

กลุ่ม Expelliarmus

ระบบจัดการโรงแรม(Hotel Management System)

ชื่อ GitHub repository

Hotel-Management

สมาชิก

นายมุขัมมรีย์ ปือราเฮง	รหัสประจำตัว 55010995	กลุ่ม 2
นายวิกรม ภูจันทร์	รหัสประจำตัว 55011136	กลุ่ม 2

วิชา Object - Oriented Analysis and Design
ภาคการศึกษาที่ 1 ปีการศึกษา 2557

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สารบัญ

บทคัดย่อ	1
บทนำและรายละเอียดการวิเคราะห์หัวข้อ	2
งานที่เกี่ยวข้อง	3
ผลการวิเคราะห์ความต้องการของผู้ใช้ระบบ	4
- Functional Requirement	4
- Non – Functional Requirement	4
แนวทางการใช้งาน	5
- Use Case Diagram	5
- Use Case Specific	6
- Activity Diagrams	8
สถาปัตยกรรมของระบบ	10
- Problem Analysis	10
Abstraction	10
Component	10
- Application Architecture	11
- Subsystems / Components	12
แผนภาพของคลาสหลัก	15
- Domain classes	15
การพัฒนาซอฟต์แวร์	17
- Deployment	17
การทดสอบ	18
- Unit Test	18
- ผลการ Test ทั้งสอง	19
- Evaluate	20
บทสรุป	22
บรรณานุกรม	23

บทคัดย่อ

ถึงแม้ว่าในประเทศไทย ซอฟต์แวร์ระบบจัดการภายในโรงแรม จะไม่ได้ถือว่าเป็นนวัตกรรมใหม่ก็ตาม แต่ถึงอย่างไร ในโรงแรมขนาดกลางถึงขนาดเล็กที่บางแห่งที่อยู่ตามต่างจังหวัด ยังมีการรับเช็คอิน หรือทำงานต่างๆ ด้วยวิธีการบันทึกลงในสมุดอยู่ และการร้องขอใช้บริการอื่นๆ ของลูกค้า จำเป็นต้องหาผ่านการโทร หรือติดต่อกับเจ้าหน้าที่โดยตรงเท่านั้น ซึ่งระบบการจัดการโรงแรมนี้ จะอำนวยความสะดวกให้แก่ผู้ให้บริการ เพิ่มขีดความสามารถในการให้บริการ และลดการใช้ทรัพยากรบุคคล เป็นการเพิ่มประสิทธิภาพในการบริหารจัดการโรงแรม รวมถึงลดข้อผิดพลาดที่อาจเกิดขึ้นจากระบบที่ไม่รัดกุม หละหลวมได้ นอกไปจากนี้ ซอฟต์แวร์ที่พวกเราจะออกแบบ จะเป็น web application ที่ทำงานผ่าน web browser ทำให้สามารถทำงานข้าม platform ต่างๆ ได้ อีกทั้งยังออกแบบการทำงานของ user แต่ละประเภท ที่มีหน้าที่แตกต่างกัน ให้ทำงานเฉพาะส่วนของตน ทำให้ภาระงานการจัดการทั้งหมด ไม่ตกอยู่ที่พนักงานคนใดคนหนึ่งเพียงคนเดียว อีกประการหนึ่ง เนื่องจากเราทำซอฟต์แวร์นี้เป็น web application ทำให้สามารถทำงานบนอุปกรณ์พกพา เช่น สมาร์ทโฟนได้ ทำให้พนักงานส่วนที่ไม่ได้ปฏิบัติการอยู่กับที่ เช่น แม่บ้าน สามารถเข้าใช้งานในขณะที่ปฏิบัติงานผ่านสมาร์ทโฟนได้อีกด้วย

บทนำและรายละเอียดการวิเคราะห์หัวข้อ

ปัจจุบันธุรกิจโรงแรมมีการแข่งขันสูง อันเนื่องมาจากการที่ประเทศไทยมีแหล่งท่องเที่ยวที่สำคัญหลายแห่ง ซึ่งธุรกิจโรงแรมในประเทศไทยนั้น มีตั้งแต่โรงแรมขนาดเล็กไปจนถึงโรงแรมขนาดใหญ่ที่รองรับบุคคลสำคัญ ฉะนั้นแล้วระบบการจัดการในโรงแรมมีส่วนที่ช่วยให้บุคลากรที่เกี่ยวข้องกับการบริการ มีความสะดวกสบายในการปฏิบัติงาน โดยระบบจัดการโรงแรมเป็นการนำเทคโนโลยีเข้ามามีส่วนช่วยในการบริหารและจัดการโรงแรมให้มีประสิทธิภาพเพิ่มขึ้น เพิ่มความสะดวกสบายแก่ผู้ใช้บริการและผู้ให้บริการ

ระบบจัดการโรงแรมเป็นการลดปัญหาการที่ข้อมูลสูญหายในกรณีที่โรงแรมบางแห่งยังมีการรับเช็คอินโดยการบันทึกลงสมุดโดยข้อมูลอาจสูญหายได้จากหลายสาเหตุไม่ว่าจะเป็นจากตัวผู้ปฏิบัติงานเองหรือสาเหตุอื่นๆ และเพิ่มความสะดวกให้แก่ลูกค้าในกรณีที่ลูกค้ามีความต้องการพิเศษ ยกตัวอย่างการใช้งาน

- ลูกค้าต้องการที่จะทำการสั่งอาหารเข้าโดยให้มาบริการที่ห้องพัก โดยปกติแล้วลูกค้าต้องทำการโทรเจ้าหน้าที่ที่ Lobby จากนั้นเจ้าหน้าที่จะทำการกรอกข้อมูลสิ่งที่ลูกค้าต้องการ และฝ่าย Cuisine จะได้รับการแจ้งว่ามีการสั่งอาหารหลักจากนั้นผู้ที่เกี่ยวข้องจะปฏิบัติงานและนำไปเสิร์ฟแก่ลูกค้าในช่วงเวลาที่ได้รับไว้

- ในกรณีที่ลูกค้าทำการเช็คเอาท์ เจ้าหน้าที่จะทำการกรอกข้อมูลแล้วผู้ปฏิบัติงานฝ่าย Maid จะได้รับการแจ้งเตือนให้ไปตรวจสอบความเสียหายให้ห้องพักแล้วทำการกรอกข้อมูลในระบบว่ามีสิ่งใดเสียหายบ้าง และข้อมูลส่วนนี้เจ้าหน้าที่ Lobby ได้จะรับแล้วทำการคิดค่าเสียหาย

งานที่เกี่ยวข้อง

ระบบที่คล้ายคลึงกับระบบจัดการโรงแรมนั้น คือ KP Hotel Professional ซึ่งระบบนี้มีฟังก์ชันการทำงานหลากหลายที่อำนวยความสะดวกโดยการกรอกข้อมูลทุกอย่างจะทำโดยเจ้าหน้าที่ที่ Lobby ซึ่งยังมีข้อเสียให้เห็น ดังนี้

- User Interface ใช้งานลำบากเนื่องจากระบบมีฟังก์ชันการทำงานที่เยอะ จึงทำให้เกิดความซับซ้อนในการทำงาน
- การกรอกข้อมูลทุกอย่างต้องกระทำโดยเจ้าหน้าที่ที่ Lobby เท่านั้น
- การทำงานของระบบยังไม่ Support บน Mobile และ Run บน Window เท่านั้น
- ระบบนี้เกี่ยวข้องกับส่วนงานอื่นมากเกินไป เช่น งานด้านการเงิน บัญชี

โดยการทำงานของระบบซอฟต์แวร์ที่ผู้จัดทำจะทำการแยก User แต่ละฝ่ายจะทำหน้าที่ของตัวเองที่ทำเท่านั้น โดยมี Admin จะทำการจัดการเกี่ยวกับ User ต่างๆ เมื่อ User ใด User หนึ่งมีการกรอกข้อมูล User ที่เกี่ยวข้องกับงานเหล่านั้นจะได้รับ การแจ้งเตือนเพื่อให้ปฏิบัติงานต่อไป ระบบมีการรองรับทุก Flat form ของการรองรับ Web Browser และรองรับ Responsive Web Design ซึ่งการใช้งานซอฟต์แวร์ที่ผู้จัดทำได้ทำขึ้นมานั้นมี User Interface ที่ใช้งานง่ายลดความซับซ้อน ผู้ที่ไม่มีพื้นฐานด้านคอมพิวเตอร์และเทคโนโลยีสามารถใช้งานได้ และการนำส่วนที่ไม่เกี่ยวข้องหรือไม่จำเป็นออก เพื่อลดความซับซ้อนการทำงานของพนักงาน

ผลการวิเคราะห์ความต้องการของผู้ใช้ระบบ

Functional Requirement

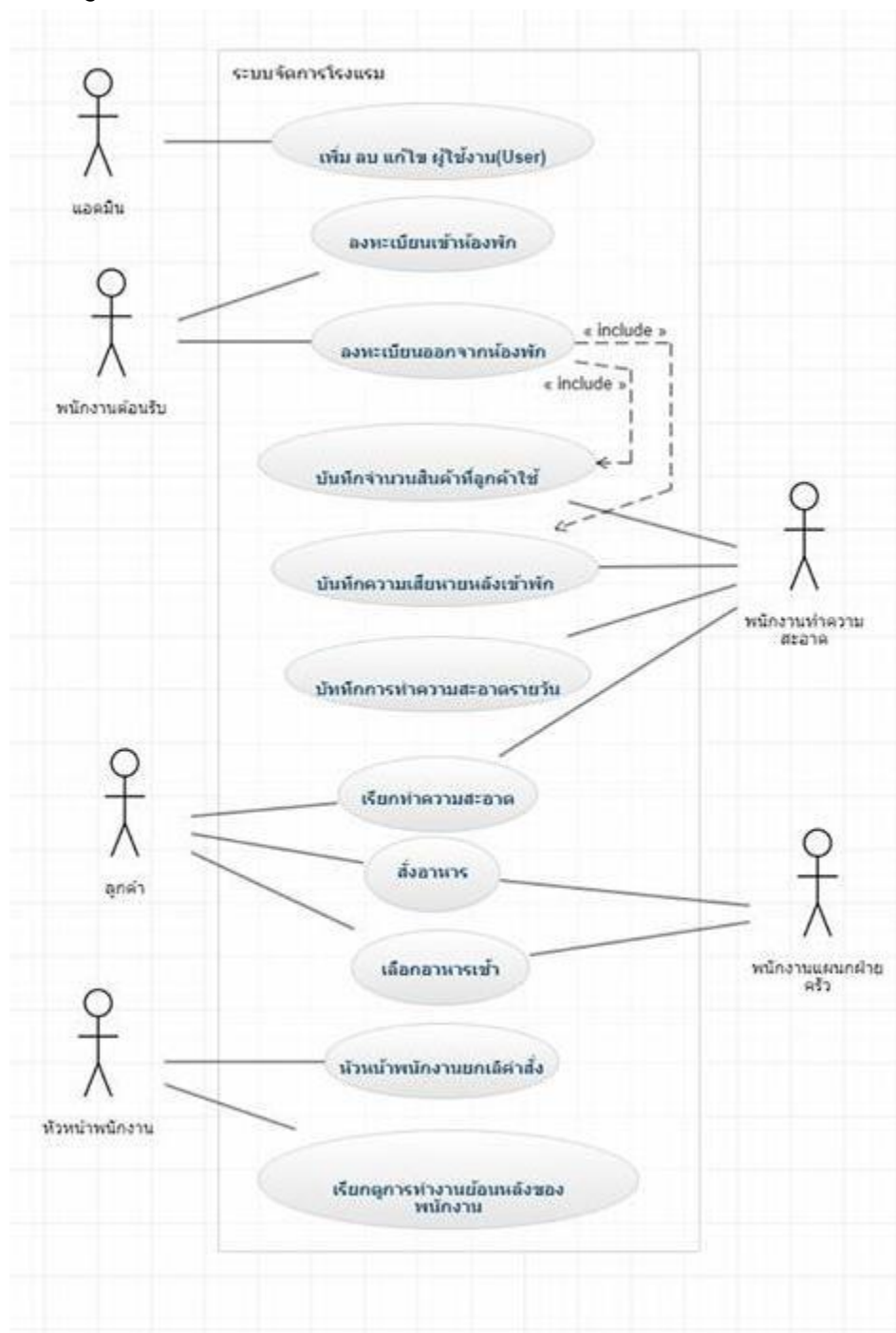
1. พนักงานสามารถลงทะเบียนเข้าพัก/ออกจากห้องพัก ให้กับลูกค้าได้
2. พนักงานสามารถบันทึกจำนวนสินค้าที่ลูกค้าใช้ตลอดการเข้าพักได้
3. พนักงานสามารถบันทึกความเสียหายหลังการเข้าพักของลูกค้าได้
4. พนักงานสามารถดึงข้อมูลการใช้สินค้าและค่าเสียหายของลูกค้าสำหรับคิดเงินได้
5. พนักงานสามารถบันทึกการทำความสะอาดรายวันได้
6. ลูกค้าสามารถระบุอาหารเข้าได้
7. ลูกค้าสามารถเรียกบริการทำความสะอาด หรือบริการอาหารถึงห้องได้
8. พนักงานภายในครัวสามารถดึงข้อมูลที่เป็นพื้นฐานข้อมูลมาได้
9. หัวหน้าพนักงาน สามารถตรวจเช็ครายการทำงานย้อนหลังทั้งหมดของพนักงานได้
10. ต้องจำแนกประเภทผู้ใช้งานตามหน้าที่ ให้เข้าถึงได้เฉพาะคำสั่ง และข้อมูลที่จำเป็นเท่านั้น
- 11.สามารถเพิ่ม ลบ แก้ไข ผู้ใช้งาน (User) ได้
- 12.พนักงานครัวสามารถเพิ่ม ลบ แก้ไข เมนูอาหารได้
- 13.ผู้จัดการสามารถเพิ่ม ลบ แก้ไข ห้องพักได้

Non – Functional Requirement

1. รองรับการทำงานบนทุกขนาดหน้าจอ
2. ทำงานบนอุปกรณ์ทุกแพลตฟอร์มที่สนับสนุนการทำงานของเว็บเบราว์เซอร์ได้
3. มีการยืนยันตัวตนเข้าใช้งานของผู้ใช้งานแต่ละคนด้วยวิธีใส่ ชื่อผู้ใช้งาน และ รหัสผ่าน
4. ผู้ใช้งานประเภทลูกค้าจะมีบัญชีผู้ใช้งานห้องละหนึ่งบัญชี มีการปิดกั้นการใช้งานขณะที่ห้องพักนั้นยังไม่มีลูกค้าเข้าพัก และเปิดให้ใช้งานเมื่อมีลูกค้าเข้าพัก โดยสั่มรหัสผ่านใหม่ทุกครั้งที่มีการเปิดการใช้งาน
5. มีการแจ้งเตือนเมื่อมีการเปลี่ยนแปลงข้อมูล
6. สามารถส่งออกข้อมูลให้ซอฟต์แวร์ที่ทำงานด้านอื่นนำไปใช้งานต่อได้
- 7.ระบบมีการอัปเดตข้อมูลแบบเรียลไทม์ได้
- 8.สามารถรองรับผู้ใช้งานได้อย่างน้อย 10 คน

แนวทางการใช้งาน

Use Case Diagram



Use Case Specific

Use case name: ลงทะเบียนเข้าห้องพัก

Use case purpose: เป็นการให้ลูกค้ามาลงทะเบียนเข้าห้องพัก และเปิดการใช้งาน user customer

Precondition: User ของลูกค้าปิดการใช้งานอยู่

Post condition: บัญชีลูกค้าเปิดให้ใช้งานเพื่อใช้งานบริการต่างๆระหว่างการเข้าพัก

Limitation: ข้อมูลบัญชีลูกค้าถูกบันทึกในฐานข้อมูลของโรงแรม

User ลูกค้า 1 User ต่อห้องพักหนึ่งห้อง

Assumption: ระบบจะมีการบันทึกข้อมูลการใช้บริการต่างๆของลูกค้าเพื่อทำการคิดเงินในวันที่ลงทะเบียนออกจากห้องพัก ดังนั้นลูกค้าต้องใช้บัญชีตัวเองเท่านั้น ในการใช้บริการต่างๆ

Primary scenario:

- A. ระบบแสดงรายการการทำงานให้พนักงานต้อนรับเลือก
- B. พนักงานต้อนรับเลือก check-in
- C. ระบบแสดงหน้าต่างให้พนักงานต้อนรับกรอกข้อมูลผู้เข้าพัก
- D. พนักงานต้อนรับกรอกข้อมูลผู้เข้าพักลงในแบบฟอร์ม
- E. พนักงานต้อนรับกดส่งข้อมูลให้ระบบ
- F. ระบบเปิดการใช้งาน (activate) customer user ห้องที่ลูกค้าเข้าพัก
- G. ระบบสร้าง password ใหม่ของ customer user
- H. ระบบแสดง password ใหม่ของ customer user
- I. พนักงานต้อนรับมอบ password และ username ให้ลูกค้า
- J. เสร็จสิ้นการทำงาน

Alternate Scenario:

- กรณีที่ 1 : พนักงานต้อนรับกรอกข้อมูลลูกค้าไม่ครบ หรือผิดจากแบบฟอร์ม
- D1: พนักงานกรอกข้อมูลผู้เข้าพักผิดจากฟอร์ม หรือกรอกไม่ครบ
 - D2: ระบบแสดงข้อความว่าผู้ใช้งานผิดพลาด
 - D3: ระบบย้อนกลับไปหน้ากรอกข้อมูล

Use Case Specific

Use case name: ระบบเรียกบริการทำความสะอาด

Use case purpose: ระบบนี้เพื่อให้ลูกค้าใช้เรียกพนักงานทำความสะอาดมาทำความสะอาดให้ห้องพักนอกเหนือจากการทำความสะอาดรายวัน

Precondition: พนักงานและลูกค้ามี User อยู่แล้ว

Post condition: ระบบกลับสู่สถานะปกติและมีการบันทึกการทำงานลงฐานข้อมูล

Limitation: ระบบจะเปิดให้ใช้ในเวลาทำการ

Assumption: การที่จะเรียกให้พนักงานมาทำความสะอาด ต้องเรียกผ่าน Application เท่านั้น

Primary scenario:

- A. ระบบแสดงรายการตัวเลือกบริการให้ลูกค้าเลือก
- B. ลูกค้าเลือก เรียกบริการทำความสะอาด
- C. ระบบเก็บข้อมูลลงฐานข้อมูล
- D. ระบบแจ้งเตือนไปยัง user พนักงานทำความสะอาด
- E. พนักงานทำความสะอาดเลือกคำสั่งยืนยันว่า กำลังไป
- F. ลูกค้าเลือกการทำงานเสร็จสิ้น
- G. ระบบบันทึกว่าการเรียกใช้บริการเสร็จสิ้น
- H. สิ้นสุดการทำงาน

Alternate Scenario:

กรณีที่ 1 หากมีพนักงานทำความสะอาด เลือกยืนยันว่า กำลังไป แล้ว แต่มีพนักงานบางคนที่หน้าต่างให้กดยืนยันยังไม่หายไป กดยืนยันเข้ามาอีกครั้ง

- E1. พนักงานทำความสะอาดเลือกคำสั่งยืนยันว่า กำลังไปมากกว่าหนึ่งคน
- E2. ระบบลงข้อมูลพนักงานที่เลือกคนแรกลงฐานข้อมูล
- E3. ระบบแสดงข้อความให้กับพนักงานคนที่เลือกที่หลังว่า มีพนักงานกำลังไปแล้ว
- E4. ระบบปิดหน้าต่างข้อความ
- E5. สิ้นสุดการทำงาน

กรณีที่ 2 พนักงานทำความสะอาดไม่กดยืนยันคำสั่งในเวลาที่กำหนดไว้

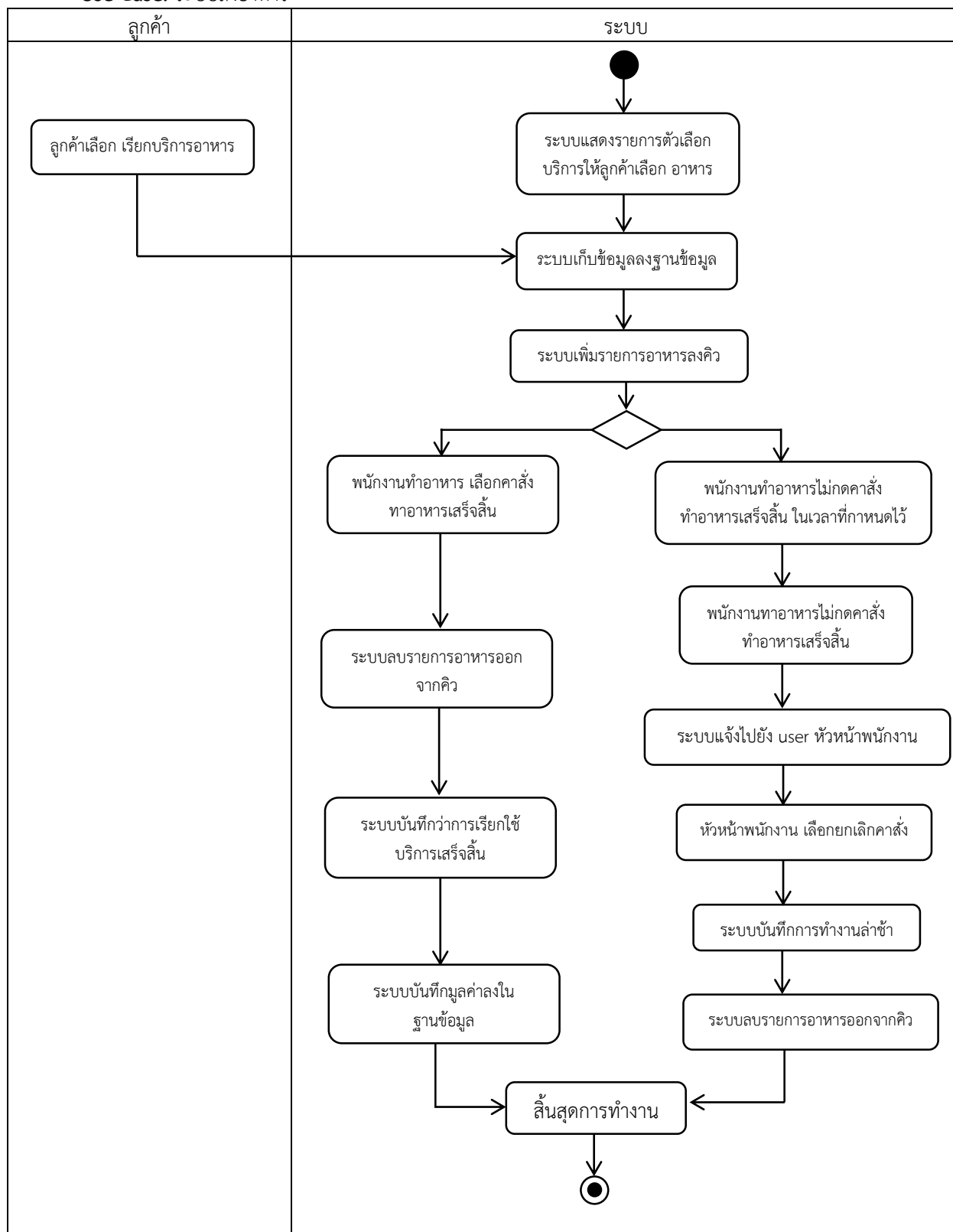
- E1. ไม่มีพนักงานทำความสะอาดเลือกยืนยันคำสั่ง
- E2. ระบบแจ้งไปยัง user หัวหน้าพนักงาน
- E3. หัวหน้าพนักงาน หรือ พนักงานเลือกยกเลิกคำสั่ง
- E4. ระบบบันทึกการทำงานล่าช้า
- E5. ระบบหยุดการแจ้งเตือน
- E6. สิ้นสุดการทำงาน

กรณีที่ 3 พนักงานทำความสะอาดไม่กดสิ้นสุดการทำงานภายในเวลาที่กำหนดไว้

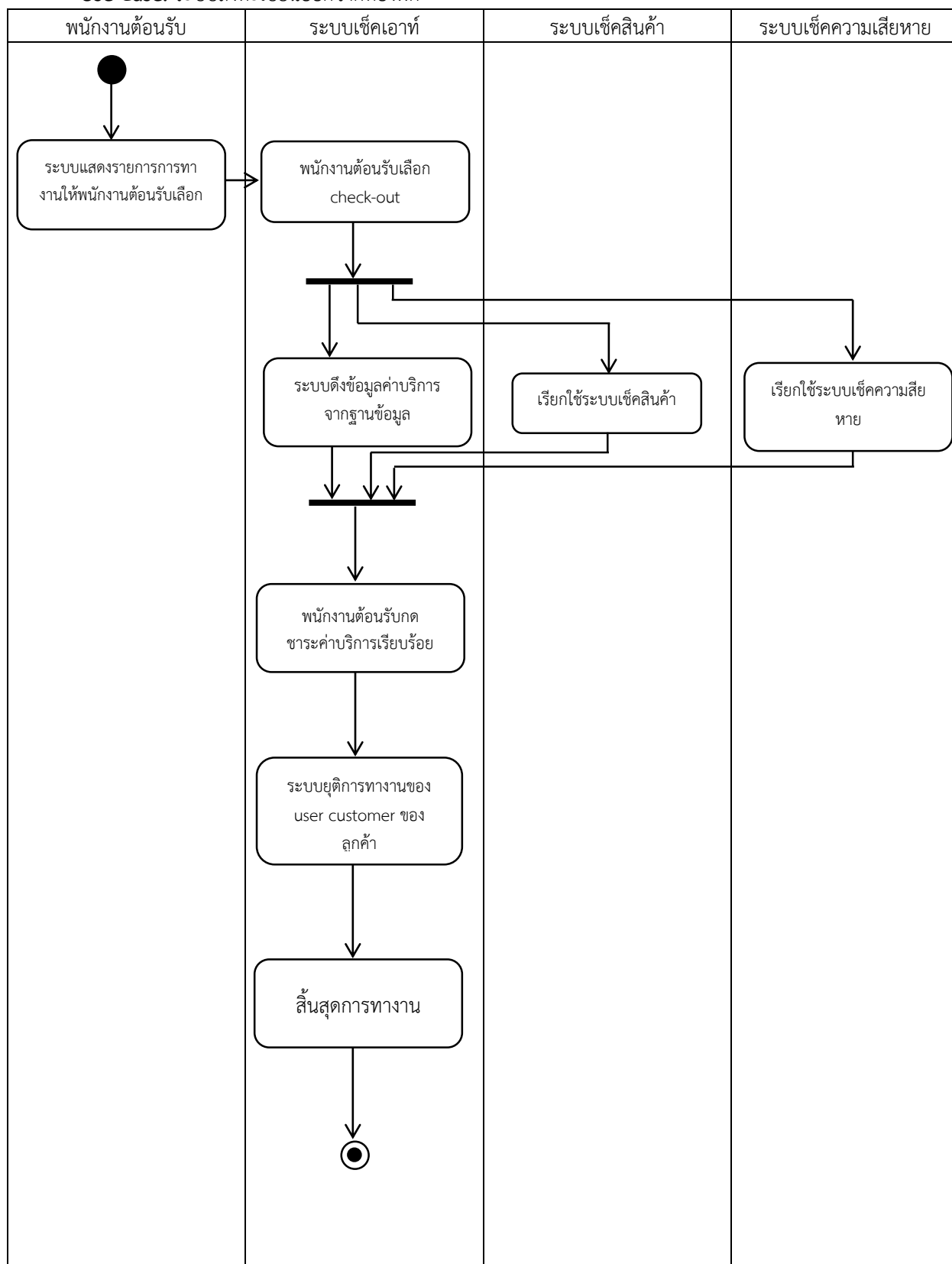
- F1. พนักงานทำความสะอาดไม่กดสิ้นสุดการทำงาน
- F2. ระบบแจ้งไปยัง user หัวหน้าพนักงาน
- F3. หัวหน้าพนักงาน หรือ พนักงานเลือกยกเลิกคำสั่ง
- F4. ระบบบันทึกการทำงานล่าช้า
- F5. ระบบหยุดการแจ้งเตือน
- F6. สิ้นสุดการทำงาน

Activity Diagrams

Use Case: ระบบสั่งอาหาร



Use Case: ระบบลงทะเบียนเป็นสมาชิก



สถาปัตยกรรมของระบบ

Problem Analysis

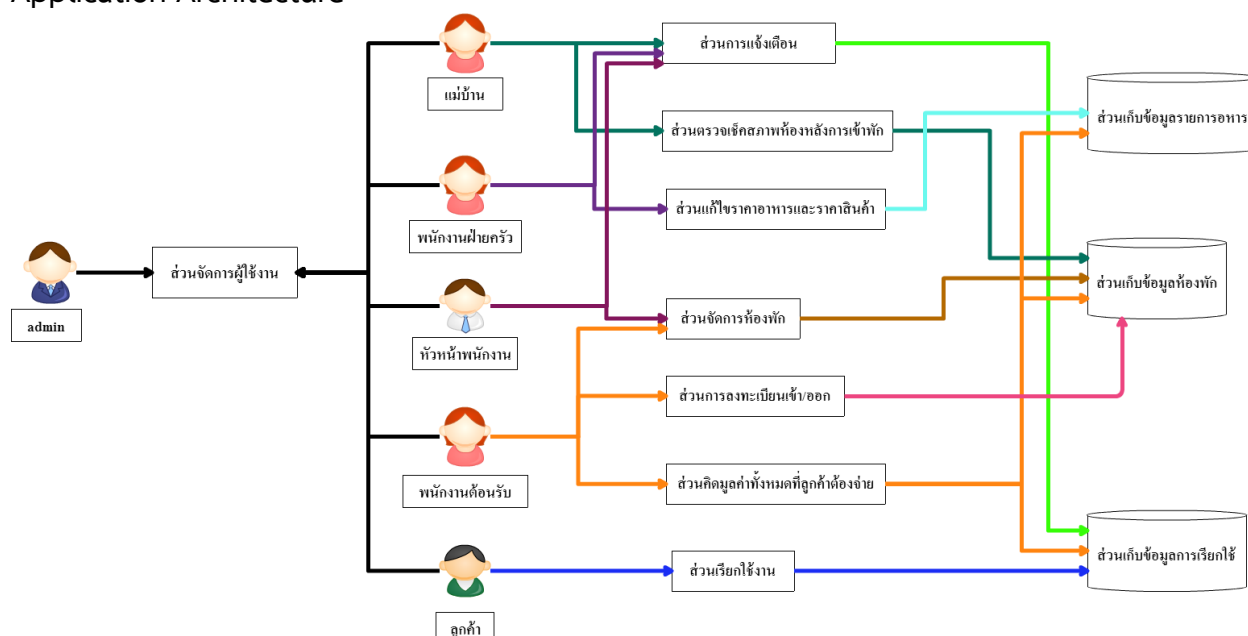
Abstraction

1. ห้องพัก จะประกอบไปด้วยเลขห้อง ราคา สถานะ สินค้าในห้อง และอุปกรณ์ในห้อง
2. พนักงาน จะประกอบไปด้วยชื่อ ตำแหน่ง ซึ่งจะใช้ตำแหน่งจำแนกการเข้าถึงเมนูและข้อมูลได้แตกต่างกัน
3. ลูกค้า ประกอบไปด้วย ชื่อ ที่อยู่ เบอร์ติดต่อ ห้องที่พัก
4. สถิติ ใช้แทนข้อมูลการทำงานของพนักงานแต่ละคน การตอบสนองคำร้องขอของลูกค้า และประวัติอื่นๆ ทั้งหมด
5. รายการอาหารที่ต้องทำ ประกอบไปด้วยอาหารที่ต้องทำ เวลา ลำดับที่ ผู้สั่ง
6. บริการ เป็นคำร้องขอบริการต่างๆ ของลูกค้า
7. บัญชีค่าใช้จ่าย ใช้สำหรับบอกว่า ลูกค้าแต่ละห้อง มีค่าใช้จ่ายรวมเท่าไร

Component

1. ส่วนจัดการผู้ใช้ ใช้สำหรับสร้าง ลบ แก้ไข ผู้ใช้ต่างๆ
2. ส่วนแจ้งเตือน จะใช้แจ้งเตือนไปยังพนักงานเมื่อมีเหตุการณ์เกิดขึ้น หรือใช้แจ้งเตือนการทำงานรายวัน
3. ส่วนลงทะเบียนการเข้าพัก ใช้สำหรับลงทะเบียนเข้าพัก (check in) และลงทะเบียนออก (check out) ให้ลูกค้า
4. ส่วนตรวจเช็คสภาพห้องหลังการเข้าพัก สำหรับนำไปคิดราคา
5. ส่วนคิดมูลค่าทั้งหมดที่ลูกค้าต้องจ่าย จะคำนวณจากราคาห้องพัก วันที่พัก สินค้าที่ใช้ ความเสียหาย อาหาร
6. ส่วนเรียกใช้บริการ สำหรับให้ลูกค้าเรียกใช้บริการต่างๆ จากโรงแรม มีสองอย่างคือ บริการทำความสะอาด และ บริการเสิร์ฟอาหาร
7. ส่วนแก้ไขรายการอาหาร และราคา
8. ส่วนจัดการห้องพัก เพิ่ม ลบ แก้ไข รายละเอียดห้องพักต่างๆ

Application Architecture



- ส่วนจัดการผู้ใช้งาน ประกอบไปด้วยการแก้ไข เพิ่ม ลบ user ผู้ใช้งาน จะติดต่อกับพื้นที่เก็บข้อมูลของผู้ใช้งาน
- ส่วนแก้ไขรายการอาหารและราคา ในส่วนนี้ทำให้พนักงานเข้ามาทำการแก้ไขรายการสินค้าและรายการอาหาร ในกรณีที่มีการเปลี่ยนแปลงของราคา โดยส่วนนี้จะติดต่อกับพื้นที่เก็บข้อมูลรายการอาหารและราคา
- ส่วนจัดการห้องพัก ประกอบไปด้วยการ Check In และ Check Out เป็นการลงทะเบียนเข้าพักและลงทะเบียนออกจากห้องพัก โดยจะติดต่อกับพื้นที่เก็บข้อมูลของห้องพักแต่ละห้อง
- ส่วนจัดการลงทะเบียนเข้าพัก ในส่วนนี้จะมี Check In และ Check Out เป็นการลงทะเบียนเข้าพัก และลงทะเบียนออกจากห้องพัก ก็จะติดต่อกับพื้นที่เก็บข้อมูลของห้องพัก
- ส่วนตรวจสอบเช็คสภาพห้องพักหลังการเข้าพัก เป็นส่วนที่จะเก็บข้อมูลความเสียหายในห้องพัก โดยจะดึงข้อมูลมาจาก พื้นที่เก็บข้อมูลห้องพัก
- ส่วนคิดค่าใช้จ่ายทั้งหมดที่ลูกค้าต้องจ่าย เป็นส่วนที่เก็บข้อมูลค่าใช้จ่ายที่ลูกค้าต้องจ่ายอันเกิดจากค่าเสียหาย และค่าบริการต่างๆ โดยดึงข้อมูลจากพื้นที่เก็บข้อมูลการเรียกใช้
- ส่วนการแจ้งเตือน การแจ้งเตือนของระบบจะถูกแจ้งเตือนเมื่อเกิดการเปลี่ยนแปลงของข้อมูลในพื้นที่เก็บข้อมูลการเรียกใช้ โดยจะแจ้งเตือนไปยังพนักงานทุกคนรวมทั้งหัวหน้าพนักงาน
- ส่วนเรียกใช้บริการ ประกอบไปด้วย ส่วนเรียกใช้ในการสั่งอาหาร และส่วนเรียกใช้ทำความสะอาดห้องพัก
- พื้นที่เก็บข้อมูลการเรียกใช้ ประกอบไปด้วยข้อมูลของรายการอาหารที่ลูกค้าสั่ง รายการการทำความสะอาดที่ลูกค้าได้ขอม่า และหน้าที่ของพนักงานแต่ละคน เมื่อเกิดการเรียกใช้งานพนักงานฝ่ายใดที่ต้องปฏิบัติตามคาขอของลูกค้า
- พื้นที่เก็บข้อมูลห้องพัก จะประกอบไปด้วยข้อมูลห้องพัก และสถานะของห้องพักนั้นๆเพื่อดูว่างหรือไม่
- พื้นที่เก็บข้อมูลรายการอาหารและราคา ประกอบไปด้วยข้อมูลรายการสินค้า และราคาของอาหารชนิดต่างๆ

Subsystems / Components

- ส่วนจัดการผู้ใช้งาน

รายละเอียด :

- การแก้ไขบัญชี User เกิดขึ้นในกรณีที่บัญชีผู้ใช้งานมีการเปลี่ยนแปลงข้อมูลของผู้ใช้งานนั้นๆ
- การเพิ่มบัญชี User เกิดขึ้นในกรณีเมื่อมีการรับพนักงานมาใหม่หรือเพิ่มจำนวนพนักงาน
- การลบบัญชี User ใช้ในกรณีที่พนักงานลาออก หรือเปลี่ยนแผนกที่ เนื่องจากแต่ละหน้าแผนกจะมีบัญชีการใช้งานที่แตกต่างกัน
- โดยในส่วนนี้จะเป็นส่วนที่เก็บข้อมูลพื้นฐานของพนักงานแต่ละคน เช่น ข้อมูลที่อยู่ หมายเลขโทรศัพท์ เป็นต้น และเป็นการกำหนดสิทธิในการใช้งานบัญชีของพนักงานแต่ละคน

การใช้งาน :

- ในส่วนของการแก้ไข เพิ่ม ลบ บัญชีผู้ใช้งานนั้นผู้ที่เข้ามาแก้ไขหรือมาเพิ่ม ลบ ข้อมูลต่างในส่วนนี้มีแค่ Admin คนเดียวเท่านั้น พนักงานในส่วนอื่นไม่มีสิทธิ์เข้าใช้งานในส่วนนี้ได้
- ส่วนแก้ไขรายการอาหารและราคา

รายละเอียด :

- ในส่วนเป็นที่แก้ไขข้อมูลหรือรายการอาหารไม่ว่าจะเป็นการเพิ่ม เมนูอาหาร ลบเมนูอาหาร การแก้ไขราคาอาหาร หรือเพิ่มเมนูแนะนำให้แก่ลูกค้า

การใช้งาน :

- การเข้ามาใช้งานในส่วนนี้ผู้ที่เข้ามาแก้ไขรายการต่างๆในส่วนนี้คือ พนักงานฝ่ายครัว เนื่องฝ่ายครัวเป็นผู้กำหนดรายการอาหาร และราคาของอาหารหรือสินค้าต่างๆ

- ส่วนจัดการห้องพัก

รายละเอียด :

- เป็นส่วนที่จัดการ เพิ่ม ลบ แก้ไข ข้อมูลของห้อง เนื่องจากในห้องพักแต่ละห้องมีประเภทการใช้งานที่ต่าง เช่น ห้องที่มีเตียงเดี่ยว ห้องที่มีเตียงคู่ หรือลูกค้ามีการขอเพิ่มบางอย่างในห้องพัก บางครั้งห้องพักที่ลูกค้าต้องการอาจจะไม่ว่างต้องมีการปรับห้องที่ว่างให้เป็นไปตามที่ลูกค้าต้องการ และการที่โรงแรมมีการปรับปรุงหรือแก้ไขห้องพัก

การใช้งาน :

- การเข้าใช้งานในส่วนนี้จะมีแค่พนักงานต้อนรับเนื่องจากเป็นผู้ที่ติดต่อลูกค้าโดยตรง และหัวหน้าพนักงาน ในกรณีที่มีการปรับปรุงห้องพัก
- ส่วนจัดการลงทะเบียนเข้าพัก

รายละเอียด :

- ใช้สำหรับการลงทะเบียนเข้าพัก และออกจากห้องพัก
- โดยในส่วนนี้จะเป็นการเก็บข้อมูลของลูกค้า เช่น ข้อมูลที่อยู่ หมายเลขโทรศัพท์ อีเมลล์และข้อมูลห้องพักที่ลูกค้าเข้าพัก
- เก็บข้อมูลสถานะห้องพักแต่ละห้อง เพื่อตรวจสอบว่ามีลูกค้าเข้าพักหรือไม่ มีการจองห้องพักล่วงหน้าหรือไม่

การใช้งาน :

- การใช้งานในส่วนนี้ ผู้ที่รับผิดชอบคือ พนักงานต้อนรับ โดยพนักงานในส่วนนี้จะมี User การใช้งานคนละหนึ่งบัญชี

- ส่วนแก้ไขรายการอาหารและราคา

รายละเอียด :

- ในส่วนที่เป็นที่ใช้แก้ไขข้อมูลหรือรายการอาหารไม่ว่าจะเป็นการเพิ่ม เมนูอาหาร ลบเมนูอาหาร การแก้ไขราคาอาหาร หรือเพิ่มเมนูแนะนำให้แก่ลูกค้า

การใช้งาน :

- การเข้ามาใช้งานในส่วนนี้ผู้ที่เข้ามาแก้ไขรายการต่างๆในส่วนนี้คือ พนักงานฝ่ายครัว เนื่องฝ่ายครัวเป็นผู้กำหนดรายการอาหาร และราคาของอาหารหรือสินค้าต่างๆ

- ส่วนจัดการลงทะเบียนเข้าพัก

รายละเอียด :

- ใช้สำหรับการลงทะเบียนเข้าพัก และออกจากห้องพัก
- โดยในส่วนนี้จะเป็นการเก็บข้อมูลของลูกค้า เช่น ข้อมูลที่อยู่ หมายเลขโทรศัพท์ อีเมลล์และข้อมูลห้องพักที่ลูกค้าเข้าพัก
- เก็บข้อมูลสถานะห้องพักแต่ละห้อง เพื่อตรวจสอบว่ามีลูกค้าเข้าพักหรือไม่ มีการจองห้องพักล่วงหน้าหรือไม่

การใช้งาน :

- การใช้งานในส่วนนี้ ผู้ที่รับผิดชอบคือ พนักงานต้อนรับ โดยพนักงานในส่วนนี้จะมี User การใช้งานคนละหนึ่งบัญชี

- ส่วนตรวจสอบเช็คสภาพห้องพักหลังการเข้าพัก

รายละเอียด :

- ส่วนนี้คือการตรวจสอบสภาพห้องพักหลังจากที่ลูกค้าออกจากห้องพักเพื่อลงทะเบียนออกจากห้องพัก เนื่องจากการเข้าพักของลูกค้าแต่ละคนอาจเกิดความเสียหายในส่วนต่างๆของห้องพัก

การใช้งาน :

- แม่บ้านจะเป็นผู้เข้าไปตรวจสอบความเสียหายต่างๆในห้องพัก จากนั้นแม่บ้านก็จะเลือกรูปแบบความเสียหาย กระดาษแตก ลูกบิดเสีย ลงในระบบ และข้อมูลส่วนนี้จัดเก็บลงในพื้นที่เก็บข้อมูลห้องพัก เพื่อให้พนักงานต้อนรับนำข้อมูลส่วนนี้มาใช้ในการคิดค่าความเสียหายก่อนที่ลูกค้าลงทะเบียนออกจากห้องพัก

- ส่วนคิดค่าใช้จ่ายทั้งหมดที่ลูกค้าต้องจ่าย

รายละเอียด :

- ก่อนที่ลูกค้าต้องการลงทะเบียนออกจากห้องพัก พนักงานต้อนรับจะทำการคิดค่าใช้จ่ายต่างๆที่ลูกค้าได้ใช้บริการ อาจจะประกอบไปด้วยค่าห้องพัก ค่าอาหาร ค่าความเสียหายอันเกิดจากการกระทำของลูกค้า เป็นต้น

การใช้งาน :

- พนักงานต้อนรับจะทำการคิดค่าใช้จ่ายทั้งหมดมาโดยการดึงข้อมูลจากส่วนเก็บข้อมูลห้องพัก ซึ่งระบุประเภทห้องพัก และข้อมูลจากพื้นที่เก็บข้อมูลการเรียกใช้โดยจะประกอบข้อมูลที่ลูกค้าเรียกใช้ต่างๆ

- ส่วนเรียกใช้บริการ

รายละเอียด :

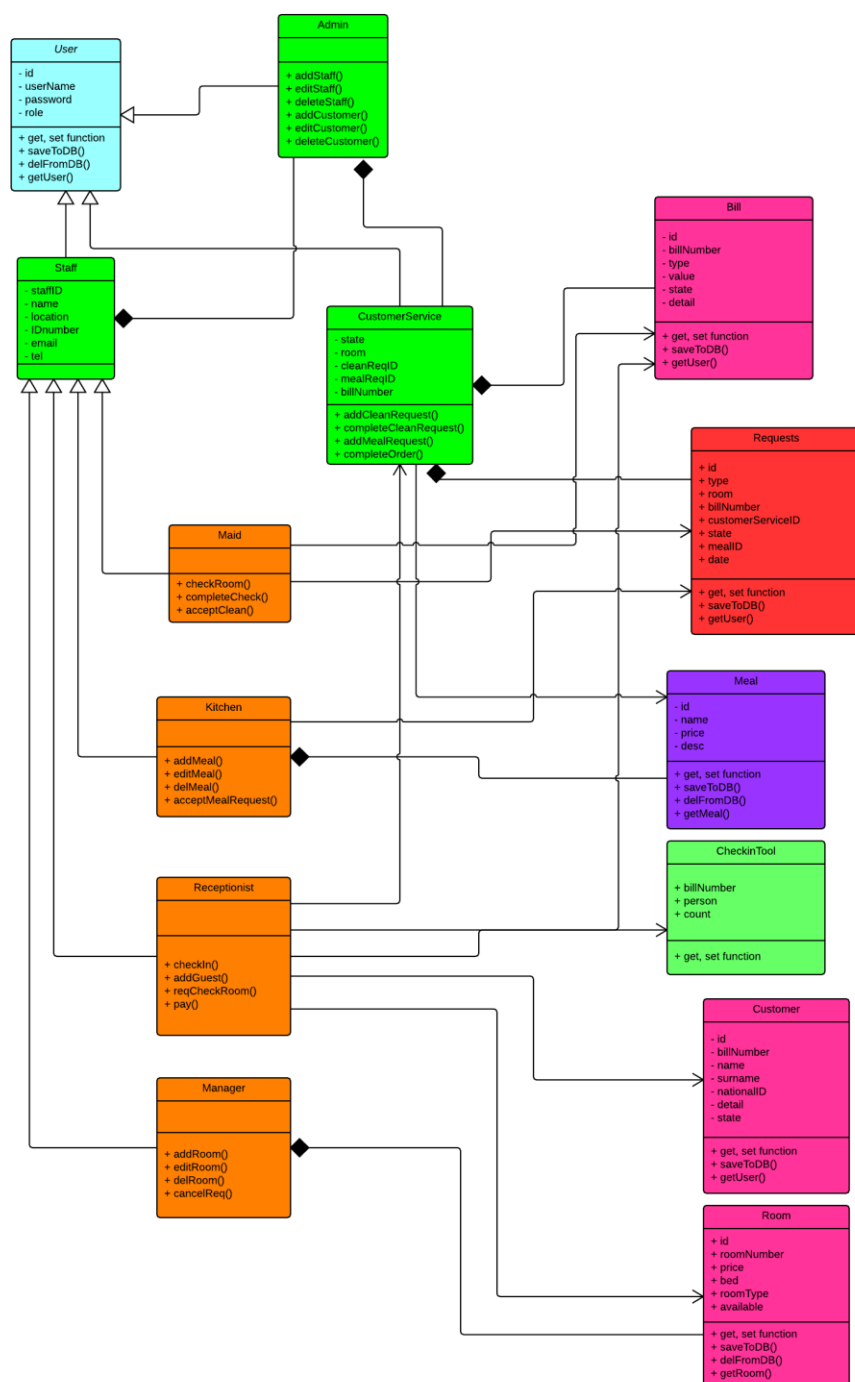
- เมื่อลูกค้ามีความต้องการที่จะเรียกใช้บริการพิเศษ เช่น ต้องการให้แม่บ้านทำความสะอาดห้องเพิ่ม นอกเหนือจากการทำความสะอาดทุกวัน หรือการสั่งอาหารแล้วต้องการให้เสิร์ฟอาหารที่ห้องพัก

การใช้งาน :

- มีการเรียกใช้งานของลูกค้าข้อมูลจะถูกเก็บในพื้นที่เก็บข้อมูลการเรียกใช้ จากนั้นระบบแจ้งเตือนจะ ทาการแจ้งเตือนพนักงานที่เกี่ยวข้องหรือต้องปฏิบัติหน้าที่นั้นๆ เพื่อให้มาปฏิบัติงาน

แผนภาพของคลาสหลัก

Domain classes



อธิบายฟังก์ชันต่างๆ

- **User** เป็น abstract class ข้อมูลเบื้องต้นของผู้ที่ต้อง log in
- **People** เป็น abstract class ข้อมูลจำพวก ชื่อ ที่อยู่ เลขประจำตัวประชาชน ฯลฯ โดย inherit ข้อมูลมาจาก User
- **Admin** เป็นคลาสที่ทำหน้าที่จัดการพนักงาน
- **Customer** เป็นคลาสของลูกค้า ถูกสร้างเมื่อมีการลงชื่อเข้าพัก และ ลบเมื่อลงชื่อออก (แต่ข้อมูลการเข้าพักจะเก็บไว้) สามารถส่งคำร้องขอทำความสะอาด หรืออาหารได้
- **Staff** เป็นคลาสของพนักงาน ที่ inherit มาจาก People และเป็น Composite Class ของ Admin จะแบ่งเป็น sub class ตามแต่ละตำแหน่งดังนี้
 - **Manager** ผู้จัดการ หรือหัวหน้าพนักงาน สามารถยกเลิกคำสั่งที่ได้รับการเพิกเฉย และตั้งค่าสถานะซ่อมบำรุงห้องพักได้
 - **Receptionist** พนักงานต้อนรับ ทำหน้าที่หลักคือลงชื่อเข้าและออกให้ลูกค้า ซึ่งจะมีการเรียกเก็บเงินด้วย
 - **Kitchen** สามารถเรียกดูรายการอาหารที่ต้องทำ แล้วตั้งค่าแล้วเสร็จ เพื่อให้พนักงานเสิร์ฟนำไปเสิร์ฟต่อได้
 - **Maid** พนักงานทำความสะอาด มีหน้าที่รับการร้องขอทำความสะอาด และ ตรวจเช็คสภาพห้องหลังการเข้าพักของลูกค้าด้วย
- **Meal** เป็น class รายการอาหาร พนักงานฝ่ายครัวสามารถแก้ไข เพิ่ม ลบ ได้ และลูกค้าสามารถเข้าถึงได้ว่ามีรายการอะไรบ้าง ผ่านการสั่งอาหาร
- **CustomerService** เป็นคลาสที่ให้ลูกค้าทำการเรียกใช้บริการทำความสะอาดและสั่งอาหาร
- **Bill** เป็นคลาสสำหรับสร้าง และจัดการรายการค่าใช้จ่ายสำหรับนำไปคิดมูลค่าของลูกค้า
- **CheckInTool** เป็นคลาสที่เข้าถึงโดย Receptionist มีหน้าที่จัดการสร้างและลบ user ให้ customer ตรวจสอบ และการคิดค่าใช้จ่ายต่างๆของลูกค้าตอนลงทะเบียนออก
- **Room** เป็นคลาสของห้องพัก มีรายละเอียดต่างๆ ของห้องพัก รวมทั้งสถานะของห้องพักด้วย

รายละเอียดการพัฒนาซอฟต์แวร์

Deployment

Software ที่ใช้งาน:

- MySQL ใช้งานจัดการ Data base
- PHP จัดการส่วนของงานเบื้องหลัง (Back End)
- CSS และ HTML ใช้ในการออกแบบส่วนการแสดงผล
- Laravel Framework
- Bootstrap ใช้งานจัดการการแสดงผลเพื่อให้รองรับทุกขนาดหน้าจอในการแสดงผล
- Apache 2 และ Ubuntu 14.04 ใช้สำหรับเป็นระบบปฏิบัติการ

Server:

- Digital Ocean
- Virtual Machine 1 เครื่อง (IP 128.199.152.130 ชื่อเครื่อง: ce-hms) memory 512 MB
Disk SSD 20 GB
- <http://www.ce-hms.tk/>

ผลการทดสอบซอฟต์แวร์

Unit Test

- ทดสอบระบบการจัดการการเรียกใช้บริการ

code/app/tests/ RequestTest.php

```
<?php
class RequestTest extends TestCase{
    public function testBasicExample()
    {
        $id = '5';
        $type = '2';
        $room = '101';
        $billNumber = '16';
        $customerServiceID = '5';
        $state = '1';
        $mealID = '0';
        $date = '141110';

        //$test = TestRequests::mockRequest($id, $type, $room, $billNumber , $customerServiceID, $state, $mealID, $date);

        $req = new Requests();
        $req->setId($id);
        $req->setType($type);
        $req->setRoom($room);
        $req->setBillNumber($billNumber);
        $req->setCustomerServiceID($customerServiceID);
        $req->setState($state);
        $req->setMealID($mealID);
        $req->setDate($date);

        $this->assertEquals($id,$req->getId());
        $this->assertEquals($type,$req->getType());
        $this->assertEquals($room,$req->getRoom());
        $this->assertEquals($billNumber,$req->getBillNumber());
        $this->assertEquals($customerServiceID,$req->getCustomerServiceID());
        $this->assertEquals($state,$req->getState());
        $this->assertEquals($mealID,$req->getMealID());
        $this->assertEquals($date,$req->getDate());
    }
}
```

- ทดสอบระบบการจัดการห้องห้องพักลูกค้า

code/app/tests/RoomTest.php

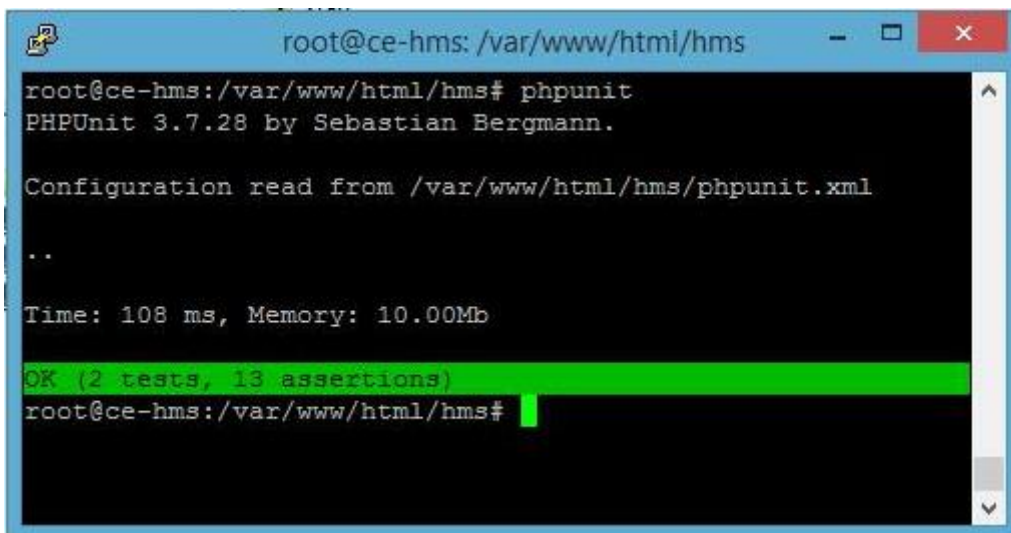
```
<?php
class RoomTest extends TestCase{
    public static function mockEditRoom($id, $roomNumber, $roomType, $bed, $price, $available){
        $room = new Room();
        $room->setId($id);
        $room->setRoomType($roomType);
        $room->setRoomNumber($roomNumber);
        $room->setBed($bed);
        $room->setPrice($price);
        $room->setAvailable($available);
        return $room;
    }

    public function testBasicExample()
    {
        $id = '5';
        $roomNumber = '101';
        $roomType = '2';
        $bed = '101';
        $price = '16';
        $available = 'true';

        $roomObj = RoomTest::mockEditRoom($id, $roomNumber, $roomType, $bed, $price, $available);

        $this->assertEquals($id,$roomObj->getId());
        $this->assertEquals($roomNumber,$roomObj->getRoomNumber());
        $this->assertEquals($roomType,$roomObj->getRoomType());
        $this->assertEquals($bed,$roomObj->getBed());
        $this->assertEquals($price,$roomObj->getPrice());
        $this->assertEquals($available,$roomObj->getAvailable());
    }
}
?>
```

ผลการ Test ทั้งสอง



```
root@ce-hms: /var/www/html/hms
root@ce-hms:/var/www/html/hms# phpunit
PHPUnit 3.7.28 by Sebastian Bergmann.

Configuration read from /var/www/html/hms/phpunit.xml

..

Time: 108 ms, Memory: 10.00Mb

OK (2 tests, 13 assertions)
root@ce-hms:/var/www/html/hms#
```

Evaluate

การทดลองที่ 1 ระบบเพิ่ม ลบ และแก้ไขรายการอาหาร

จุดประสงค์ของการทดลอง (Goal of the experiment)

- เพื่อตรวจสอบว่าระบบสามารถเพิ่มรายการอาหารได้
- เพื่อตรวจสอบว่าระบบสามารถแก้ไขรายการอาหารได้
- เพื่อตรวจสอบว่าระบบสามารถลบรายการอาหารได้
- เพื่อตรวจสอบความสะดวกในการใช้

สิ่งที่วัด (Measurement metrics)

- ความถูกต้องในการเพิ่ม ลบ แก้ไข รายการอาหาร

วิธีการทดลองและสิ่งที่ต้องใช้ในการทดลอง

1. login เข้าใช้งานแอปพลิเคชันด้วย username และ password ของฝ่าย kitchen
2. เลือกเมนูเพิ่มรายการอาหาร
3. กรอกข้อมูลรายการอาหารลงไปตามแบบฟอร์ม
4. กดยืนยันการส่งข้อมูล
5. เลือกคำสั่ง ดูรายการอาหาร แล้วตรวจสอบความถูกต้อง
6. เลือกเมนูแก้ไขรายการอาหาร
7. เลือกรายการอาหารที่ต้องการแก้ไข
8. เลือกหัวข้อเป็นการเปลี่ยนแปลงรายการ
9. แก้ไขข้อมูลที่ต้องการเปลี่ยน
9. กดยืนยันการแก้ไข
10. เลือกเมนูแก้ไขรายการอาหาร
11. เลือกรายการอาหารที่ต้องการแก้ไข
12. เลือกรายการอาหารที่ต้องการลบ
13. เลือกหัวข้อเป็นลบรายการ
14. กดยืนยันลบรายการอาหาร
15. เลือกคำสั่ง ดูรายการอาหาร แล้วตรวจสอบความถูกต้อง

สิ่งที่ต้องใช้ในการทดลอง

- ข้อมูล username และ password ของพนักงานฝ่ายครัว
- ข้อมูลและฐานข้อมูลรายการอาหาร

ผลที่ได้จากการทดลอง (Experimental results)

สรุปและสิ่งที่ได้จากการทดลองนี้ (Conclusion)

การทดลองที่ 2 ระบบเรียกทำความสะอาด

จุดประสงค์ของการทดลอง (Goal of the experiment)

- เพื่อตรวจสอบว่าระบบสามารถเพิ่มคำร้องขอทำความสะอาดได้
- เพื่อตรวจสอบว่าพนักงานสามารถตอบรับคำร้องขอได้
- เพื่อตรวจสอบว่า ลูกค้าที่ร้องขอ สามารถยืนยันคำร้องขอแล้วเสร็จได้
- เพื่อตรวจสอบว่า หากพนักงานทำความสะอาดไม่เกิดภายในเวลาที่กำหนด ระบบจะส่งคำร้องขอไปให้ผู้จัดการยกเลิกได้

สิ่งที่วัด (Measurement metrics)

- ความถูกต้องของการร้องขอการทำความสะอาด

วิธีการทดลอง

1. เปิดใช้งานแอปพลิเคชันพร้อมกันสามคนขึ้นไป โดยอย่างน้อยมีหนึ่งคนเข้าใช้ด้วยบัญชีลูกค้า มีอย่างน้อยหนึ่งคนเข้าใช้ด้วยบัญชีพนักงานทำความสะอาด และมีหนึ่งคนเข้าใช้ด้วยบัญชีผู้จัดการ
2. บัญชีลูกค้า เลือกเพิ่มคำร้องขอทำความสะอาด
3. บัญชีพนักงาน ตอบรับคำร้องขอในเวลาที่กำหนด
4. บัญชีลูกค้า เลือกคำร้องขอสิ้นสุด
5. บัญชีลูกค้า ลองเพิ่มคำร้องขออีกครั้ง
6. บัญชีพนักงาน ไม่ตอบรับคำร้องขอในเวลาที่กำหนด
7. บัญชีผู้จัดการ ยกเลิกคำร้องขอ
8. ตรวจสอบผล

สิ่งที่ต้องใช้ในการทดลอง

- ข้อมูล username และ password ของพนักงานทำความสะอาด ลูกค้า และผู้จัดการ
- ข้อมูลและฐานข้อมูลคำร้องขอ

ผลที่ได้จากการทดลอง (Experimental results)

สรุปและสิ่งที่ได้จากการทดลองนี้ (Conclusion)

บทสรุป

เว็บไซต์ hotel management system เป็นเว็บแอพลิเคชันที่ใช้ในการจัดการ และการบริหารของโรงแรม ซึ่งทางผู้จัดทำมีจุดประสงค์เพื่อเนื่องจากเล็งเห็นว่าประเทศไทยเป็นประเทศที่แหล่งท่องเที่ยวส่วนใหญ่ และอุตสาหกรรมการท่องเที่ยวของประเทศไทยมีส่วนสำคัญในการพัฒนาประเทศ ซึ่งโรงแรมเป็นปัจจัยสำคัญสิ่งหนึ่งในการมาท่องเที่ยวของนักท่องเที่ยวไม่ว่าจะเป็นชาวต่างชาติหรือคนไทยด้วยกันเอง และแต่ละโรงแรมของแต่ละจังหวัดจะมีลักษณะการจัดการ การทำงานของพนักงานที่แตกต่างกัน ซึ่งโรงแรมแถวต่างจังหวัดที่มีขนาดเล็กบางแห่งนั้นจะมีรูปแบบการทำงานแบบเก่าอยู่ ซึ่งทางผู้จัดทำเล็งเห็นว่าสามารถนำความรู้ที่เรียนมาประกอบกับการนำซอฟต์แวร์มาพัฒนาเพื่อช่วยในการอำนวยความสะดวกของการทำงานของพนักงาน จะสามารถใช้พนักงานที่น้อยลงและการจัดการงานที่เพิ่มมากขึ้น จะช่วยลดต้นทุนของการบริหารลดลงและการใช้งานซอฟต์แวร์นั้นจะทำได้โดยง่าย ลดความซับซ้อน เพื่อให้สะดวกต่อการใช้งานของพนักงานที่ไม่ถนัดในด้านการใช้เทคโนโลยี

สำหรับสิ่งที่ได้จากการทำงานชิ้นนี้นับเป็นประสบการณ์ที่ได้นำความรู้ที่ได้เรียนจากวิชา object oriented analysis and design ตั้งแต่การกำหนดความต้องการของลูกค้า การออกแบบสถาปัตยกรรมของระบบ และการพัฒนาไปจนถึงการทดสอบของระบบ รวมไปถึงทางผู้จัดทำมีความรู้การจากใช้ Framework Laravel , PHP , HTML , CSS , Bootstrap และอีกหลายอย่างที่ไม่ได้กล่าวถึง ซึ่งในช่วงแรกนับเป็นเรื่องที่ต้องศึกษาหลายอย่างพร้อมกันเนื่องจากมีเวลาจำกัดเนื่องด้วยการเรียน สุดท้ายก็สามารถพัฒนาเว็บนี้จนสำเร็จคล่อง แม้ในระหว่างการทำงานอาจจะพบกับปัญหาที่ไม่เคยเจอที่ต้องแก้ไข แต่นั่นนับการฝึกทักษะการแก้ปัญหาให้กับทางผู้จัดทำ และต้องทำการศึกษากระบวนการทำงานของโรงแรมเพื่อให้ระบบที่ได้มีความเหมือนจริงมากที่สุด และทางผู้จัดทำมีความเป็นอยู่อย่างยิ่งว่าระบบที่ได้ออกมานั้นจะสามารถแก้ปัญหาการทำงานและการจัดการต่างๆโรงแรมเพื่อให้มีความสะดวกในการใช้งานอย่างยิ่ง

บรรณานุกรม

Bootstrap

แหล่งที่มา <http://getbootstrap.com/>

Bootstrap Template

แหล่งที่มา <http://bootstrapzero.com/>

Cooler Theme

แหล่งที่มา <https://color.adobe.com/>

การแบ่งแผนกงานในโรงแรม

แหล่งที่มา <http://th.jobsdb.com/thth/articles/แบ่งแผนกงานในโรงแรม>

โครงสร้างโรงแรม

แหล่งที่มา <http://nokza405.blogspot.com/2012/01/blog-post.html>

แผนภาพ class diagram

แหล่งที่มา <https://www.lucidchart.com/>

แผนภาพ Application Architecture

แหล่งที่มา <https://cacoo.com/>

แผนภาพ use case diagram

แหล่งที่มา <https://app.genmymodel.com/>

PHP Framework

แหล่งที่มา <http://laravel.com/>

Login Template

แหล่งที่มา <http://www.bootply.com/101498>

Random Password

แหล่งที่มา <http://stackoverflow.com/questions/4356289/php-random-string-generator>

Auto Refresh

แหล่งที่มา <http://www.sitepoint.com/auto-refresh-div-content-jquery-ajax/>