

**รายงาน**

**เรื่อง ระบบร้านขายของ**

**จัดทำโดย กลุ่ม 11**

นาย จิตรเทพ คชพงษ์ 6110450073 sec 200

นาย อเมริกา ญาณาคม 6110451037 sec 200

นาย พงศกร ดีประมาณ 6110450227 sec 200

นาย จักรภัทร ผลยงค์ 6110450669 sec 200

นาย ทยุต สุวรรณบัตร 5910450280 sec 200

**เสนอ**

อาจารย์สมโชค เรืองอิทธินันท์

**รายงานนี้เป็นส่วนหนึ่งของวิชา 01418321 Systems Analysis and Design**

**ภาคปลายปีการศึกษา 2562**

**มหาวิทยาลัยเกษตรศาสตร์**

**วิทยาเขตบางเขน**

สมาชิกกลุ่มที่ 11

1.นาย พงศกร ดีประมาณ 6110450227

หน้าที่รับผิดชอบ:

-ทำ application

-ปรับปรุงรายงาน

-ตรวจสอบความถูกต้องข้อมูล

2.นาย จิตรเทพ คชพงษ์ 6110450073

หน้าที่รับผิดชอบ:

-เขียน ER diagram

-เขียน CRUD diagram

-เขียน business flow

-เขียน use case description

-เขียนรายงาน

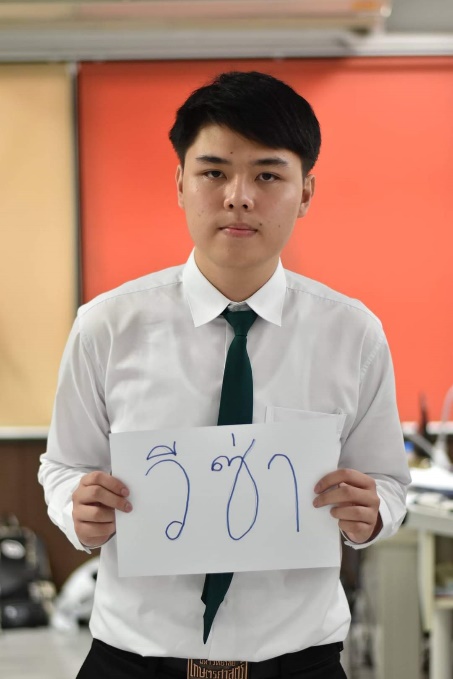
3. นาย จักรภัทร ผลยงค์ 6110450669

หน้าที่รับผิดชอบ:

-ช่วยทำ application

-เขียน schema structure

-เขียน use case diagram



4.นาย อเมริกา ญาณาคม 6110451037

หน้าที่รับผิดชอบ

-ออกแบบ data base

-ช่วยทำ application

5.นาย ทยุต สุวรรณบัตร 5910450280

หน้าที่รับผิดชอบ

 - สร้าง sequence diagram

- สร้าง collaboration diagram

- ทำ power point

หน้าให้คะแนน

นาย พงศกร ดีประมาณ 6110450227

หัวหน้ากลุ่ม

* เป็นคนทำงาน,มอบหมายงาน, คอยช่วยเหลือเพื่อนให้เดินงานต่อ,คอยติดต่อระหว่างคนเขียน code กับ คนทำรายงาน

นาย อเมริกา ญาณาคม 6110451037

ให้คะแนน 8/8

* เป็นผู้นำคนเขียน application , มีความอดทน ขยันทำงาน, ติดตามความคืบหน้าของงาน

นายจิตรเทพ คชพงษ์ 6110450073

ให้คะแนน 7/8

* เป็นคนทำรายงาน คอยเตือนคนในกลุ่มเรื่องการทำงาน และออกแบบ diagram

นาย จักรภัทร ผลยงค์ 6110450669

ให้คะแนน 7/8

* ตอนแรกจะให้คะแนนมากกว่านี้เพราะว่า ขยันทำงาน คอยช่วยเหลือเวลาทำ application แต่ชอบสร้าง ตัวแปร ที่ทำให้เข้าใจยากจนต้องให้ เพื่อนมาเปลี่ยนใหม่

นาย ทยุต สุวรรณบัตร 5910450280

ให้คะเนน 7/8

* เป็นคนทำงานเก่ง คอยเตือนจุดที่ต้องแก้ไขในรายงานและตัว diagram ต่างๆ

บทที่ 1 บทนำ

1.1 บทนำ

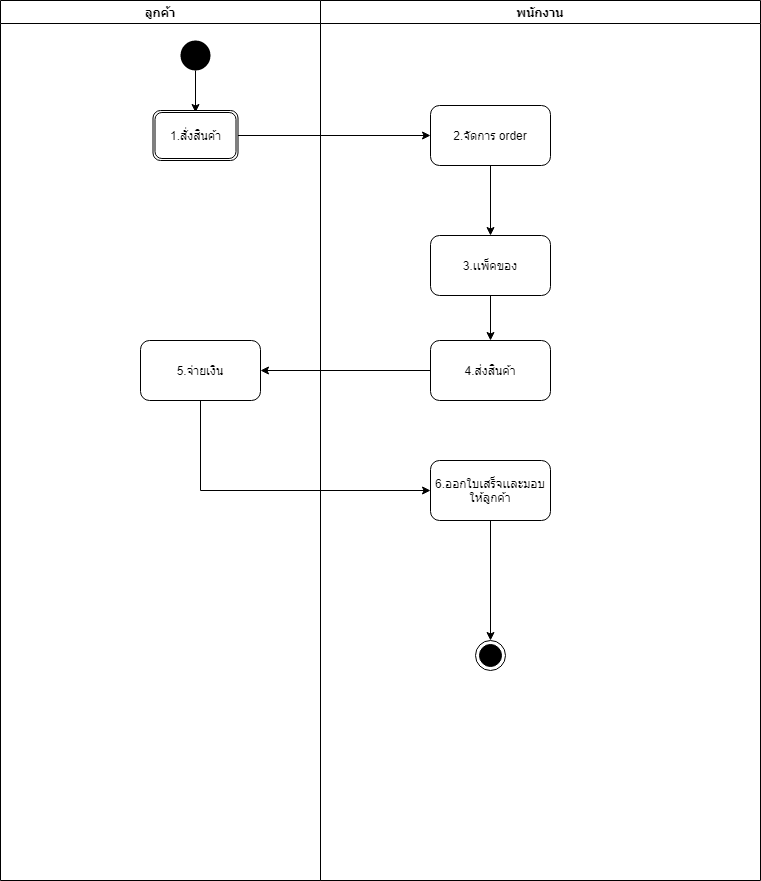
ในปัจจุบัน ร้านค้าขายของทางบ้าน จิตรเทพ คชพงษ์ ยังไม่มีการจัดการด้านการเก็บข้อมูลอย่างเป็นระเบียบทำให้ทางร้านค้านั้น เช็คใบเสร็จ และ ใบสั่งยาก นอกจากนี้เวลาลูกค้าสั่งของ ต้องสั่งผ่านการโทรศัพท์อย่างเดียวด้วย

ดังนั้นเพื่อให้เกิดความสะดวกในการเช็คใบสั่ง กับ ใบเสร็จ จึงมีการสร้างโปรแกรมขึ้นมาเพื่อให้ลูกค้าสั่งผ่าน app ได้เลยและสามารถเช็คใบเสร็จ กับ ใบสั่งได้อย่างสะดวก

1.2 วัตถุประสงค์

เพื่อสร้างโปรแกรมค้าขายที่อำนวยความสะดวกให้แก่ผู้ใช้งาน โดยโปรแกรมนี้นำโครงสร้างการทำงานมาจาก บริษัทสมัชล์แสน ซึ่งบริษัทนี้ไม่มีโปรแกรมหรือsoftware เป็นของตัวเอง เราจึงสร้างโปรแกรมนี้ขึ้นมาเพื่อทำให้เกิดประโชน์แก่บริษัท

1.3 Business flow เก่า

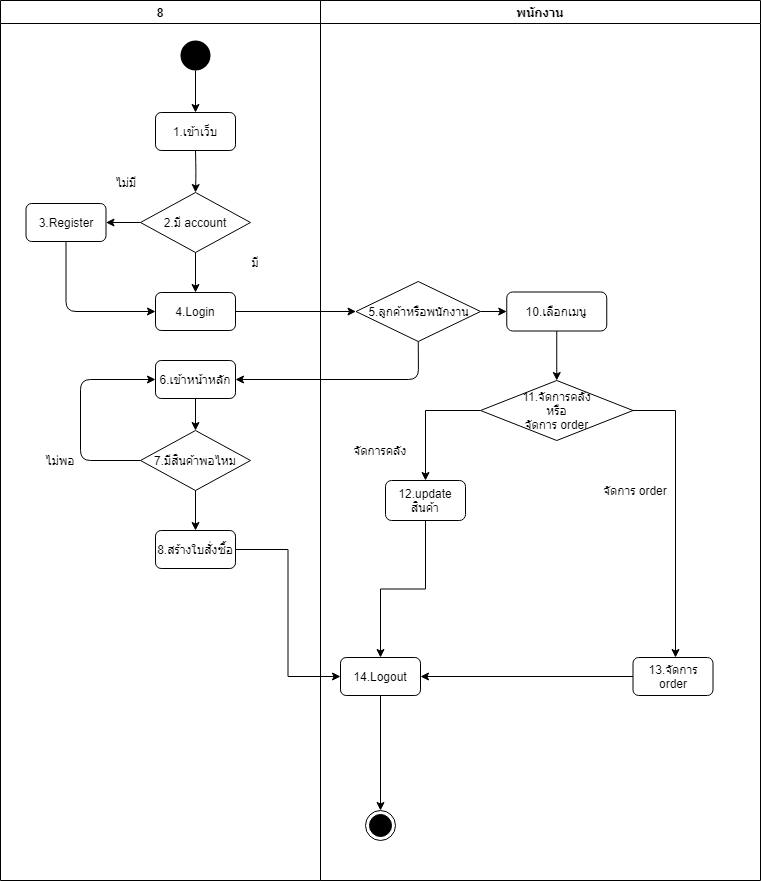


1.4 ปัญหาในระบบปัจจุบัน

1. ลูกค้าสั่งของได้เฉพาะผ่านการโทรศัพท์เท่านั้น

2. ไม่สามารถเช็คว่าลูกค้าคนไหนเป็นคนสั่ง order

**1.5** Business flow ใหม่



**การแก้ไขปัญหาของระบบใหม่**

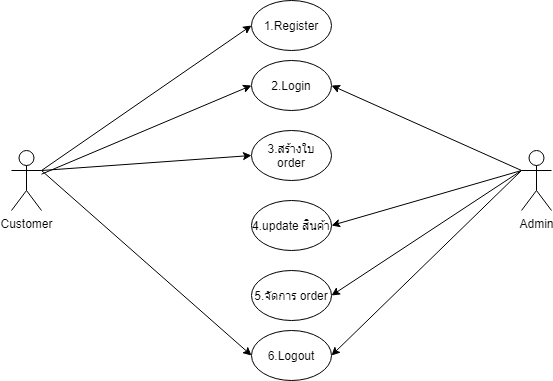
- จาก ปัญหาข้อ 1. แก้โดยสร้างระบบเพื่อเพิ่มช่องทางการสั่งของของลูกค้าเพิ่มขึ้น

- จาก ปัญหาข้อ 2. แก้โดยสร้างฐานข้อมูลสำหรับการเก็บข้อมูลของ order เพื่อทำให้สามารถตรวจสอบในภายหลังได้

1.6 แปลง Business flow เป็น Use case

|  |  |
| --- | --- |
| Usecase ID | Process ID |
| 1.Register | 1,2,3 |
| 2.Login | 4,5 |
| 3.สร้าง order | 6,7,8 |
| 4.update สินค้า | 9,10,11 |
| 5.จัดการ order | 12 |
| 6.Logout | 13 |

1.7 Use Case Diagram



1.8 state diagram



1.9 Crud Table ใหม่

|  |  |  |  |
| --- | --- | --- | --- |
|  | Customer | Product | Order |
| 1.Register | C,R | - | - |
| 2.Login | R | - | - |
| 3.สร้าง order | R | R,U | C |
| 4.update สินค้า | - | C,U | - |
| 5.จัดการ order | R | R | R,U |
| 6.Logout | R | - | - |

1.9 Use Case Description

บอกรายละเอียดการทำงานแต่ละขั้นตอนการทำงานของโปรแกรม โดยจะมีรายละเอียดต่างกันตาม การทำงานของแต่ละขั้นตอน

Use case 1

Use case name : Register

Actor : Customer

Description : นำรายละเอียดของลูกค้ามากรอกลงในระบบ

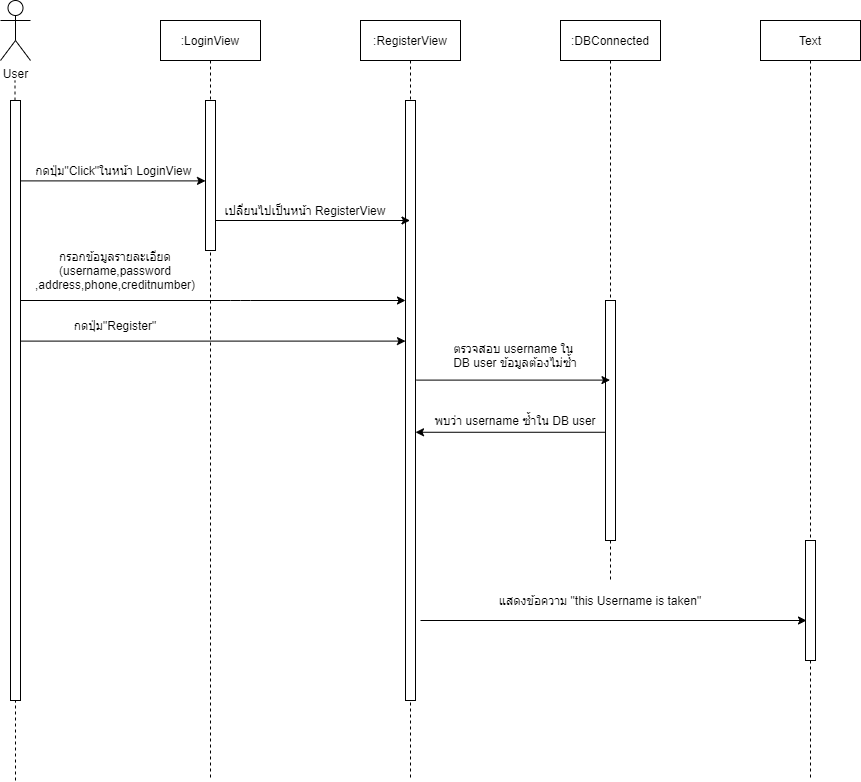
|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| Normal Flow | 1.กดปุ่ม “Click”ในหน้า LoginView  3.กรอกข้อมูลรายละเอียด (username,password,address,phone,creditnumber)  4.กดปุ่ม “Register” | 2.เปลี่ยนไปหน้า RegisterView  5.ตรวจสอบ username ใน DB user ข้อมูลต้อง  ไม่ซ้ำ  6.บันทึกข้อมูลลง DB User ด้วยคำสั่ง INSERT INTO user (username,password,address,phone,credit card number)  7.แสดงหน้าต่าง LoginView |
| Alternative flow |  | A5A ตรวจสอบพบเจอข้อมูลซ้ำใน DB User  แล้ว แสดงข้อความ “this Username is taken”  A5A ตรวจสอบ User กรอกข้อมูลไม่ครบ แล้วแสดงข้อความ “Please fill everything” |

Sequence Diagram : สมัครสมาชิก

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

กรณีมีสมาชิกอยู่แล้ว



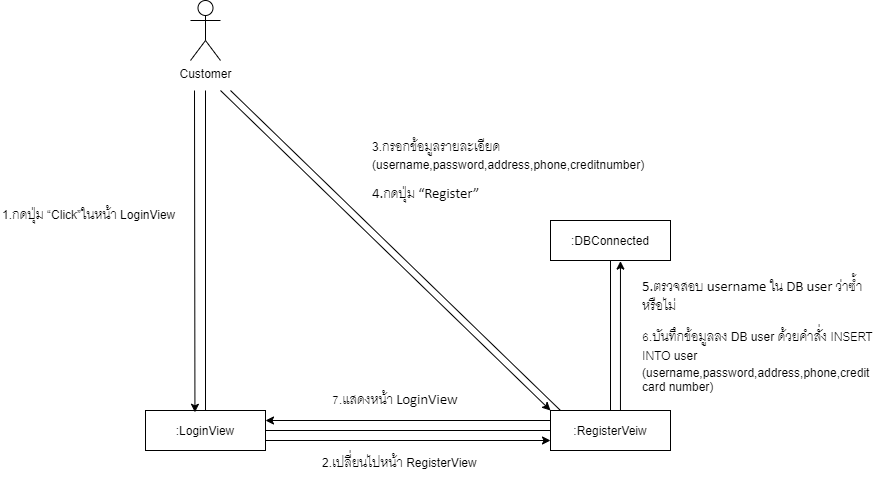
กรณีใส่ข้อมูลไม่ครบ

รูปภาพประกอบด้วย ข้อความ

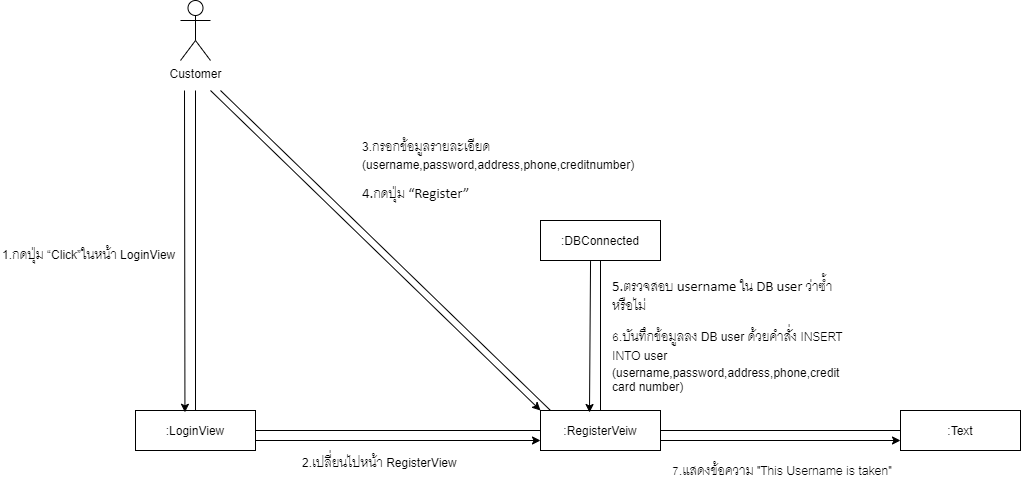
คำอธิบายที่สร้างโดยอัตโนมัติ

Collaboration Diagram:

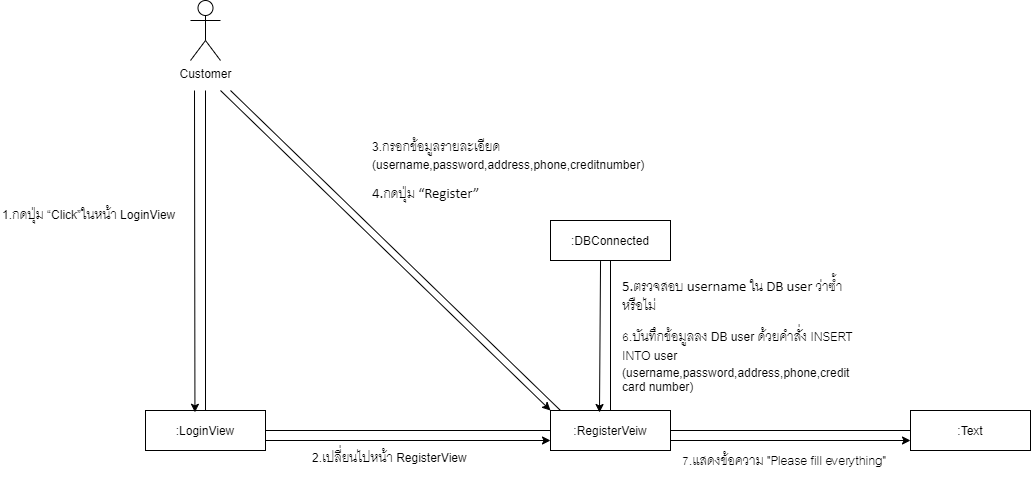
ลงทะเบียนผู้ใช้งาน



กรณีมีข้อมูลในระบบแล้ว



กรณีใส่ข้อมูลไม่ครบ



Use case 2

Use case name : Login

Actor : User

Description : เข้าสู่ระบบ

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| Normal Flow | 1.กรอก Username และ Password  2.กดปุ่ม “Login” | 3.ตรวจสอบ Username,Password  ต้องเป็น not null  4.ตรวจสอบ Username และ Password ใน DB User  5.ดึงข้อมูลด้วยคำสั่ง SELECT\*FROM user Where username = username and password = password  6.ไปหน้า OrderView |
| Alternative flow |  | A3A พบข้อมูลเป็น null แสดงข้อความ “Missing Username or Password”  A5A ข้อมูลไม่ตรงกับฐานข้อมูล แล้วแสดงข้อความ “Wrong Username or Password”  A6A เมื่อใส่ username กับ password ของ Admin จะไปหน้า AdminView |

Sequence Diagram : เข้าสู่ระบบ

กรณีในฐานะ Customer

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

กรณีเข้าสู่ระบบในฐานะ Admin

รูปภาพประกอบด้วย ข้อความ

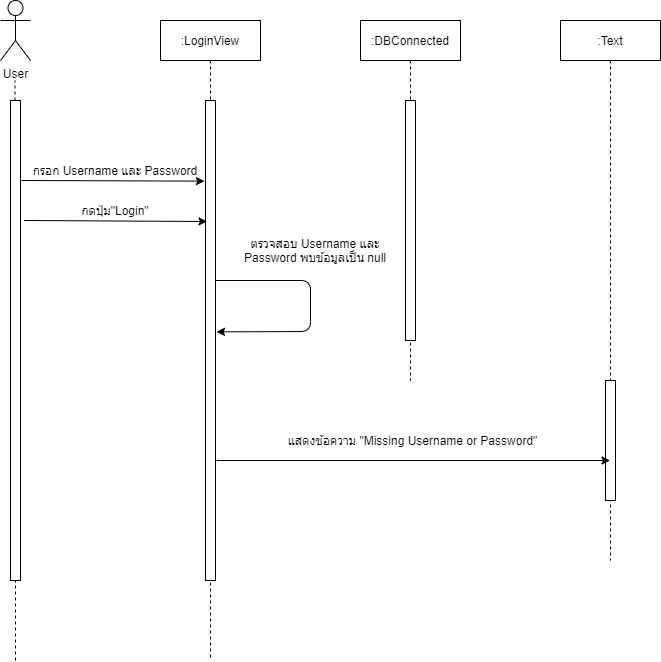
คำอธิบายที่สร้างโดยอัตโนมัติ

กรณีใส่ข้อมูลผิด

รูปภาพประกอบด้วย ข้อความ

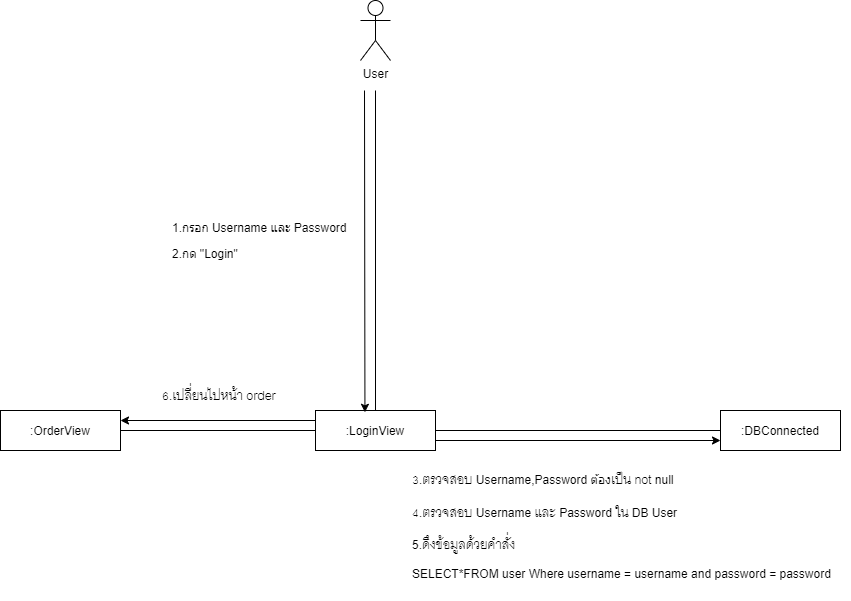
คำอธิบายที่สร้างโดยอัตโนมัติ

กรณีข้อมูลไม่ครบ

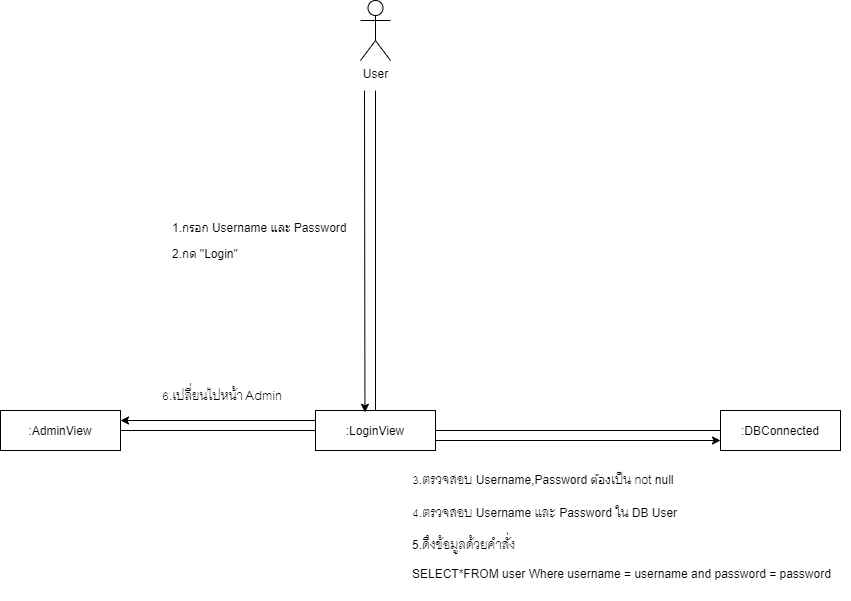


Collaboration Diagram : การเข้าสู่ระบบ

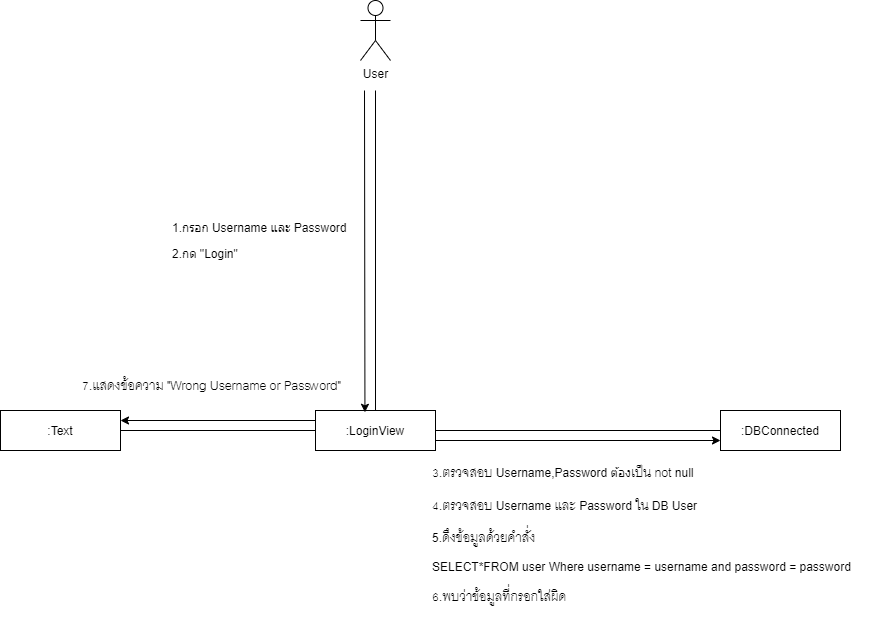
ในฐานะ Customer



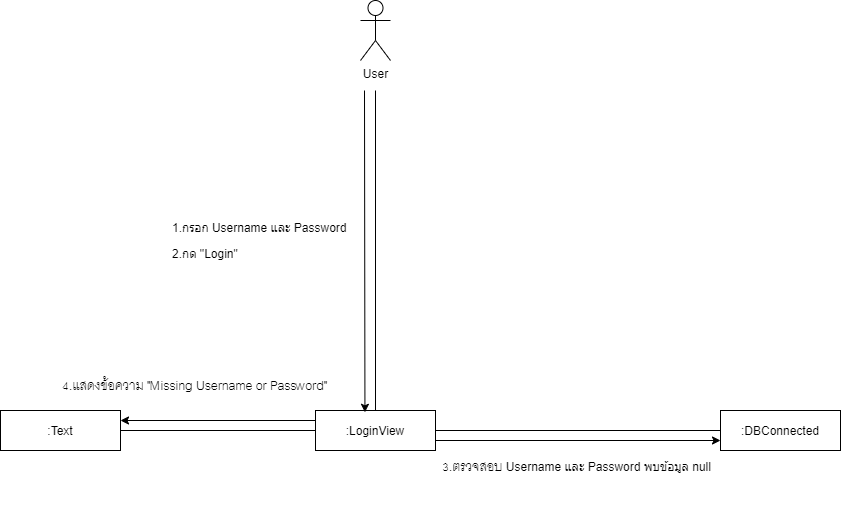
ในฐานะ Admin



กรณี ใส่ข้อมูลผิด



กรณี ใส่ข้อมูลไม่ครบ



Use case 3

Use case name : สร้าง order

Actor : Customer

Precondition : Orderview

Description : สร้างใบ order,ลบ order

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| Normal Flow | 1.เลือกสินค้าและใส่จำนวน  2.กดปุ่ม”Add”  7.กดปุ่ม “Cart”  9.กดปุ่ม”Confirm” | 3.ตรวจสอบจำนวนต้อง มากกว่า 0 หรือ ไม่เป็นทศนิยม  4.ตรวจสอบข้อมูลที่สั่งใน DB product  5.เอาข้อมูลเก็บไว้ใน text file  6.นำข้อมูลจาก text file มาใส่ tableview ใน cartview  8.แสดงหน้า cartview  10.บันทึกข้อมูลลง DB orderlist โดยคำสั่ง INSERT INTO orderlist (username,P\_name,O\_quantity,O\_price,status)  11.ลบข้อมูลออกจาก text file  12.แสดง popup ข้อความ “Confirmed”  13.กดปุ่ม “OK”  14.แสดงหน้า Orderview |
| Alternative flow | A8A กดปุ่ม”remove” แล้วสินค้าที่อยู่ใน tableview ในหน้า cartview ที่ถูกเลือกจะถูกลบออกไป | A3A เช็คข้อมูลแล้วพบว่า จำนวนนั้น น้อยกว่า 1 หรือ เป็นทศนิยม แล้วแสดง Pop-up “โปรดระบุจำนวนใหม่” กดปุ่ม “ OK “ แล้ว Pop-up จะหายไป  A4A เช็ครายการพบว่าสินค้าไม่พอ แล้วแสดง Pop-up “Not enough item” กดปุ่ม “OK”แล้ว Pop-up จะหายไป |

Sequence Diagram : สร้าง Order

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

กรณี : จำนวนสินค้าไม่พอกับจำนวนที่ลูกค้าสั่ง

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

กรณี: จำนวนสินค้าที่ต้องการ น้อยกว่า 1 หรือ เป็นทศนิยม

รูปภาพประกอบด้วย ข้อความ

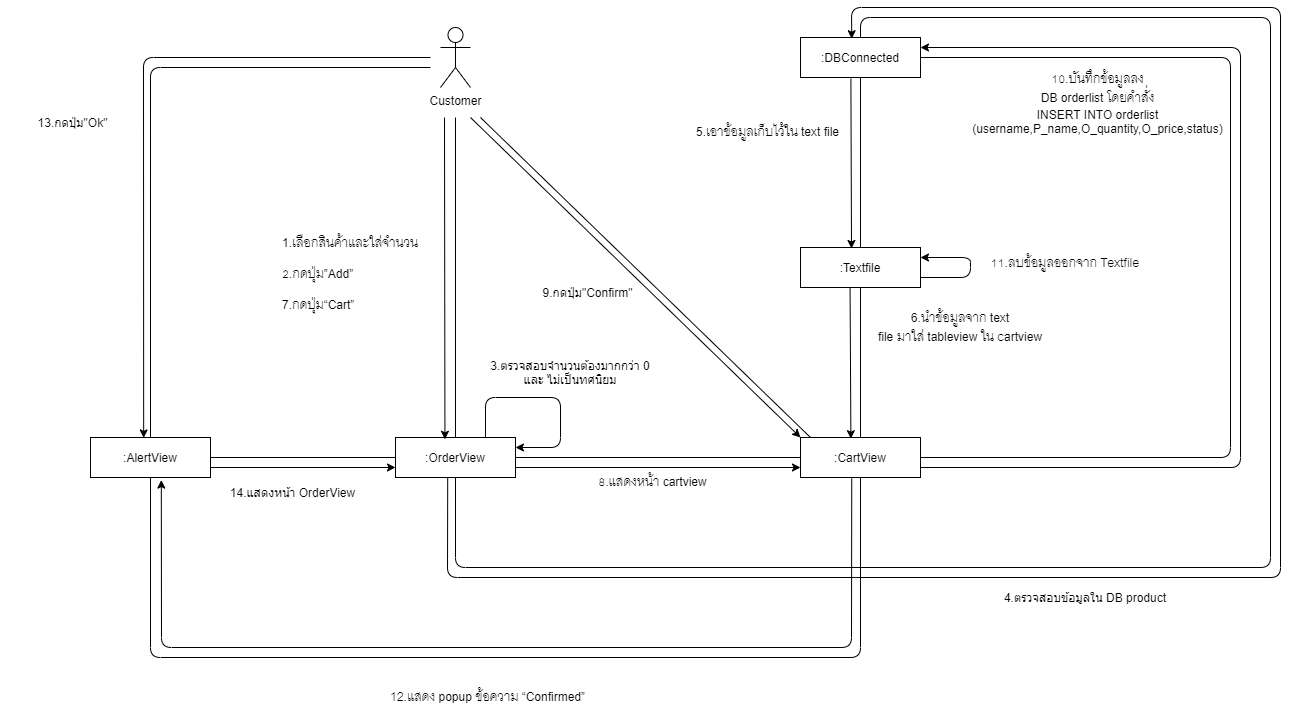
คำอธิบายที่สร้างโดยอัตโนมัติ

กรณี : ลบใบ Order

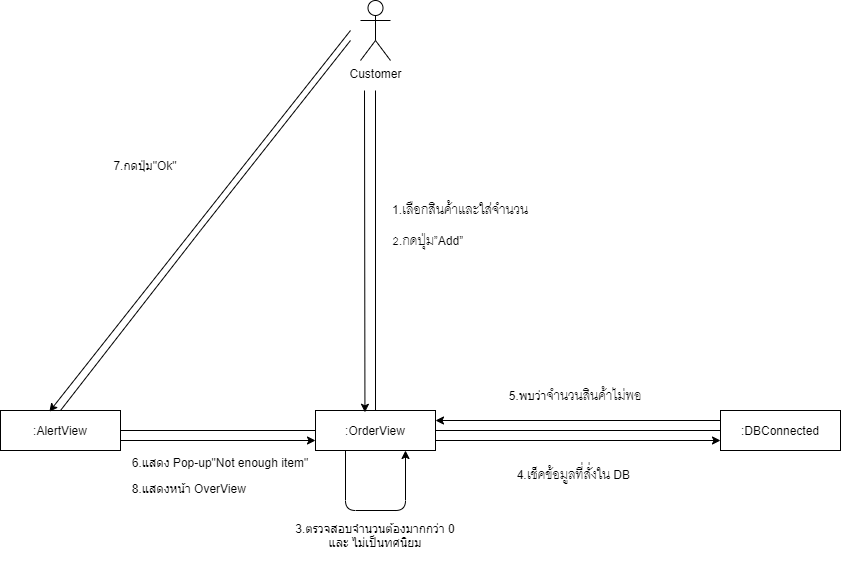
รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

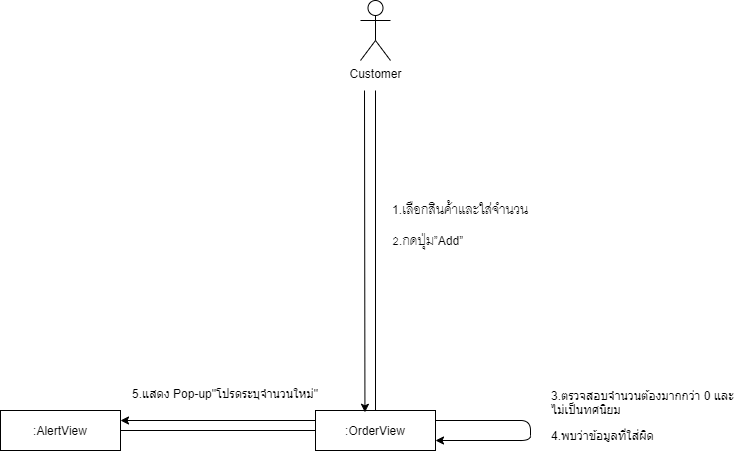
Collaboration Diagram : สร้าง order



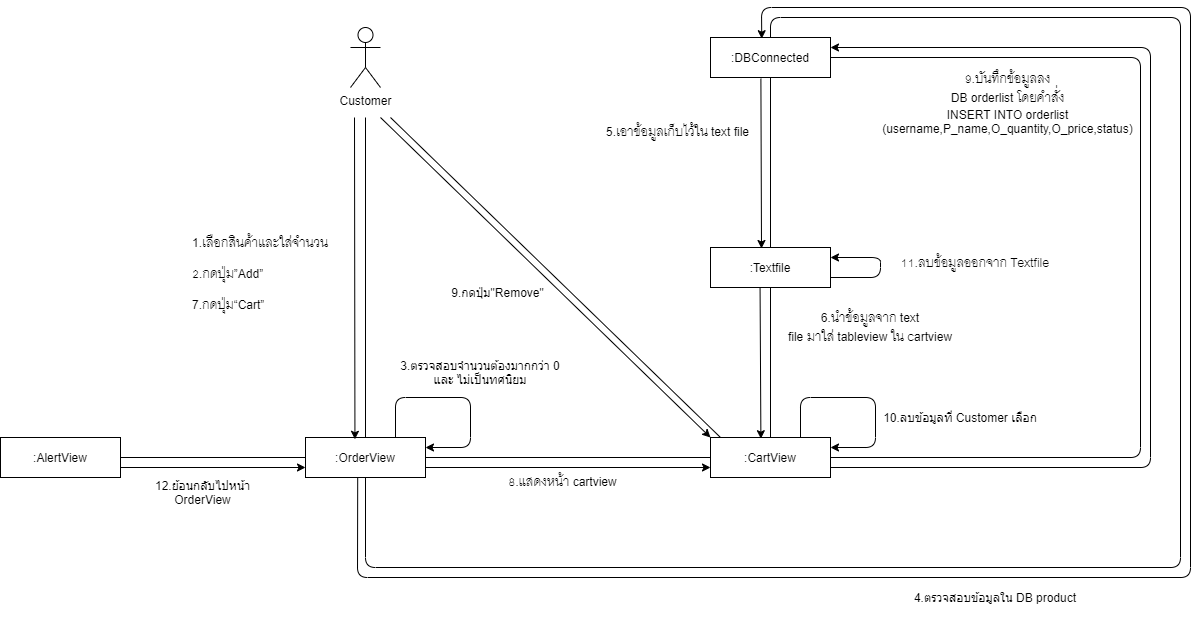
กรณี จำนวนสินค้าไม่พอกับจำนวนที่ลูกค้าสั่ง



กรณี: จำนวนสินค้าที่ต้องการ น้อยกว่า 1 หรือ เป็นทศนิยม



กรณี : ลบใบ Order



Use case 4

Use case name : update สินค้า

Actor : Admin

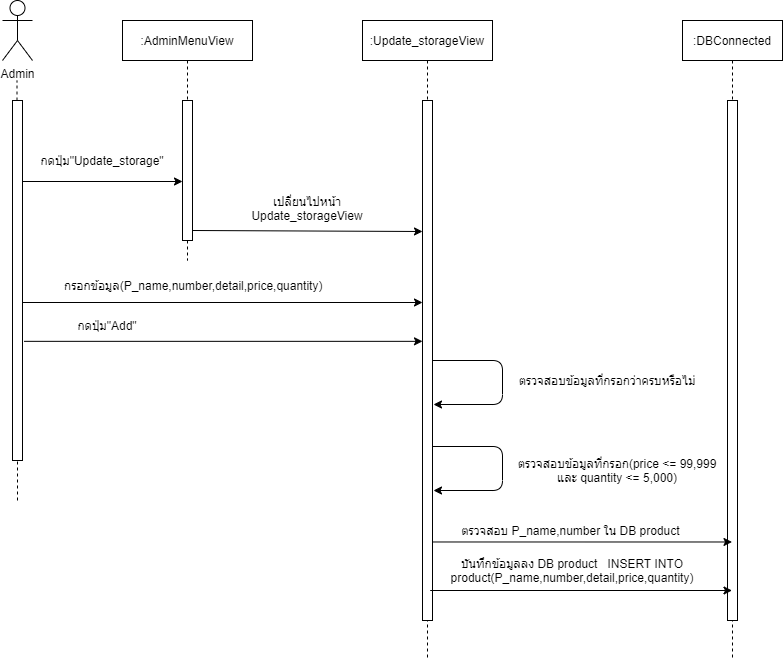
Precondition : Adminview

Description : เพิ่ม,ลด สินค้ากับจำนวนสินค้า

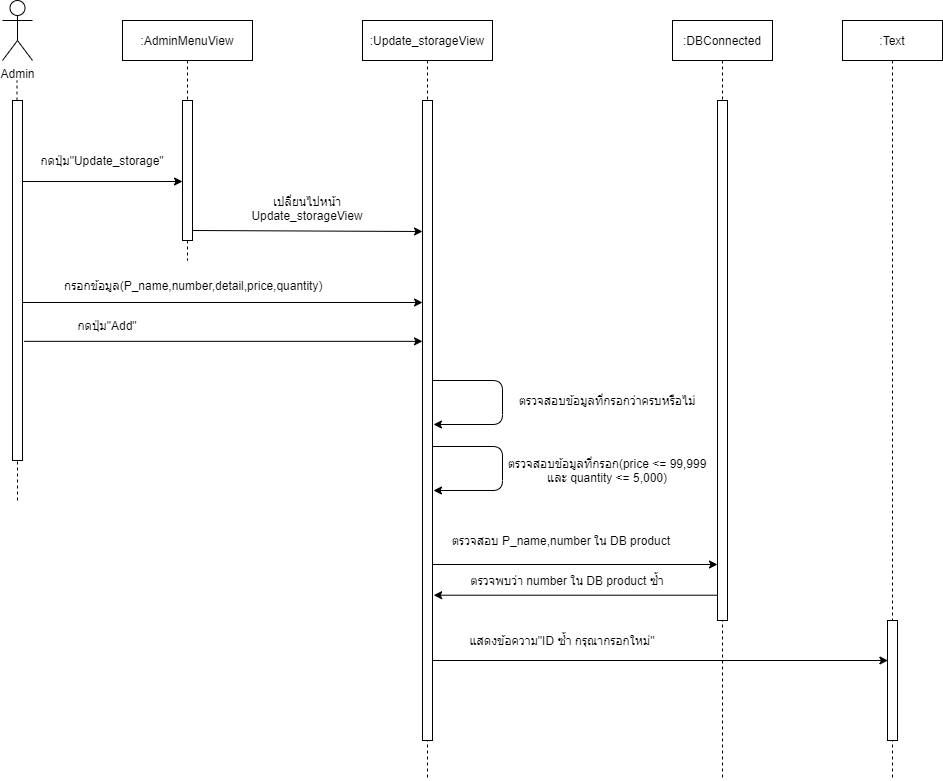
|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| Normal Flow | 1.กดปุ่ม “Update\_storage”  3.กรอกข้อมูล(P\_name,number,detail,price,quantity)  4.กดปุ่ม Add | 2.ไปหน้า Update\_storageview  5.ตรวจสอบข้อมูลที่กรอกเข้ามาว่าครบหรือไม่  6.ตรวจข้อมูลที่กรอกว่า price <=99,999 และ quantity <= 5000 หรือไม่  7.ตรวจสอบ (P\_name,number) ใน DB product  8.บันทึกข้อมูลลง DB product INSERT INTO product(P\_name,number,detail,price,quantity) |
| Alternative flow |  | A5A ข้อมูลกรอกไม่ครบ แสดงข้อความ “Please fill Everything”  A6A ข้อมูลที่กรอกเข้ามาพบว่า เกินมากกว่าที่กำหนด แล้วแสดงข้อความ “กรอกข้อมูลผิดพลาด”  A7A ข้อมูลที่กรอกเข้ามาพบว่ามีID,Name ซ้ำจะบันทึกข้อมูลลง DB product Update product(price,quantity)  A7A ข้อมูล number ซ้ำ แสดงข้อความ “ID ซ้ำกรุณากรอกใหม่” |

Sequence Diagram : Update สินค้า

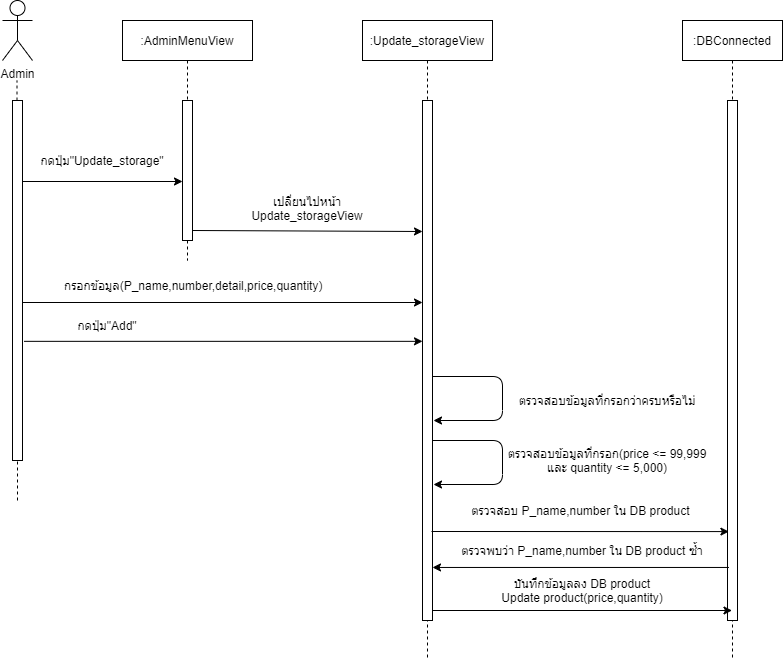
กรณี : เพิ่มสินค้า



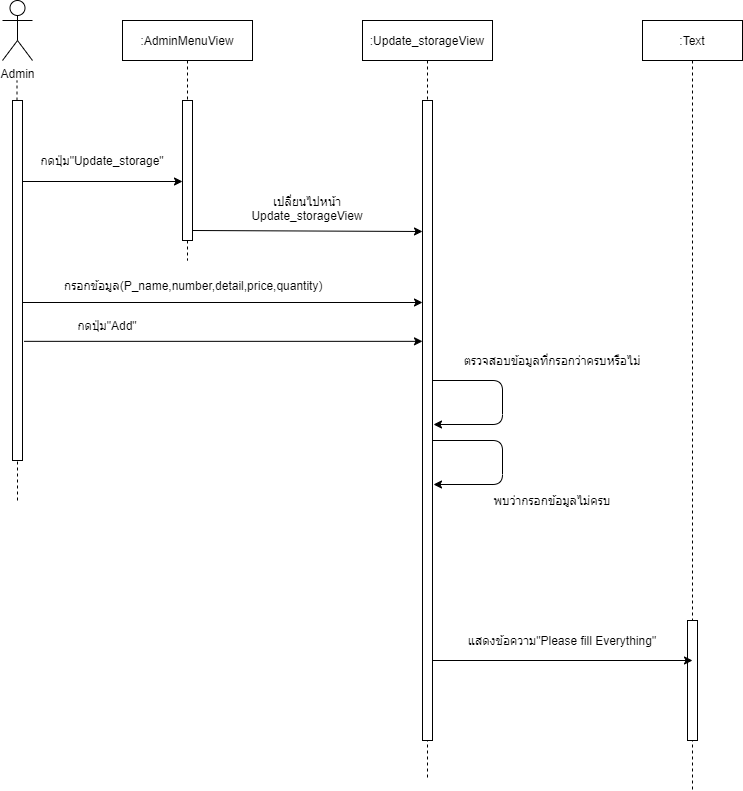
กรณี : ID ซ้ำ



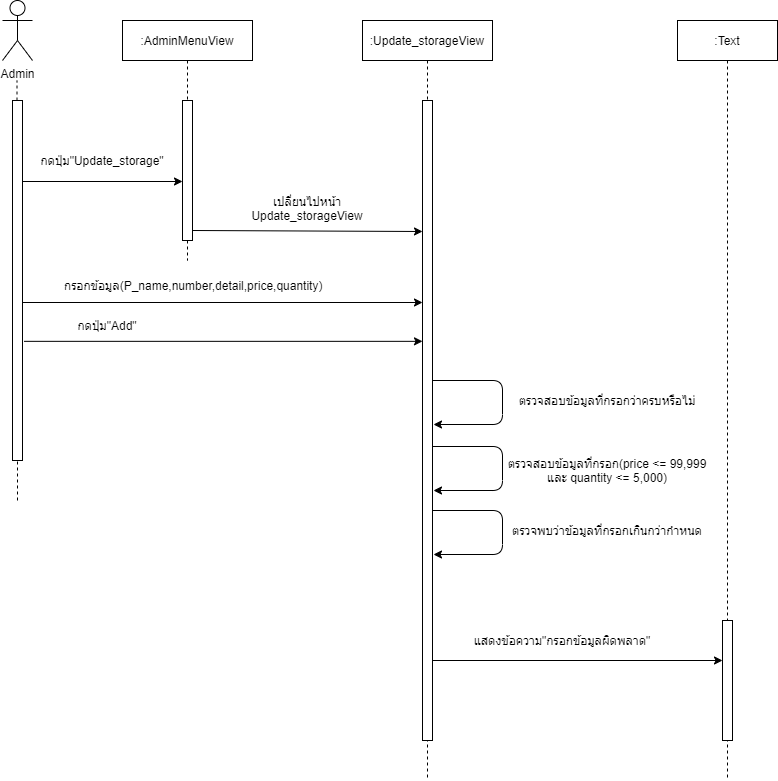
กรณี : เพิ่มจำนวนของสินค้า



กรณี : กรอกข้อมูลไม่ครบ

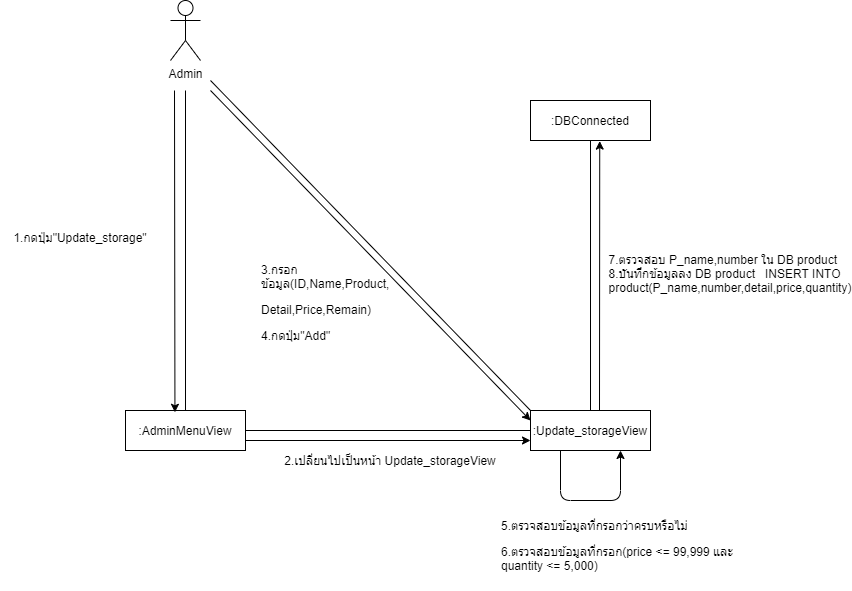


.กรณี กรอกจำนวนสินค้าไม่ถูกต้อง

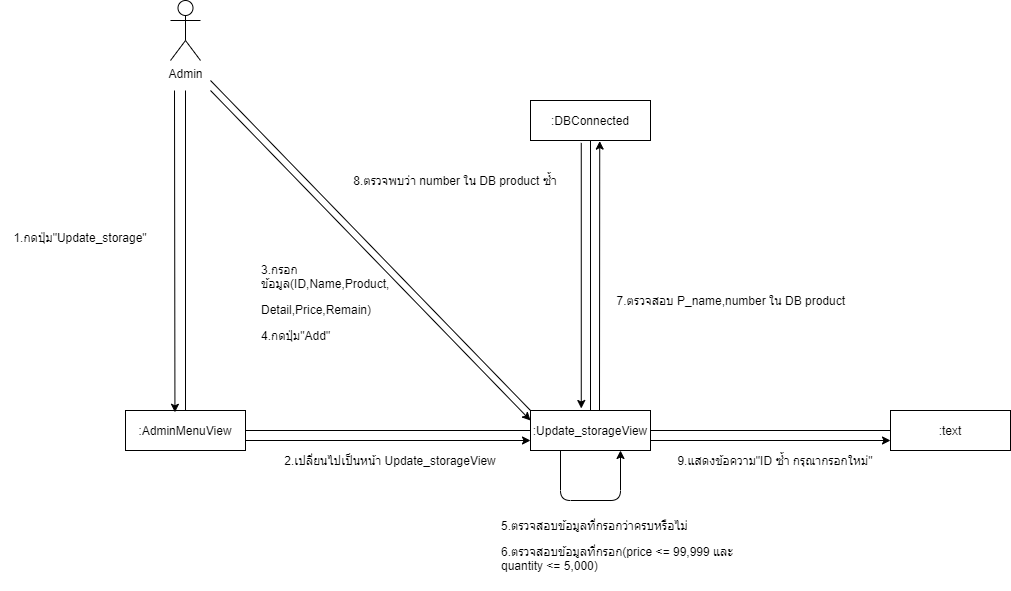


Collaboration Diagram : Update สินค้า

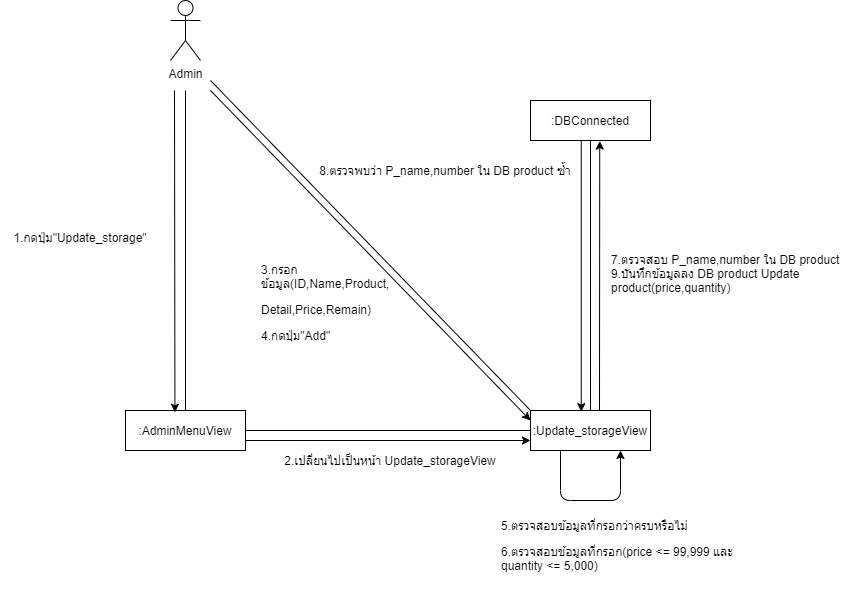
กรณี : เพิ่มสินค้า



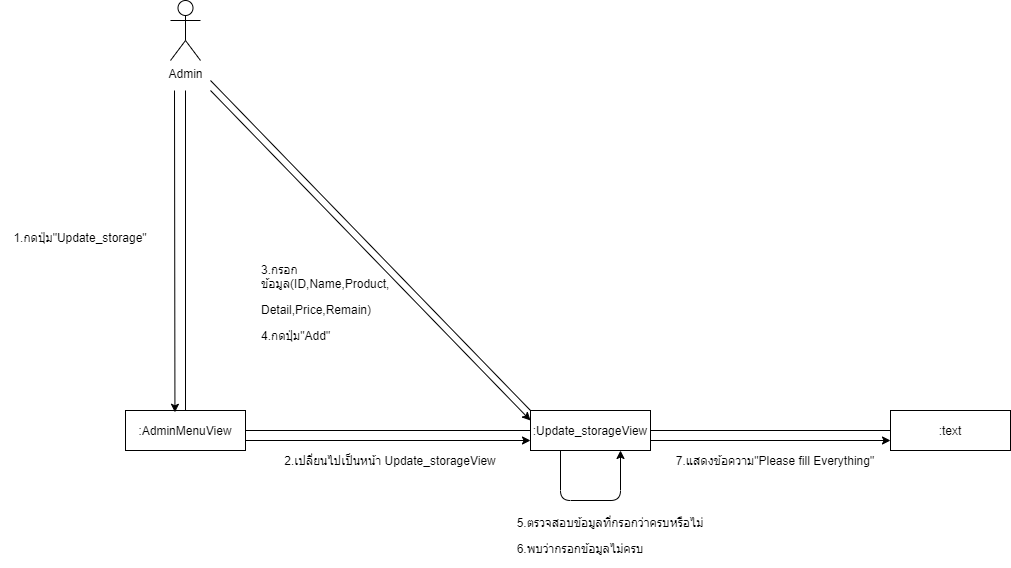
กรณี : ID ซ้ำ



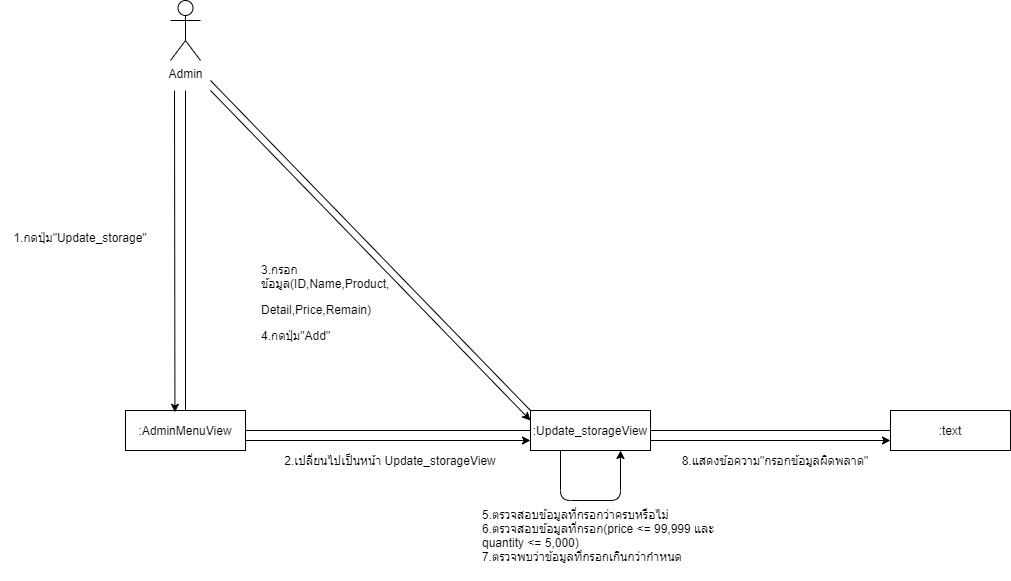
กรณี : เพิ่มจำนวนสินค้า



กรณี:กรอกข้อมูลไม่ครบ



กรณี กรอกจำนวนสินค้าไม่ถูกต้อง



Use case 5

Use case name : จัดการ order

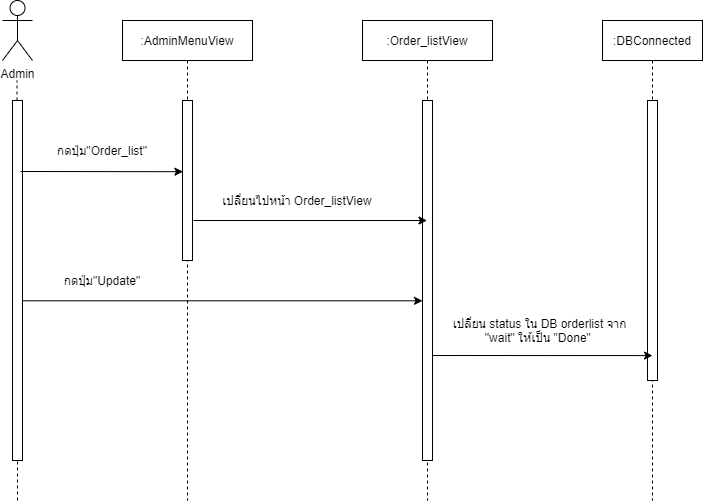
Actor : Admin

Precondition : Adminview

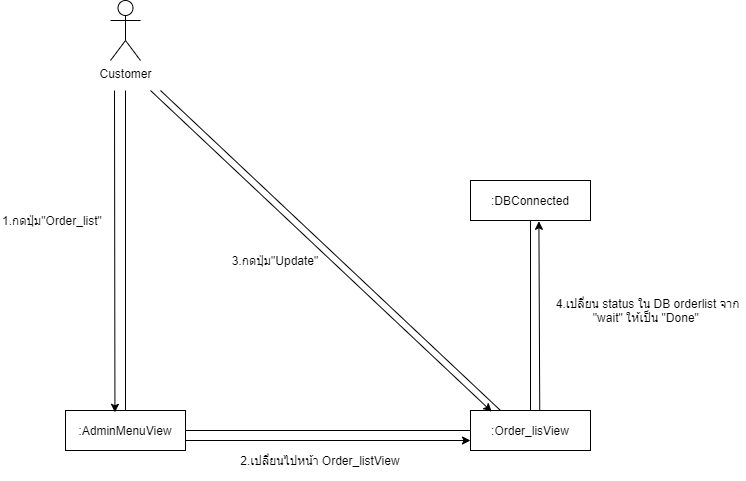
Description : จัดการสถานะ order

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| Normal Flow | 1.กดปุ่ม “Order\_list”  3.กดปุ่ม “Update” | 2.ไปหน้า Order\_listView  4.เปลี่ยน status ใน DB orderlist จาก “wait” เป็น “Done” |
| Alternative flow |  |  |

Sequence Diagram : จัดการ order



Collaboration Diagram : จัดการ order



Use case 6

Use case name : Logout

Actor : User

Precondition : หน้า orderview หรือ adminview ขึ้นอยู่กับ user ที่ใช้

Description : ออกจากระบบ

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| Normal Flow | 1.กดปุ่ม “Logout” | 2.แสดงหน้า Loginview |
| Alternative flow |  |  |

Sequence Diagram : ออกจากระบบ

กรณี ออกจากระบบในฐานะ Customer



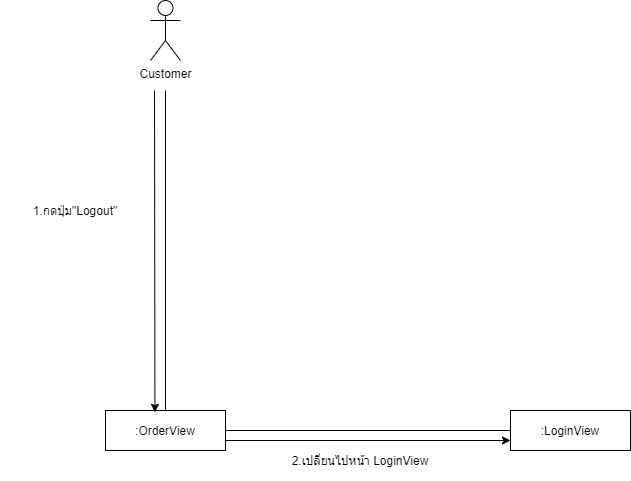
กรณี ออกจากระบบในฐานะ Admin

รูปภาพประกอบด้วย ข้อความ

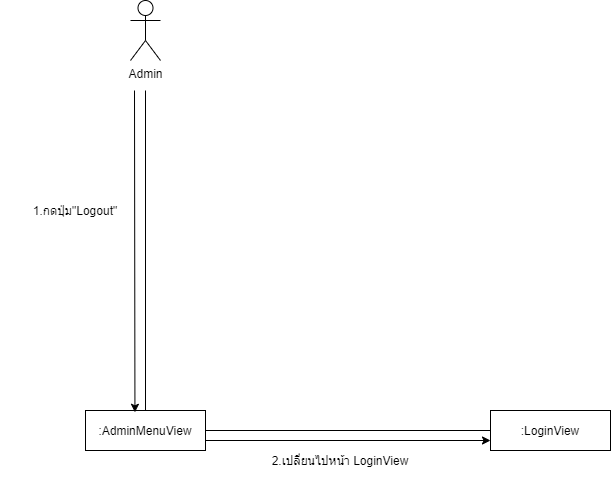
คำอธิบายที่สร้างโดยอัตโนมัติ

Collaboration Diagram : ออกจากระบบ

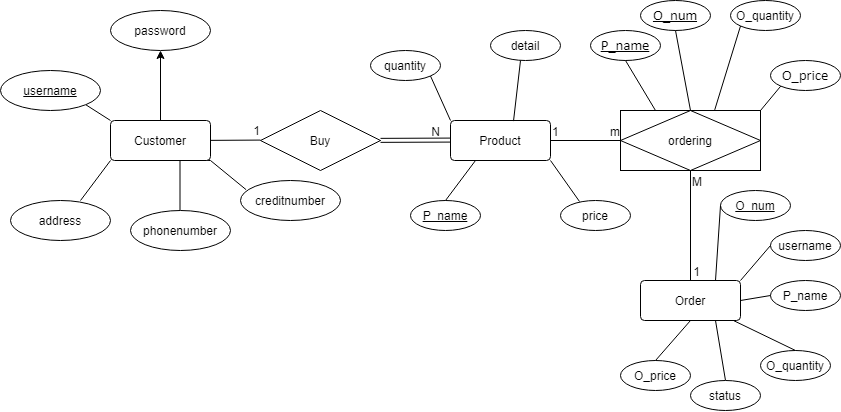
กรณี ออกจากระบบในฐานะ Customer



กรณี ออกจากระบบในฐานะ Admin



2.Data Design

2.1 ER Design

2.2.1 Schema Structure

ตาราง Customer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field | Description | Type | Size | Default | PK | FK |
| username | ชื่อลูกค้า | TEXT | 20 | NONE | Y | - |
| password | รหัสลูกค้า | TEXT | 10 | NONE | - | - |
| address | ที่อยู่ | TEXT | 30 | NONE | - | - |
| phonenumber | เบอร์โทร | INT | 10 | NONE | - | - |
| creditnumber | รหัสบัตรเครดิต | INT | 16 | NONE | - | - |

ตาราง Product

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field | Description | Type | Size | Default | PK | FK |
| number | รหัสสินค้า | INT | 3 | NONE | Y | - |
| P\_name | ชื่อสินค้า | TEXT | 30 | NONE | - | - |
| detail | รายละเอียด | TEXT | 50 | NONE | - | - |
| price | ราคาต่อหน่วย | INT | 5 | NULL | - | - |
| quantity | จำนวนสินค้า | INT | 5 | NONE | - | - |

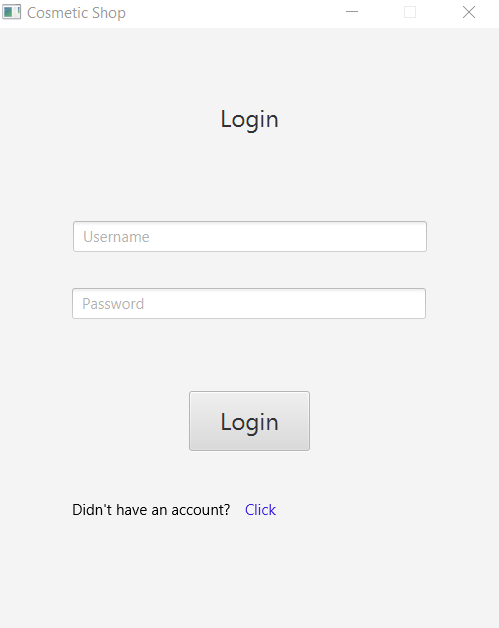
ตาราง Ordering

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field | Description | Type | Size | Default | PK | FK | CK |
| P\_name | ชื่อสิ้นค้า | TEXT | 30 |  | - | - | Y |
| O\_num | รหัสใบสั่ง | INT | 5 |  | - | - | Y |
| O\_quantity | จำนวนที่สั่ง | INT | 5 |  | - | - | - |
| O\_price | ราคารวม | INT | 10 |  | - | - | - |

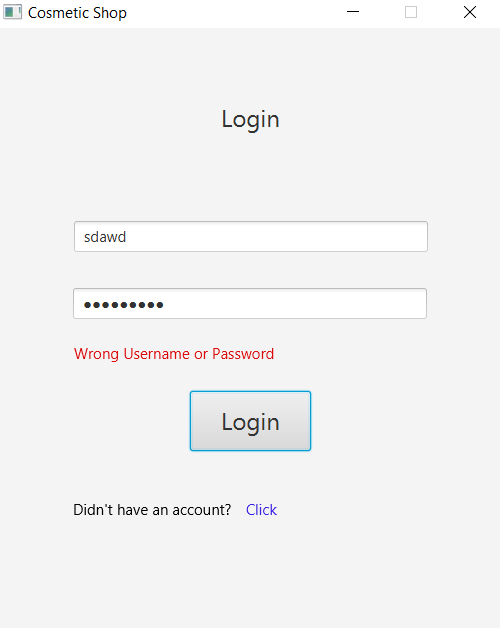
ตาราง Order

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field | Description | Type | Size | Default | PK | FK |
| O\_num | รหัสใบสั่งซื้อ | INT | 5 | NONE | Y | - |
| P\_name | ชื่อสินค้า | TEXT | 30 | NONE | - | - |
| O\_quantity | จำนวนสินค้าในใบสั่ง | INT | 5 | NONE | - | - |
| O\_price | จำนวนเงินที่ต้องชำระ | INT | 10 | NONE | - | - |
| Status | สถานะของใบ order | TEXT | 10 | NONE | - | - |
| username | ชื่อลูกค้า | TEXT | 20 | NONE | - | - |

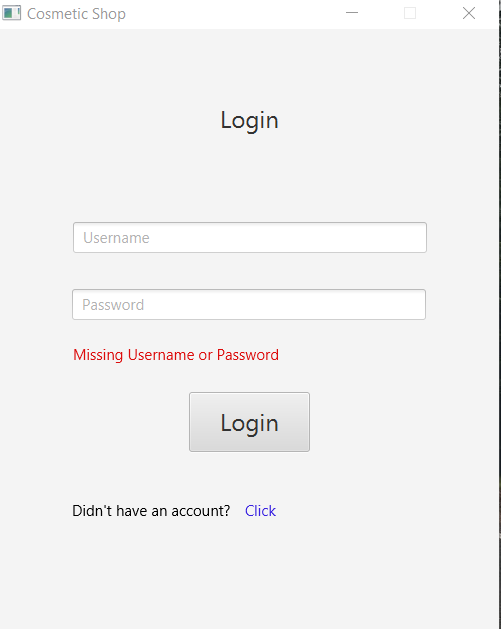
2.3 โครงสร้าง menu ,Screen,ผลการทดลองที่เกิดขึ้น



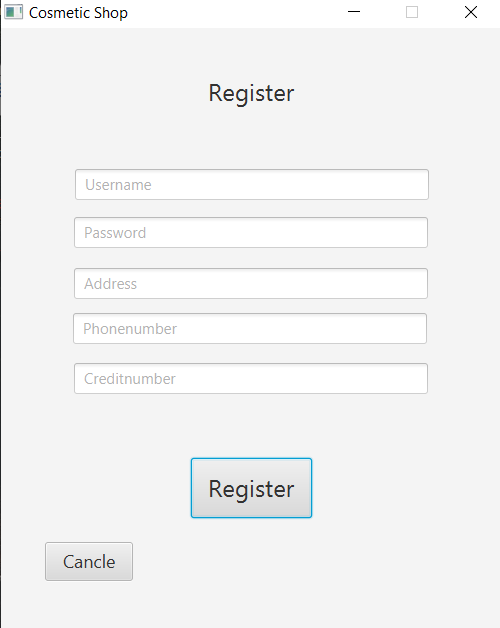
รูปของหน้า Loginview เป็นหน้าแรกของโปรแกรมนี้



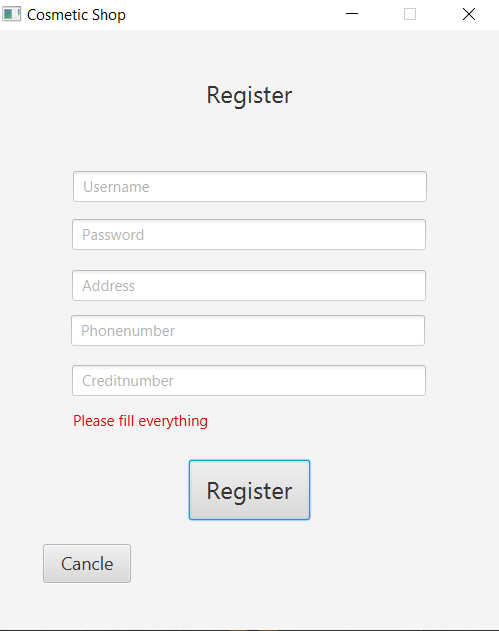
รูปของกรณีที่ใส่ข้อมูล Username หรือ Password ไม่ถูกต้องแล้วกดปุ่ม Login

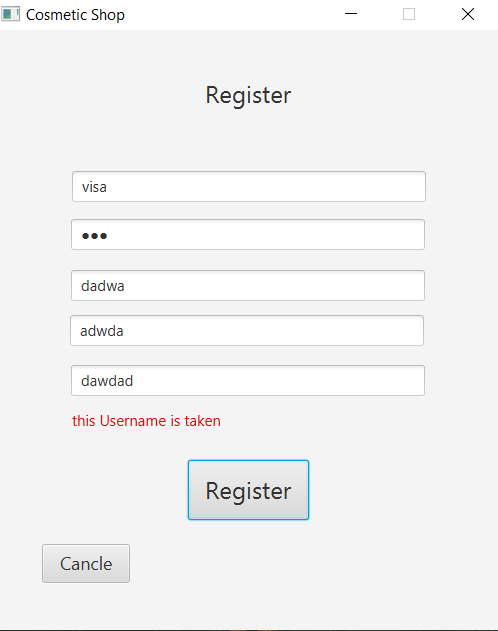


รูปของกรณีที่ไม่ได้ใส่ข้อมูลแล้วกดปุ่ม Login

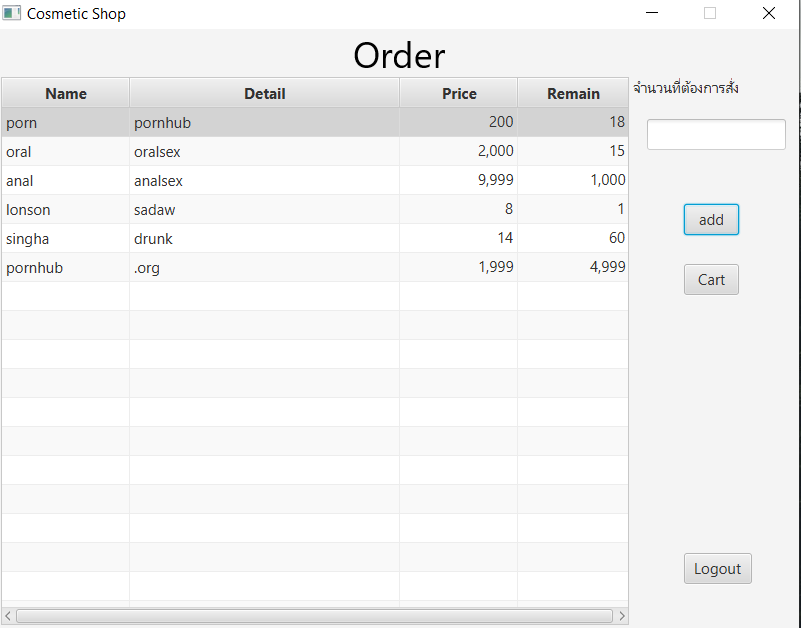


รูปของหน้า Register โดยเข้ามาจากหน้า Login โดยกดปุ่ม “Click”

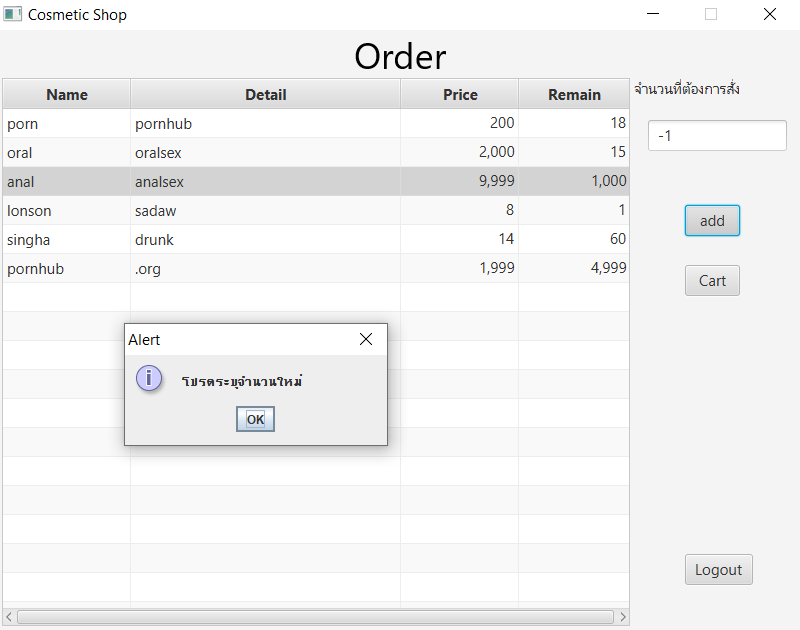


รูปของหน้า Register กรณีไม่ได้ใส่ข้อมูลอะไรเลยแล้วกดปุ่ม Register

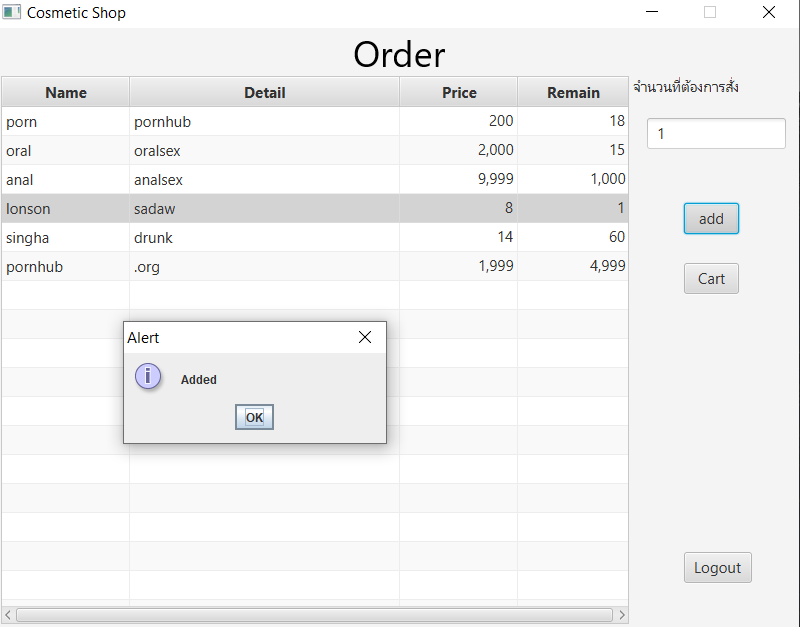
รูปของหน้า Register หลังจากกดปุ่ม Register ตอนใส่ข้อมูล แล้ว Username ไปซ้ำกับข้อมูลใน DB



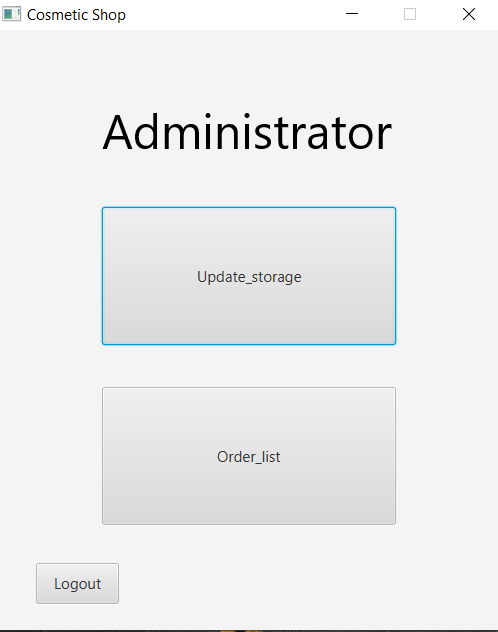
รูปของ หน้าสั่งสินค้า หลังจาก Login ในฐานะ Customer



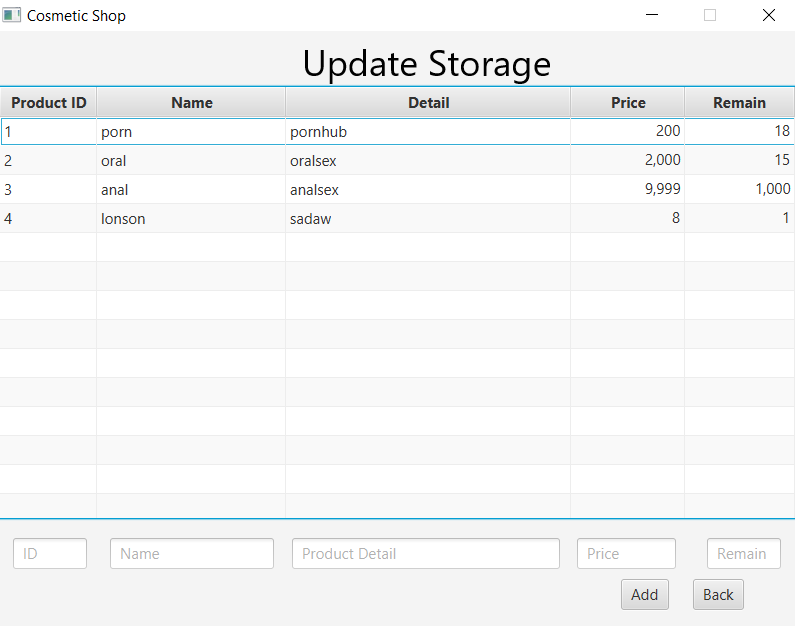
รูปของ หน้าสั่งสินค้า หลังจากกดปุ่ม add โดยใส่จำนวนสินค้าเป็น น้อยกว่า 1 หรือ ติดทศนิยม หรือ ใส่จำนวนที่ต้องการมากกว่าจำนวนสินค้าที่มีอยู่



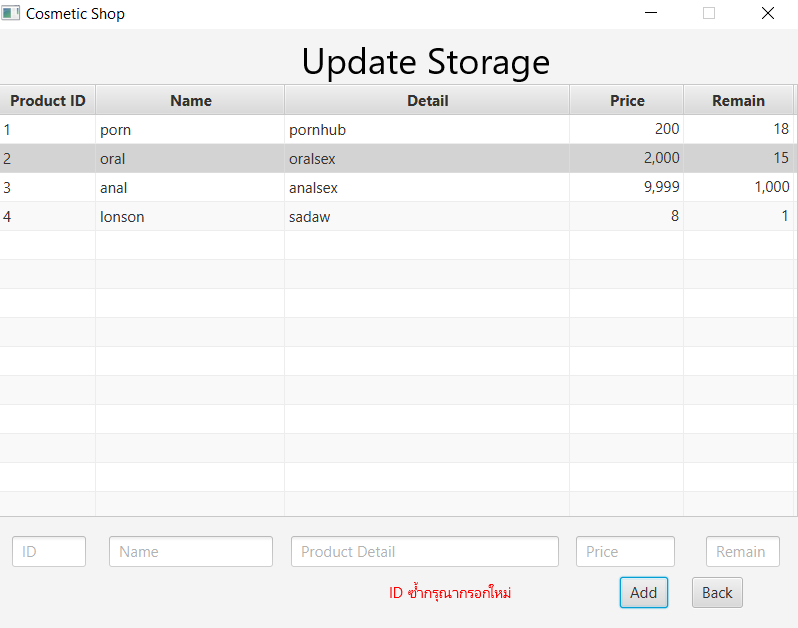
รูปของ หน้าสั่งสินค้า หลังจากเพิ่มจำนวนสินค้าแล้วกดปุ่ม add



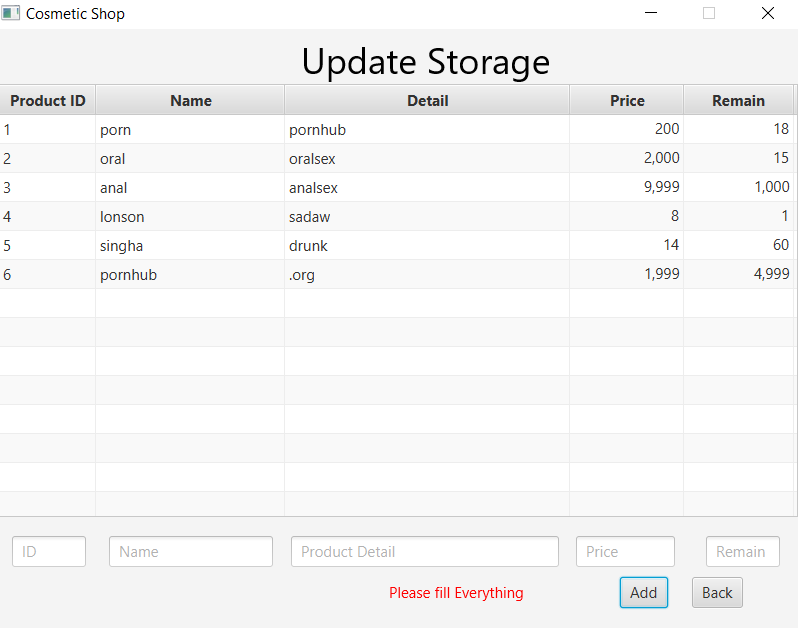
รูปของหน้า Admin เวลาเข้าสู่ระบบจากหน้า Login ด้วย User ที่ถูกกำหนดไว้ใน DB



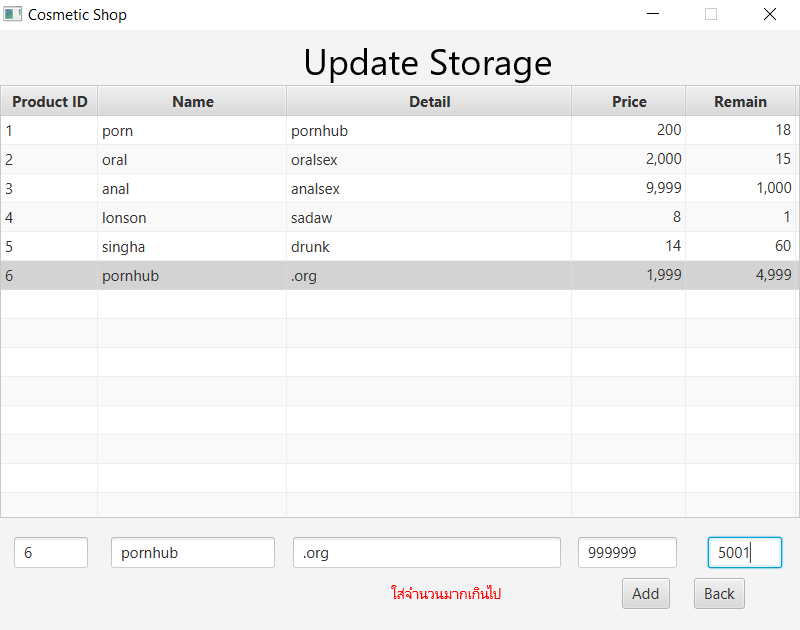
รูปของหน้า update สินค้า โดยเข้ามาหน้านี้จากการกดปุ่ม Update\_storage จากหน้า Admin



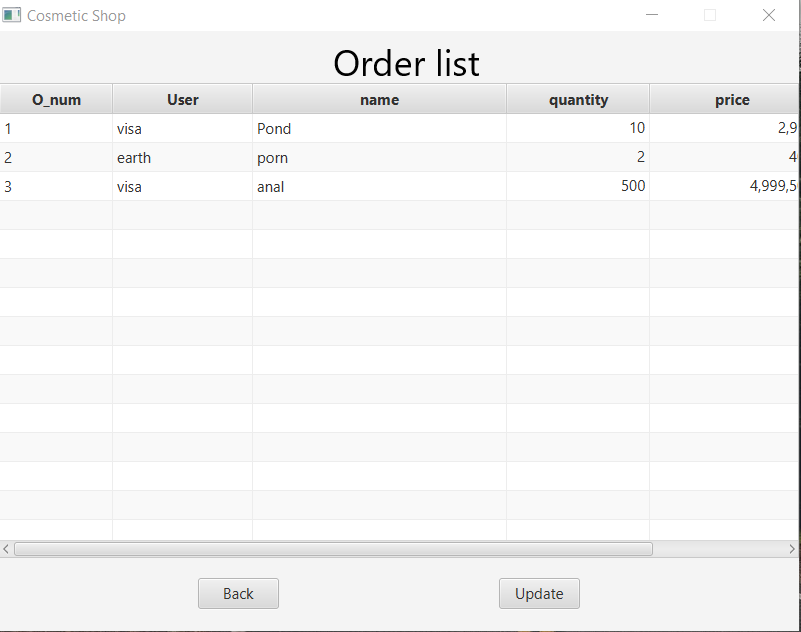
รูปของหน้า update สินค้า กรณีที่ใส่ ID ซ้ำกับในระบบของ DB แล้วกดปุ่ม Add



รูปของหน้า update สินค้า กรณีที่ใส่ข้อมูลไม่ครบ แล้วกดปุ่ม Add

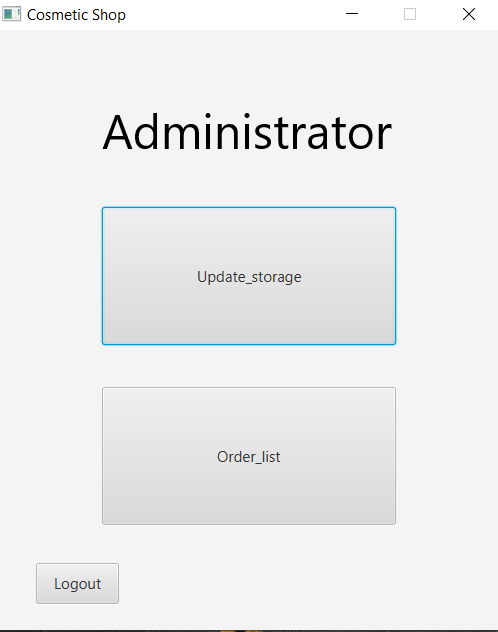


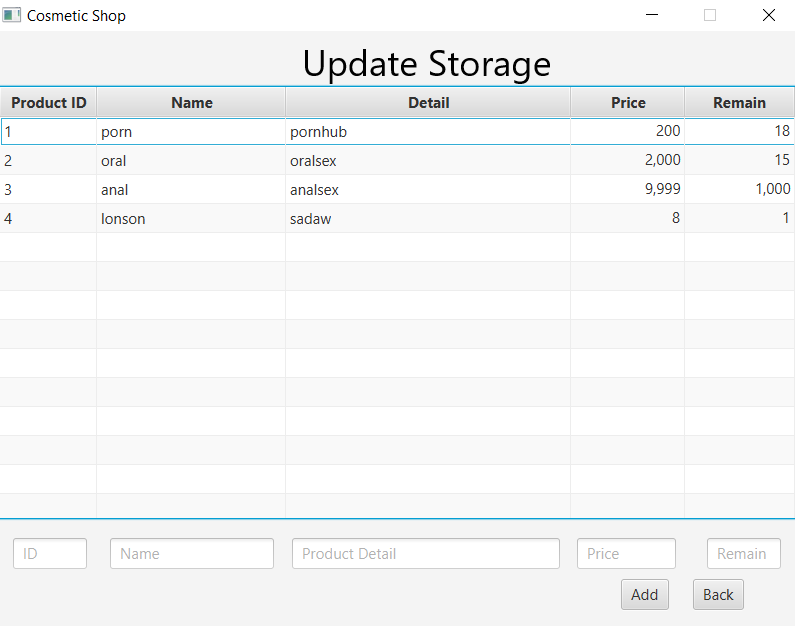
รูปของหน้า update สินค้า กรณีใส่จำนวนสินค้ามากเกินไป



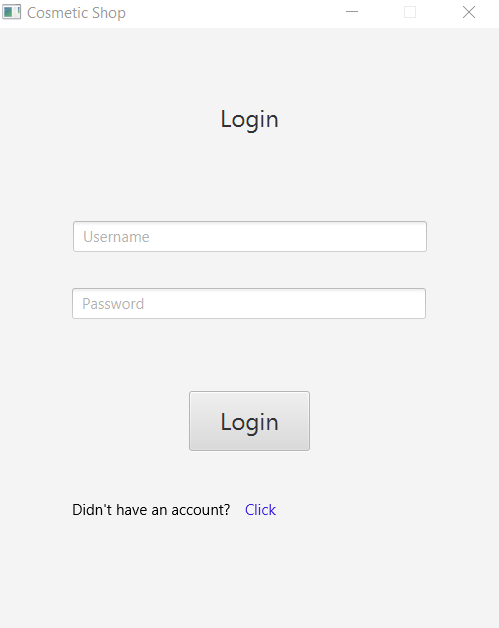
รูปของ หน้า Order\_list หลังจากกดปุ่ม Order\_list ผ่านหน้า Admin

โดยเมื่อกดปุ่ม update จะทำการเปลี่ยนสถานะ wait เป็น done





รูปภาพของก่อนจะกดปุ่ม Log out



รูปภาพหลังจากกดปุ่ม Logout

สรุปผล

2 ปัญหาอุปสรรค

สมาชิกทุกคนในกลุ่มได้มีการคุยกันในด้านของระบบและวางโครงสร้างนานเกินไป ทำให้การทำงานในด้านของตัวโปรแกรมมีเวลาน้อย นอกจากนี้ผู้ที่ทำส่วนโปรแกรมมีความรู้ในด้านการเขียน code ไม่เยอะมากจึงทำให้เสียเวลาในการศึกษาเพิ่ม นอกจากนี้เนื่องจากการเขียน code ที่ใช้เวลานานแล้ว การทำรายงานก็ไม่สามารถทำอย่างรวดเร็วได้เนื่องจาก ต้องมีการแก้รายงานโดยอ้างอิงจากโปรแกรมบางส่วนจึงทำให้การทำรายงานช้าตามๆไป

แนวทางการพัฒนา

1.การวิเคราะห์ปัญหา

สอบถามผู้ใช้งานในด้านความถูกต้อง,ความสะดวก,เพื่อนำไปแก้ไขข้อผิดพลาด

2. การออกแบบระบบ

ระบบต้องใช้งานง่ายและสามารถบันทึก,แก้ไข,เพิ่ม,ลด ข้อมูลได้อย่างครบถ้วนตามวัตถุประสงค์ของผู้ใช้งานและโปรแกรม

ข้อสรุปและข้อเสนอแนะ

1.เป้าหมาย = เพื่อให้ผู้ใช้งานใช้ได้ง่ายและสามารถเก็บข้อมูลการทำงานได้

2.เวลา = เวลาในการออกแบบและเวลาในการสร้างระบบ

3.ต้นทุน = ไม่มีเพราะมีโปรแกรมและอุปกรณ์การทำงานพร้อมอยู่แล้ว

4.คุณภาพ = เมื่อนำระบบจัดเก็บข้อมูลไปใช้พบว่ามีความถูกต้องและแม่นยำ

5.ประสิทธิภาพ = ในการออกแบบหน้าจอ เน้นที่สามารถทำให้ใช้ง่ายและทำความเข้าใจง่าย

6.ประสิทธิผล = ระบบเก็บข้อมูลได้อย่างครบถ้วน

7.ความถูกต้อง = เมื่อกรอกข้อมูล ระบบจะตรวจสอบและปรับปรุงให้ข้อมูลเป็นปัจจุบันที่สุด

8.การยอมรับ = ผู้ใช้งานมีความพึงพอใจกับระบบ

9.การควบคุม = ข้อมูลในระบบสามารถป้องกันการสูญหายและมีการตรวจสอบข้อมูลก่อนนำเข้าสู่ระบบ