

## Architectural Design, UML diagrams & Implementation Plan

### ระบบจองที่นั่งร้านอาหารแบบออนไลน์

สมาชิกกลุ่ม :	นางสาว รินทร์ภัส	เลิศธนกุลพงษ์	รหัสนักศึกษา 55011049	Sec 2
	นางสาว วชิรญาณ	ตันติวัฒนารมย์	รหัสนักศึกษา 55011067	Sec 2

---

## Design

### Problem Analysis

เหตุผล : เนื่องจากคนในปัจจุบันอยู่กับการใช้งาน Internet เป็นเวลานาน และการจองร้านอาหารต้องมีการโทรไปเพื่อจองกับที่ร้าน หรือต้องไปจองที่ร้านด้วยตนเอง จึงได้มีการแก้ปัญหานี้โดยการจำลองสิ่งต่างๆที่ใช้ในการจองมาเป็น abstraction ดังนี้

#### Abstraction

- ปฏิทิน ซึ่งประกอบด้วย วัน/เวลา ที่มีคนจอง และ วัน/เวลา ที่ที่นั่งว่าง
- ร้านอาหาร ซึ่งประกอบด้วย ชื่อร้าน, ที่อยู่ร้าน, จำนวนที่นั่งในร้าน, เวลาเปิด/ปิดร้าน และ เบอร์โทรศัพท์ติดต่อร้าน
- จัดการการจอง ประกอบไปด้วยรายละเอียดการจองต่างๆ โดยสามารถจองที่นั่ง, แก้ไขการจอง, ลบการจอง และ

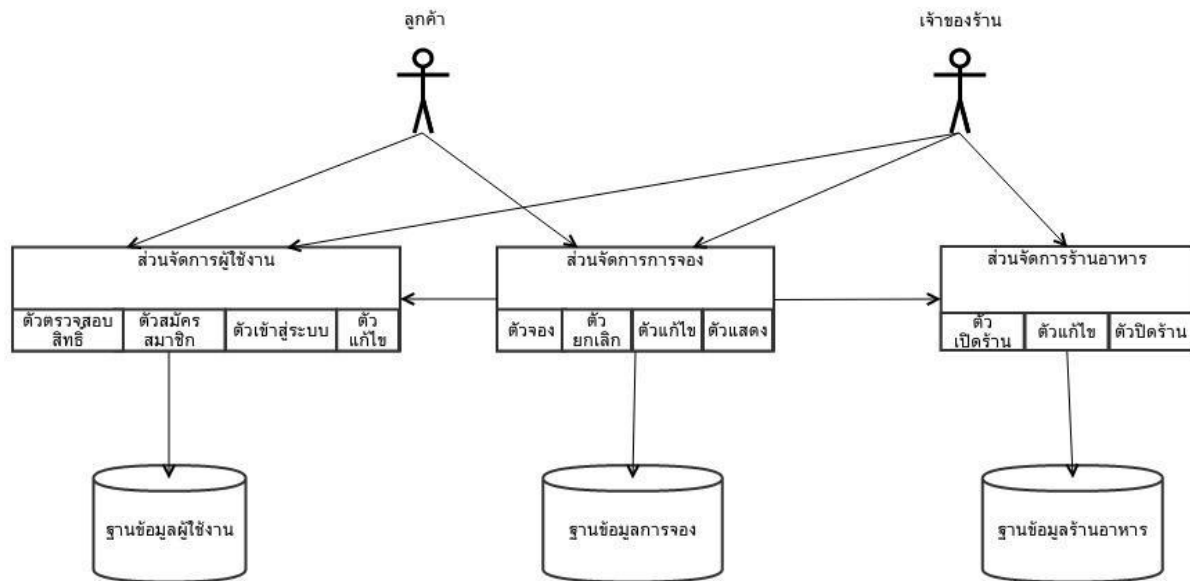
แสดงรายการจองได้

- ลูกค้า ซึ่งประกอบด้วย ชื่อ, E-mail และรายการจองของลูกค้า

#### Component

- ตัวจัดการการสมัครสมาชิก
- ตัวจัดการการเข้าสู่ระบบ
- ตัวตรวจสอบสิทธิ์ผู้ใช้
- ตัวจัดการการจอง
- ตัวยกเลิกการจอง
- ตัวดูรายการจอง
- ตัวแก้ไขการจอง
- ตัวจัดการการแก้ไขข้อมูลส่วนตัว
- ตัวจัดการการแก้ไขข้อมูลร้าน
- ตัวจัดการการเปิดร้าน
- ตัวจัดการการปิดร้าน

## Application Architecture



แผนภาพอธิบายภาพรวมของระบบ

- ส่วนจัดการผู้ใช้งาน

ในส่วนนี้จะประกอบด้วย ตัวสมัครสมาชิก ตัวจัดการการเข้าสู่ระบบ ตัวตรวจสอบสิทธิ์ผู้ใช้ และ ตัวจัดการการแก้ไขข้อมูลของผู้ใช้งาน โดยส่วนจัดการผู้ใช้งานนี้จะเป็นตัวติดต่อฐานข้อมูลของผู้ใช้งาน และเชื่อมต่อกับส่วนจัดการการจอง(ต้องใช้ข้อมูลจากส่วนนี้เพื่อทำการจอง) โดยที่ทั้งลูกค้าและเจ้าของร้านสามารถจัดการส่วนนี้ได้

- ส่วนจัดการการจอง

ในส่วนนี้จะประกอบด้วย ตัวจอง ตัวจัดการการยกเลิกการจอง ตัวจัดการการแก้ไขการจอง และตัวแสดงรายการจองที่ยังไม่ถึงเวลา ซึ่งส่วนนี้จะติดต่อกับฐานข้อมูลการจอง และติดต่อกับส่วนจัดการผู้ใช้งาน และส่วนจัดการร้านอาหาร เพราะในการจัดการการจองจำเป็นต้องใช้ข้อมูลจากทั้ง 2 ส่วนที่ได้กล่าวไป โดยส่วนนี้สามารถถูกใช้บริการได้จากทั้งลูกค้าและเจ้าของร้าน

- ส่วนจัดการร้านอาหาร

ในส่วนนี้จะประกอบด้วยตัวจัดการการเปิดร้าน ตัวจัดการการแก้ไขข้อมูลร้าน และตัวจัดการการปิดร้าน(ลบร้านออกจากระบบ) ซึ่งส่วนนี้จะติดต่อกับฐานข้อมูลร้านอาหารและส่วนจัดการการจอง(ต้องใช้ข้อมูลจากส่วนนี้เพื่อทำการจอง) และสามารถถูกใช้บริการได้จากเจ้าของร้านคนเดียว

- ฐานข้อมูลผู้ใช้งาน

ประกอบด้วยชื่อผู้ใช้ รหัสผ่าน E-mail และเบอร์โทรศัพท์ของผู้ใช้งาน โดยเชื่อมต่อกับส่วนจัดการผู้ใช้งาน

- ฐานข้อมูลการจอง

ประกอบด้วยชื่อผู้จอง และข้อมูลการจองของผู้ใช้นั้นๆ โดยเชื่อมต่อกับส่วนจัดการการจอง

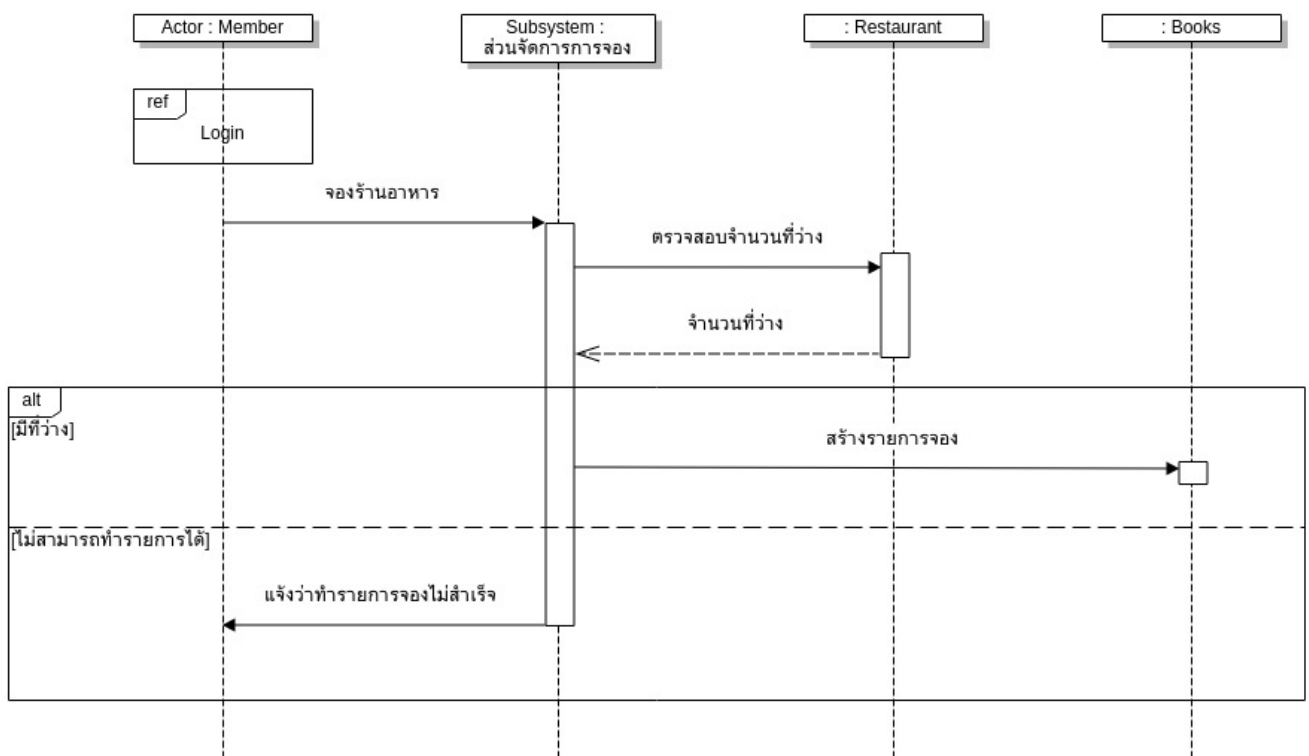
- ฐานข้อมูลร้านอาหาร

ประกอบด้วยชื่อร้านอาหาร จำนวนที่นั่งในร้านอาหาร มุมในร้านอาหาร ชื่อเจ้าของร้าน ที่อยู่ร้าน เวลาเปิด/ปิดร้าน และเบอร์โทรศัพท์ที่ติดต่อร้าน โดยเชื่อมต่อกับส่วนจัดการร้านอาหาร

## Subsystem / Component

- ส่วนจัดการผู้ใช้งาน
  - ตัวจัดการการสมัครสมาชิก : มีหน้าที่จัดการเก็บข้อมูลต่างๆของการสมัครสมาชิกของผู้ใช้งาน เข้าสู่ฐานข้อมูลผู้ใช้งาน
  - ตัวจัดการการเข้าสู่ระบบ : มีหน้าที่เปรียบเทียบบัญชีผู้ใช้งานกับระบบฐานข้อมูลผู้ใช้งานเพื่อให้ผู้ใช้เข้าถึงข้อมูลส่วนตัวของตัวเองอย่างถูกต้อง
  - ตัวจัดการการแก้ไขข้อมูลผู้ใช้ : มีหน้าที่รับข้อมูลที่เปลี่ยนแปลงหรือข้อมูลเพิ่มเติมของผู้ใช้ส่งไปแก้ไขในฐานข้อมูลของผู้ใช้นั้น
  - ตัวตรวจสอบสิทธิ์ผู้ใช้ : มีหน้าที่ตรวจสอบสิทธิ์ของผู้ใช้ว่าอยู่ในฐานะลูกค้าหรือเจ้าของร้าน โดยตรวจสอบจากจำนวนร้านในฐานข้อมูลผู้ใช้ ถ้าหากเป็นผู้ใช้ที่จำนวนร้านเป็น 0 จะได้สิทธิ์เป็นลูกค้าซึ่งไม่สามารถจัดการร้านใดๆได้ แต่ถ้าหากมีจำนวนร้านมากกว่า 1 จะได้สิทธิ์เจ้าของร้าน มีสิทธิ์ในการจัดการร้านของตัวเองได้
- ส่วนจัดการการจอง
  - ตัวจอง : มีหน้าที่เก็บข้อมูลการจองร้านอาหารของผู้ใช้งานนั้นๆเข้าสู่ระบบฐานข้อมูลการจอง และเปรียบเทียบเวลาของการจอง หากเวลาที่จองมีการตรงกันกับข้อมูลที่มีอยู่แล้วในระบบ หรือจำนวนที่นั่งไม่พอในช่วงเวลานั้น จะไม่อนุญาตให้การจองสำเร็จได้
  - ตัวจัดการการยกเลิกการจอง : มีหน้าที่ลบข้อมูลการจองที่มีในระบบที่ผู้ใช้ไม่ต้องการออกจากระบบฐานข้อมูลการจอง ในส่วนนี้มีผลกระทบกับการแสดงรายการจองของผู้ใช้งาน ทั้งลูกค้าและเจ้าของร้าน
  - ตัวจัดการการแก้ไขการจอง : มีหน้าที่รับข้อมูลที่เปลี่ยนแปลงจากผู้ใช้งาน เช่น วัน เวลา จำนวนโต๊ะ และมุมที่ต้องการส่งไปแก้ไขฐานข้อมูลการจองของผู้ใช้ ซึ่งทำให้มีผลกระทบกับการแสดงรายการการจองของผู้ใช้งาน ทั้งลูกค้าและเจ้าของร้าน
  - ตัวดูรายการจอง : มีหน้าที่แสดงข้อมูลจากฐานข้อมูลการจอง โดยในฝั่งของลูกค้าจะแสดงรายการที่ลูกค้าได้จองไว้เท่านั้น (ไม่รวมการจองที่ผ่านมา) ส่วนในฝั่งของเจ้าของร้านจะแสดงรายการที่มีลูกค้าจองในแต่ละวันและแต่ละช่วงเวลาพร้อมทั้งรายละเอียดต่างๆที่ลูกค้าระบุ
- ส่วนจัดการร้านอาหาร
  - ตัวจัดการการเปิดร้าน : มีหน้าที่จัดการเก็บข้อมูลต่างๆของร้านอาหารที่ต้องการเปิดในระบบ เข้าสู่ฐานข้อมูลร้านอาหาร
  - ตัวจัดการการแก้ไขข้อมูลร้าน : มีหน้าที่รับข้อมูลที่เปลี่ยนแปลงหรือข้อมูลเพิ่มเติมของร้านอาหาร เช่น เวลาเปิด/ปิดร้าน วันทำการ หรือที่อยู่ร้าน(โดยที่ไม่สามารถแก้ไขชื่อร้านได้) ส่งไปแก้ไขในฐานข้อมูลร้านอาหาร
  - ตัวจัดการการปิดร้าน : มีหน้าที่ลบร้านอาหารการออกจากฐานข้อมูลร้านอาหาร
- ฐานข้อมูลผู้ใช้งาน
  - ตารางข้อมูลผู้ใช้งาน : ในตารางนี้จะประกอบไปด้วย ชื่อผู้ใช้ รหัสผ่าน(ที่เข้ารหัสแล้ว) E-mail และเบอร์โทรศัพท์ของผู้ใช้งาน ซึ่งใช้เป็น ฐานข้อมูลในการเข้าถึงข้อมูลของผู้ใช้นั้นๆ
- ฐานข้อมูลการจอง
  - ตารางข้อมูลการจอง (join ระหว่างตารางผู้ใช้งานกับตารางร้านอาหาร) : ในตารางนี้จะประกอบด้วย id ผู้จอง, id ร้านอาหาร และข้อมูลการจองของผู้ใช้นั้นๆ ได้แก่ วัน เวลาเข้า เวลาออก จำนวนที่นั่ง และมุมที่ต้องการ ซึ่งใช้เป็น ฐานข้อมูลของการจองของลูกค้า แต่ตารางนี้สามารถเข้าถึงได้จากทั้งสองฝั่งของผู้ใช้งาน คือ ลูกค้า และ ร้านอาหาร โดยลูกค้าจะเข้าถึงข้อมูลที่เป็นของตัวเอง ส่วนร้านอาหารจะเข้าถึงข้อมูลที่ตรงกับร้านอาหารของตนเองได้
- ฐานข้อมูลร้านอาหาร
  - ตารางจัดการรูปภาพของร้าน : ในตารางนี้จะเป็นส่วนจัดการการเก็บรูปภาพของร้านอาหารต่างๆ โดยมี id ร้านอาหาร และ path ไปยังที่เก็บรูปนั้นๆใน server
  - ตารางเก็บข้อมูลมุมในร้าน : ในตารางนี้จะเป็นตารางที่ประกอบด้วย rid ร้าน และ มุมที่มีในร้านอาหารนั้นๆ โดยมีจำนวนโต๊ะที่แน่นอนในแต่ละมุม

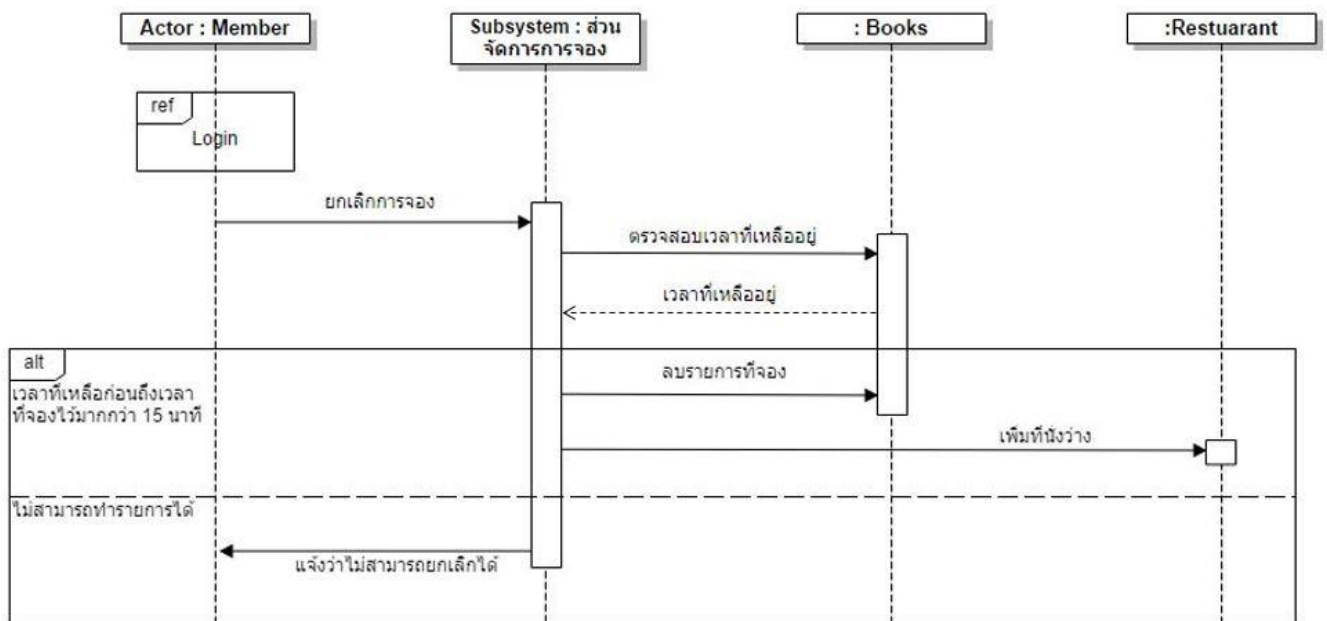
## Sequence diagram การจอง



จาก Sequence diagram ด้านบน อธิบายถึงสถานการณ์การจอง โดยมี Actor ที่เป็นสมาชิก, ส่วนจัดการการจอง ที่เป็น subsystem, Restaurant และ Books ที่เป็น Data mapper ซึ่งแทนข้อมูลจาก database

โดยเมื่อลูกค้าทำการจองร้านอาหาร ส่วนจัดการการจองจะไปตรวจสอบจำนวนที่ว่างในร้านอาหารจาก Restaurant ซึ่งเป็น Data mapper แทนข้อมูลของ Restaurant ใน Database ซึ่ง Restaurant ก็จะตอบกลับมาเป็นจำนวนที่ว่างในร้าน ถ้ามีที่ว่างส่วนจัดการการจองก็จะไปสร้างรายการจองเก็บลงใน Database ผ่านทาง Books ซึ่งเป็น Data mapper แต่ถ้าจำนวนที่ว่างไม่พอ ทำให้ไม่สามารถทำรายการจองได้ ส่วนจัดการการจองก็จะแจ้งว่าทำรายการจองไม่สำเร็จ

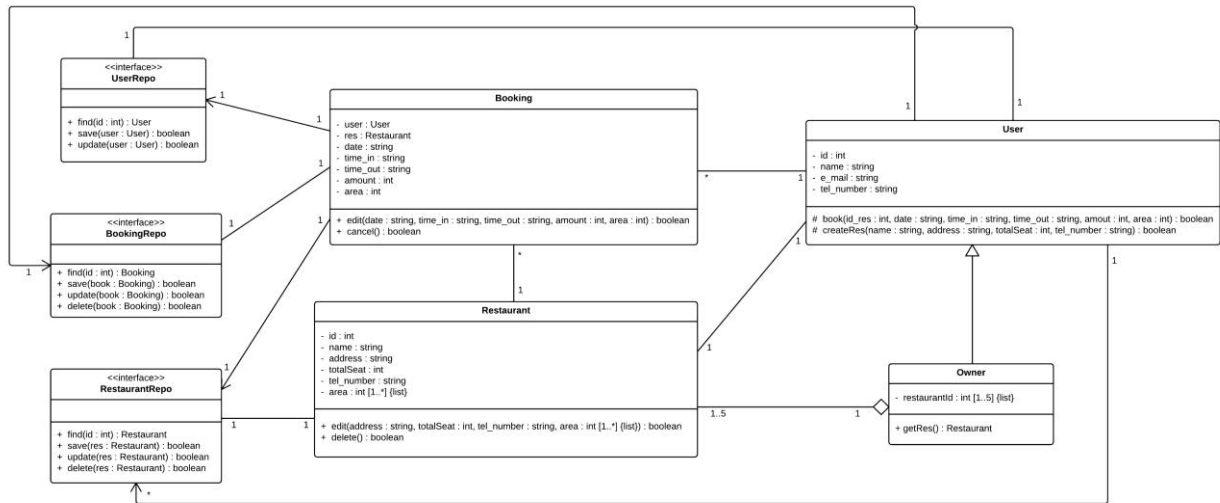
## Sequence diagram ยกเลิกการจอง



จาก Sequence diagram ด้านบน อธิบายถึงสถานการณ์ในการยกเลิกการจอง โดยมี Actor ที่เป็นสมาชิก, ส่วนจัดการการจอง ที่เป็น subsystem, Restaurant และ Books ที่เป็น Data mapper ซึ่งแทนข้อมูลจาก database โดยเมื่อลูกค้าทำการยกเลิกรายการจอง ส่วนจัดการการจองจะไปตรวจสอบเวลาที่เหลือก่อนถึงเวลาที่ได้จองไว้ใน Books ที่เป็น Data mapper แทนข้อมูลใน Database หลังจากนั้นหากตรวจสอบพบว่ามียังเวลาที่เหลือก่อนถึงเวลาที่ได้จองไว้มากกว่า 15 นาที ส่วนจัดการการจองก็จะไปลบรายการที่ได้จองไว้ใน Database ผ่านทาง Books และก็จะไปเพิ่มที่นั่งว่างในส่วนของRestaurant แต่ถ้าหากเวลาที่เหลืออยู่ก่อนถึงเวลาที่ได้จองไว้ไม่ถึง 15 นาที ส่วนจัดการการจองจะทำหน้าที่ส่งข้อความแจ้งแก่ Actor ว่าไม่สามารถยกเลิกการจองนี้ได้

## Domain classes

### Class diagrams



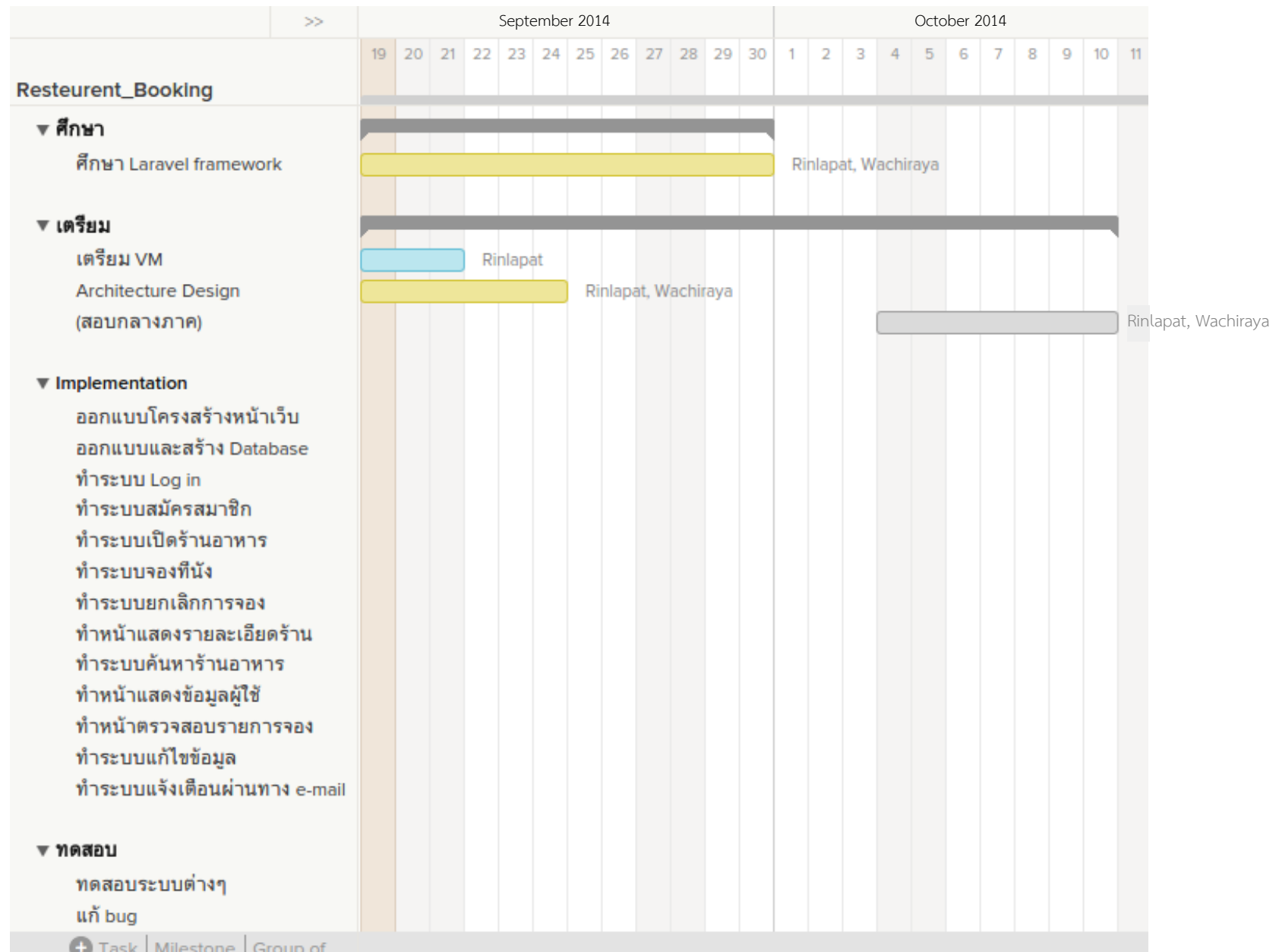
จาก class diagram ด้านบน อธิบายได้ว่า

- ส่วน **BookingRepo**, **RestaurantRepo** และ **UserRepo** เป็นตัวเชื่อมต่อกับส่วน Database
- ส่วน **User**, **Books** และ **Restaurant** เป็น Model ของข้อมูลที่เราสนใจใน problem domain ซึ่งสามารถเอาข้อมูลข้างในออกมาได้ผ่าน method `get` (ซึ่งในที่นี้ไม่ได้เขียนลงไป) และเขียนข้อมูลผ่าน method `set` (ซึ่งในที่นี้ไม่ได้เขียนลงไป)
- ระหว่าง Model และ Repo จะมีเส้นเชื่อมถึงกันทุกคู่ เพื่อให้ Repo รู้จัก Object ชนิดเดียวกับ Model
- **Owner**(เจ้าของร้าน) inherit **User**(ผู้ใช้) เนื่องจาก เจ้าของร้านเป็นผู้ใช้ที่มีการเปิดร้านอาหารอยู่ในบัญชีผู้ใช้
- **Restaurant** เป็น part of **Owner** โดยที่ เจ้าของร้าน(**Owner**) 1 คน มีร้านอาหารได้ไม่เกิน 5 ร้าน
- ลูกค้า(**User**) 1 คน สามารถมีได้หลายการจอง(**Books**)

## Deployment

- ใช้ VM 1 ตัว
  - ลง LAMP stack server บน VM (ลง Ubuntu server บน VM, ใช้ apache2 เป็น Application server, ใช้ MySQL เป็นตัวจัดการฐานข้อมูล และลง php5 (libapache2-mod-php5 php5-mcrypt)
  - ใช้ phpmyadmin เป็นตัวช่วยจัดการฐานข้อมูล
  - ใช้ Laravel เป็น Framework ที่ใช้พัฒนา (PHP5)
  - ใช้ Composer เป็น PHP Dependency manager
  - ใช้ Bootstrap เป็น Frontend Framework (HTML, CSS, และ JavaScript)
  - ระบบทุกระบบทำงานอยู่บน VM Domain name : eatwme.cloudapp.net
-

## Implementation plan



(มีต่อ)



