**บทที่ 1**

**บทนำ**

* 1. **ที่มาและความสำคัญของปริญญานิพนธ์**

ร้านถวิล เป็นสถานบันเทิงที่ เป็นที่นิยมของวัยรุ่น และวัยทำงาน การรับรายการอาหาร และการจัดการคลังสินค้ายังมีความยุ่งยากซับซ้อน เพราะทางร้านถวิลยังใช้การรับรายการอาหารแบบเขียนใส่กระดาษแล้วนำไปให้ครัว และการจัดการคลังสินค้าแบบเก่าที่ยังไม่ได้นำเทคโนโลยีเข้ามาใช้ทำให้เกิดปัญหา เช่น การคำนวณรายการอาหารผิดพลาด รายการอาหารไม่ครบถ้วน วัตถุดิบไม่เพียงพอ อาหารที่ได้ไม่เป็นไปตามลำดับเมนูที่สั่ง และเกิดความยุ่งยากในการเช็คสต็อกสินค้าทำให้เกิดความผิดพลาดในการจัดการสต๊อกสินค้า ซึ่งเป็นผลทำให้ความนิยมลดลง

เนื่องจากเครื่อง POS ของทางร้านใช้อยู่สามารถคำนวณเงิน ดูอาหารและเครื่องดื่ม รวมโต๊ะได้ รับเมนูอาหารและเครื่องดื่มได้ แต่ว่าเครื่องที่ทางร้านใช้อยู่ทำงานช้า จึงทำให้เกิดความล้าช้าในการสั่งอาหารและเครื่องดื่ม บางครั้งรอนานเกินไปอาจทำให้ลูกค้าหงุดหงิดได้ หรือ บางครั้งพนักงานต้องจำหรือจดใส่กระดาษเพื่อรับเมนูอาหารและเครื่องดื่ม ทำให้เกิดความผิดในการสั่งอาหารและเครื่องดื่ม

ดังนั้นทางคณะผู้จัดทำจึงมีแนวคิดแก้ไขปัญหาโดยการจัดทำและพัฒนาระบบบริหารจัดการร้านถวิล ผ่านเว็บแอปพลิเคชัน เพื่อการทำงานที่เป็นระเบียบ และความสะดวกรวดเร็วให้กับทางร้านถวิล

**1.2 วัตถุประสงค์ของปริญญานิพนธ์**

1.2.1 เพื่อศึกษาเกี่ยวกับการเขียนเว็บไซต์ด้วย รีแอค จาวาสคริปต์ สปริงบูท ในการเขียนหน้าเว็บและการจัดการฐานข้อมูล

1.2.1เพื่อพัฒนาและออกแบบระบบจัดการร้านอาหาร

1.2.3 เพื่อจัดการสต๊อกสินค้าร้านถวิล

1.2.2 เพื่อสร้างความสะดวกแก่พนักงานร้านถวิล

**1.3 ขอบเขตของปริญญานิพนธ์**

1.3.1 ผู้ดูแลระบบ(Casheir) สามารถใช้งานในระบบได้ดังนี้

- จัดการข้อมูลรายการอาหาร และเครื่องดื่ม

- จัดการข้อมูลโปรโมชั่น

- จัดการข้อมูลวัตถุดิบ และต้นทุน

- รายงานรายรับ-รายจ่ายปัจจุบันและย้อนหลัง รายวัน รายเดือน รายปี

- รายงานวัตถุดิบคงเหลือและต้นทุนปัจจุบันและย้อนหลัง รายวัน รายเดือน รายปี

- ตรวจสอบสถานะรายการอาหารของแต่ละโต๊ะ

- สรุปยอดรายการอาหารขายดี

- สามารถแสดงเมนูที่ขายดี และเมนูแนะนำ

- ออกใบเสร็จให้กับลูกค้า

- บันทึกการชำระค่าอาหาร

- ยกเลิกรายการอาหารที่สั่ง

- คำนวณค่าอาหาร

- จัดการการย้ายโต๊ะ

- จัดการการรวมโต๊ะ

1.3.2 พนักงานรับรายการอาหาร สามารถใช้งานระบบได้ดังนี้

- รับรายการสั่งอาหาร

- ตรวจสอบสถานะรายการอาหารของแต่ละโต๊ะ

- ยกเลิกรายการอาหารที่สั่ง

- คำนวณค่าอาหาร

- เรียกดูข้อมูลรายการอาหาร

- เรียกดูข้อมูลสต๊อก

- เรียกดูข้อมูลโปรโมชั่น

- จัดการการย้ายโต๊ะ

- จัดการการรวมโต๊ะ

1.3.3 พนักงานครัว สามารถใช้งานระบบได้ดังนี้

- ดูรายการอาหารที่สั่งตามลำดับ

- เปลี่ยนสถานะอาหารที่สั่ง (กำลังทำ พร้อมเสิร์ฟ เสิร์ฟแล้ว ยกเลิก ของเสีย)

**1.4 ประโยชน์ที่คาดว่าจะได้รับ**

1.5.1 ได้รับความรู้เกี่ยวกับการเขียนเว็บไซต์ด้วย รีแอค จาวาสคริปต์ สปริงบูท ในการเขียนหน้าเว็บและการจัดการฐานข้อมูล

1.5.2 สามารถรับสั่งรายการอาหารได้ถูกต้องและเป็นระบบ

1.5.3 สามารถวิเคราะห์รายรับ-รายจ่ายและจัดการสต๊อกสินค้าได้เป็นระเบียบมากขึ้น

1.5.4 สามารถจัดการสต๊อกสินค้าได้เป็นระเบียบมากขึ้น

1.5.5 ได้ระบบบริหารจัดการร้านถวิล ผ่านเว็บแอพพลิเคชัน

**1.5 ขั้นตอนการดำเนินงาน**

1.4.1 ศึกษาระบบงานของร้านถวิลว่ามีการทำงานเป็นอย่างไร ตั้งแต่เริ่มเข้ารับบริการจนถึงการชำระค่าบริการ เพื่อทราบกระบวนการทำงานและความต้องการหรือปัญหา

1.4.2 วิเคราะห์ระบบการทำงานของร้านอาหาร เช่น ใช้การเก็บข้อมูลแบบกระดาษ ทำให้การค้นหาข้อมูลเป็นไปได้ยากลำบากและอาจทำให้ข้อมูลสูญหาย

1.4.3 ศึกษาเกี่ยวกับการเขียนเว็บไซต์ด้วย รีแอค จาวาสคริปต์ สปริงบูท ในการเขียนหน้าเว็บและการจัดการฐานข้อมูล

1.4.4 ทำการพัฒนาระบบตามรูปแบบโดยในส่วนของ Web Application ใช้ รีแอค จาวาสคริปต์ ในการพัฒนาเว็บไซต์ และในส่วนของการจัดการฐานข้อมูลใช้ สปริงบูท

1.4.5 ทดสอบระบบเพื่อหาข้อผิดพลาดในโปรแกรม รวมถึงตรวจสอบว่าโปรแกรมสามารถให้ผลตรงตามที่ต้องการหรือไม่ เพื่อให้เกิดความสมบูรณ์ในโปรแกรม

1.4.6 สรุปผลการทดลอง

1.4.7 จัดทําเล่มปริญญานิพนธ์

**1.6 แผนการดําเนินงาน**

**ตารางที่ 1.1** แผนการดําเนินงานโครงงานวิศวกรรมคอมพิวเตอร์ 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **การดําเนินงาน** | **ก.ค.65** | | | | **ส.ค.65** | | | | **ก.ย.65** | | | | **ต.ค.65** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| ศึกษาระบบงานของร้านถวิลว่ามีการทำงานเป็นอย่างไร ตั้งแต่เริ่มเข้ารับบริการจนถึงการชำระค่าบริการ เพื่อทราบกระบวนการทำงานและความต้องการหรือปัญหา |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| วิเคราะห์ระบบการทำงานของร้านอาหาร เช่น ใช้การเก็บข้อมูลแบบกระดาษ ทำให้การค้นหาข้อมูลเป็นไปได้ยากลำบากและอาจทำให้ข้อมูลสูญหาย |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ศึกษาข้อมูลการเขียนโปรแกรมด้วย React, Java Script, Java Spring Boot, API เกี่ยวกับการเขียนโปรแกรมและการจัดการฐานข้อมูล |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**ตารางที่ 1.2** แผนการดําเนินงานโครงงานวิศวกรรมคอมพิวเตอร์ 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **การดําเนินงาน** | **ต.ค.65** | | | | **พ.ย.65** | | | | **ธ.ค.65** | | | | **ม.ค.66** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| ทำการพัฒนาระบบตามรูปแบบโดยในส่วนของ Web Application ใช้ React, Java Script ในการพัฒนาเว็บไซต์ และในส่วนของการจัดการฐานข้อมูลใช้ Java Spring Boot |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ทดสอบระบบเพื่อหาข้อผิดพลาดในโปรแกรม รวมถึงตรวจสอบว่าโปรแกรมสามารถให้ผลตรงตามที่ต้องการหรือไม่ เพื่อให้เกิดความสมบูรณ์ในโปรแกรม |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| สรุปผลการทดลอง |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| จัดทําเล่มปริญญานิพนธ์ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**บทที่ 2**

**ทฤษฎีและงานวิจัยที่เกี่ยวข้อง**

ในการทำปริญญานิพนธ์ระบบบริหารจัดการร้านถวิลผ่านเว็บแอปพลิเคชัน กรณีศึกษาจำเป็นต้องใช้ความรู้ และทฤษฎีที่หลากหลายเพื่อใช้ในการปรับปรุงการทำงานของร้านถวิล และช่วยอำนวยความสะดวกในการดำเนินงาน โดยคณะผู้จัดทำได้ทำการศึกษารายละเอียด และรวบรวมเนื้อหาข้อมูลต่างๆ ที่เกี่ยวข้องสรุปได้ดังนี้

2.1 ภาษาจาวาสคริปต์ (JavaScript)

2.2 ภาษาจาวา (Java)

2.3 จาวาสปริงบูท (Java Spring Boot)

2.4. รีแอค (React)

2.5 ภาษาเอสคิวแอล (Structured Query Language : SQL)

2.6 แผนภาพกระแสข้อมูล (Data Flow Diagram)

2.7 แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล (Entity Relation Diagrams : ER–Diagrams)

**2.1 ภาษาจาวาสคริปต์ (JavaScript)**

ภาษาจาวาสคริปต์[1] คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ที่กำลังได้รับความนิยมอย่างสูง จาวาสคริปต์ เป็น ภาษาสคริปต์เชิงวัตถุ ซึ่งในการสร้างและพัฒนาเว็บไซต์ เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ ดำเนินงานไปทีละคำสั่ง หรือเรียกว่า อ็อบเจ็กโอเรียลเต็ด (Object Oriented Programming : OOP) ที่มีเป้าหมายในการ ออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server

**2.2 ภาษาจาวา (Java)**

ภาษาจาวา[2] หรือ Java programming language คือภาษาโปรแกรมเชิงวัตถุ (Object-Oriented Programming : OOP) พัฒนาโดย เจมส์ กอสลิง และคณะ ที่บริษัท ซันไมโครซิสเต็มส์ ภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้สถานที่ทำงานของ เจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ จาวา ซึ่งเป็นชื่อกาแฟแทน จุดเด่นของภาษาจาวา อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของ การเขียนโปรแกรมเชิงวัตถุ มาพัฒนาโปรแกรมของตน ภาษาจาวาเป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้นคลาสคือที่เก็บเมทอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม

**2.3 สปริงบูท (Spring Boot)**

จาวาสปริงบูท[3] เป็นหนึ่งในเฟรมเวิร์ค (Framework) ที่ได้รับความนิยมมากที่สุดสำหรับพัฒนาแอพพลิเคชั่นด้วยภาษาจาวา แรกเริ่มนักพัฒนาใช้ จาวาสปริงบูท เพื่อพัฒนาเว็บแอปพลิเคชัน แต่เราสามารถนำ จาวาสปริงบูท มาใช้กับการพัฒนาแอพพลิเคชั่นสำหรับคอมพิวเตอร์ตั้งโต๊ะหรือแม้แต่แอปพลิเคชันที่ทำงานในแบบบรรทัดคำสั่งได้ จาวาสปริงบูท เป็นเฟรมเวิร์คที่รวบรวมเฟรมเวิร์คหลาย ๆ เฟรมเวิร์คมาทำงานร่วมกันเพื่อสร้างเป็นฟังก์ชันมาตรฐานให้กับแอปพลิเคชัน ในบางครั้งจึงเรียก จาวาสปริงบูท ว่าแพลตฟอร์ม ตัวอย่างของสิ่งที่ จาวาสปริงบูท สามารถทำได้ เช่น component container รองรับการจัดการวงจรชีวิตของออบเจกต์ภายในแอพพลิเคชั่น, data access framework รองรับการทำงานกับฐานข้อมูลได้หลายแบบ, security framework มีเฟรมเวิร์ครองรับเรื่องความปลอดภัย เช่น การรับรองผู้ใช้งาน การควบคุมการใช้งาน testing framework มีเฟรมเวิร์ครองรับการทดสอบแอพพลิเคชั่น เนื่องจาก จาวาสปริงบูท มีความสามารถมากมาย ดังนั้นจึงมีความซับซ้อนอย่างมากในการตั้งค่าการใช้งาน ทางผู้พัฒนา จาวาสปริงบูท จึงพัฒนาเฟรมเวิร์ค จาวาสปริงบูท ขึ้นมาเพื่อแก้ปัญหาเรื่องความซับซ้อนในการตั้งค่าการใช้งานของ จาวาสปริงบูท ให้เป็นอัตโนมัติมากขึ้น ทำให้นักพัฒนาไม่ต้องตั้งค่าทุกอย่างด้วยตัวเองโดย จาวาสปริงบูท เฟรมเวิร์คจะใช้ค่าตั้งต้นที่กำหนดแล้วในการกำหนดค่าต่าง ๆ ให้แทน นอกจากนี้ จาวาสปริงบูท ยังรองรับการใช้งานในแบบบรรทัดคำสั่ง การติดตามสถานะการทำงานของแอพพลิเคชั่น มี application server ในตัวเพื่อให้ผู้พัฒนาเรียกใช้แอพพลิเคชั่นที่สร้างขึ้นมา

**2.4 รีแอค (React)**

รีแอค[4] คือ จาวาสคริปต์ ไลบรารี (JavaScript Library) ที่ใช้สำหรับสร้าง หน้าต่างผู้ใช้งาน (User Interface : UI) ที่ให้เราสามารถเขียนโค้ดในการสร้าง UI ที่มีความซับซ้อนแบ่งเป็นส่วนเล็ก ๆ ออกจากกันได้ ซึ่งแต่ละส่วนสามารถแยกการทำงานออกจากกันได้อย่างอิสระ และทำให้สามารถนำชิ้นส่วน UI เหล่านั้นไปใช้ซ้ำได้อีก

**2.5 ภาษาเอสคิวแอล (SQL)**

ภาษาเอสคิวแอล ย่อมาจาก (Structured Query Language : SQL) [5]  คือ ภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตราฐานบนระบบฐานข้อมูลเชิงสัมพันธ์และเป็นระบบเปิดหมายถึงเราสามารถใช้คำสั่ง เอสคิวแอล กับฐานข้อมูลชนิดใดก็ได้ และ คำสั่งงานเดียวกันเมื่อสั่งงานผ่าน  ระบบฐานข้อมูลที่แตกต่างกันจะได้ ผลลัพธ์เหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูล ชนิดใดก็ได้โดยไม่ติดยึดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง นอกจากนี้แล้ว เอสคิวแอล ยังเป็นชื่อโปรแกรมฐานข้อมูล ซึ่งโปรแกรม เอสคิวแอล เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม เอสคิวแอล จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่ง ซึ่งแบ่งการทำงานได้เป็น 4 ประเภท ดังนี้ Select query ใช้สำหรับดึงข้อมูลที่ต้องการ, Update query ใช้สำหรับแก้ไขข้อมูล,Insert query ใช้สำหรับการเพิ่มข้อมูล, Delete query ใช้สำหรับลบข้อมูลออกไป

**2.6 แผนภาพกระแสข้อมูล (Data Flow Diagram)**

แผนภาพกระแสข้อมูล (Data Flow Diagram : DFD) [6] เป็นเครื่องมือเชิงโครงสร้างที่ใช้บรรยายภาพรวม ของระบบโดยแสดงขั้นตอนการทำงานของระบบหรือโพรเซส (Process) ระบุแหล่งกำเนิดของข้อมูล การไหลของข้อมูล ปลายทางข้อมูล การเก็บข้อมูลและการประมวลผลข้อมูล กล่าวง่าย ๆ คือ แผนภาพกระแสข้อมูล จะช่วยแสดงแผนภาพว่า ข้อมูลมาจากไหน จะไปไหน เก็บข้อมูลไว้ที่ไหน มีอะไรเกิดขึ้นกับข้อมูลระหว่างทางเรียกว่าแผนภาพกระแสข้อมูล หรือ แผนภาพแสดงความเคลื่อนไหวของข้อมูลโดย แผนภาพกระแสข้อมูล

**2.7 แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล (Entity Relation Diagrams : ER–Diagrams)**

แผนภาพ (E-R Diagram) [7] คือ แบบจำลองที่ใช้อธิบายโครงสร้างของฐานข้อมูลซึ่งเขียนออกมาในลักษณะของรูปภาพ การอธิบายโครงสร้างและความสัมพันธ์ของข้อมูล (Relationship) ประกอบด้วย เอนทิตี้ (Entity) เป็นวัตถุ หรือสิ่งของที่เราสนใจในระบบงานนั้น ๆ แอททริบิว (Attribute) เป็นคุณสมบัติของวัตถุที่เราสนใจ

**บทที่ 3**

**วิธีการดำเนินงาน**

จากการศึกษาข้อมูลเกี่ยวกับการทำระบบบริหารจัดการร้านถวิลผ่านเว็บแอปพลิเคชัน ผู้จัดทำได้รวบรวมข้อมูลต่าง ๆ ซึ่งมีขั้นตอนการดำเนินงานดังต่อไปนี้

3.1 วิเคราะห์ระบบบริหารจัดการร้านถวิลผ่านเว็บแอปพลิเคชัน

3.2 การออกแบบแผนภาพกระแสการไหลของข้อมูล (Data Flow Diagrams)

3.3 พจนานุกรมข้อมูล (Data Dictionary)

3.4 ตัวอย่างขั้นตอนการทำงานของระบบ

**3.1 วิเคราะห์และออกแบบระบบฐานข้อมูล**

การวิเคราะห์และออกแบบโครงสร้างการดำเนินงานภายในระบบนั้นเป็นการวิเคราะห์ขั้นตอนเกี่ยวกับการรับรายการอาหาร และการจัดการสต๊อก รวมทั้งการวิเคราะห์ขั้นตอนต่าง ๆ ทั้งหมดที่เกิดขึ้นภายในระบบแสดงโดยใช้ผังงาน (Flow Chart) ได้ดังนี้

Diagram

Description automatically generated

**รูปที่ 3.1** ผังงานเบื้องต้นของระบบบริหารจัดการร้านถวิลผ่านเว็บแอปพลิเคชัน

จากรูปที่ 3.1 ผู้ใช้งานสามารถเข้าสู่ระบบโดยตรงผ่านหน้าเว็บแอปพลิเคชัน โดยระบบมีสิทธิ์ผู้ใช้ทั้งหมด 2 สิทธิ์ คือ สิทธิ์ของพนักงานรับOrder และพนักงานCashier

1. สิทธิ์ของพนักงานรับ Order

- รับรายการสั่งอาหาร

- ตรวจสอบสถานะรายการอาหารของแต่ละโต๊ะ

- ยกเลิกรายการอาหารที่สั่ง

- คำนวณค่าอาหาร

- เรียกดูข้อมูลรายการอาหาร

- เรียกดูข้อมูลสต๊อก

- เรียกดูข้อมูลโปรโมชั่น

- จัดการการย้ายโต๊ะ

- จัดการการรวมโต๊ะ

2. สิทธิ์ของพนักงาน Cashier

- จัดการข้อมูลรายการอาหาร และเครื่องดื่ม

- จัดการข้อมูลโปรโมชั่น

- จัดการข้อมูลวัตถุดิบ และต้นทุน

- ดูรายงานรายได้ย้อนหลัง รายวัน รายเดือน รายปี

- ดูรายงานวัตถุดิบคงเหลือ และต้นทุน ย้อนหลัง รายวัน รายเดือน รายปี

- ดูรายงานรายรับ - รายจ่าย รายวัน รายเดือน รายปี

- ตรวจสอบสถานะรายการอาหารของแต่ละโต๊ะ

- สรุปยอดรายการอาหารขายดี

- สามารถแสดงเมนูที่ขายดี และเมนูแนะนำ

- ออกใบเสร็จให้กับลูกค้า

- บันทึกการชำระค่าอาหาร

- ยกเลิกรายการอาหารที่สั่ง

- คำนวณค่าอาหาร

- จัดการการย้ายโต๊ะ

- จัดการการรวมโต๊ะ

3 สิทธิ์ของครัว

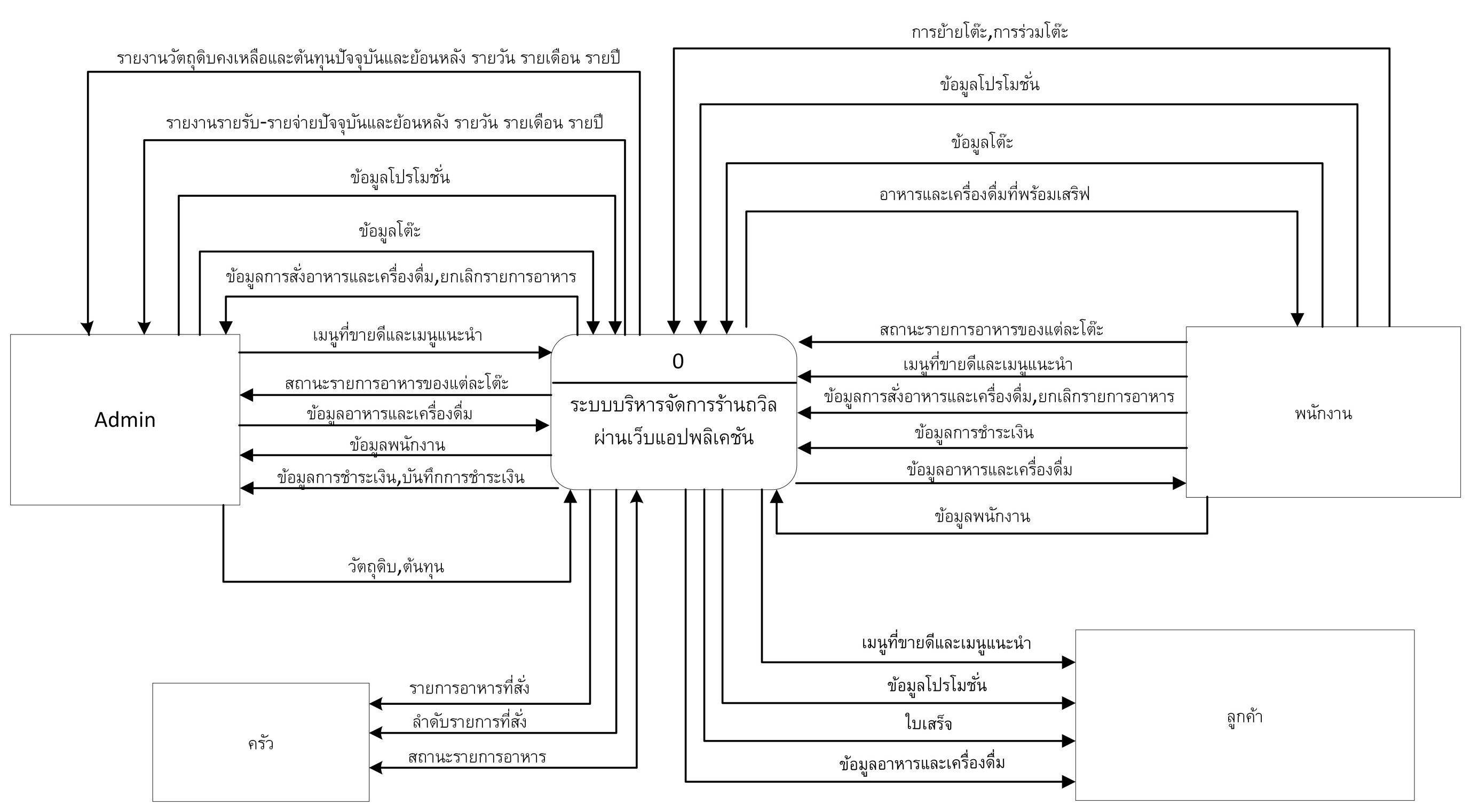
- ดูรายการอาหารที่สั่ง

- ดูลำดับรายการที่สั่ง

- รายงานอาหารที่ทำเสร็จ

**3.2 การออกแบบแผนภาพกระแสไหลของข้อมูล (Data Flow Diagrams)**

แผนภาพกระแสการไหลของข้อมูลระดับสูงสุดได้นําข้อมูลจากการวิเคราะห์ และออกแบบโครงสร้างระบบงาน มาใช้ในการออกแบบแผนภาพกระแสการไหลของข้อมูลระดับสูงสุดโดยอธิบายกระบวนการทํางาน ดังรูปที่ 3.2



**รูปที่ 3.2** แผนภาพกระแสการไหลของข้อมูลระดับสูงสุด (Context Diagram)

3.2.1 การออกแบบแผนภาพกระแสการไหลของข้อมูลระดับที่ 1

แผนภาพกระแสการไหลข้อมูล ได้นําข้อมูลการออกแบบแผนภาพกระแสการไหลของข้อมูลระดับสูงสุดมาใช้ในการออกแบบแผนภาพกระแสการไหลของข้อมูลระดับที่ 1 โดยอธิบายกระบวนการทํางานกระแสการไหลของข้อมูลแยกเป็นกระบวนการทํางานต่าง ๆ แสดงดังรูป 3.3 ดังนี้

กระบวนการที่ 1.0 กระบวนการข้อมูลพนักงาน ทําหน้าที่ในการเรียกดูเพิ่มลบแก้ไขข้อมูลพนักงาน

กระบวนการที่ 2.0 กระบวนการจัดการสต๊อก ทําหน้าที่ในการเรียกดูเพิ่มลบแก้ไขข้อมูลสต๊อก

กระบวนการที่ 3.0 กระบวนการเรียกดูข้อมูลในสต๊อก ทําหน้าที่ แสดงสถานะของสต๊อก

กระบวนการที่ 4.0 กระบวนการรับอาหารและเครื่องดื่ม ทําหน้าที่ในรับอาหารและเครื่องดื่มกับลูกค้า

กระบวนการที่ 5.0 กระบวนการรายงาน ทําหน้าที่ในการเรียกดู รายรับรายจ่าย ออกรายงานใบชําระเงิน สรุปยอดรายเดือน รายปี

Diagram

Description automatically generated

**รูปที่ 3.3** แผนภาพกระแสการไหลข้อมูลระดับที่ 1

**3.3 พจนานุกรมข้อมูล**

3.3.1 แผนภาพแสดงความสัมพันธ์เอนทิตีของระบบ

Diagram

Description automatically generated

**รูปที่ 3.4** แผนภาพความสัมพันธ์เอนทิตีของระบบ

3.3.2 พจนานุกรมโครงสร้างข้อมูล

จากแผนภาพความสัมพันธ์เอนทิตีในรูปที่ 3.4 นั้นได้แสดงแผนภาพความสัมพันธ์เอนทิตีออกเป็นตารางที่ใช้ในระบบ สามารถอธิบายลักษณะข้อมูลในแต่ละตาราง ดังต่อไปนี้

**ตารางที่ 3.1** พนักงาน (employee)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ชื่อคอลัมน์ | คำอธิบาย | ประเภทข้อมูล | ค่าว่าง | คีย์ | หมายเหตุ |
| emp\_ID | รหัสพนักงาน | Int(11) | No | PK |  |
| emp\_Name | ชื่อพนักงาน | Varchar(100) | No |  |  |
| emp\_Phone | เบอร์โทรพนักงาน | Varchar(10) | No |  |  |
| emp\_Username | ชื่อผู้ใช้งานพนักงาน | Varchar(45) | No |  |  |
| emp\_Password | รหัสเข้าใช้งานพนักงาน | Varchar(45) | No |  |  |
| emp\_Address | ที่อยู่พนักงาน | Varchar(255) | No |  |  |
| emp\_Slip | สลิปโอนเงิน | Varchar(255) | Yes |  |  |
| emp\_Line | ไอดีไลน์พนักงาน | Varchar(45) | No |  |  |

ตารางที่ 3.1 ตารางพนักงาน ใช้เก็บข้อมูลพนักงานภายในร้าน คอลัมน์ emp\_ID ใช้เก็บรหัสพนักงาน ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง ให้เป็นคีย์หลัก, คอลัมน์ emp\_Name ใช้เก็บชื่อของพนักงานร้าน ประเภทข้อมูลเป็น Varchar ไม่อนุญาติให้มีค่าว่าง, คอลัมน์ emp\_Phone ใช้เก็บเบอร์โทรของพนักงาน ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ emp\_Username ใช้เก็บชื่อผู้ใช้งานของพนักงาน ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ emp\_Password ใช้เก็บรหัสเข้าใช้งานของพนักงาน ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ emp\_Address ใช้เก็บที่อยู่ของพนักงาน ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ emp\_Slip ใช้เก็บสลิปโอนเงิน ประเภทข้อมูลเป็น Varchar อนุญาตให้มีค่าว่าง, คอลัมน์ emp\_Line ใช้เก็บไอดีไลน์พนักงาน ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง

**ตารางที่ 3.2** สิทธิ์ของพนักงาน (employee\_role)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ชื่อคอลัมน์ | คำอธิบาย | ประเภทข้อมูล | ค่าว่าง | คีย์ | หมายเหตุ |
| empRole\_ID | รหัสสถานะพนักงาน | Int(1) | No | PK |  |
| emp\_ID | รหัสพนักงาน | Int(11) | No | FK | emp\_ID  (emp\_ID) |

ตารางที่ 3.2 ตารางสิทธิ์ของพนักงาน ใช้เก็บสิทธิ์ของพนักงาน คอลัมน์ empRole\_ID ใช้เก็บรหัสสิทธิ์ของพนักงาน ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง ให้เป็นคีย์หลัก, คอลัมน์ emp\_ID ใช้เก็บรหัสพนักงาน ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง เป็นคีย์นอกจาก ตารางที่ 3.1 ตารางพนักงาน คอลัมน์ emp\_ID

**ตารางที่ 3.3** โต๊ะภายในร้าน (table)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ชื่อคอลัมน์ | คำอธิบาย | ประเภทข้อมูล | ค่าว่าง | คีย์ | หมายเหตุ |
| table\_ID | รหัสโต๊ะ | Int(11) | No | PK |  |
| table\_Zone | พื้นที่โต๊ะ | Varchar(45) | No |  |  |

ตารางที่ 3.3 ตารางโต๊ะภายในร้าน ใช้เก็บข้อมูลของโต๊ะภายในร้าน คอลัมน์ Table\_ID ใช้เก็บรหัสโต๊ะ ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง ให้เป็นคีย์หลัก, คอลัมน์ Table\_Zone ใช้เก็บพื้นที่โต๊ะ หรือตำแหน่งที่อยู่ของโต๊ะ ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง

**ตารางที่ 3.4** แคชเชียร์ (cashier)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ชื่อคอลัมน์ | คำอธิบาย | ประเภทข้อมูล | ค่าว่าง | คีย์ | หมายเหตุ |
| cashier\_ID | รหัสแคชเชียร์ | Int(11) | No | PK |  |

ตารางที่ 3.4 ตารางแคชเชียร์ ใช้เก็บรหัสแคชเชียร์ คอลัมน์ cashier\_ID ใช้เก็บรหัสแคชเชียร์ ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง ให้เป็นคีย์หลัก

**ตารางที่ 3.5** ออเดอร์รวม (total\_order)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ชื่อคอลัมน์ | คำอธิบาย | ประเภทข้อมูล | ค่าว่าง | คีย์ | หมายเหตุ |
| totalOrder\_ID | รหัสออเดอร์รวม | Int(11) | No | PK |  |
| totalPrice | จำนวนเงินรวม | Int(11) | No |  |  |
| totalOrder\_TimeStamp | เวลาที่คิดเงิน | DateTime | No |  |  |
| totalOrder\_Status | สถานะการคิดเงิน | Varchar (45) | No |  |  |

**ตารางที่ 3.5** ออเดอร์รวม (total\_order) (ต่อ)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ชื่อคอลัมน์ | คำอธิบาย | ประเภทข้อมูล | ค่าว่าง | คีย์ | หมายเหตุ |
| cashier\_ID | รหัสแคชเชียร์ | Int(11) | No | FK | cashier\_ID (cashier\_ID) |
| table\_ID | รหัสโต๊ะ | Int(11) | No | FK | table\_ID (table\_ID) |

ตาราง 3.5 ตารางออเดอร์รวม ใช้เก็บรายการสั่งอาหารต่าง ๆ ที่พนักงานรับเข้ามา คอลัมน์ totalOrder\_ID ใช้เก็บรหัสออเดอร์รวม ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง ให้เป็นคีย์หลัก, คอลัมน์ totalPrice ใช้เก็บจำนวนเงินรวม ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ totalOrder\_TimeStamp ใช้เก็บเวลาที่คิดเงิน ประเภทข้อมูลเป็น DateTime ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ totalOrder\_Status ใช้เก็บสถานการณ์คิดเงิน ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ cashier\_ID ใช้เก็บรหัสแคชเชียร์ ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง เป็นคีย์นอกจาก ตารางที่ 3.4 ตารางแคชเชียร์ คอลัมน์ cashier\_ID, คอลัมน์ table\_ID ใช้เก็บรหัสโต๊ะ ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง เป็นคีย์นอกจาก ตารางที่ 3.3 ตารางโต๊ะ คอลัมน์ table\_ID

**ตารางที่ 3.6** สต๊อกสินค้า (stock)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ชื่อคอลัมน์ | คำอธิบาย | ประเภทข้อมูล | ค่าว่าง | คีย์ | หมายเหตุ |
| stock\_ID | รหัสสต๊อก | Int(11) | No | PK |  |
| stock\_Name | ชื่อสินค้า | Varchar(50) | No |  |  |
| stock\_Qty | จำนวนสินค้า | Int(10) | No |  |  |
| stock\_Cost | ต้นทุนสินค้า | Int(10) | No |  |  |
| stock\_Min | จำนวนแจ้งเตือน | Int(5) | No |  |  |
| stock\_Status | สถานะสินค้า | Varchar(50) | No |  |  |
| stock\_TimeStamp | เวลาที่ตัดสินค้า | DateTime | No |  |  |

ตารางที่ 3.6 ตารางสต๊อกสินค้า ใช้เก็บข้อมูลวัตถุดิบต่าง ๆ ที่อยู่ในร้าน คอลัมน์ stock\_ID ใช้เก็บรหัสสต๊อกสินค้า ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง ให้เป็นคีย์หลัก, คอลัมน์ stock\_Name ใช้เก็บชื่อสินค้า ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ stock\_Qty ใช้เก็บจำนวนสินค้า ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ stock\_Cost ใช้เก็บต้นทุนของสินค้า ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ stock\_Min ใช้เก็บค่าของสต๊อกที่ต่ำที่สุดที่ต้องการให้แจ้งเตือน ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง , คอลัมน์ stock\_Status ใช้เก็บสถานะของสินค้า ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง, stock\_TimeStamp ใช้เก็บเวลาของสินค้า ประเภทข้อมูลเป็น DateTime ไม่อนุญาตให้มีค่าว่าง

**ตารางที่ 3.7** เมนูสินค้า และอาหาร (menu)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ชื่อคอลัมน์ | คำอธิบาย | ประเภทข้อมูล | ค่าว่าง | คีย์ | หมายเหตุ |
| menu\_ID | รหัสเมนู | Int(11) | No | PK |  |
| menu\_Name | ชื่อเมนู | Varchar(50) | No |  |  |
| menu\_Price | ราคาเมนู | Int(11) | No |  |  |
| menu\_Pic | รูปภาพเมนู | Varchar(255) | No |  |  |
| menu\_Qty | จำนวนเมนู | Int(10) | No |  |  |
| menu\_Cost | ต้นทุนเมนู | Int(10) | No |  |  |
| menu\_Status | สถานะเมนู | Varchar(45) | No |  |  |
| menu\_TimeStamp | เวลาที่สั่งเมนู | DateTime | No |  |  |
| typeMenu\_ID | ชนิดของเมนู | Int(11) | No | FK | typeMenu\_ID  (typeMenu\_ID) |

ตารางที่ 3.7 ตารางเมนูสินค้า และอาหาร ใช้เก็บข้อมูลเมนูต่าง ๆ คอลัมน์ menu\_ID ใช้เก็บรหัสของเมนู ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง ให้เป็นคีย์หลัก, คอลัมน์ menu\_Name ใช้เก็บชื่อเมนู ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ menu\_Price ใช้เก็บราคาเมนู ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ menu\_Pic ใช้เก็บรูปภาพเมนู ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ menu\_Qty ใช้เก็บจำนวนของเมนู ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ menu\_Cost ใช้เก็บต้นทุนของเมนู ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ menu\_Status ใช้เก็บสถานะของเมนู ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ menu\_TimeStamp ใช้เก็บเวลาที่สั่งเมนู ประเภทข้อมูลเป็น DateTime ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ typeMenu\_ID ใช้เก็บชนิดของเมนู ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง เป็นคีย์นอกจากตารางที่ 3.8 ตารางชนิดเมนู คอลัมน์ typeMenu\_ID

**ตารางที่ 3.8** ชนิดเมนู (type\_Menu)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ชื่อคอลัมน์ | คำอธิบาย | ประเภทข้อมูล | ค่าว่าง | คีย์ | หมายเหตุ |
| typeMenu\_ID | รหัสชนิดเมนู | Int(11) | No | PK |  |
| typeMenu\_Name | ชื่อชนิดเมนู | Varchar(45) | No |  |  |

ตารางที่ 3.8 ตารางชนิดเมนู ใช้เก็บข้อมูลของชนิดเมนูภายในร้าน คอลัมน์ typeMenu\_ID ใช้เก็บรหัสชนิดเมนู ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง ให้เป็นคีย์หลัก, คอลัมน์ typeMenu\_Name ใช้เก็บชื่อชนิดเมนู ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง

**ตารางที่ 3.9** สต๊อกสินค้า และเมนู (stock\_Menu)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ชื่อคอลัมน์ | คำอธิบาย | ประเภทข้อมูล | ค่าว่าง | คีย์ | หมายเหตุ |
| stockMenu\_ID | รหัสสต๊อกเมนู | Int(11) | No | PK |  |
| stockMenu\_Qty | จำนวนสต๊อกเมนู | Int(10) | No |  |  |
| stockMenu\_Status | สถานะสต๊อกเมนู | Varchar(45) | No |  |  |
| stockMenu\_TimeStamp | เวลาที่เข้าถึงสต๊อก | DateTime | No |  |  |
| stock\_ID | รหัสสต๊อก | Int(11) | No | FK | stock\_ID (stock\_ID) |
| menu\_ID | รหัสเมนู | Int(11) | No | FK | menu\_ID (menu\_ID) |

ตารางที่ 3.9 ตารางสต๊อกสินค้า และเมนู ใช้ในการเรียกดูข้อมูลวัตถุดิบจากตารางสต๊อกสินค้า เพื่อมารวมเป็น หนึ่งเมนู ในตารางของเมนู คอลัมน์ stockMenu\_ID ใช้เก็บรหัสสต๊อกเมนู ประเภทข้อมูล Int ไม่อนุญาตให้มีค่าว่าง ให้เป็นคีย์หลัก, คอลัมน์ stockMenu\_Qty ใช้เก็บจำนวนของสต๊อกเมนู ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ stockMenu\_Status ใช้แสดงสถานะของสต๊อกเมนู ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ stockMenu\_TimeStamp ใช้เก็บเวลาที่เข้าถึงสต๊อกเมนู ประเภทข้อมูลเป็น DateTime ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ stock\_ID ใช้เก็บข้อมูล รหัสสต๊อก ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง เป็นคีย์นอกจาก ตารางที่ 3.6 ตารางสต๊อกสินค้า คอลัมน์ stock\_ID, คอลัมน์ Menu\_ID ใช้เก็บข้อมูลรหัสเมนู ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง เป็นคีย์นอกจากตารางที่ 3.7 ตารางเมนูสินค้า และอาหาร คอลัมน์ menu\_ID

**ตารางที่ 3.10** ออเดอร์เมนู (order\_menu)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ชื่อคอลัมน์ | คำอธิบาย | ประเภทข้อมูล | ค่าว่าง | คีย์ | หมายเหตุ |
| order\_ID | รหัสออเดอร์เมนู | Int(11) | No | PK |  |
| order\_Qty | จำนวนออเดอร์เมนู | Int(10) | No |  |  |
| orderMenu\_Status | สถานะออเดอร์เมนู | Varchar(45) | No |  |  |
| orderMenu\_TimeStamp | เวลาที่สั่งเมนู | DateTime | No |  |  |
| menu\_ID | รหัสเมนู | Int(11) | No | FK | menu\_ID (menu\_ID) |
| emp\_ID | รหัสพนักงาน | Int(11) | No | FK | emp\_ID (emp\_ID) |
| totalOrder\_ID | รหัสออเดอร์รวม | Int(11) | No | FK | totalOrder\_ID  (totalOrder\_ID) |

ตารางที่ 3.10 ออเดอร์เมนู ใช้เก็บข้อมูลพนักงาน แต่ละคนสั่งเมนูอะไร คอลัมน์ order\_ID ใช้เก็บรหัสออเดอร์เมนู ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง ให้เป็นคีย์หลัก, คอลัมน์ order\_Qty ใช้เก็บจำนวนออเดอร์เมนูที่สั่ง ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ orderMenu\_Status ใช้เก็บสถานะออเดอร์เมนู ประเภทข้อมูลเป็น Varchar ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ orderMenu\_TimeStamp ใช้เก็บเวลาที่สั่งเมนู ประเภทข้อมูลเป็น DateTime ไม่อนุญาตให้มีค่าว่าง, คอลัมน์ menu\_ID ใช้เก็บข้อมูลรหัสเมนู ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง เป็นคีย์นอกจากตารางที่ 3.7 ตารางเมนูสินค้า และอาหาร คอลัมน์ menu\_ID, คอลัมน์ emp\_ID ใช้เก็บรหัสพนักงาน ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง เป็นคีย์นอกจากตารางที่ 3.1 ตารางพนักงาน คอลัมน์ emp\_ID, คอลัมน์ totalOrder\_ID ใช้เก็บรหัสออเดอร์รวม ประเภทข้อมูลเป็น Int ไม่อนุญาตให้มีค่าว่าง เป็นคีย์นอกจาก ตารางที่ 3.5 ตารางออเดอร์รวม คอลัมน์ totalOrder\_ID

**3.4 ตัวอย่างขั้นตอนการทำงานของระบบ**

การออกแบบหน้าจัดการพนักงานและการจัดการสต๊อกของระบบ คือหน้าของ Admin

ประกอบด้วยหน้าจอเมนูย่อย ๆ ดังนี้

3.4.1หน้าจอจัดการพนักงาน

ส่วนหน้าจอที่ใช้ในการจัดการพนักงาน โดยสามารถ เพิ่ม,ลบ,แก้ไข,ค้นหา ได้

โดยสิทธิ์ที่จะเข้ามาหน้าจัดการพนักงานได้จะต้องเป็น Admin ของระบบเท่านั้น

Graphical user interface

Description automatically generated

**รูปที่ 3.5 หน้าจอส่วนจัดการพนักงาน**

3.4.2 หน้าจอส่วนของการเพิ่มพนักงาน

ส่วนหน้าจอแสดงการเพิ่มพนักงาน เป็นส่วนที่เพิ่มข้อมูลพนักงานต่างๆ เช่น สิทธิ์ของพนักงาน ชื่อ Username Password เบอร์โทร ที่อยู่ Line ของพนักงาน

Table

Description automatically generated

**รูปที่ 3.6 หน้าจอส่วนของการเพิ่มพนักงาน**

3.4.3 หน้าจอส่วนของการแก้ไขพนักงาน

ส่วนหน้าจอแสดงการแก้ไขพนักงาน เป็นส่วนที่แก้ไขข้อมูลพนักงานต่างๆ เช่น สิทธิ์ของพนักงาน ชื่อ Username Password เบอร์โทร ที่อยู่ Line ของพนักงาน โดยที่ดึงข้อมูลที่เคยเพิ่มมาแล้วมาแก้ไข

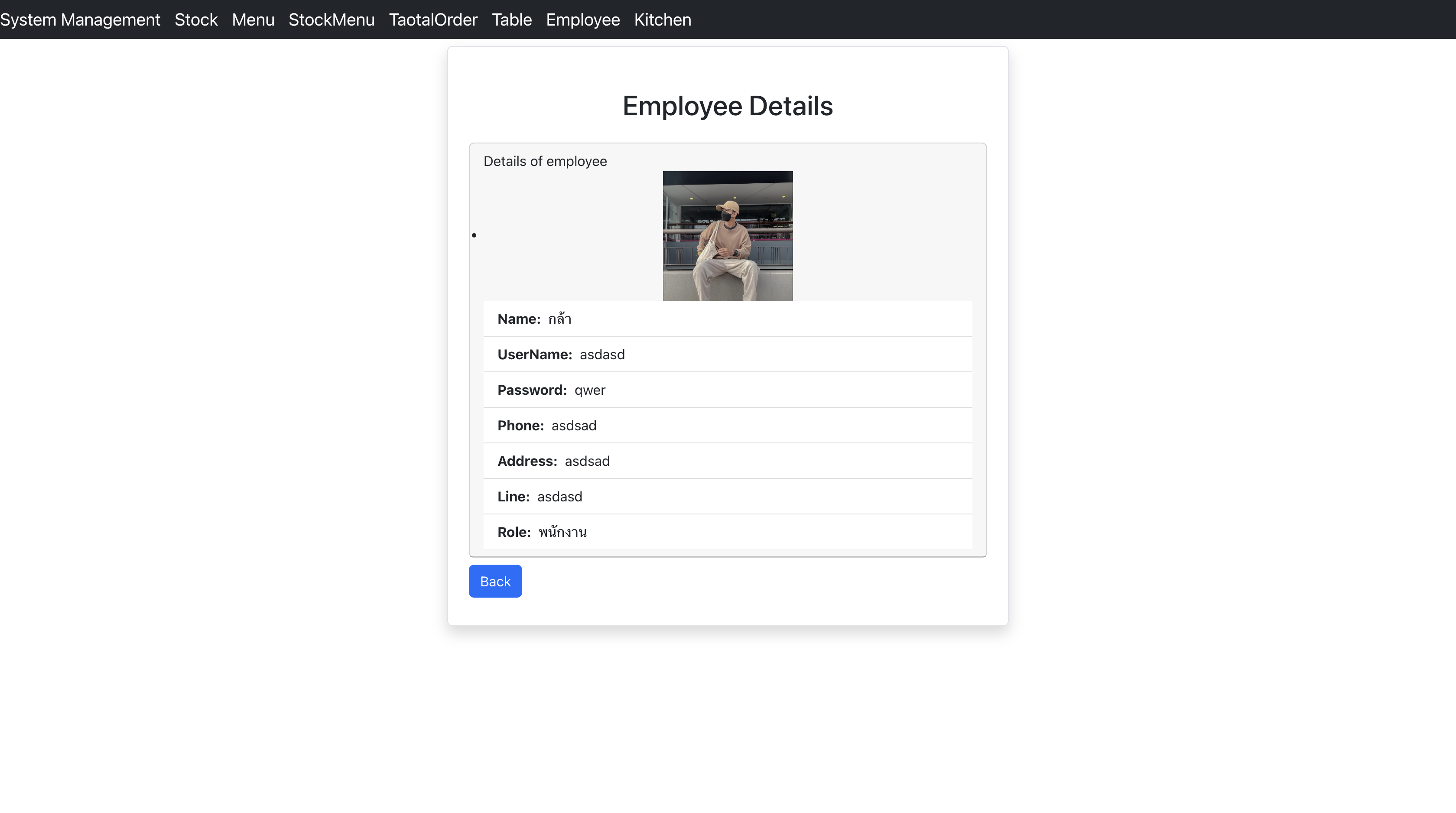
Graphical user interface, application

Description automatically generated

**รูปที่ 3.7 หน้าจอส่วนของการแก้ไขพนักงาน**

3.4.4 หน้าจอส่วนของการแสดงข้อมูลพนักงาน

ส่วนหน้าจอแสดงข้อมูลพนักงาน เป็นส่วนที่แสดงข้อมูลพนักงานต่างๆ เช่น สิทธิ์ของพนักงาน ชื่อ Username Password เบอร์โทร ที่อยู่ Line ของพนักงาน



**รูปที่ 3.8 หน้าจอส่วนของการแสดงข้อมูลพนักงาน**

3.4.5 หน้าจอส่วนของการแสดงสต๊อก

ส่วนหน้าจอแสดงสต๊อก โดยสามารถ เพิ่ม,ลบ,แก้ไข,ค้นหา ได้ และแสดงข้อมูลสต๊อก ต่างๆ เช่น ชื่อประเภท จำนวน ต้นทุน จำนวนการแจ้งเตือนสต๊อกเหลือน้อย วันและเวลา

Graphical user interface, application

Description automatically generated

**รูปที่ 3.9 หน้าจอส่วนของการแสดงสต๊อก**

3.4.6 หน้าจอส่วนของการเพิ่มสต๊อก

ส่วนหน้าจอการเพิ่มสต๊อก เป็นในส่วนการเพิ่มสต๊อกต่างๆ เช่น ชื่อประเภท จำนวน ต้นทุน จำนวนการแจ้งเตือนสต๊อกเหลือน้อย วันและเวลา

**Graphical user interface, application

Description automatically generated**

**รูปที่ 3.10 หน้าจอส่วนของการเพิ่มสต๊อก**

3.4.7 หน้าจอส่วนของการจัดการเมนู

ส่วนหน้าจอแสดงการจัดการเมนู โดยสามารถ เพิ่ม,ลบ,แก้ไข,ค้นหา ได้ และแสดงข้อมูลเมนูต่างๆ เช่น ชื่อเมนู ราคา จำนวน ต้นทุน สถานะ ประเภทเมนู วันและเวลา

**Graphical user interface, application, table

Description automatically generated**

**รูปที่ 3.11 หน้าจอส่วนของการจัดการเมนู**

3.4.8 หน้าจอส่วนของการเพิ่มเมนู

ส่วนหน้าจอการเพิ่มเมนู เป็นในส่วนการเพิ่มเมนูต่างๆ เช่น ชื่อเมนู ราคา จำนวน ต้นทุน สถานะ ประเภทเมนู วันและเวลา

Graphical user interface

Description automatically generated

**รูปที่ 3.12 หน้าจอส่วนของการเพิ่มเมนู**

3.4.9 หน้าจอส่วนของการจัดการสต๊อกเมนู

ส่วนหน้าจอแสดงการจัดการสต๊อกเมนู สามารถดูส่วนผสมของเมนูได้ โดยสามารถ เพิ่ม,ลบ,แก้ไข

ค้นหา ได้ และแสดงข้อมูลสต๊อกเมนูต่างๆ เช่น จำนวน สถานะ วันและเวลา ชื่อเมนู ชื่อสต๊อก

Graphical user interface, application

Description automatically generated

**รูปที่ 3.13 หน้าจอส่วนของการจัดการสต๊อกเมนู**

3.4.10 หน้าจอส่วนของการเพิ่มสต๊อกเมนู

ส่วนหน้าจอการเพิ่มสต๊อกเมนู เป็นในส่วนการเพิ่มสต๊อกเมนูต่างๆ เช่น จำนวน สถานะ วันและเวลา ชื่อเมนู ชื่อสต๊อก

**Graphical user interface, application

Description automatically generated**

**รูปที่ 3.14 หน้าจอส่วนของการเพิ่มสต๊อกเมนู**

3.4.11 หน้าจอส่วนของโต๊ะ

ส่วนหน้าจอแสดงการจัดการโต๊ะ สามารถดูโต๊ะโดยสามารถ เพิ่ม,ลบ,ย้ายโต๊ะ,รวมโต๊ะ

,เปิดโต๊ะ,ค้นหา ได้ และสามารถสั่งอาหารได้

Diagram

Description automatically generated

**รูปที่ 3.15 หน้าจอส่วนของโต๊ะ**

3.4.12 หน้าจอส่วนหน้าแคชเชียร์

ส่วนหน้าจอแสดงการจัดการโต๊ะ สามารถดูโต๊ะโดยสามารถ เพิ่ม,ลบ,ย้ายโต๊ะ,รวมโต๊ะ

,เปิดโต๊ะ,ค้นหา ได้ และสามารถดูสถานะของแต่ละโต๊ะว่าชำระเงินหรือไม่

**Table

Description automatically generated**

**รูปที่ 3.16 หน้าจอส่วนของแคชเชียร์**

3.4.13 หน้าจอส่วนหน้าพนักงานครัว

ส่วนหน้าจอแสดงการจัดการโต๊ะ สามารถดูโต๊ะโดยสามารถ เพิ่ม,ลบ,ย้ายโต๊ะ,รวมโต๊ะ

,เปิดโต๊ะ,ค้นหา ได้ และสามารถดูสถานะของแต่ละโต๊ะว่าชำระเงินหรือไม่

Graphical user interface, table

Description automatically generated

**รูปที่ 3.17 หน้าจอส่วนของแคชเชียร์**

**โปรแกรมที่ 3.1** การสร้างเทเบิลดาต้าเบสด้วย Spring Boot

1. @Entity

2. @Table(name = "employee")

3. public class Employee {

4. @Id

5. @GeneratedValue(strategy = GenerationType.IDENTITY)

6. private long id;

7. @Column(name = "name\_Emp")

8. private String name\_Emp;

9. @Column(name = "username")

10. private String username;

11. @Column(name = "password")

12. private String password;

13. @Column(name = "phone")

14. private String phone;

15. @Column(name = "address")

16. private String address;

17. @Column(name = "line")

18. private String line;

19. public Employee() { }

20. @ManyToOne

21. @JoinColumn(name = "idRole")

22. private Role idRole;

**โปรแกรมที่ 3.2** การสร้างเทเบิลดาต้าเบสด้วย Spring Boot

1. @RequestMapping("/api/v1/")

2. public class EmployeeController {

3. @Autowired

4. private EmployeeRepository employeeRepository;

5. @GetMapping("/employees")

6. public List<Employee> getAllEmployees(){

7. return employeeRepository.findAll(); }

8. @PostMapping("/employees")

9. public Employee createEmployee(@RequestBody Employee employee) {

10. return employeeRepository.save(employee); }

11. @GetMapping("/employees/{id}")

12. public ResponseEntity<Employee> getEmployeeById(@PathVariable Long id) {

13. Employee employee = employeeRepository.findById(id)

14. .orElseThrow(() -> new ResourceNotFoundException("Employee not 15exist with 15. id :" + id));

16. return ResponseEntity.ok(employee);

**โปรแกรมที่ 3.2** การสร้าง API ด้วย Spring Boot

(ต่อ)

17. @PutMapping("/employees/{id}")

18. public ResponseEntity<Employee> updateEmployee(@PathVariable Long id,

19. @RequestBody Employee employeeDetails){

20. Employee employee = employeeRepository.findById(id)

21. .orElseThrow(() -> new ResourceNotFoundException("Employee not

22. exist with id :" + id));

23. employee.setName\_Emp(employeeDetails.getName\_Emp());

24. employee.setUsername(employeeDetails.getUsername());

25. employee.setPassword(employeeDetails.getPassword());

26. employee.setPhone(employeeDetails.getPhone());

27. employee.setAddress(employeeDetails.getAddress());

28. employee.setLine(employeeDetails.getLine());

29. employee.setIdRole(employeeDetails.getIdRole());

30. Employee updatedEmployee = employeeRepository.save(employee);

31. return ResponseEntity.ok(updatedEmployee); }

32. @DeleteMapping("/employees/{id}")

33. public ResponseEntity<Map<String, Boolean>> deleteEmployee(@PathVariable 34. Long id){

35. Employee employee = employeeRepository.findById(id)

36. .orElseThrow(() -> new ResourceNotFoundException("Employee not

37. exist with id :" + id));

38. employeeRepository.delete(employee);

39. Map<String, Boolean> response = new HashMap<>();

40. response.put("deleted", Boolean.TRUE);

41. return ResponseEntity.ok(response); }}

**โปรแกรมที่ 3.3** การสร้างหน้าเรียกใช้ API ด้วย React

1. import axios from 'axios'

2. const EMPLOYEE\_BASE\_REST\_API\_URL = 'http://localhost:8080/api/v1/employees';

3. class EmployeeService{

4. getAllEmployees(){

5. return axios.get(EMPLOYEE\_BASE\_REST\_API\_URL)

6. }

7. createEmployee(employee){

8. return axios.post(EMPLOYEE\_BASE\_REST\_API\_URL, employee)

9. }

10. getEmployeeById(employeeId){

11. return axios.get(EMPLOYEE\_BASE\_REST\_API\_URL + '/' + employeeId);

12. }

13. updateEmployee(employeeId, employee ){

14. return axios.put(EMPLOYEE\_BASE\_REST\_API\_URL + '/' +employeeId,

15. employee );

16. }

17. deleteEmployee(employeeId){

18. return axios.delete(EMPLOYEE\_BASE\_REST\_API\_URL + '/' + employeeId);

19. }

20. }

21. export default new EmployeeService();

**โปรแกรมที่ 3.3** การเรียกใช้ API แสดงข้อมูลพนักงาน ด้วย React

1.const getEmployeeId = (employeeId) => {

2. EmployeeService.getEmployeeById(employeeId).then(response => {

3. setEmployee(response.data);

4. }).catch(error => {

5. console.log('Something went wrong', error);

6. });

7. }

8. EmployeeService.createEmployee(employee).then((response) => {

9. console.log(response.data)

10. console.log(response.data.role)

11. history('/employees');

12. }).catch(error => {

13. console.log(employee)

14. console.log(employee.idRole)

15. console.log(error)

16. })

17. EmployeeService.updateEmployee(id,employee).then((response) => {

18. history('/employees')

19. console.table(employee)

20. }).catch(error => {

21. console.log(employee.idRole)

22. console.log(error)

23. })

24. const deleteEmployee = (employeeId) => {

25. if (window.confirm('Are you sure you want to delete this employee?')) {

26. EmployeeService.deleteEmployee(employeeId).then((response) => {

27. getAllEmployees();

28. }).catch(error => {

29. console.log(error); })}

}

**จากโปรแกรมที่ 3.1** สามารถอธิบายการทำงานได้ดังนี้

บรรทัดที่ 1 : สร้างตาราง

บรรทัดที่ 2 : กำหนดชื่อตาราง

บรรทัดที่ 3 : สร้างคลาส Employee

บรรทัดที่ 4-18 : สร้าง Attribute ในตาราง

บรรทัดที่ 19 : สร้าง Constructor

บรรทัดที่ 20 : สร้าง Relationship ManyToOne

บรรทัดที่ 21 : Join กับ Column ชื่อ idRole

บรรทัดที่ 22 : สร้างตัวแปร

บรรทัดที่ 27 : Attribute ชื่อ idRole

**จากโปรแกรมที่ 3.2** สามารถอธิบายการทำงานได้ดังนี้

บรรทัดที่ 1 : กำหนดการเข้าถึง API

บรรทัดที่ 2 : สร้างคลาส EmployeeController

บรรทัดที่ 3-4 : การกำหนดให้ Attribute ของ Class EmployeeRepository

บรรทัดที่ 5 : กำหนดการเข้าถึง API แบบ GetMapping เพื่อนเรียกดูข้อมูลทั้งหมดใน Data base

บรรทัดที่ 6-7 : เรียกดูข้อมูลทั้งหมด

บรรทัดที่ 8-10 : เพิ่มข้อมูลลงใน Data base

บรรทัดที่ 11-16 : เรียกดูข้อมูลเฉพาะ id

บรรทัดที่ 17-31 : เป็นการ Update ข้อมูลลงใน Data base

บรรทัดที่ 32-41 : ลบข้อมูลใน Data base

**จากโปรแกรมที่ 3.3** สามารถอธิบายการทำงานได้ดังนี้

บรรทัดที่ 1 : import axios

บรรทัดที่ 2 : สร้างตัวแปรเพื่อเก็บ url ของ API

บรรทัดที่ 3: สร้างคลาส EmployeeService

บรรทัดที่ 4-6 : สร้างฟังก์ชัน getAllEmployees เพื่อแสดงข้อมูลทั้งหมด

บรรทัดที่ 7-9 : สร้างฟังก์ชัน createEmployee เพื่อเพิ่มข้อมูลพนักงาน

บรรทัดที่ 10-12 : เสร้างฟังก์ชัน getEmployeeById เพื่อแสดงข้อมูลพนักงาน

บรรทัดที่ 13-16 : สร้างฟังก์ชัน updateEmployee เพื่อแก้ไขข้อมูล

บรรทัดที่ 17-20 : สร้างฟังก์ชัน deleteEmployee เพื่อลบข้อมูล

**จากโปรแกรมที่ 3.3** สามารถอธิบายการทำงานได้ดังนี้

บรรทัดที่ 1-7 : สร้างฟังก์ชันเพื่อเรียกดูข้อมูลพนักงาน โดยเรียกใช้ฟังก์ชัน getEmployeeById เพื่อแสดงที่หน้าเว็บ

บรรทัดที่ 8-16 : สร้างฟังก์ชันเพื่อเพิ่มข้อมูลพนักงาน โดยเรียกใช้ฟังก์ชัน createEmployee

บรรทัดที่ 17-23: สร้างฟังก์ชันเพื่อแก้ไขข้อมูลพนักงาน โดยเรียกใช้ฟังก์ชัน updateEmployee

บรรทัดที่ 24-29 : สร้างฟังก์ชันเพื่อลบข้อมูลพนักงาน โดยเรียกใช้ฟังก์ชัน deleteEmployee

**บทที่ 4**

**การทดลองและผลการดำเนินงาน**

หลังจากได้ทำการออกแบบและสร้างระบบบริหารจัดการร้านถวิล ขั้นตอน ต่อไปคือ การทดลองเพื่อหาประสิทธิภาพการทำงานของระบบบริหารจัดการร้านถวิล ซึ่งการทดลองได้แบ่งออกเป็น 12 ส่วนคือ

4.1 ทดสอบความสามารถตามขอบเขตที่กำหนด

4.2 การทดสอบการใช้งานของระบบ

4.3 การทดสอบระบบสั่งอาหาร

4.4 การทดสอบระบบบันทึกการชำระค่าอาหาร

4.5 การทดสอบระบบรายงานรายรับ-รายจ่ายปัจจุบันและย้อนหลัง รายวัน รายเดือน รายปี

4.6 การทดสอบระบบรายงานวัตถุดิบคงเหลือและต้นทุนปัจจุบันและย้อนหลัง รายวัน รายเดือน รายปี

**4.1 การทดสอบการทํางานของระบบตามขอบเขตที่กําหนด**

การทดสอบการทํางานของระบบตามขอบเขตที่กําหนด เพื่อตรวจสอบและประเมินผล

ความสามารถว่าระบบที่พัฒนามีความสามารถทําตามขอบเขต โดยแสดงรายเอียดของขอบเขต

ที่กําหนด และความสามารถของระบบพร้อมทั้งระบุหัวข้อการทดลองเพื่อแสดงผลการทดลอง

ได้ดังตารางที่ 4.1

**ตารางที่ 4.1** การทดสอบความสามารถตามขอบเขตที่กำหนด

|  |  |  |
| --- | --- | --- |
| ขอบเขต | ความสามารถ | หมายเหตุ |
| จัดการข้อมูลรายการอาหาร และเครื่องดื่ม |  |  |
| จัดการข้อมูลวัตถุดิบ และต้นทุน |  |  |
| รายงานรายรับ-รายจ่ายปัจจุบันและย้อนหลัง รายวัน รายเดือน รายปี |  |  |
| รายงานวัตถุดิบคงเหลือและต้นทุนปัจจุบันและย้อนหลัง รายวัน รายเดือน รายปี |  |  |
| ตรวจสอบสถานะรายการอาหารของแต่ละโต๊ะ |  |  |

**ตารางที่ 4.1** การทดสอบความสามารถตามขอบเขตที่กำหนด (ต่อ)

|  |  |  |
| --- | --- | --- |
| ขอบเขต | ความสามารถ | หมายเหตุ |
| สรุปยอดรายการอาหารขายดี |  |  |
| สามารถแสดงเมนูที่ขายดี และเมนูแนะนำ |  |  |
| ออกใบเสร็จให้กับลูกค้า |  |  |
| บันทึกการชำระค่าอาหาร |  |  |
| ยกเลิกรายการอาหารที่สั่ง |  |  |
| คำนวณค่าอาหาร |  |  |
| จัดการการย้ายโต๊ะ |  |  |
| จัดการการรวมโต๊ะ |  |  |
| รับรายการสั่งอาหาร |  |  |
| เรียกดูข้อมูลรายการอาหาร |  |  |
| เรียกดูข้อมูลสต๊อก |  |  |
| เรียกดูข้อมูลโปรโมชั่น |  |  |
| ดูรายการอาหารที่สั่งตามลำดับ |  |  |
| เปลี่ยนสถานะอาหารที่สั่ง (กำลังทำ พร้อมเสิร์ฟ เสิร์ฟแล้ว ยกเลิก ของเสีย) |  |  |

จากตารางที่ 4.1 สรุปได้ว่า ระบบคุมคุมการให้แสงพืชผ่านแอปพลิเคชัน สามารถทำงานได้ ตามขอบเขตที่ระบุไว้ทุกประการ โดยการทดลองในแต่ละหัวข้อการทดลองหรือตารางผลการทดลอง แต่ละตารางนั้นสะท้อนให้เห็นถึงความสามารถที่ตอบสนองต่อการทำงานตามขอบเขตที่กำหนดไว

**4.2 การทดสอบการใช้งานของระบบ**

ในการทดสอบการใช้งานของระบบบริหารจัดการร้านถวิล สามารถแบ่งการทดสอบตามปัจจัยที่เกี่ยวข้องดังนี้

4.2.1 การทดสอบการใช้งานระบบของแคชเชียร์

การทดสอบการใช้งานระบบของของแคชเชียร์ มีวัตถุประสงค์เพื่อศึกษา

ระบบบริหารจัดการร้านถวิล ในการจัดเก็บรายรับ-รายจ่ายปัจจุบันและย้อนหลัง รายวัน รายเดือน รายปี

ของร้านถวิล เพื่อการทำงานที่เป็นระเบียบ และความสะดวกรวดเร็วให้กับทางร้านถวิล โดยทดลองด้วยการใช้แสดงผลการทดลองดังตารางที่ 4.2

**ตารางที่ 4.2** การทดสอบการใช้งานระบบ

|  |  |  |  |
| --- | --- | --- | --- |
| **รายการทดสอบ** | **ความสามารถในการใช้งาน** | | |
| **ได้** | **ไม่ได้** | **หมาเหตุ** |
| 1.1 สามารถสมัครสมาชิก |  |  |  |
| 1.2 สามารถลงชื่อเข้าสู่ระบบได้ |  |  |  |
| 1.3 สามารถเพิ่ม แก้ไข ค้นหา ลบข้อมูลพนักงาน |  |  |  |
| 1.4 สามารถเพิ่ม แก้ไข ค้นหา ลบข้อมูลสต๊อก |  |  |  |
| 1.5 สามารถเพิ่ม แก้ไข ค้นหา ลบข้อมูลเมนูอาหาร |  |  |  |
| 1.6 สามารถเพิ่ม แก้ไข ค้นหา ลบข้อมูลโต๊ะ |  |  |  |
| 1.7 สามารถเปิดโต๊ะ แก้ไข ค้นหา ลบข้อมูลโต๊ะ |  |  |  |
| 1.8 สามารถรวมโต๊ะ แก้ไขการรวมโต๊ะ ค้นหา ลบข้อมูลการรวมโต๊ะ |  |  |  |

จากตารางที่ 4.2 สามารถอธิบายผลการทดสอบการใช้งานในความสามารถของระบบบริหารจัดการร้านถวิล สามารถทํางานได้ตามที่กําหนดทุกหัวข้อการทดลอง

**4.3 การทดสอบระบบสั่งอาหาร**

ในการทดสอบการใช้งานของระบบบริหารจัดการร้านถวิล สามารถแบ่งการทดสอบตามปัเกี่ยวข้องดังนี้

4.3.1 การทดสอบการใช้งานการทดสอบระบบสั่งอาหาร

การทดสอบการใช้งานระบบของของแคชเชียร์ มีวัตถุประสงค์เพื่อศึกษา

ระบบบริหารจัดการร้านถวิล ในดูรายการอาหาร สั่งอาหาร ยกเลิกรายการอาหารที่สั่ง คำนวณค่าอาหาร เรียกดูข้อมูลรายการอาหาร เรียกดูข้อมูลสต๊อก เรียกดูข้อมูลโปรโมชั่นจัดการการย้ายโต๊ะ จัดการการรวมโต๊ะ ของร้านถวิล เพื่อการทำงานที่เป็นระเบียบ และความสะดวกรวดเร็วให้กับทางร้านถวิล โดยทดลองด้วยการใช้แสดงผลการทดลองดังตารางที่ 4.3

**ตารางที่ 4.3** การทดสอบระบบสั่งอาหาร

|  |  |  |  |
| --- | --- | --- | --- |
| **รายการทดสอบ** | **ความสามารถในการใช้งาน** | | |
| **ได้** | **ไม่ได้** | **หมาเหตุ** |
| 1.1 ดูรายการอาหาร |  |  |  |
| 1.2 สั่งอาหาร |  |  |  |
| 1.3 แก้ไขการสั่งอาหาร |  |  |  |
| 1.4 ยกเลิกการสั่งอาหาร |  |  |  |
| 1.5 ดูรายการอาหารที่สั่ง |  |  |  |
| 1.6 คำนวนค่าอาหาร |  |  |  |

4.4 การทดสอบระบบบันทึกการชำระค่าอาหาร

(\* ยังไม่ทำเลยคิดไม่ออกครับ)

4.5 การทดสอบระบบรายงานรายรับ-รายจ่ายปัจจุบันและย้อนหลัง รายวัน รายเดือน รายปี

(\* ยังไม่ทำเลยคิดไม่ออกครับ)

4.6 การทดสอบระบบรายงานวัตถุดิบคงเหลือและต้นทุนปัจจุบันและย้อนหลัง รายวัน รายเดือน รายปี

(\* ยังไม่ทำเลยคิดไม่ออกครับ)

**บทที่ 5**

**บทสรุปผลการดําเนินงานและข้อเสนอแนะ**

**5.1 สรุปผลการทดลอง**

ปริญญานิพนธ์เรื่องระบบบริหารจัดการร้านถวิล ถูกพัฒนาขึ้นบนเว็บแอปพลิเคชัน สามารถแสดงผลข้อมูลในรูปแบบตาราง / การสั่งอาหาร เพื่ออํานวยความสะดวกในการใช้งานระบบ โดยตัวระบบสามารถแก้ปัญหาความผิดพลาดของการสั่งอาหาร การทำงานที่เป็นระเบียบ และความสะดวกรวดเร็วให้กับทางร้านถวิล และลดความยุ่งยากในการในการจัดการสต๊อก อีกทั้งระบบยังสามารถแจ้งเตือนสต๊อกวัตถุดิบเวลาใกล้จะหมด โดยสามารถสรุปผลได้ดังต่อไปนี้

5.1.1 สรุปผลการทดลองการทดสอบการใช้งานของระบบ ซึ่งจากผลการทดลองของระบบสามารถสรุปผลได้ดังต่อไปนี้

1) สรุปผลการทดสอบการใช้งานระบบของผู้ใช้งานระดับผู้ดูแลระบบ ซึ่งจากการ

ทดลองใช้งานระบบบริหารจัดการร้านถวิล พบว่า มีความสะดวกในการจัดการการรวมโต๊ะ ย้ายโต๊ะ

เช็คบิล สามารถทำได้รวดเร็วขึ้น

2) สรุปผลการทดสอบการใช้งานระบบของผู้ใช้งานระดับพนักงานรับรายการอาหาร ซึ่งจากการทดลองใช้งานระบบบริหารจัดการร้านถวิล พบว่า โปรแกรมมีความสะดวกในการรับรายการอาหารของแต่ละโต๊ะ มีการรับรายการอาหารเป็นลำดับ ไม่ซับซ้อน

3) สรุปผลการทดสอบการใช้งานระบบของผู้ใช้งานระดับพนักงานครัว ซึ่งจากการ

ทดลองใช้งานระบบบริหารจัดการร้านถวิล พบว่า โปรแกรมมีความสะดวกในการดูรายการอาหารที่สั่งมาจากพนักงานรับรายการอาหารแบบเป็นลำดับ

**5.2 ปัญหาและอุปสรรคในการดําเนินงาน**

5.2.1 เนื่องจากระบบบริหารจัดการร้านถวิล ที่คณะผู้จัดทําได้พัฒนาขึ้นต้องการข้อมูลรายการอาหารและวัตถุดิบจริง เลยต้องขออนุญาตจากเจ้าของร้านไปเก็บข้อมูลภายในร้าน เนื่องจากเวลาว่างไม่ตรงกันจึงทำให้เกิดความล่าช้าของงาน

5.2.2 เนื่องจากระบบบริหารจัดการร้านถวิล ที่คณะผู้จัดทําได้พัฒนาขึ้นมีฟังก์ชันหลายอย่างจึงทำให้การทดสอบฟังก์ชันต่าง ๆ เป็นไปด้วยความล่าช้า เพราะว่าเกิดบัคเกิดขึ้นทำให้แก้ต้องแก้เป็นเวลานาน

**5.3 แนวทางการแก้ไขปัญหา**

5.3.1 สมมตเมนูอาหารและเครื่องดื่มขึ้นมาใช้ก่อน

5.3.2 เร่งทำชิ้นงานให้เสร็จ เพื่อจะได้ทดสอบระบบว่าเกิดบัคอะไรขึ้น จะได้แก้ทันเวลา

**5.4 แนวทางในการพัฒนาและข้อเสนอแนะ**

แนวทางการพัฒนาของระบบบริหารจัดการร้านถวิล ผ่านเว็ปแอปพลิเคชั่น ในระดับต่อไปควรมีการพัฒนาระบบดังนี้

5.4.1 พัฒนาในส่วนของหน้าเว็บต้องมีความสวยงาม

5.4.2 ควรพัฒนาระบบให้สามารถแจ้งเตือนวัตถุดิบเหลือน้อย

5.4.3 พัฒนาในส่วนของเมนู Dashboard ให้สามารถรายงานสรุปผลตามช่วง วัน เดือน ปีที่

กําหนด

5.4.4 ควรพัฒนาให้มีการยืนยันบัญชีผู้ใช้งาน สําหรับผู้ใช้งานบุคคลภายนอกที่เข้าสู่ระบบด้วย

การสมัครสมาชิก

5.4.5 พัฒนาในส่วนของการรายงานสรุปผลข้อมูลต่าง ๆ ภายในระบบ อาทิเช่น ข้อมูลการ

พิจารณาอนุมัติการชำระเงิน ข้อมูลผู้ใช้งาน เป็นต้น

**เอกสารอ้างอิง**

[1] www.mindphp.com/คู่มือ/73-คืออะไร/2187-java-javascript-คืออะไร.html

,สืบค้นเมื่อ: 28 กรกฎาคม 2565.

[2] www.mindphp.com/คู่มือ/73-คืออะไร/2185-java-คืออะไร.html

,สืบค้นเมื่อ: 28 กรกฎาคม 2565.  
[3] www.marupatnote.home.blog/2021/08/01/5110/

,สืบค้นเมื่อ: 28 กรกฎาคม 2565.

[4] www.mindphp.com/คู่มือ/73-คืออะไร/2088-sql-คืออะไร.html

,สืบค้นเมื่อ: 28 กรกฎาคม 2565.  
[5] www.blog.rmutl.ac.th/natchasit/files/Lab2\_WebProg\_63.pdf

,สืบค้นเมื่อ: 28 กรกฎาคม 2565.

[6] www.9experttraining.com/articles/บทความ-e-r-diagram-คืออะไร

,สืบค้นเมื่อ: 28 กรกฎาคม 2565.

[7] https://www.9experttraining.com/livetraining

,สืบค้นเมื่อ: 28 กรกฎาคม 2565.