# บทที่ 4 ผลการพัฒนาระบบ (Results)

โปรเจคพัฒนาโปรแกรมจัดการร้านอาหารเดลิเวอรี่ถือว่าสำเร็จลุล่วงได้ดี โดยตัวโปรแกรมสามารถใช้ได้อย่างถูกต้องและตรงกับเป้าหมายที่ผู้ใช้ต้องการ เช่น การจัดเก็บสูตรอาหารได้อย่างแม่นยำ การจัดเก็บราคาอาหารแยกตามแพลตฟอร์ม การบันทึกรายจ่ายได้อย่างละเอียด และมีระบบการค้นหาหลายชั้น การบันทึกคำสั่งซื้อได้สะดวกรวดเร็ว ไม่จำเป็นต้องป้อนค่าจากแป้นพิมพ์ การคำนวณและแสดงผลสถิติที่เจ้าของร้านต้องการ โดยรวมแล้วผู้ใช้นั้นพึงพอใจอย่างมาก ทั้งยังมีความต้องการที่จะนำโปรแกรมนี้ไปใช้งานและต่อยอดพัฒนาร้านค้าต่อไป

## 1. ผลการทดสอบ (Test report)

วิธีการทดสอบที่เลือกใช้คือการทดสอบแบบ Black-Box Testing โดยให้ผู้ที่ทดสอบเป็นที่ไม่มีความรู้ความเข้าใจในเบื้องหลังการทำงานของระบบภายใน (เจ้าของร้าน) เพื่อทดสอบทุกส่วนที่เป็นจุดประสงค์ของการพัฒนาโปรแกรม จากผู้ที่จะนำไปใช้จริง โดยได้ผลการทดสอบดังนี้

|  |  |  |  |
| --- | --- | --- | --- |
| **PROJECT :** | Delivery Restaurant Manager | | |
| **TESTER :** | Thanyaporn Anekawieng | **TEST DATE :** | 25/10/2022 |
| **PREPARED BY :** | Thachin Wisutimarkul | **DOCUMENTED BY :** | Thachin Wisutimarkul |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Login | | **TEST OBJECTIVE :** Authentication test | | |
| **TEST DATA :** email = admin@smiledong.com, password = 00000000 | | | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Empty input |  | Toast message “Invalid credentaisl” | Pass |
| 2 | Empty email | password = 00000000 | Toast message “Invalid credentaisl” | Pass |
| 3 | Empty password | email = admin@smiledong.com | Toast message “Invalid credentaisl” | Pass |
| 4 | Wrong email | email = ad@smiledong.com, password = 00000000 | Toast message “Invalid credentaisl” | Pass |
| 5 | Wrong password | email = admin@smiledong.com, password = 0000 | Toast message “Invalid credentaisl” | Pass |
| 6 | Correct email and password | email = admin@smiledong.com, password = 00000000 | Redirected to dashboard | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Dashboard | | **TEST OBJECTIVE :** Sales report display test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Default display | Display = default  Rider = default  Time = default | Show tables with sales of every rider in span of 1 month | Pass |
| 2 | Select rider | Display = default  Rider = GRAB  Time = default | Show tables with sales of rider grab in span of 1 month | Pass |
| 3 | Select time range | Display = default  Rider = default  Time = Y | Show tables with sales of every rider in span of 1 year | Pass |
| 4 | Toggle graph view | Display = graph  Rider = default  Time = default | Show bar chart with sales of every rider in span of 1 month | Pass |
| 5 | Graph select rider | Display = graph  Rider = GRAB  Time = default | Show bar chart with sales of rider grab in span of 1 month | Pass |
| 6 | Graph select time range | Display = graph  Rider = default  Time = Y | Show graph with sales of every rider in span of 1 year | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Order | | **TEST OBJECTIVE :** Add new order test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Empty input |  | Nothing happened | Pass |
| 2 | Select item without select rider |  | Unable to click item therefore nothing happened | Pass |
| 3 | Only send time data | Time = 25/10/2022 | Nothing happened | Pass |
| 4 | Only select rider | Rider = Grab | Nothing happened | Fail :  Saved with empty items |
| 5 | Select rider then select item | Item = pickled shrimp  Rider = Grab | Saved with Grab as rider and 1 pickled shrimp as items | Pass |
| 6 | Select rider then select item 3 times | Item = pickled shrimp  Rider = Grab | Saved with Grab as rider and 3 pickled shrimp as items | Pass |
| 7 | Select rider then select several items | Item = [pickled shrimp, rice, seaweed]  Rider = Lineman | Saved with Lineman as rider and 1 pickled shrimp, 1 rice, 1 seaweed as items | Pass |
| 8 | 1. Select rider 2. Select several items 3. Select first item 2 more times | Item = [pickled shrimp, rice, seaweed]  Rider = Lineman | Saved with Lineman as rider and 2 pickled shrimp, 1 rice, 1 seaweed as items | Pass |
| 9 | 1. Select rider 2. Select several items 3. Reduce first item by 1 | Item = [pickled shrimp, rice, seaweed]  Rider = Lineman | Saved with Lineman as rider and 1 rice, 1 seaweed as items | Pass |
| 10 | 1. Select rider 2. Select several items 3. Select first item 4 more times 4. Reduce first item by 2 | Item = [pickled shrimp, rice, seaweed]  Rider = Lineman | Saved with Lineman as rider and 3 pickled shrimp, 1 rice, 1 seaweed as items | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Order | | **TEST OBJECTIVE :** Delete order test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click delete icon on order |  | Order removed from table | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Expense Category | | **TEST OBJECTIVE :** Add new category test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Empty input |  | Nothing happened | Pass |
| 2 | Empty name | Parent = Business | Nothing happened | Pass |
| 3 | Empty parent | Name = testing | Add new main category name ‘testing’ | Pass |
| 4 | Duplicate name | Name = Business | Nothing happened | Pass |
| 5 | Select Business as parent | Name = testing  Parent = Business | Add new sub category name ‘testing’ under ‘Business’ category | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Expense Category | | **TEST OBJECTIVE :** Delete category test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Delete sub-category |  | Category removed from the list | Pass |
| 2 | Delete main category |  | All sub-category under main category and the main category itself is removed from the list | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Expense | | **TEST OBJECTIVE :** Add new expense test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Empty input  Save & Go Back |  | Toast message with error that occurred | Pass |
| 2 | Only send category input  Save & Go Back | Category = ingredient | Redirect to expense table  Toast message with error that occurred | Pass |
| 3 | Only send description input  Save & Go Back | Description = testing | Redirect to expense table  Toast message with error that occurred | Pass |
| 4 | Only send amount input  Save & Go Back | Amount = 5 | Redirect to expense table  Toast message with error that occurred | Pass |
| 5 | Only send total input  Save & Go Back | Total = 200 | Redirect to expense table  Toast message with error that occurred | Pass |
| 6 | Only send date&time input  Save & Go Back | Date&Time =10/04/2022 00:00 | Redirect to expense table  Toast message with error that occurred | Pass |
| 7 | Select category ingredient  Save & Go Back | Category = ingredient | Redirect to expense table  Description input form turn into searchable drop-down input with ingredient list | Pass |
| 8 | Select category packaging  Save & Go Back | Category = packaging | Redirect to expense table  Description input form turn into searchable drop-down input with packaging list | Pass |
| 9 | Select category with sub-categogy | Category = Business | New searchable drop-down input appear beside category input call Sub-category input with Business’s sub category list | Pass |
| 10 | Select ingredient category  Select ingredient  Empty amount, total and date  Save & Go Back | Category = ingredient  Ingredient = Shrimp | Redirect to expense table  Toast message with error that occurred | Pass |
| 11 | Select ingredient category  Select ingredient  Enter amount  Empty total and date  Save & Go Back | Category = ingredient  Ingredient = Shrimp  Amount = 100 | Redirect to expense table  Toast message with error that occurred | Pass |
| 12 | Select ingredient category  Select ingredient  Enter amount  Enter total  Empty date  Save & Go Back | Category = ingredient  Ingredient = Shrimp  Amount = 100  Total = 345 | Redirect to expense table  Saved as Ingredient category, Descript as Shrimp, Amount 100, Total 345 and Date&Time is current time. | Pass |
| 13 | Fill every input  Save & Go Back | Category = ingredient  Ingredient = Shrimp  Amount = 100  Total = 345  Date&Time = 05/10/2022, 00:00 | Redirect to expense table  Saved as Ingredient category, Descript as Shrimp, Amount 100, Total 345 and Date&Time is 05/10/2022, 00:00 | Pass |
| 14 | Fill every input  Save & Add new expense | Category = ingredient  Ingredient = Shrimp  Amount = 100  Total = 345  Date&Time = 05/10/2022, 00:00 | Saved as details  Clear every input except Category and Date&Time | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Expense | | **TEST OBJECTIVE :** Delete expense test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click delete icon on expense |  | Expense removed from table | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Expense | | **TEST OBJECTIVE :** Repeat expense test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click duplicate icon on expense |  | New expense add to the table with same details as select order except Date; which save as current time  Toast message “Create successfully” | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Expense | | **TEST OBJECTIVE :** Edit expense test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click edit icon on expense | Category = Equipment  Description = Glove  Amount = 50  Total = 49  Date&Time = 10/27/2022 16:36 | Open edit form with current expense data | Pass |
| 2 | Modify input  Save & Go back | Description = Scissors  Amount = 2  Total = 69 | Redirect to expenses table  Glove expense at 10/27/2022 16:36 change to Scissors with amount of 2 and total of 69 | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Materials | | **TEST OBJECTIVE :** Add new material test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Empty input  Click ADD |  | Toast message with error that occurred | Pass |
| 2 | Empty unit  Click ADD | Name = test | Toast message with error that occurred | Pass |
| 3 | Empty name  Click ADD | unit = testUnit | Toast message with error that occurred | Pass |
| 4 | Fill all input  Click ADD | Name = test  Unit = testing  Type = materials | Add new material name test, type materials, unit testing into materials table | Pass |
| 5 | Duplicate name  Click ADD | Name = shrimp  Unit = testing  Type = materials | Toast message with error that occurred | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Materials | | **TEST OBJECTIVE :** Edit material test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click edit icon on material | Name = test  Type = Packaging  Unit = testing | Open edit form with current material data | Pass |
| 2 | Modify input  Save | Name = test2  Type = Ingredient  Unit = testing2 | test material change to test2 with type Ingredient and unit testint2 | Pass |
| 3 | Modify to duplicate name and Save | Name = Shrimp | Toast message with error that occurred | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Materials | | **TEST OBJECTIVE :** Delete material | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click delete icon on material |  | Material removed from table | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Items | | **TEST OBJECTIVE :** Add new item | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Save with empty input |  | Nothing happen | Pass |
| 2 | Save with only name | Name = test | Redirect to tables  New Item name test appear in the table | Pass |
| 3 | Enter name  Add new material  Empty material input  Save | Name = test | Nothing happen | Pass |
| 4 | Enter name  Add material with default amount  Save | Name = test  Material = Shrimp | Save with name test  and Shrimp amount 1 | Pass |
| 5 | Add several materials and save | Name = test  Material = [Shrimp, Salmon]  Amount = [1,10] | Save with name test  1 Shrimp, 10 Salmon, | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Items | | **TEST OBJECTIVE :** Edit item test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click edit icon on material | Name = test  Material = [Shrimp, Salmon]  Amount = [1,10] | Open edit form with current item data | Pass |
| 2 | Modify input  Save | Name = DoubleTrouble  Material = [Shrimp, Salmon]  Amount = [5,45] | test item name change to DoubleTrouble  Shrimp amount change to 5  Salemon amount change to 45 | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Items | | **TEST OBJECTIVE :** Delete item | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click delete icon on item |  | item removed from table | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Itemsets | | **TEST OBJECTIVE :** Add new itemsets | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Save with empty input |  | Nothing happen | Pass |
| 2 | Save with only name | Name = test | Redirect to tables  New Item name test appear in the table | Pass |
| 3 | Enter name  Add new item  Empty item input  Save | Name = test | Nothing happen | Pass |
| 4 | Enter name  Add item with default amount  Save | Name = test  Material = Pickled Shrimp | Save with name test Pickled Shrimp amount 1 workcost 0 and gascost 0 | Pass |
| 5 | Add several items and save | Name = test  Material = [PickledShrimp, PickledSalmon]  Amount = [1,10] | Save with name test  1 PickledShrimp, 10 PickledSalmon, workcost 0 and gascost 0 | Pass |
| 6 | Add several items, fill every inputs and save | Name = test  Material = [PickledShrimp, PickledSalmon]  Amount = [1,10]  workCost = 20  gasCost = 6 | Save with name test  1 PickledShrimp, 10 PickledSalmon, workCost 20 and gasCost 6 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Itemsets | | **TEST OBJECTIVE :** Edit itemset test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click edit icon on material | Name = test  Material = [PickledShrimp, PickledSalmon]  Amount = [1,10]  workCost = 20  gasCost = 6 | Open edit form with current item data | Pass |
| 2 | Modify input  Save | Name = MeatMeat  Amount = [2,2] | test item name change to MeatMeat  PickledShrimp amount change to 2  PickledSalmon amount change to 2 | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Itemsets | | **TEST OBJECTIVE :** Delete itemset test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click delete icon on itemset |  | itemset removed from table | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Riders&Pricing | | **TEST OBJECTIVE :** Add new rider test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click + icon |  | Rider create form open | Pass |
| 2 | Click - icon |  | Rider create form close without saving | Pass |
| 3 | Save with empty input |  | Toast message with error that occurred | Pass |
| 4 | Save with only name input | Name = test | Toast message with error that occurred | Pass |
| 5 | Save with only fees input | Fees = 30 | Toast message with error that occurred | Pass |
| 6 | Save with only tax input | Tax = 7 | Toast message with error that occurred | Pass |
| 7 | Save with only gas cost input | GasCost = 1 | Toast message with error that occurred | Pass |
| 8 | Save without gas cost input | Name = test  Fees = 30  Tax = 7 | Save test to rider database with fees 30, tax 7 and gasCost 0  test appear on rider list.  When access test; prices table show every menu with 0 in price | Pass |
| 9 | Save with every input filled | Name = test  Fees = 30  Tax = 7  GasCost = 1 | Save test to rider database with fees 30, tax 7 and gasCost 1  test appear on rider list.  When access test; prices table show every menu with 0 in price | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Riders&Pricing | | **TEST OBJECTIVE :** Delete rider test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click + icon |  | Rider create form open | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Riders&Pricing | | **TEST OBJECTIVE :** Set menu price test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Double click on price number |  | Enter price editing mode | Pass |
| 2 | Enter price in string | Price = abcd | Nothing happened | Pass |
| 3 | Enter price in number | Price = 100 | Price save as 100 | Pass |

## 2. ผลลัพธ์ของระบบที่พัฒนาในภาพรวม (Output)

Diagram, schematic

Description automatically generated

โดยรวมแล้วระบบสามารถรวบรวมกระบวนการทำงานทั้งหมดไว้ในแอพพลิเคชั่นเดียวได้สำเร็จ ไม่มีความจำเป็นต้องแยกการทำงานไปไว้ในหลายระบบหลายอุปกรณ์เหมือนระบบเก่าที่ทางร้านเคยใช้งาน

ผู้ใช้สามารถเก็บข้อมูลสูตรอาหารและวัตถุดิบต่างๆที่ใช้ภายในร้านได้อย่างสมบูรญ์

A screenshot of a computer

Description automatically generated with medium confidence

แต่ละแพลตฟอร์มจะมีราคาของทุกๆเมนูที่แตกต่างกัน สามารถปรับแต่งได้ตามที่ผู้ใช้ต้องการ แบบที่แอพพลิเคชั่นจัดการคำสั่งซื้อทั่วไปไม่สามารถทำได้ง่ายๆ

A screenshot of a computer

Description automatically generated with low confidence

ส่วนการลงรายจ่ายภายในร้านมีระบบช่วยเหลือให้เลือกวัตถุดิบได้ง่ายขึ้น สามารถแบ่งหมวดหมู่ได้อย่างชัดเจน สามารถสร้างหมวดหมู่ย่อยเพื่อความแม่นยำในการบันทึกรายจ่าย ระบบยังมีคำสั่งในการทำซ้ำคำสั่งซื้อเพื่อความรวดเร็วในการเพิ่มรายจ่ายที่มีการเพิ่มบ่อยๆอีกด้วย

Graphical user interface

Description automatically generated

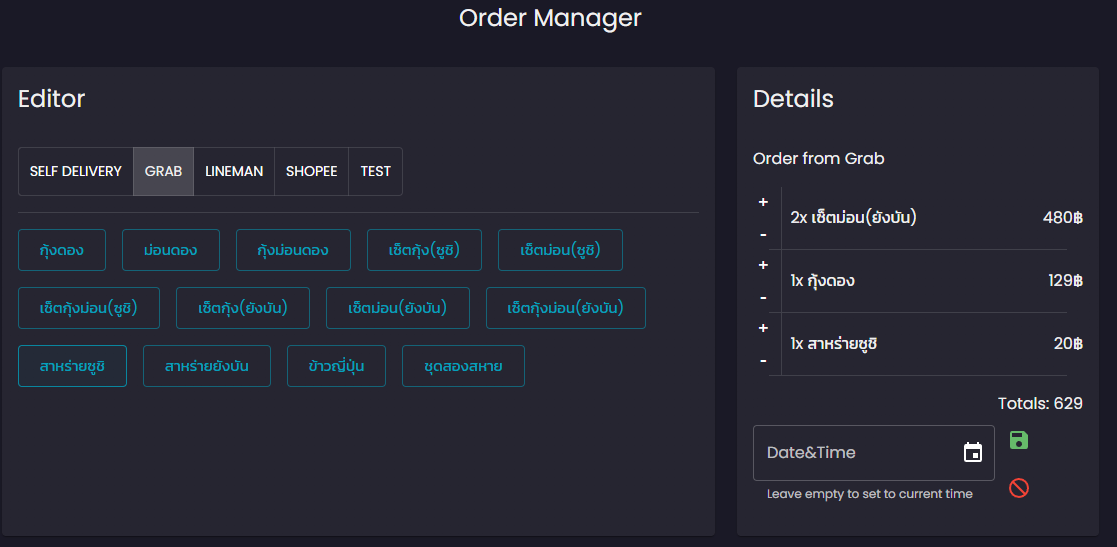
A screenshot of a computer

Description automatically generated with medium confidence

ในส่วนของการแสดงรายจ่าย ระบบสามารถคัดกรองข้อมูลได้อย่างละเอียด ค้นหาตามชื่อ ค้นหาตามประเภท ค้นหาตามจำนวน ฯลฯ รวมถึงยังสามารถจัดเรียงข้อมูลได้ตามต้องการอีกด้วย

A screenshot of a computer screen

Description automatically generated with medium confidence

ในการบันทึกคำสั่งซื้อ ผู้ใช้สามารถบันทึกคำสั่งซื้อได้โดยโดยง่ายเพียงเลือกแพลตฟอร์ม เลือกเมนูอาหารและกดบันทึก ทำให้กระบวนการในการบันทึกคำสั่งซื้อเป็นไปอย่างรวดเร็วและง่ายดายกว่าระบบเก่าของทางร้านมาก

ระบบมีการแสดงสถิติการขายที่รวบรวมผลการขายของทุกแพลตฟอร์มไว้ด้วยกัน และยังสามารถกำหนดช่วงระยะเวลาที่ต้องการดูสถิติได้อีกด้วย

A screenshot of a computer

Description automatically generated with medium confidence

## 3. ผลประเมินความพึงพอใจของผู้ใช้ระบบ (Assessment)

พนักงานร้าน

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ข้อคำถาม | ระดับความพึ่งพอใจ | | | | |
| น้อยที่สุด | น้อย | ปานกลาง | มาก | มากที่สุด |
| 1. ความง่ายในการใช้งาน |  |  |  | / |  |
| 1. กระบวนการทำงานของระบบ |  |  | / |  |  |
| 1. ความถูกต้องของระบบ |  |  |  |  | / |
| 1. ตรงตามวัตถุประสงค์ที่ต้องการ |  |  |  |  | / |
| 1. การออกแบบเข้าใจง่าย ไม่ซับซ้อน |  |  |  | / |  |
| 1. ความสะดวกในการใช้งาน |  |  |  | / |  |
| 1. ความเหมาะสมในการใช้งาน |  |  |  | / |  |
| 1. ความพึงพอใจในการใช้งาน |  |  |  | / |  |
| 1. ประโยชน์ในการใช้งาน |  |  |  | / |  |

เจ้าของร้าน

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ข้อคำถาม | ระดับความพึ่งพอใจ | | | | |
| น้อยที่สุด | น้อย | ปานกลาง | มาก | มากที่สุด |
| 1. ความง่ายในการใช้งาน |  |  |  |  | / |
| 1. กระบวนการทำงานของระบบ |  |  | / |  |  |
| 1. ความถูกต้องของระบบ |  |  |  |  | / |
| 1. ตรงตามวัตถุประสงค์ที่ต้องการ |  |  |  |  | / |
| 1. การออกแบบเข้าใจง่าย ไม่ซับซ้อน |  |  |  | / |  |
| 1. ความสะดวกในการใช้งาน |  |  | / |  |  |
| 1. ความเหมาะสมในการใช้งาน |  |  |  | / |  |
| 1. ความพึงพอใจในการใช้งาน |  |  |  | / |  |
| 1. ประโยชน์ในการใช้งาน |  |  |  |  | / |

## 4. คู่มือการติดตั้ง (Installation)

### 4.1 คุณสมบัติของเครื่องคอมพิวเตอร์ที่สามารถติดตั้งโปรแกรมนี้ได้ (Hardware และ Software requirement)

**Hardware**

1. Server
   1. CPU 3GHz or faster with 4 or more cores
   2. At least 4GB Memory
   3. Hard disk free space at least 1.2GB
2. Client
   1. Pentium 4 processor or higher
   2. At least 1GB Memory
   3. Hard disk free space at least 100MB

**Software**

1. Server
   1. Linux
   2. SmartOS
   3. macOS 10.7 or higher
   4. Windows Server 2008 or higher
   5. Node.js 16.0 or higher
   6. NPM 8.0 or higher
   7. Git 2.37 or higher
2. Client
   1. Windows 7, Windows 8, Windows 8.1, Windows 10 or later
   2. macOS High Sierra 10.13 or later
   3. 64-bit Ubuntu 18.04+, Debian 10+, openSUSE 15.2+, or Fedora Linux 32+
   4. Android 7.0 Nougat or later

### 4.2 รายละเอียดการติดตั้ง

0. จัดเตรียมฐานข้อมูล MongoDB ผู้ใช้สามารถเลือกที่จะใช้ local server ผ่าน community server หรือใช้บริการ MongoDB Atlas ก็ได้ โดยสามารถศึกษาเพิ่มเติมได้ที่ https://www.mongodb.com/docs/atlas/getting-started/

1. ตรวจสอบให้แน่ใจว่า server สามารถใช้ Node.js และ git ได้

2. เปิด command prompt หรือ terminal ในตำแหน่งที่ต้องการติดตั้ง

3. คัดลอกคำสั่งต่อไปนี้ลงใน command prompt หรือ terminal ที่ใช้แล้วรันคำสั่ง

git clone https://github.com/nickethachin/DeliveryRestaurantManager.git --branch CS499 && cd DeliveryRestaurantManager && npm install --force && npm install –prefix frontend –force && npm run build –prefix frontend



4. รอจนการติดตั้งโปรแกรมเสร็จสมบูรญ์

5. สร้างไฟล์ใหม่ภายในโฟลเดอร์ของโปรแกรม ตั้งชื่อว่า “.env”

6. แก้ไขไฟล์ .env ด้วยเนื้อหาดังนี้

NODE\_ENV = production

PORT = พอร์ตที่ต้องการให้โปรแกรมรัน

MONGO\_URI = URI สำหรับเชื่อมต่อกับ MongoDB

JWT\_SECRET = รหัสสำหรับสร้างและตรวจสอบ login Token

6.1 ตัวอย่างไฟล์ .env

Text

Description automatically generated

7. จากนั้นสามารถรันคำสั่ง npm start ได้ เมื่อต้องการรันเซิฟเวอร์



8. รอจนโปรแกรมทำงานจนเสร็จสมบูรญ์

Graphical user interface, text

Description automatically generated

9. เมื่อเซิฟเวอร์ทำงานแล้ว เราสามารถเข้าถึงโปรแกรมได้โดยการเข้าผ่าน Web browser และใช้ url ของเครื่อง และ port ที่กำหนด ในกรณีที่เครื่องที่รันเซิฟเวอร์กับเครื่องที่จะเข้าใช้โปรแกรมเป็นเครื่องเดียวกัน และ port ที่กำหนดคือ 5000 ผู้ใช้สามารถเข้าถึงโปรแกรมได้ที่ http://localhost:5000

Graphical user interface, application

Description automatically generated

10. เมื่อเข้ามาแล้วระบบจะต้องการให้เราล็อกอิน ให้เราไปที http://localhost:5000/register เพื่อสมัครสมาชิกก่อน โดยช่องแรกให้กรอกชื่อที่ต้องการ ช่องที่สองกรอกอีเมล์ ช่องสามและสี่กรอกพาสเวิร์ดที่ต้องการ จากนั้นกดปุ่ม Submit ด้านล่าง

Graphical user interface, application

Description automatically generated

11. เมื่อสมัครสมาชิกสำเร็จและเข้าสู่หน้า Dashboard ได้ ก็ถือได้ว่าติดตั้งเสร็จสมบูรญ์แล้ว

## 5. คู่มือการใช้งาน (User manual)

1. เมื่อเข้าใช้งานครั้งแรก ผู้ใช้จะต้องเข้าสู่ระบบก่อน ช่องแรกใส่ email และช่องที่สองใส่ password ที่ได้สมัครไว้ในตอนติดตั้งโปรแกรม

Graphical user interface, application

Description automatically generated

2. จากนั้น ผู้ใช้จะต้องตั้งค่าเมนูอาหาร ไรเดอร์ และราคาอาหาร โดยกระบวนการนี้ จะทำเพียงครั้งเดียวเท่านั้น เว้นเสียแต่ว่า จะมีการปรับราคาหรือสูตรอาหาร

2.1 คลิ๊กไปที่เมนู Material ทางฝั่งซ้ายของโปรแกรม

Graphical user interface, application

Description automatically generated

2.2 จากนั้น ทำการเพิ่มวัตถุดิบภายในร้านทีละอย่าง สามารถทำได้โดยการกรอกลงไปภายในช่องข้อมูลด้านบน จากซ้ายไปขวา ช่องที่ต้องกรอกจะเป็น ชื่อ ประเภท และหน่วย แล้วกดปุ่ม ADD

ชื่อ คือชื่อของวัตถุดิบที่จะเพิ่ม

ประเภท คือ ประเภทว่าเป็น Ingredient (วัตถุดิบ) หรือ Packaging (บรรจุภัณฑ์)

หน่วย คือ หน่วยเรียกของวัตถุดิบนั้นๆ เช่น ชิ้น ตัว กรัม ขวด มิลลิกรัม A screenshot of a computer

Description automatically generated with medium confidence

2.2.1 เมื่อเพิ่มสำเร็จข้อมูลจะถูกเพิ่มไปยังตารางด้านล่าง

2.2.2 ผู้ใช้สามารถกดปุ่มดินสอสีเขียวทางด้านขวาเพื่อแก้ไขข้อมูลได้

2.2.3 ผู้ใช้สามารถกดปุ่มถังขยะสีแดงทางด้านขวาเพื่อลบข้อมูลได้

2.3 ทำซ้ำไปเรื่อยๆจนกว่าจะลงทะเบียนวัตถุดิบและบรรจุภัณฑ์ที่ใช้ภายในร้านจนหมด

2.4 ไปที่เมนู Item ทางด้านซ้ายของโปรแกรม Graphical user interface, application

Description automatically generated

2.5 กดปุ่ม Create ทางด้านขวาเพื่อเริ่มสร้างอาหาร Graphical user interface, application

Description automatically generated

2.6 สร้างอาหารโดยกรอกข้อมูลดังนี้

2.6.1 ช่องว่างช่องแรก กรอกชื่ออาหารที่ต้องการ

2.6.2 กรอกวัตถุดิบที่ใช้

2.6.2.1 กดปุ่ม  เพื่อเพิ่มวัตถุดิบ

2.6.2.2 ในช่อง Material สามารถกดปุ่มสามเหลี่ยมเล็กๆหรือพิมพ์หาของที่ใช้

Graphical user interface

Description automatically generated

2.6.2.2.1 หากไม่มีวัตถุดิบที่ต้องการ กลับไปทำข้อ 2.2

2.6.2.3 ในช่องจำนวน ให้กรอกจำนวนของที่ใช้ต่ออาหาร 1 ชุด

2.6.2.4 สามารถกดปุ่ม  เพื่อเปิดเครื่องคิดเลขได้

2.6.3 กดปุ่ม  ทางด้านขวา เพื่อบันทึกอาหาร

2.6.4 เมื่อเพิ่มสำเร็จข้อมูลจะถูกเพิ่มไปยังตาราง

2.6.5 ผู้ใช้สามารถกดปุ่มดินสอสีเขียวทางด้านขวาเพื่อแก้ไขข้อมูลได้

2.6.6 ผู้ใช้สามารถกดปุ่มถังขยะสีแดงทางด้านขวาเพื่อลบข้อมูลได้

2.6.7 ทำซ้ำจนมีอาหารเท่าที่ต้องการ

2.7 สร้างเมนูอาหารโดยกรอกข้อมูลดังนี้

2.7.1 ช่องว่างช่องแรก กรอกชื่อเมนูอาหารที่ต้องการ

2.7.2 ช่อง Work Cost กรอกต้นทุนค่าแรงที่ใช้ในการทำเมนูนี้

2.7.3 ช่อง Gas Cost กรอกต้นทุนค่าน้ำมันรถที่ใช้ในการทำเมนูนี้

2.7.4 กรอกอาหารที่ใช้ประกอบเมนูนี้

2.7.4.1 กดปุ่ม  เพื่อเพิ่มวัตถุดิบ

2.7.4.2 ในช่อง Item สามารถกดปุ่มสามเหลี่ยมเล็กๆหรือพิมพ์หาของที่ใช้

A screenshot of a computer

Description automatically generated with medium confidence

2.7.4.2.1 หาไม่มีอาหารที่ต้องการ กลับไปทำข้อ 2.6

2.7.4.3 ในช่องจำนวน ให้กรอกจำนวนของที่ใช้ต่อเมนู 1 เมนู

2.7.4.4 สามารถกดปุ่ม  เพื่อเปิดเครื่องคิดเลขได้

2.7.5 กดปุ่ม  ทางด้านขวา เพื่อบันทึกอาหาร

2.7.6 เมื่อเพิ่มสำเร็จข้อมูลจะถูกเพิ่มไปยังตาราง

2.7.7 ผู้ใช้สามารถกดปุ่มดินสอสีเขียวทางด้านขวาเพื่อแก้ไขข้อมูลได้

2.7.8 ผู้ใช้สามารถกดปุ่มถังขยะสีแดงทางด้านขวาเพื่อลบข้อมูลได้

2.8 ไปที่เมนู Rider&Pricing เพื่อตั้งค่าไรเดอร์ Graphical user interface, application

Description automatically generated

2.8.1 กดปุ่ม  ทางซ้ายเพื่อเปิดเมนูสร้างไรเดอร์

A screenshot of a computer

Description automatically generated with medium confidence

2.8.2 กรอกข้อมูลดังนี้ เพื่อสร้างไรเดอร์

Name: ชื่อไรเดอร์

Fees: ค่าธรรมเนียม (%)

Tax: ภาษีค่าธรรมเนียม (%)

Gas Cost: ต้นทุนค่าน้ำมันสำหรับทุกเมนูในไรเดอร์นี้

2.8.3 กดปุ่ม  ทางขวาของหน้าจอเพื่อบันทึกไรเดอร์ใหม่

2.8.4 ทำซ้ำจนได้ไรเดอร์ครบที่ต้องการ

2.9 ตั้งค่าราคา

2.9.1 เลือกไรเดอร์ที่ต้องการแก้ไขราคา

A screenshot of a computer

Description automatically generated with medium confidence

2.9.2 ดับเบิ้ลคลิ๊กที่ราคาของเมนูที่ต้องการแก้ไข

Application

Description automatically generated with low confidence

2.9.3 แก้ไขราคาเป็นราคาที่ต้องการแล้วกด Enter หรือคลิ๊กที่พื้นที่ว่างข้างนอก

2.9.4 ทำซ้ำจนแก้ไขราคาที่ต้องการจนครบ

2.10 ตั้งค่าหมวดหมู่รายจ่าย

2.10.1 ไปที่เมนู Expense Graphical user interface, application

Description automatically generated

2.10.2 กดที่ Category Edit ทางด้านขวาของหน้าจอ

A screenshot of a computer screen

Description automatically generated with medium confidence

2.10.3 หากต้องการเพิ่มหมวดหมู่หลัก ให้ใส่ชื่อหมวดหมู่ที่ช่อง Name แล้วกด 

2.10.4 หากต้องการเพิ่มหมวดหมู่รอง

2.10.4.1 กรอกชื่อที่ต้องการ

2.10.4.2 เลือกหมวกหมู่หลักจากช่อง Parent

2.10.4.3 กดบันทึก 

2.10.5 ผู้ใช้สามารถแก้ไขหมวดหมู่ได้โดยการกดที่ชื่อหมวดหมู่ด้านล่าง

2.10.6 ผู้ใช้สามารถลบหมวดหมู่ได้โดยการกด  ทางด้านขวา

2.11 เสร็จสิ้นขั้นตอนการตั้งค่าร้านค้าครั้งแรก

3. การจัดการรายจ่าย

3.1 ไปที่เมนู Expense Graphical user interface, application

Description automatically generated

3.2 กดปุ่ม Create ทางด้านขวาของหน้าจอเพื่อเพิ่มรายจ่าย

Graphical user interface, application

Description automatically generated

3.2.1 เลือกหมวดหมู่ที่ช่อง Category

3.2.1.1 กรณีที่หมวดหมู่ที่เลือกมีหมวดหมู่รอง สามารถเลือกหมวดหมู่รองได้

3.2.2 เลือก material ในกรณีที่เลือกหมวดหมู่เป็น Ingredient หรือ Packaging

3.2.3 กรอกชื่อหรือคำอธิบายเมื่อเลือกหมวดหมู่อื่น

3.2.4 กรอกจำนวนที่ซื้อ

3.2.5 กรอกยอดรวมทั้งหมด

3.2.6 สามารถเลือกวันเวลาทึ่ค่าใช้จ่ายเกิดขึ้นได้ หากปล่อยว่างจะใช้วันเวลาปัจจุบัน

3.2.7.1 กด Save & Add new expense เพื่อบันทึกและเตรียมสร้างรายการถัดไป

3.2.7.2 กด Save & Go back เพื่อบันทึกและกลับไปยังหน้าตารางรายการ

3.3 ผู้ใช้สามารถกดปุ่ม  เพื่อแก้ไขรายการที่เลือกได้

3.4 ผู้ใช้สามารถกด  เพื่อสร้างรายการใหม่จากข้อมูลของรายการที่เลือก

3.5 ผู้ใช้สามารถกด  เพื่อลบรายการที่เลือกได้

4. การจัดการคำสั่งซื้อ

4.1 กดที่ Order เมนูทางด้านซ้ายของโปรแกรม Graphical user interface, application

Description automatically generated

4.2 สร้างคำสั่งซื้อใหม่

4.2.1 เลือกไรเดอร์ที่ต้องการ

Graphical user interface

Description automatically generated with medium confidence

4.2.2 คลิ๊กซ้ายที่เมนูที่ต้องการเพื่อเพิ่มเมนูเข้าสู่คำสั่งซื้อ

Graphical user interface

Description automatically generated

4.2.3 ผู้ใช้สามารถกดที่เมนูซ้ำได้เพื่อเพิ่มจำนวนเมนูเดิมลงในคำสั่งซื้อ

4.2.4 ผู้ใช้สามารถใช้ปุ่ม  เพื่อเพิ่มหรือลดจำนวนเมนูอาหารได้

4.2.5 ผู้ใช้สามารถเลือกวันเวลาที่คำสั่งซื้อเกิดหรือปล่อยว่างเพื่อใช้เวลาปัจจุบันได้

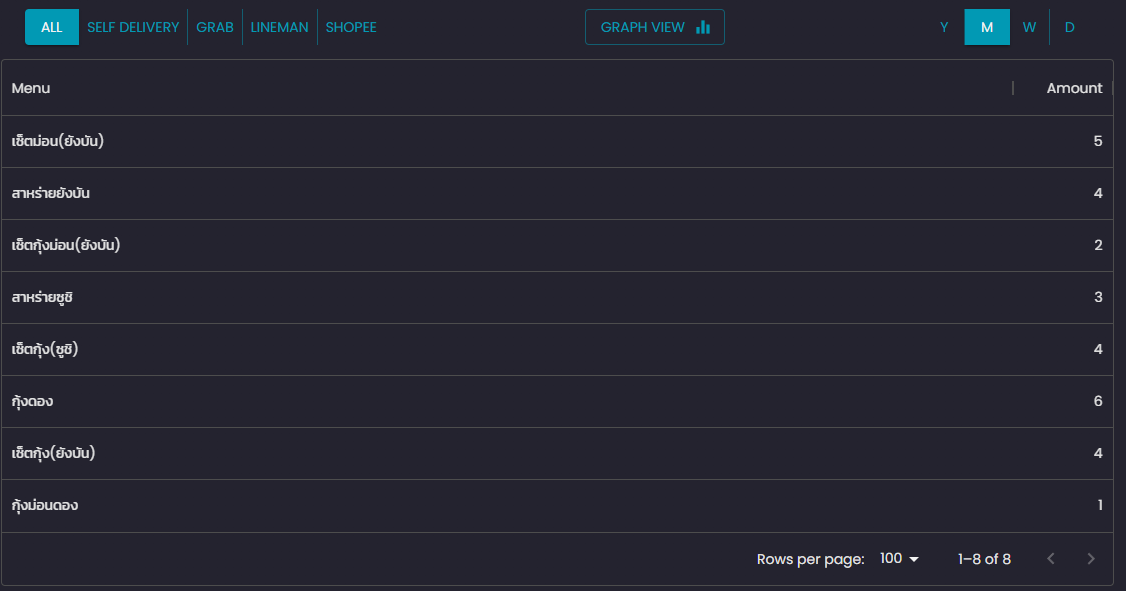
4.2.6 กดปุ่ม  เพื่อบันทึกคำสั่งซื้อลงในระบบ

4.3 ผู้ใช้สามารถกดปุ่ม  เพื่อลบคำสั่งซื้อที่ไม่ต้องการได้

5. สถิติการขาย

5.1 ไปที่เมนู Dashboard Graphical user interface, application

Description automatically generated



5.2 ผู้ใช้สามารถเลือกไรเดอร์ที่ต้องการได้

A picture containing graphical user interface

Description automatically generated

5.3 ผู้ใช้สามารถเลือกขอบเขตระยะเวลาได้ Graphical user interface, application

Description automatically generated

D = คำสั่งซื้อที่เกิดขึ้นใน 24 ชั่วโมงที่ผ่านมา

W = คำสั่งซื้อที่เกิดขึ้นใน 7 วันที่ผ่านมา

M = คำสั่งซื้อที่เกิดขึ้นใน 1 เดือนที่ผ่านมา

Y = คำสั่งซื้อที่เกิดขึ้นใน 1 ปีที่ผ่านมา

5.4 ผู้ใช้สามารถกดปุ่ม เพื่อเปิดหรือปิดโหมดแสดงสถิติผ่านกราฟแท่งได้

Chart

Description automatically generated