข้อมูลปลอดภัย และตอบสนองรวดเร็ว

โครงสร้างของการพัฒนาแอปพลิเคชันนี้ประกอบด้วย การวางแผนโครงการ การวิเคราะห์ความต้องการ การออกแบบ การพัฒนา และการทดสอบ ซึ่งครอบคลุมถึงการจัดทำ System Request, การสร้าง Use-Case Diagram, Activity Diagram และ Class Diagram ที่แสดงโครงสร้างหลักของแอป เช่น คลาส Recipe, User และ ImageAttachment พร้อม Behavioral State Machine ที่ช่วยให้การจัดการสถานะของข้อมูลในคลาส Recipe เป็นไปอย่างถูกต้อง

## การวางแผน (Planning)

### คำร้องขอระบบ (System Request)

ระบบ Recipe Organizer ได้รับการพัฒนาเพื่อตอบสนองความต้องการของผู้ใช้ในการจัดเก็บและจัดการสูตรอาหารส่วนตัวอย่างมีประสิทธิภาพ โดยมีเป้าหมายหลักเพื่อ

1. ให้ผู้ใช้สามารถเพิ่ม แก้ไข และลบข้อมูลสูตรอาหารของตนได้สะดวก รวมถึงแนบภาพประกอบและบันทึกส่วนตัว
2. มีฟังก์ชันค้นหาสูตรอาหารตามประเภทและวัตถุดิบ เพื่อช่วยให้ผู้ใช้ค้นหาสูตรที่ต้องการได้อย่างรวดเร็ว
3. รองรับการแชร์สูตรอาหารผ่านโซเชียลมีเดียต่าง ๆ เพื่อให้ผู้ใช้สามารถแบ่งปันสูตรที่ตนชื่นชอบกับผู้อื่นได้
4. ใช้งานได้ในระบบปฏิบัติการ Desktop และจัดเก็บข้อมูลที่ปลอดภัย เพื่อให้แอปพลิเคชันสามารถเข้าถึงได้สะดวก

### แผนงาน (Workplan)

#### Simple Work Breakdown Structure

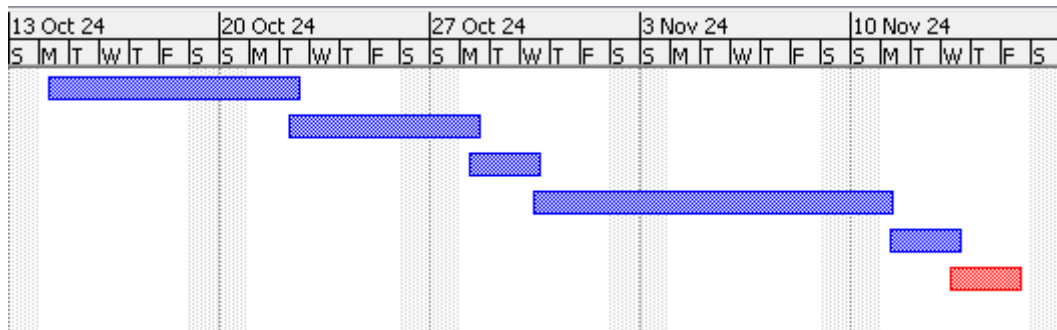
Name	Duration	Start	Finish
Requirement Analysis	7 days?	10/13/24 8:00 AM	10/22/24 5:00 PM
System	5 days?	10/22/24 8:00 AM	10/28/24 5:00 PM
Design	3 days?	10/28/24 8:00 AM	10/30/24 5:00 PM
Development	8 days?	10/30/24 11:00 AM	11/11/24 11:00 AM
Testing & Verification	3 days?	11/11/24 8:00 AM	11/13/24 5:00 PM
Deployment & Maintenance	3 days?	11/13/24 8:00 AM	11/15/24 5:00 PM

โครงสร้างการแบ่งงาน (WBS) สำหรับโครงการพัฒนาแอป Recipe Organizer ประกอบด้วยงานหลัก ๆ ดังนี้

1. การวิเคราะห์ความต้องการ (Requirement Analysis)
  - เก็บรวบรวมความต้องการจากผู้ใช้
  - กำหนด Functional และ Non-Functional Requirements
  - สร้าง Use-Case Diagram และ Activity Diagram
2. การออกแบบระบบ (System Design)
  - ออกแบบ User Interface (UI) ของแอปพลิเคชัน

- ### 3. การพัฒนา (Development)

## Workplan: Gantt Chart



## การวิเคราะห์ (Analysis)

## Requirement Definitions

## Functional Requirements

1. ผู้ใช้สามารถเพิ่ม แก้ไข และลบสูตรอาหารของตนเองได้ในแอปพลิเคชัน
2. ระบบสามารถบันทึกและเรียกดูข้อมูลสูตรอาหารได้ รวมถึงรายละเอียดของส่วนผสมและขั้นตอนการทำ
3. ผู้ใช้สามารถค้นหาสูตรอาหารตามประเภทหรือวัตถุดิบที่ต้องการได้

4. แอปพลิเคชันมีฟีเจอร์แชร์สูตรอาหารผ่านช่องทางโซเชียลมีเดีย
5. ระบบมีการแนบรูปภาพประกอบสูตรอาหารเพื่อเพิ่มความชัดเจน

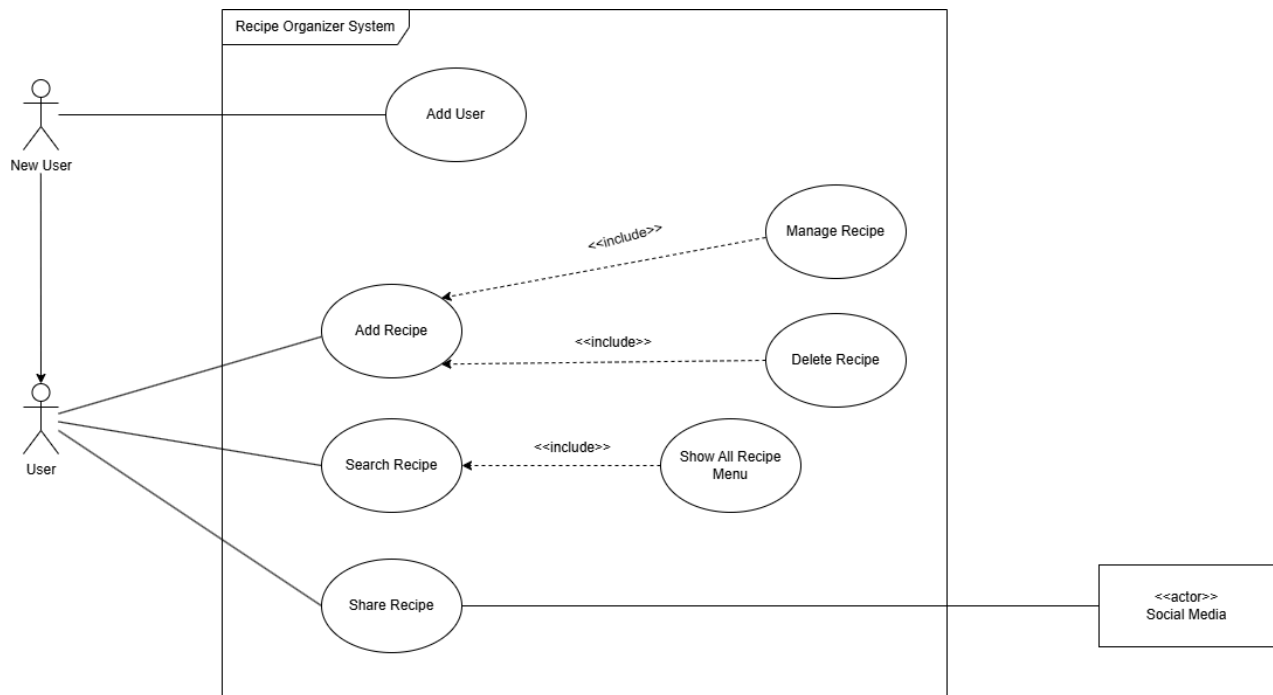
### Non-Functional Requirements

1. แอปพลิเคชันต้องสามารถใช้งานได้บนระบบ Desktop
2. ข้อมูลผู้ใช้ต้องถูกเก็บไว้อย่างปลอดภัยและเข้าถึงได้รวดเร็ว
3. ระบบต้องรองรับผู้ใช้หลายคนและสามารถจัดการข้อมูลได้ดี โดยไม่ทำให้ระบบช้าลง
4. แอปพลิเคชันต้องมีประสบการณ์ใช้งานที่เรียบง่ายและตอบสนองได้รวดเร็ว

### Use Case Diagrams

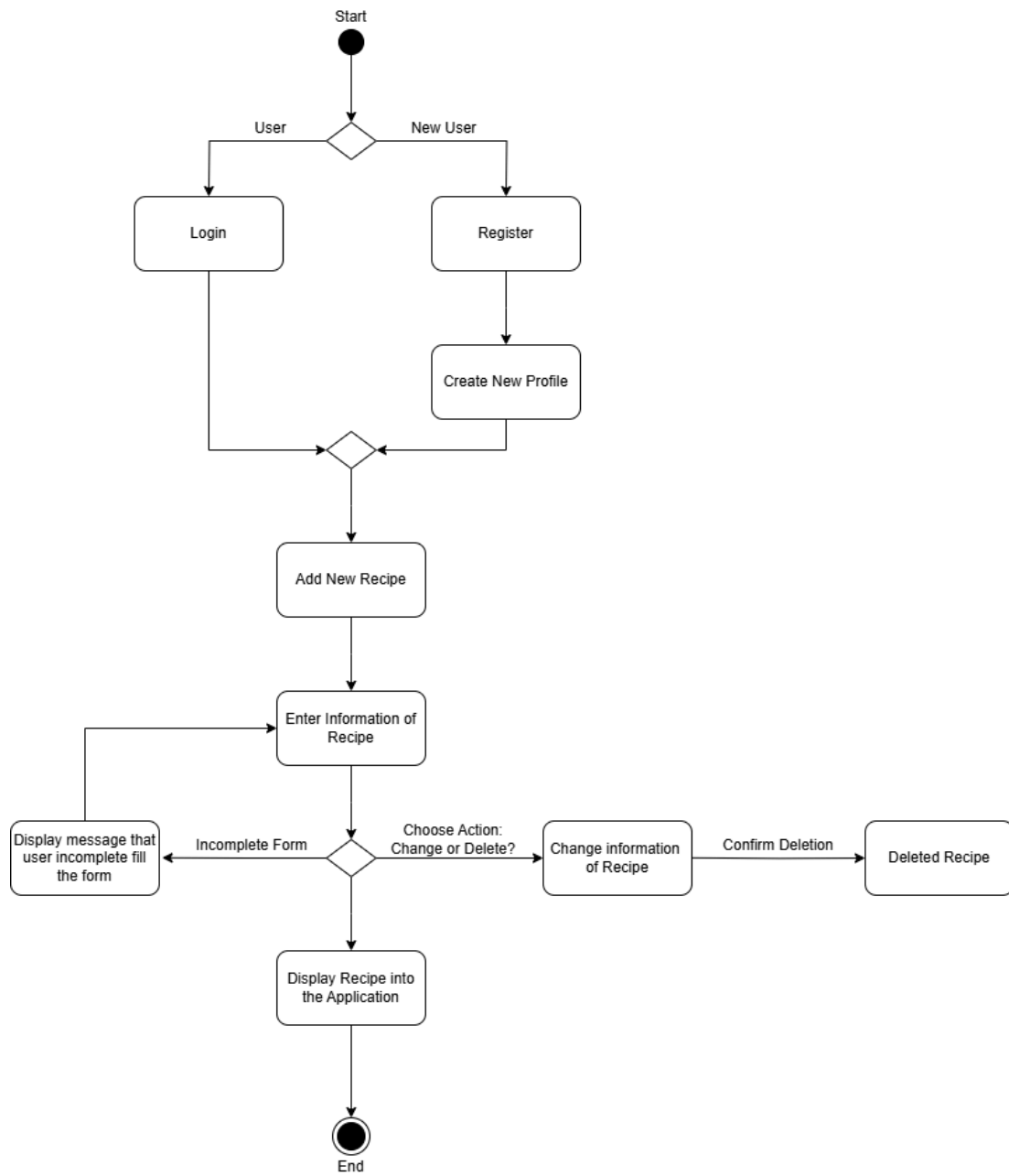
#### Major Use Cases

1. **Add Recipe** : ผู้ใช้เพิ่มสูตรอาหารใหม่ลงในระบบ
2. **Search Recipe** : ผู้ใช้ค้นหาสูตรอาหารตามประเภทหรือวัตถุดิบ
3. **Share Recipe** : ผู้ใช้แชร์สูตรอาหารไปยังโซเชียลมีเดีย



## Activity Diagram

สำหรับ Use Case "Add New Recipe"





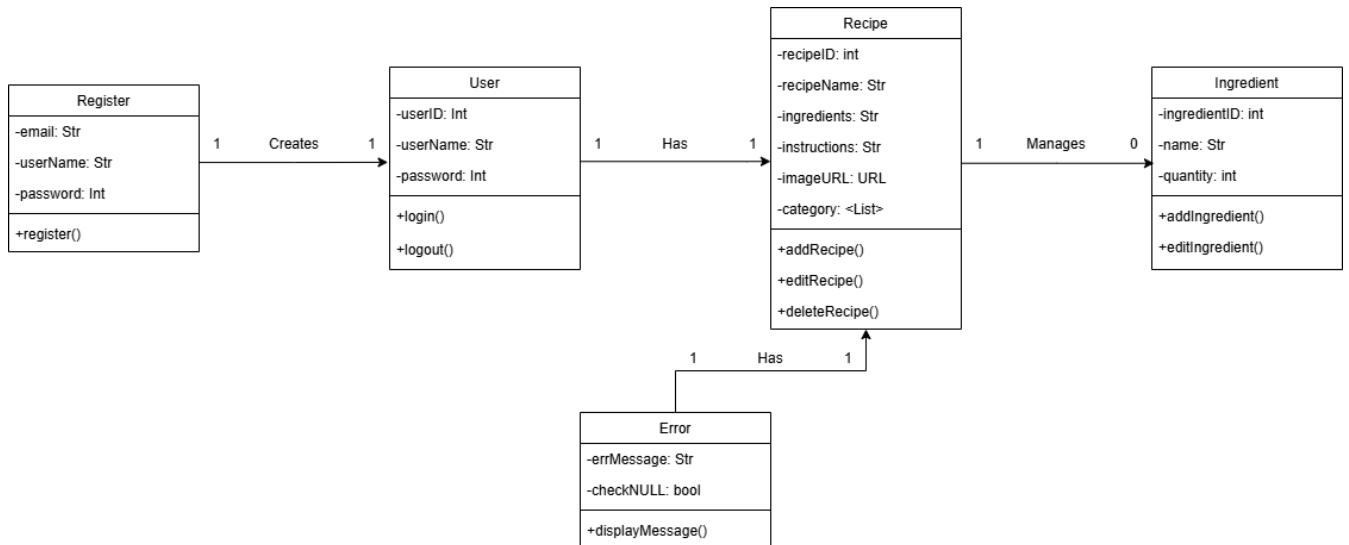
## Use Case Description

สำหรับ Use Case "Add New Recipe"

Use Case Name: Add New Recipe	ID: 1	Important Level: Primary
Primary Actor: User	Use Case Type: Primary, Essential	
Stakeholders and interests:  User: ต้องการเพิ่ม หรือแก้ไข/ลบเมนูสูตรอาหาร		
Brief Description:  Use case ช่วยให้ผู้ใช้งานสามารถเพิ่มสูตรอาหารได้โดยกรอกข้อมูลที่จำเป็น		
Trigger: ผู้ใช้งานต้องการเพิ่มสูตรอาหารใหม่		
Type of Trigger: User Initiated		
Relationships:  Association: ผู้ใช้ Include: อัปโหลดภาพและการตรวจสอบความครบถ้วนของข้อมูล Extend: การแจ้งเตือนการกรอกข้อมูลไม่ครบถ้วน Generalization: -		
Normal Flow of Events:  1. ผู้ใช้ทำการเข้าสู่ระบบ 2. ผู้ใช้อัปโหลดรูปภาพที่เกี่ยวข้อง 3. ผู้ใช้อัปโหลดรูปภาพที่เกี่ยวข้อง 4. ผู้ใช้กรอกข้อมูลชื่อสูตรอาหาร ส่วนผสม และวิธีทำ 5. ผู้ใช้กดปุ่ม "Submit" และระบบจะมีการเช็คค่าผู้ใช้กรอกข้อมูลครบถ้วนหรือไม่ ถ้าไม่ครบระบบจะแจ้งเตือนให้ผู้ใช้กรอกข้อมูลให้ครบ 6. ระบบบันทึกข้อมูลสูตรอาหารลงในเซฟเวอร์		
SubFlows:  บัญชีผู้ใช้งาน -ผู้ใช้ต้องทำการเข้าสู่ระบบ การเพิ่มสูตรอาหาร -หากผู้ใช้ไม่ได้ใส่รูปภาพ ก็ยังสามารถที่จะบันทึกได้ตามปกติ แก้ไขสูตรอาหาร -กดที่แก้ไข -แก้ไขสูตรอาหาร -กดที่ปุ่มยืนยัน ลบสูตรอาหาร -กดที่แก้ไข -ระบบจะทำการแสดงข้อความเพื่อยืนยันกับผู้ใช้ใช้งานว่าต้องการที่จะลบหรือไม่ -ผู้ใช้ยืนยันที่จะลบ -ระบบจะนำสูตรอาหารนั้นออกจากฐานข้อมูล		
Alternative/Exceptional Flows:  บัญชีผู้ใช้งาน -ผู้ใช้งานเคยลงทะเบียนกับแอปพลิคेशनแล้วหรือไม่ -ผู้ใช้ใส่ชื่อไม่ถูกต้อง -ผู้ใช้ใส่รหัสไม่ถูกต้อง การเพิ่มสูตรอาหาร -หากผู้ใช้กรอกข้อมูลไม่ครบ ระบบจะแสดงข้อความแจ้งเตือนให้ผู้ใช้กรอกข้อมูลให้ครบถ้วน -หากผู้ใช้ไม่ได้ใส่หัวข้อ "หมวดหมู่" ให้กับข้อมูลของสูตรอาหาร จะไม่สามารถเพิ่มลงในแอปพลิคेशनได้		

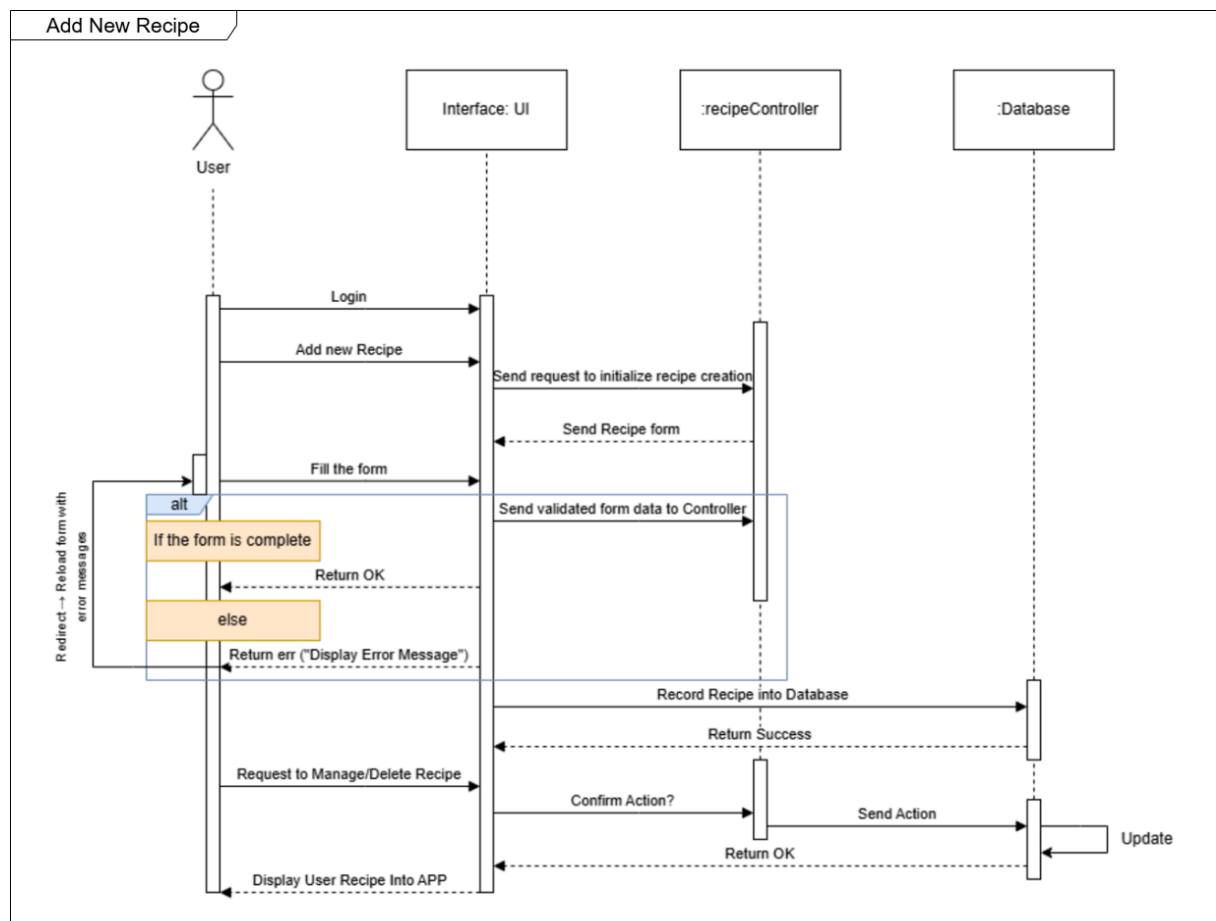
## Analysis Class Diagram

สำหรับ Use Case "Add New Recipe"



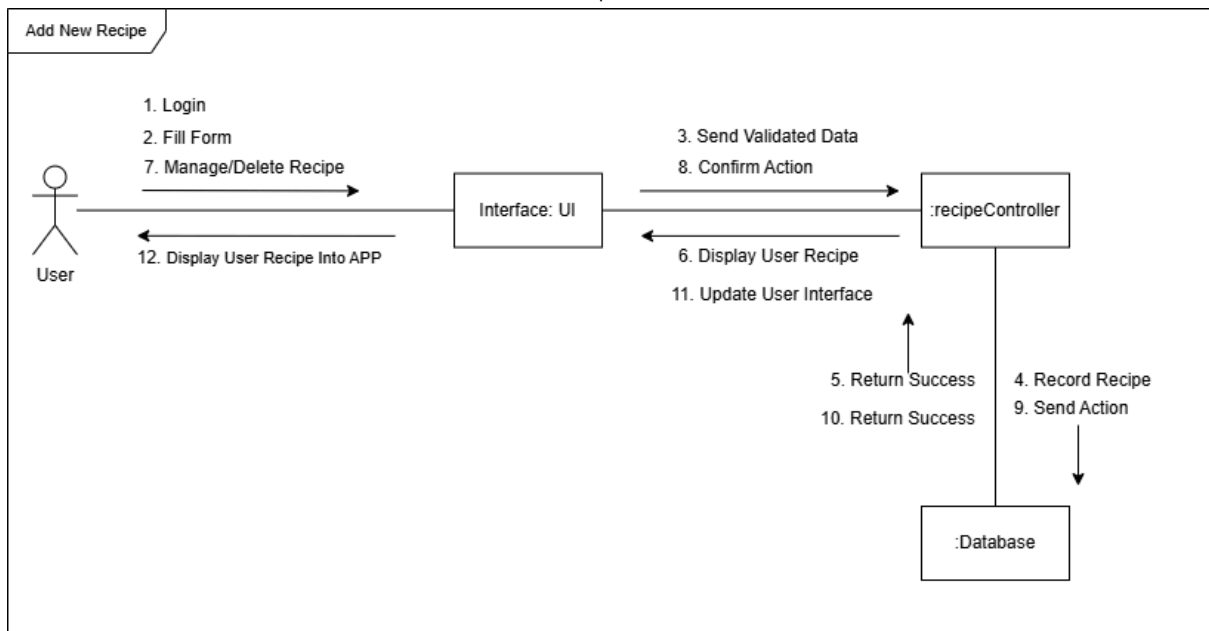
## Sequence Diagram

สำหรับ Use Case "Add New Recipe"



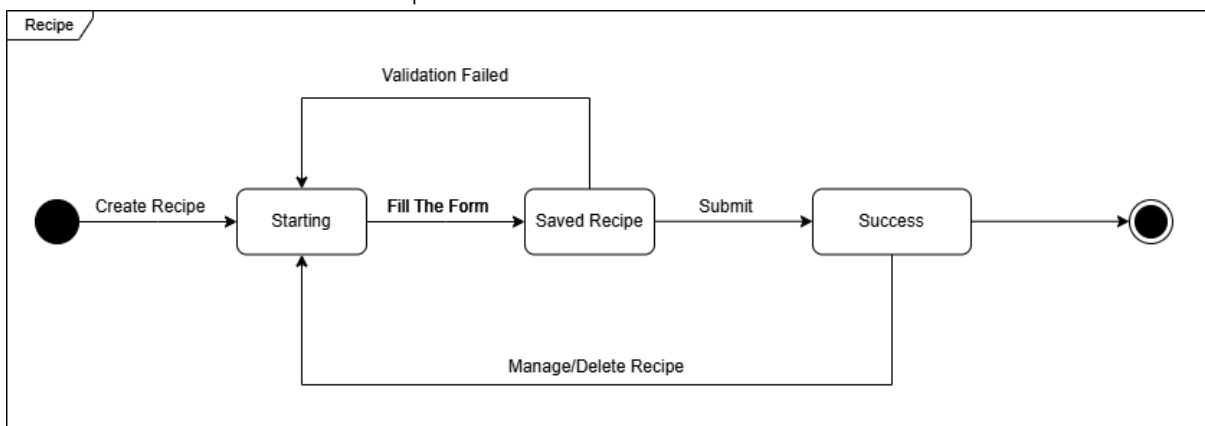
## Communication Diagram

สำหรับ Use Case "Add New Recipe"



## Behavioral State Machine

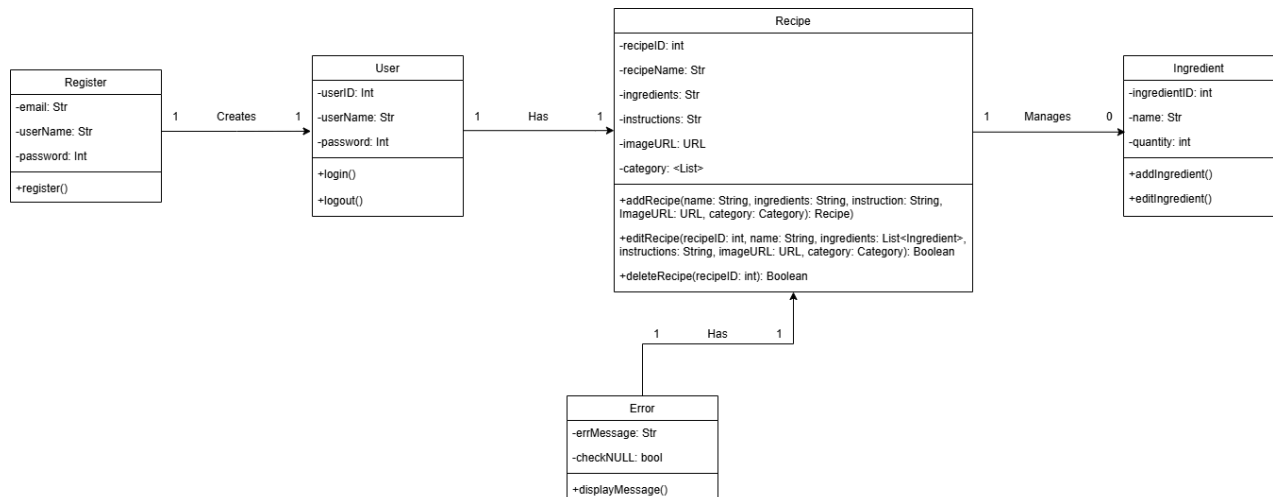
สำหรับคลาส "Recipe"



## การออกแบบ (Design)

### Design Class Diagram

สำหรับคลาส “Recipe”



### Object Constraint Language (OCL)

ตัวอย่างสำหรับ Method addRecipe()

```
[ ] context Recipe::addRecipe(name: String, ingredients: List<Ingredient>, instructions: String,
                               imageUrl: String, category: Category): Recipe
pre: name <> "" and ingredients->notEmpty() and instructions <> "" and category <> null
post: Recipe.allInstances()->exists(r | r.name = name and r.ingredients = ingredients and
                                     r.instructions = instructions and r.imageUrl = imageUrl and r.category = category)
```

อัลกอริทึม (Structured English) สำหรับ Method addRecipe()

1. รับข้อมูล name, ingredients, instructions, imageUrl, และ category
2. ตรวจสอบว่าข้อมูลที่รับเข้ามาไม่ว่างเปล่า
3. สร้างออบเจกต์ Recipe ใหม่
4. เพิ่มออบเจกต์ที่สร้างเข้าไปในรายการสูตรอาหาร
5. คืนค่าออบเจกต์ Recipe ที่สร้างใหม่

## Method Contract

สำหรับ Method "editRecipe"

<b>Method Name:</b> EditRecipe	<b>Class name:</b> Recipe	<b>ID:</b> 1
<b>Clients:</b> User		
<b>Associated Use case:</b> ผู้ใช้ต้องการแก้ไขข้อมูลสูตรอาหารที่มีอยู่ในระบบ		
<b>Description of Responsibilities:</b> รับผิดชอบการอัปเดตข้อมูลของสูตรอาหารที่ระบุโดยปรับเปลี่ยนค่าต่างๆ พร้อมบันทึกการเปลี่ยนแปลงเข้าสู่ฐานข้อมูล		
<b>Arguments Received:</b> None		
<b>Type of Value Returned:</b> Boolean		
<b>Pre-conditions:</b> recipeID ต้องตรงกับสูตรที่มีอยู่ในระบบ และ name, ingredients, instructions, category ต้องไม่ว่าง		
<b>Post-conditions:</b> ข้อมูลของสูตรอาหารที่มี recipeID ตรงกันจะถูกอัปเดต และคืนค่า true หากการอัปเดตสำเร็จ		

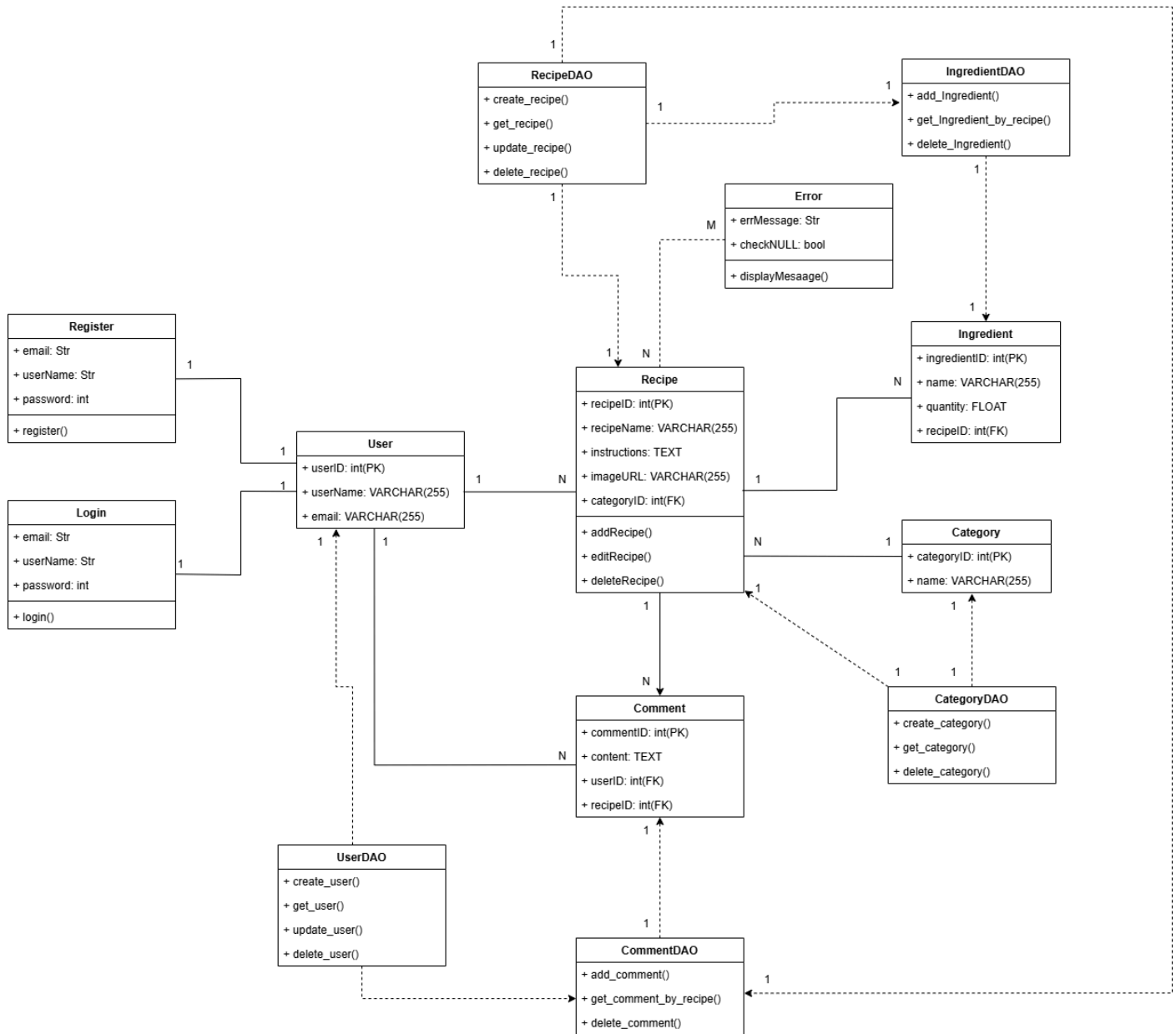
## Method Specification

สำหรับ Method "editRecipe"

Method Name: editRecipe	Class name: Recipe	ID: 1
Contract ID: User	Programmer: Recipe	Date Due: 15 November 2024
Programming Language: Electron		
Trigger/Events:  การเรียกเมธอดเพื่อแก้ไขข้อมูลสูตรอาหาร เมื่อแก้ไขสำเร็จจะส่งสถานะว่าเป็น True		
Arguments Received and Data Types:  editRecipe(Boolean)	Notes:  คำร้องที่ต้องการการแก้ไขข้อมูลจากผู้ใช้	
Messages Sent & Arguments Passed:  -	Data Type:  -	Notes:  -
Argument Returned and Data Type:  editRecipe(Boolean)	Notes:  คืนค่า True หากสถานะการอัปเดตเสร็จสิ้น	
Algorithm Specification:  -ตรวจสอบว่ามีการเข้าสู่ระบบหรือไม่ -ตรวจสอบว่ามีสูตรอาหารที่มี recipeID ตรงกันในระบบหรือไม่		
Misc.Note: ไม่มี		

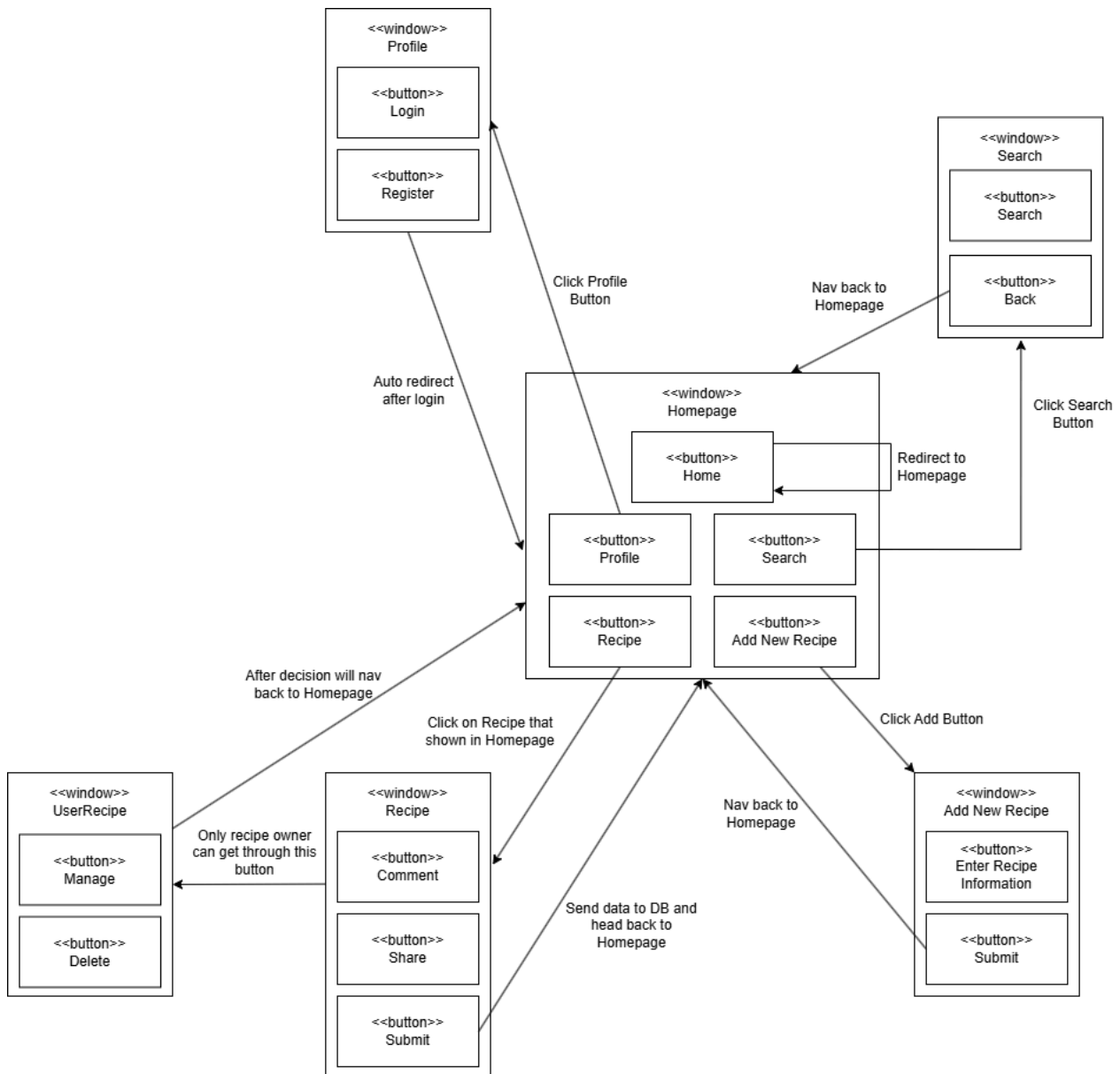
## Data Base Design

### Mapping Class Diagram to Relational Database



## User Interface — Windows Navigation Diagram (WND)

### Windows Navigation Diagram สำหรับ Recipe Organizer





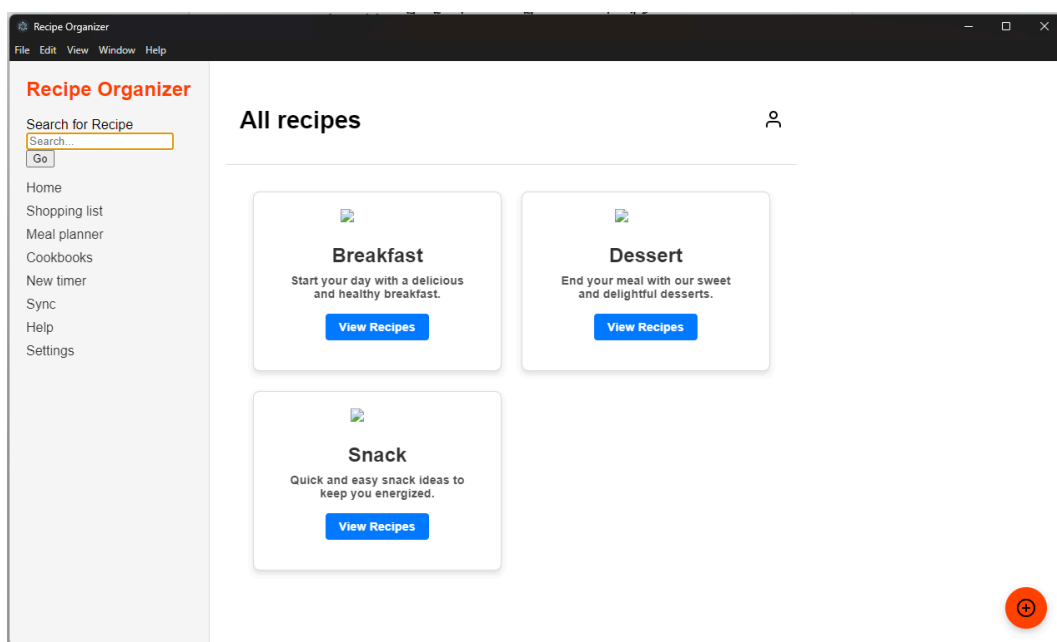
## Implementation

ทางผู้จัดทำได้ทำการสร้างแอปพลิเคชันที่มีชื่อว่า “Recipe Organizer” เพื่อจัดเก็บสูตรอาหารต่างๆ โดยให้ผู้ใช้งานทั่วไปสามารถมาแชร์สูตรอาหารร่วมกันได้ ผ่านการโพสต์ในรูปแบบของ Forum ที่สามารถโพสต์และสามารถเลือกดูสูตรอาหารต่างๆภายในแอปที่มีผู้ใช้หลากหลายท่านเข้ามาแชร์สูตรการทำอาหารภายในแอปพลิเคชันได้ และสามารถคอมเมนต์สูตรอาหารและสามารถแชร์ผ่านโซเชียลมีเดียภายนอก อย่างเช่น Facebook, X และอื่นๆ

ทางผู้จัดทำได้เลือกเทคโนโลยีอย่าง “Electron” มาใช้ เป็นเทคโนโลยีในการเขียน Desktop-Application ในสมัยใหม่ เนื่องจากมีความยืดหยุ่นที่สูงมาก และมีลูกเล่นที่หลากหลายด้วยการใช้ Javascript และภาษาอื่นๆที่ใช้ในการเขียนเว็บไซต์ เช่น HTML, CSS, Tailwind และ “Electron” เป็น Build cross-platform desktop apps ซึ่งสามารถเข้ากันได้ดีผ่านระบบปฏิบัติการระหว่าง Windows และ MacOS

แอปพลิเคชันนี้ Front-End เราเลือกใช้ HTML, CSS และ Back-End เราใช้เป็น Javascript และ Node.js

## Screenshot Application



รูปภาพที่ 1 “All Recipe Menu page”

Profile - Login

File Edit View Window Help

### Login

Username

นภคมล

Password

Login

รูปภาพที่ 2 “User Profile page”

New Recipe

File Edit View Window Help

### Add New Recipe

Upload Image

Choose File No file chosen

Recipe Name

Enter recipe name

Ingredients

Enter ingredients

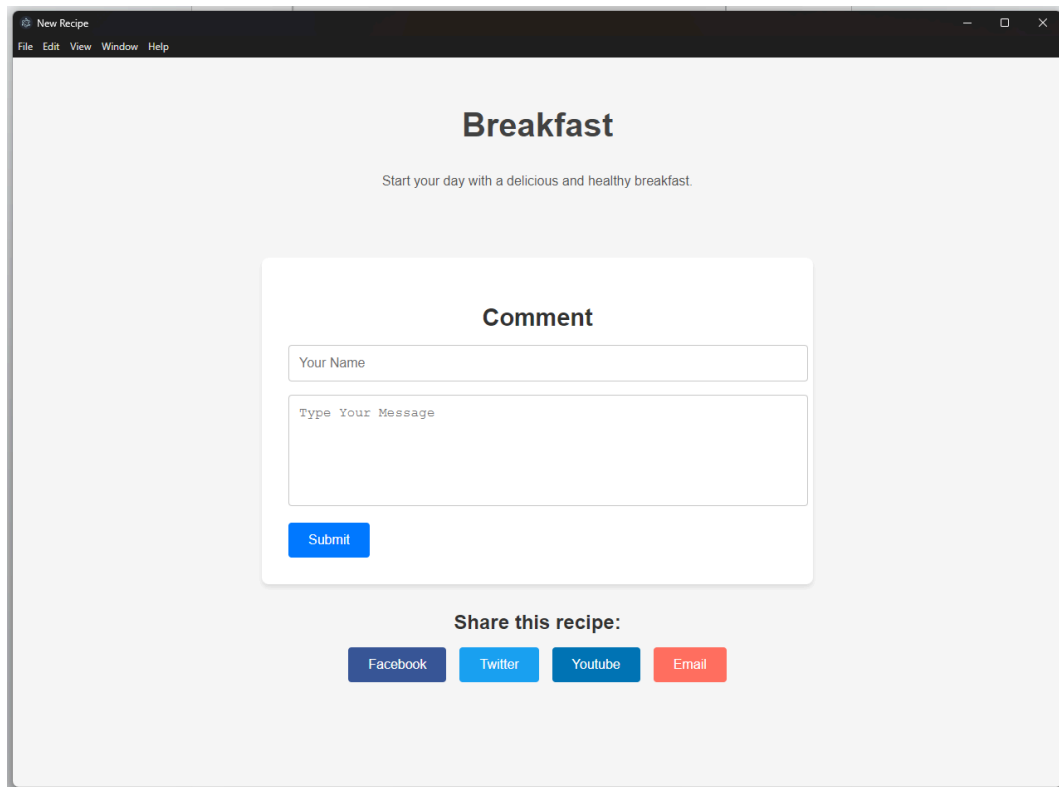
Instructions

Enter instructions

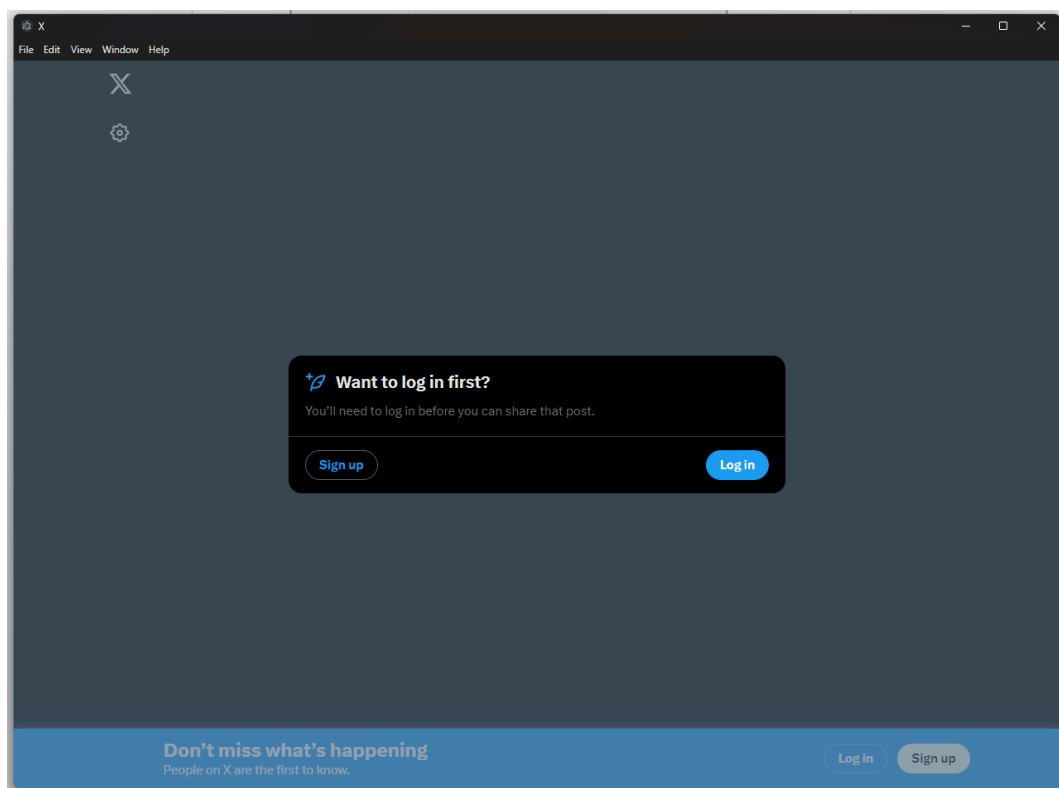
Category

Submit

รูปภาพที่ 3 “Add New Recipe page”



รูปภาพที่ 4 “Describe Menu page”



รูปภาพที่ 5 “Share to Social Media page”

## การทำงานของ Application

### รูปภาพที่ 1

“All Recipe Menu” แสดงผลของเมนูทั้งหมด และ UI ต่างๆภายในแอปพลิเคชัน  
เปรียบเสมือน index ของแอปพลิเคชัน

### รูปภาพที่ 2

“User Profile” หน้าเพจสำหรับล็อกอินเข้าสู่แอปพลิเคชัน เมื่อผู้ใช้ทำการล็อกอินสำเร็จ  
ระบบจะพาผู้ใช้กลับไปยังหน้า “All Recipe Menu” ที่เป็นหน้าแรก

### รูปภาพที่ 3

“Add New Recipe” เป็นเมนูสำหรับการเพิ่มเมนูอาหาร พร้อมรายละเอียดต่างๆ เช่น  
ชื่อเมนู วัตถุดิบ วิธีการทำอาหาร และประเภทของอาหาร

### รูปภาพที่ 4

“Describe Menu” จะปรากฏชื่อเมนู และรายละเอียดของเมนูอาหาร ถัดมาทางด้าน  
ล่าง ผู้ใช้สามารถที่จะแสดงความคิดเห็น และสามารถที่จะแชร์ไปยังโซเชียลมีเดียต่างๆ เช่น  
Facebook, X, Youtube และ Email

### รูปภาพที่ 5

“Share to Social Media” เป็นตัวอย่างการแชร์ไปยังแอปพลิเคชัน “X” หรือ  
“Twitter” แต่ในที่นี้เราจำเป็นต้องล็อกอินผ่านทางแอปพลิเคชันของ “X” ก่อน จึงจะสามารถโพ  
สต์ได้

## บทสรุป

ในการทำแอป Recipe Organizer นี้ ทำให้ได้เรียนรู้มากมายเกี่ยวกับการวิเคราะห์และออกแบบระบบในแบบเชิงวัตถุ (OOAD) ซึ่งทำให้เห็นว่าแค่การคิดแบบเป็นขั้นเป็นตอนและการออกแบบที่ดี มีส่วนสำคัญมากในการทำให้ระบบทำงานได้ตรงกับที่ตั้งใจไว้ การสร้างไดอะแกรมต่างๆ เช่น Use-Case Diagram หรือ Activity Diagram ช่วยให้เห็นภาพรวมได้ง่ายขึ้น เห็นการทำงานของแต่ละส่วนชัดเจนขึ้น แล้วยังช่วยให้เข้าใจว่าอะไรเป็นฟังก์ชันหลักของแอปและต้องใส่ใจรายละเอียดในจุดไหนบ้าง

อุปสรรคหลักๆ คือการกำหนดความสัมพันธ์ระหว่างคลาสต่างๆ ให้ชัดเจน เช่น ส่วนที่สูตรอาหารต้องเชื่อมโยงกับส่วนผสมหลายๆ อย่าง ต้องวางแผนให้ดีไม่งั้นจะซับซ้อนเกินไป ยังมีปัญหาเกี่ยวกับการสร้าง Sequence Diagram และ Communication Diagram ที่ต้องทำให้การสื่อสารระหว่างคลาสตรงกับการทำงานจริง ทำให้ต้องคิดรอบคอบมากๆ ในขั้นตอนการออกแบบ และในขั้นตอนการทำงานจริงทำให้เกิดความสับสนอย่างมาก เพราะต้องคิดให้เป็นกระบวนการ งานนี้ก็ถือว่าเป็นงานที่สเกลใหญ่ และด้วยเวลาที่จำกัดอีกด้วย

สรุปแล้ว โครงการนี้ช่วยพัฒนาทักษะการวิเคราะห์และออกแบบระบบได้เยอะมาก เรียนรู้วิธีจัดการปัญหาที่เจอไปทีละขั้น และเข้าใจหลักการทำงานของ OOAD ว่ามีส่วนช่วยให้แอปพลิเคชันทำงานได้ดีขึ้นยังไงแน่นอนว่าเป็นประสบการณ์ที่สำคัญในการทำงานพัฒนาซอฟต์แวร์ในอนาคต