

# Groot

Daily-exercise-log  
(ระบบแนะนำท่ากายบริหารออนไลน์)

Git hub repository: Daily-exercise-log

## ผู้จัดทำ

นายกฤษฎชนน	ศรียะมล	รหัสนักศึกษา	55010028	sec 1
นายวุฒิกรณ	เชยโศคา	รหัสนักศึกษา	55011178	sec 2

Object Oriented Analysis and Design  
ภาคการศึกษาที่ 1/2557

## บทคัดย่อ

**Daily exercise log web** เป็นเว็บที่ให้บริการคำแนะนำการออกกำลังกายด้วยท่ากายบริหารต่างๆ โดยผู้ใช้งานต้อง login เพื่อบันทึกข้อมูลที่เกี่ยวข้องกับการออกกำลังกาย เช่น ท่ากายบริหาร ระยะเวลาที่ใช้ในการออกกำลังกาย

## บทนำและรายละเอียดการวิเคราะห์หัวข้อ

ในปัจจุบัน คนส่วนใหญ่เริ่มหันมาสนใจการออกกำลังกายโดยใช้ท่ากายบริหาร เพราะสะดวก สามารถออกกำลังกายได้ทุกที่ ทุกเวลา จึงมีแนวคิดที่จะช่วยในการส่งเสริม สนับสนุน โดยการแนะนำท่ากายบริหารที่ตรงกับความต้องการในการบริหารส่วนต่างๆ ของร่างกายเหมาะสมกับวัย ตลอดจนแนะนำท่ากายบริหารใหม่ๆ ที่รู้จักยังไม่แพร่หลาย โดยให้บริการผ่านทางเว็บไซต์ เพียงแค่สมัครเป็นสมาชิกกับทางเว็บไซต์ก็สามารถเข้าใช้งานได้ ปัจจุบันสามารถเข้าถึงอินเทอร์เน็ตได้ง่าย ผู้ใช้จึงสามารถเข้าใช้บริการของเว็บไซต์ได้สะดวกยิ่งขึ้น

## งานที่เกี่ยวข้อง

*Daily exercise log Web* ได้รับแรงบันดาลใจมาจาก *App Train* ที่เป็น app ที่แนะนำท่าออกกำลังกาย มีการบันทึกข้อมูล เพื่อให้ผู้ใช้สามารถเลือกแนวทางการออกกำลังกายเหมาะสมกับตนเอง โดย *App Train* จะสามารถทำงานได้กับระบบ *Windows Phone 8* , *Windows Phone 8.1* ฉะนั้นขอบเขตของผู้ใช้งานก็จะจำกัดอยู่ที่ผู้ใช้งานบางกลุ่มที่ใช้ *Windows Phone 8* , *Windows Phone 8.1* เท่านั้น แต่ *Daily exercise log Web* จะสามารถใช้งานได้ผ่าน *web browser* ทุก *browser* ไม่ว่าจะทำงานบนระบบไหน ซึ่งจะทำให้สะดวกต่อผู้ใช้งานมากขึ้น

## อ้างอิงเว็บ

<http://www.windowsphone.com/th-th/store/app/trainr/7971bb5f-1b13-4ce3-86e6-141eb094d705>

## ผลการวิเคราะห์ความต้องการของผู้ใช้ระบบ

### Functional requirements

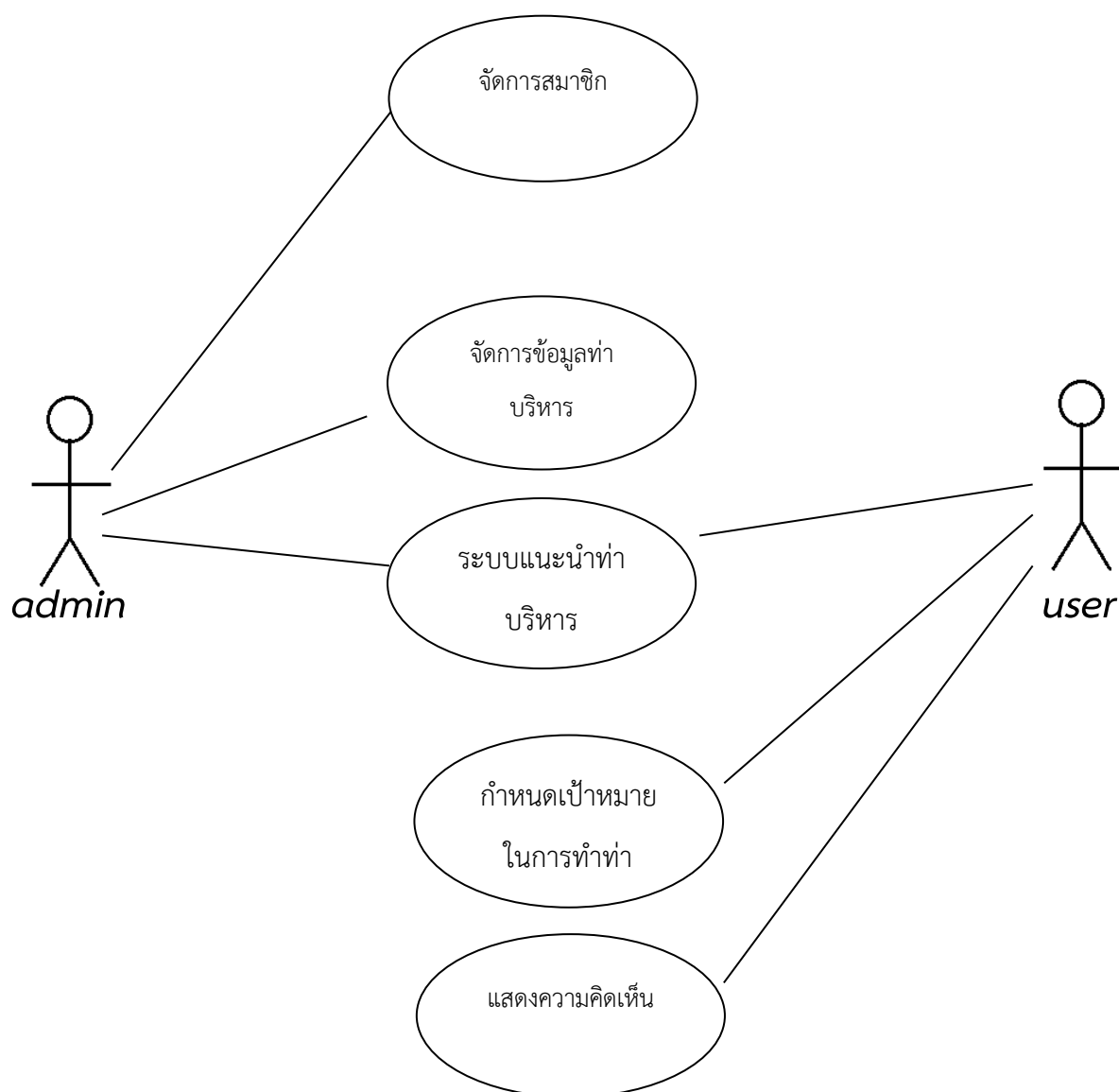
1. ระบบแสดงความคิดเห็น
2. ระบบแนะนำท่าบริหาร
3. ระบบกำหนดเป้าหมายในการทำท่าบริหาร
4. แสดงแคลอรี่ที่ใช้ไปในการทำท่าบริหาร

### Non-Functional requirements

1. พร้อมใช้งานตลอดเวลา
2. สามารถใช้งานได้ทุกแพลตฟอร์มที่รองรับบราวเซอร์
3. สามารถใช้งานได้ในทุกขนาดหน้าจอ
4. รองรับผู้ใช้งานหลายคนพร้อมกัน

## แนวทางการใช้งาน

Use case diagram



## Use case specifications

Use case name : ระบบจัดการสมาชิก

Use case purpose : ทำให้ผู้ใช้ที่เป็นผู้ดูแลระบบ จัดการสมาชิกถึงสิทธิ์การใช้งาน

Preconditions : ต้องเข้าสู่ระบบ

Postconditions : สมาชิกจะได้รับสิทธิ์การใช้งานระบบต่างๆ

Assumption : จะทำให้ระบบมีความเป็นระเบียบมากขึ้น

Limitations : ผู้ใช้ต้องอยู่ในสถานะผู้ดูแลระบบ สมาชิกจะต้องทำการลงทะเบียนไว้ก่อนแล้ว

Primary scenario :

1. ผู้ดูแลระบบ เข้าสู่ระบบ
2. เลือกการจัดการสมาชิก
3. กรอกชื่อสมาชิก
4. ระบบจะทำการลงทะเบียนเข้ากับฐานข้อมูล

Alternate scenarios :

Condition 1 : ไม่พบชื่อสมาชิกที่กรอก

- ผู้ดูแลระบบ ใส่ชื่อสมาชิกผิด
- ระบบแจ้งว่าไม่พบข้อมูลและย้อนกลับไปกรอกใหม่

Use case name : ระบบการจัดการแผนการบริหาร

Use case purpose : ทำให้สมาชิกสามารถวางแผนเรื่องทำบริหาร เพื่อที่จะบริหารได้หลายๆ ส่วน และ ทำได้อย่างถูกต้อง

Preconditions : ต้องเข้าสู่ระบบ

Postconditions : สมาชิกจะได้แผนการในการทำบริหาร

Assumption : จัดการแผนการบริหารให้เป็นระเบียบ

Limitations : -

Primary scenario :

1. เข้าสู่ระบบ
2. แผนการบริหาร
3. เลือกรูปแบบและทำบริหาร
4. ระบบจะทำการบันทึกข้อมูล

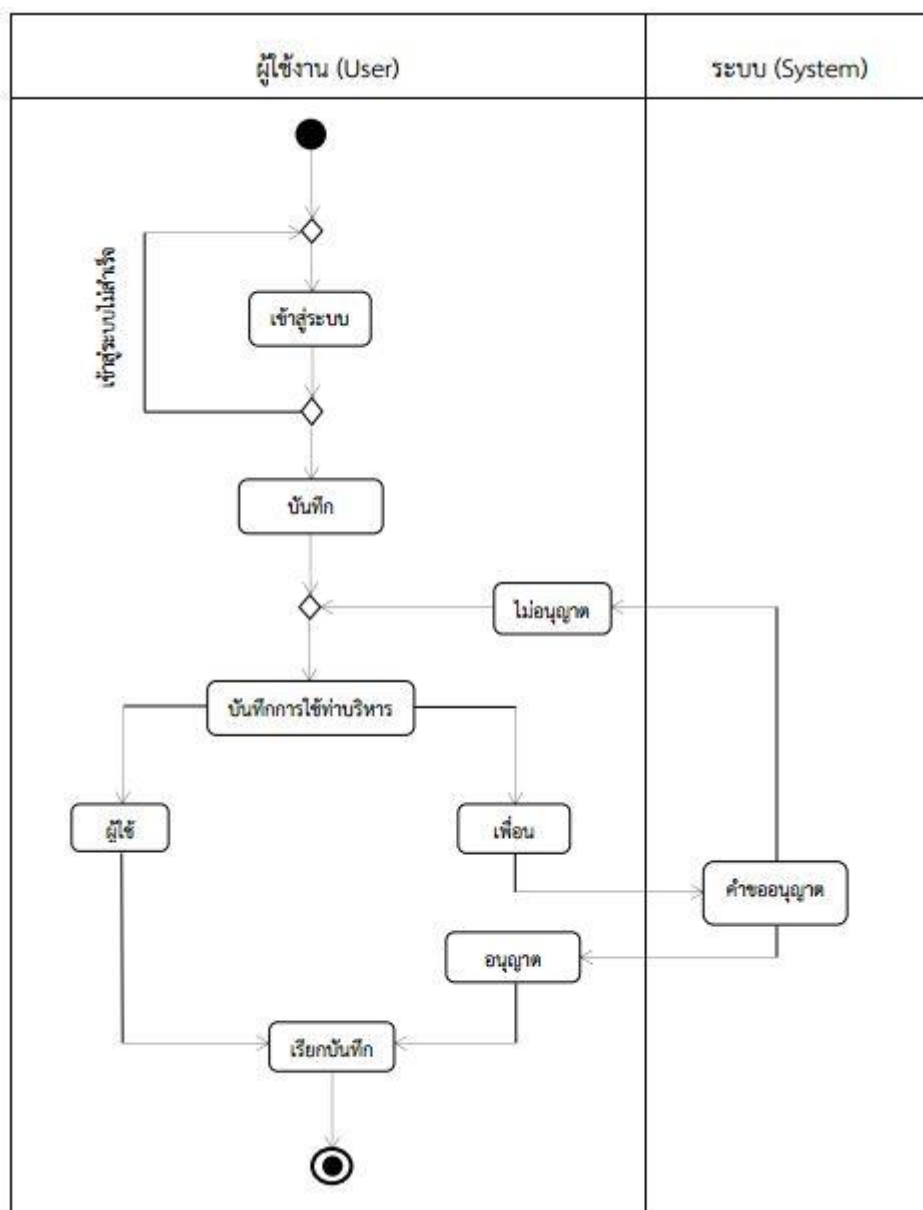
Alternate scenarios :

Condition 1 : บัญชีหรือรหัสผ่านไม่ถูกต้อง

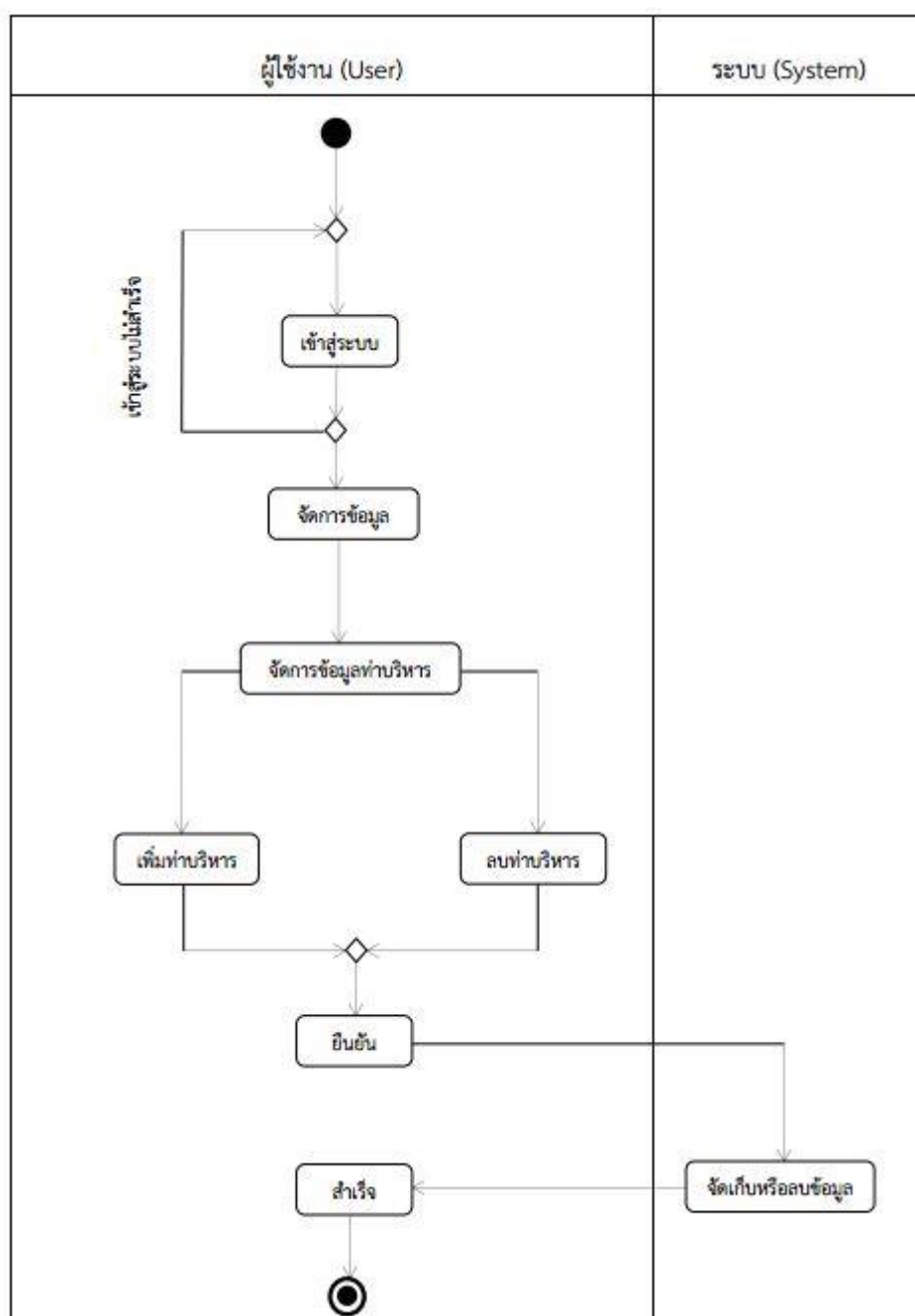
- ผู้ใช้กรอกชื่อหรือรหัสไม่ถูกต้อง
- ระบบย้อนกลับไปหน้าเข้าสู่ระบบ
- ทำตาม Primary scenario

## Activity diagram

Use case name : บันทึกการใช้ท่ากายบริหาร



Use case name : ระบบจัดการข้อมูลท่าบริหาร



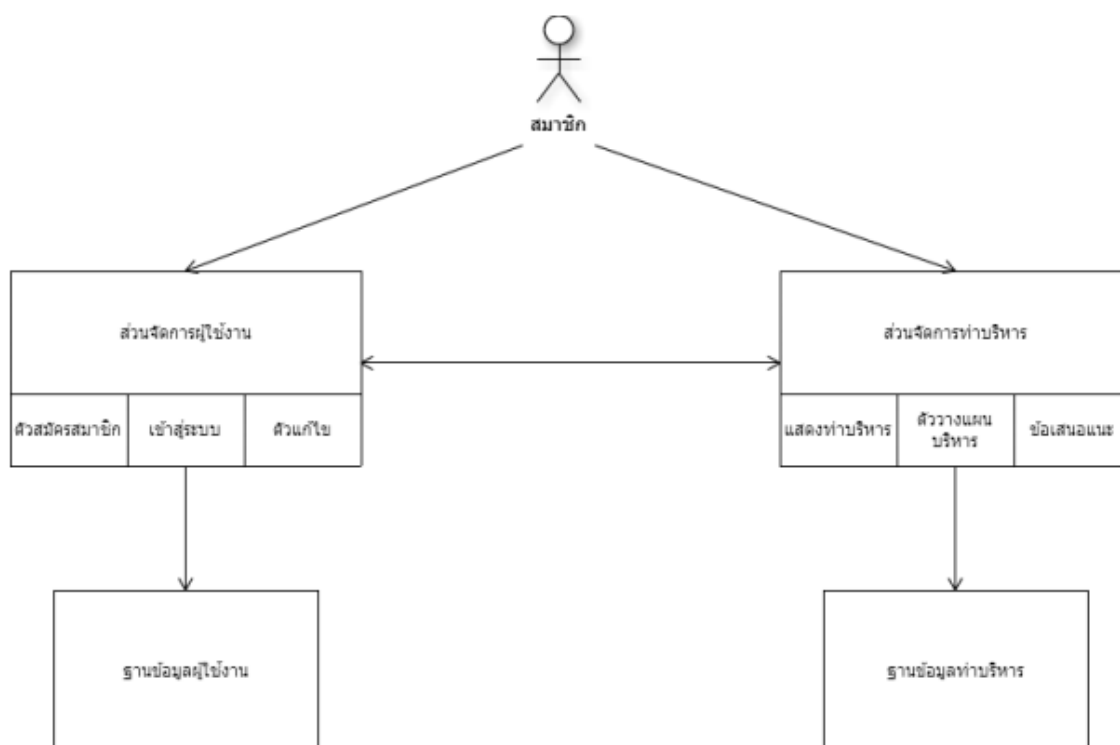


## สถาปัตยกรรมของระบบ

### Problem Analysis

- **Abstraction**
  - ทำบริหารเป็นส่วนที่ให้ผู้เลือกใช้ดูทำบริหาร
  - ผู้ใช้ ซึ่งประกอบด้วยชื่อผู้ใช้ เบอร์โทรศัพท์ facebook
  - ข้อมูลประวัติ ประกอบด้วยบันทึกการใช้ทำบริหาร
  - แผนการทำทำบริหาร เป็นตารางการทำทำบริหาร ประกอบด้วยทำบริหารซึ่งบันทึกไว้เป็นชุดๆ
- **Component**
  - ตัวจัดการการสมัครสมาชิก
  - ตัวจัดการการแก้ไขข้อมูลส่วนตัว
  - ตัวจัดการการเข้าสู่ระบบ
  - ตัวจัดการการเลือกทำบริหาร
  - ตัวจัดการแผนการทำทำบริหาร
  - ตัวจัดการทำบริหาร

## Application Architecture



- **ส่วนจัดการผู้ใช้งาน**

ประกอบด้วย ตัวจัดการสมัครสมาชิก ตัวจัดการการเข้าสู่ระบบ และตัวจัดการแก้ไขข้อมูลผู้ใช้ ส่วนผู้ใช้งานจะต้องเป็นตัวติดต่อกับฐานข้อมูล และเชื่อมต่อกับส่วนจัดการทำบริหาร

- **ส่วนจัดการทำบริหาร**

ประกอบด้วย ตัวจัดการแสดงทำบริหาร ตัวจัดการแผนการทำทำบริหาร และตัวจัดการทำบริหาร ส่วนจัดการทำบริหารจะเป็นส่วนติดต่อกับฐานข้อมูลทำบริหาร

- **ฐานข้อมูลผู้ใช้**

ประกอบด้วย ชื่อผู้ใช้ รหัสผ่าน เบอร์โทรศัพท์ Facebook โดยเชื่อมต่อกับส่วนจัดการผู้ใช้งาน

- **ฐานข้อมูลทำบริหาร**

ประกอบด้วย ทำบริหาร แผนการบริหาร ข้อเสนอแนะ โดยเชื่อมต่อกับส่วนจัดการทำบริหาร

## Subsystems / Components

### ส่วนจัดการผู้ใช้งาน

- ตัวจัดการสมัครสมาชิก : มีหน้าที่จัดการเก็บข้อมูลต่างๆของการสมัครสมาชิกของผู้ใช้งาน เข้าสู่ฐานข้อมูลผู้ใช้งาน
- ตัวจัดการการเข้าสู่ระบบ : มีหน้าที่เปรียบเทียบบัญชีผู้ใช้งานกับฐานข้อมูลผู้ใช้งานเพื่อให้ผู้ใช้งานเข้าถึงข้อมูลส่วนตัวผู้ใช้
- ตัวจัดการการแก้ไขข้อมูลผู้ใช้ : มีหน้าที่รับข้อมูลที่เปลี่ยนแปลงของผู้ใช้ส่งไปแก้ไขในฐานข้อมูลของผู้ใช้

### ส่วนจัดการทำบริหาร

- ตัวจัดการการแสดงผลทำบริหาร : มีหน้าที่เรียกดูทำบริหาร ซึ่งจะเรียกข้อมูลในฐานข้อมูลทำบริหาร
- ตัวจัดการการวางแผนบริหาร : มีหน้าที่จัดเก็บข้อมูลแผนการทำทำบริหาร ที่ผู้ใช้วางแผนไว้ เก็บลงในฐานข้อมูลทำบริหาร
- ตัวจัดการข้อเสนอแนะ : มีหน้าที่รับข้อเสนอแนะในการทำทำบริหารส่งไปเก็บในฐานข้อมูล

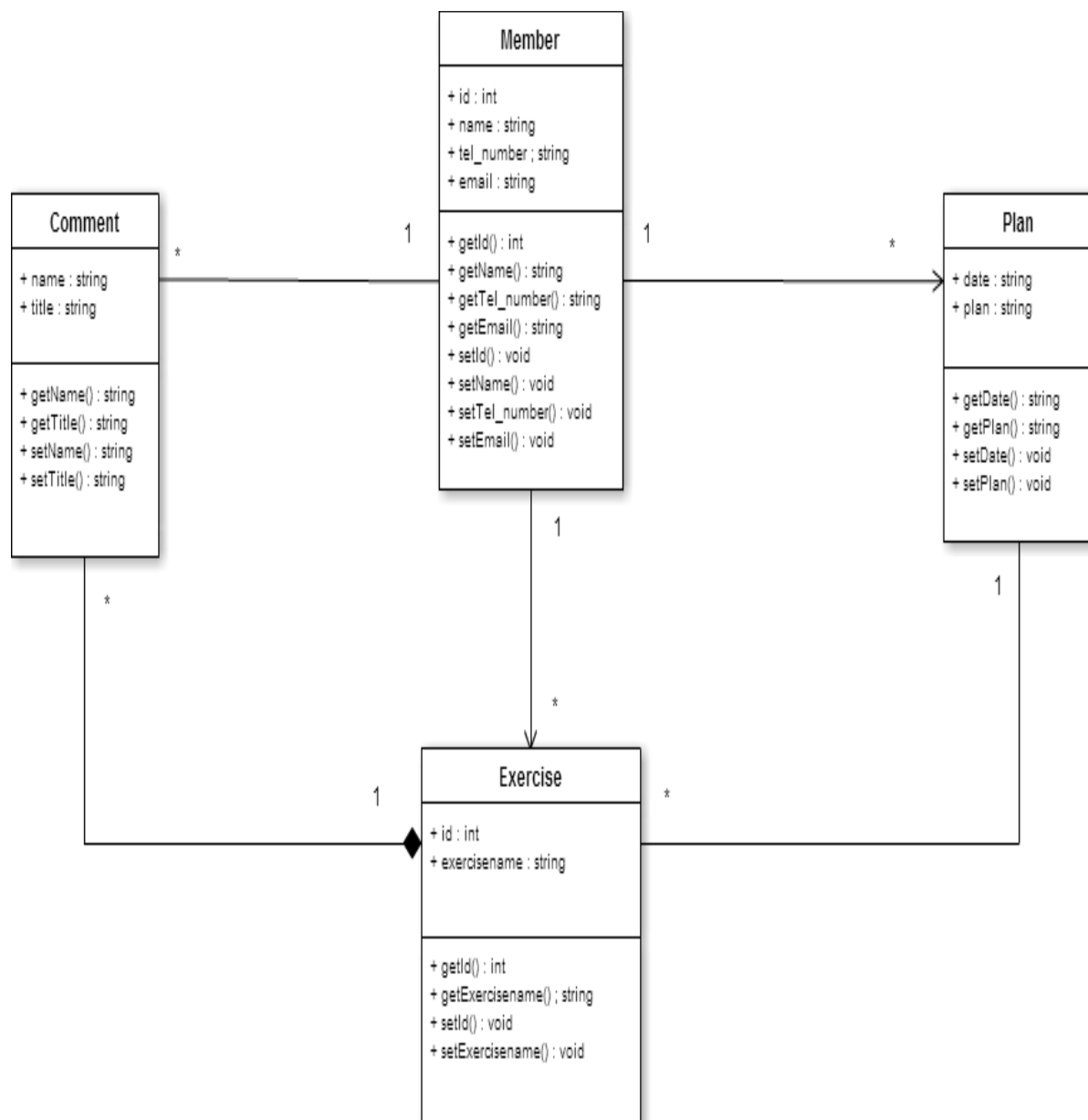
### ฐานข้อมูลผู้ใช้งาน

- ข้อมูลผู้ใช้งาน : ภายในจะเก็บข้อมูล ชื่อ รหัสผ่าน เบอร์โทรศัพท์ facebook ประวัติการทำทำบริหาร

### ฐานข้อมูลทำบริหาร

- ข้อมูลทำบริหาร : ภายในจะเก็บข้อมูล ทำบริหาร ข้อเสนอแนะ

## Class diagram for domain objects

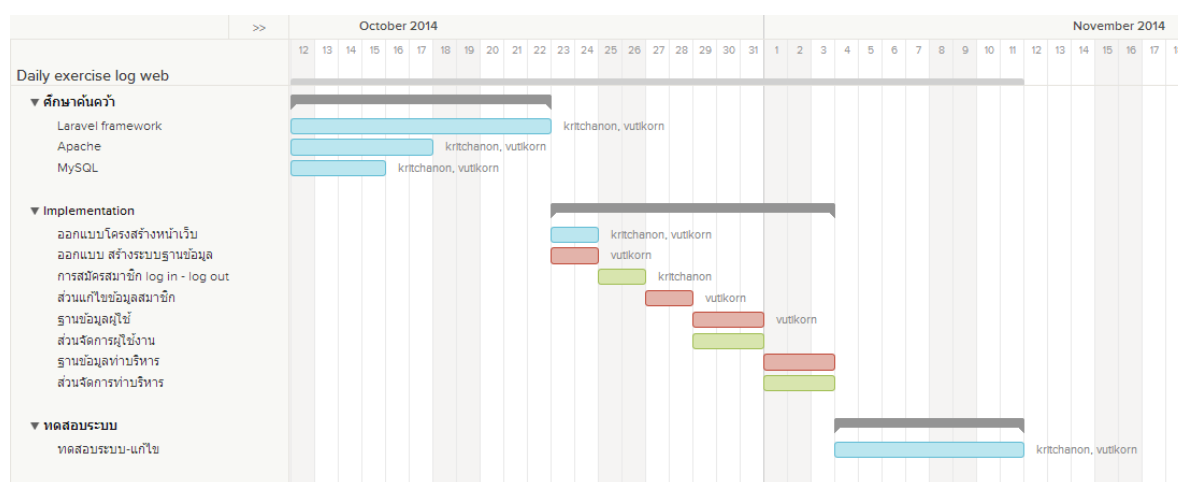


## Deployment

Software และ Technology ที่ใช้ ได้แก่

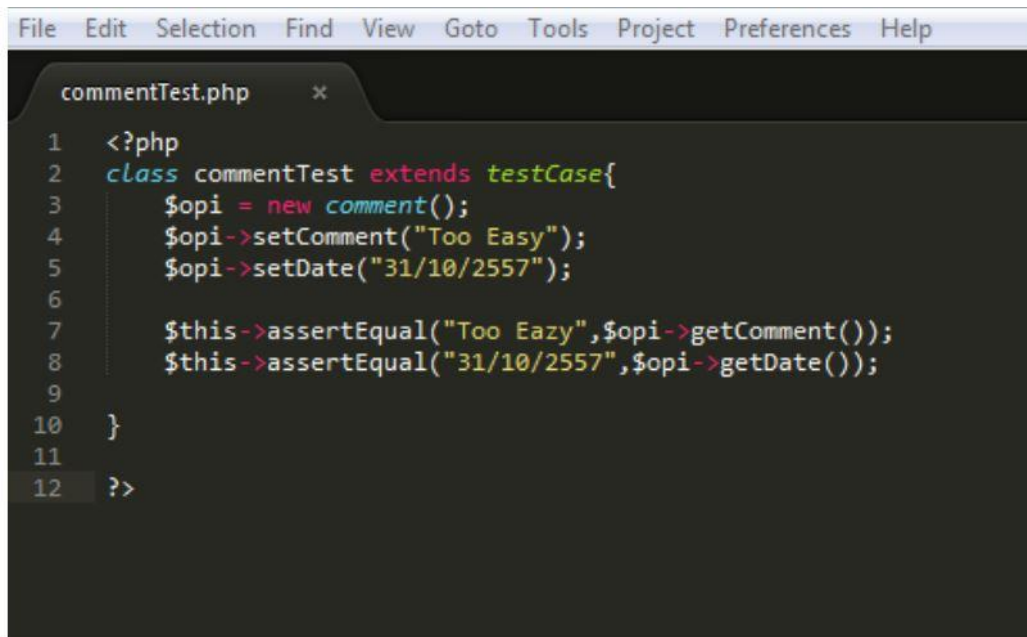
- Ubuntu 14.10 บน VM
- Apache เป็น application sever - Laravel เป็น Framework ที่ใช้พัฒนา(PHP5)
- phpMyAdmin เป็นตัวช่วยจัดการฐานข้อมูล
- Bootstrap เป็น Frontend Framework (HTML, CSS, และ JavaScript)
- Composer เป็น PHP Dependency manager

## Implementation Plan



## ผลการทดสอบซอฟต์แวร์

## Test comment bootstrap-3.2.0-dist/test/commentTest.php

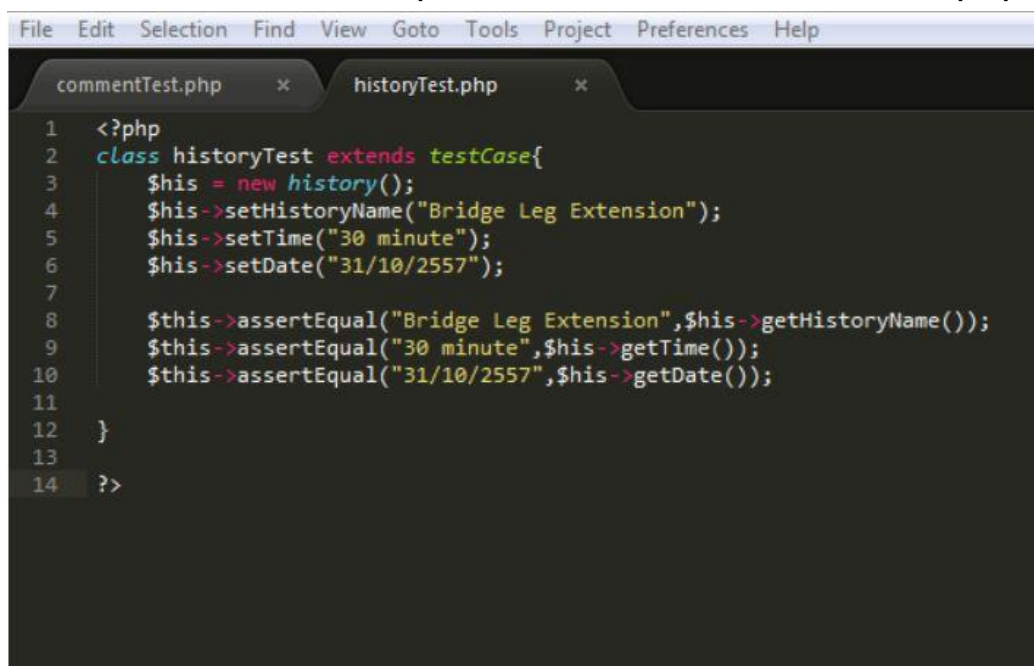


```
File Edit Selection Find View Goto Tools Project Preferences Help

commentTest.php x

1  <?php
2  class commentTest extends testCase{
3      $opi = new comment();
4      $opi->setComment("Too Easy");
5      $opi->setDate("31/10/2557");
6
7      $this->assertEqual("Too Easy",$opi->getComment());
8      $this->assertEqual("31/10/2557",$opi->getDate());
9
10 }
11
12 ?>
```

## Test comment bootstrap-3.2.0-dist/test/commentTest.php



```
File Edit Selection Find View Goto Tools Project Preferences Help

commentTest.php x  historyTest.php x

1  <?php
2  class historyTest extends testCase{
3      $his = new history();
4      $his->setHistoryName("Bridge Leg Extension");
5      $his->setTime("30 minute");
6      $his->setDate("31/10/2557");
7
8      $this->assertEqual("Bridge Leg Extension",$his->getHistoryName());
9      $this->assertEqual("30 minute",$his->getTime());
10     $this->assertEqual("31/10/2557",$his->getDate());
11
12 }
13
14 ?>
```

## Evaluation

การทดลองที่ 1 ทดลองการแสดงความคิดเห็น

- จุดประสงค์ของการทดลอง

เพื่อทดสอบว่าระบบสามารถจัดเก็บข้อมูลลงฐานข้อมูลได้

ถูกต้อง

- สิ่งที่จะวัด

การจัดเก็บข้อมูลการแสดงความคิดเห็น

- วิธีการทดลอง

- เข้าสู่ระบบ

- เลือกทำบริหาร

- แสดงความคิดเห็น

- กด “ยืนยัน”

- สิ่งที่ต้องใช้ในการทดลอง

- ชื่อ และ รหัสผ่าน เพื่อเข้าสู่ระบบ

- ผลที่ได้จากการทดลอง

- (ยังไม่สามารถทำการทดลองได้)

การทดลองที่ 2 ทดลองบันทึกประวัติการทำท่าบริหาร

- จุดประสงค์ของการทดลอง

เพื่อทดสอบว่าระบบสามารถจัดเก็บข้อมูลลงฐานข้อมูลได้

ถูกต้อง

- สิ่งที่จะวัด

การจัดเก็บข้อมูลการทำท่าบริหาร

- วิธีการทดลอง

- เข้าสู่ระบบ
- เลือกข้อมูลผู้ใช้
- เลือกบันทึกการทำท่าบริหาร
- กด “ยืนยัน”

- สิ่งที่ต้องใช้ในการทดลอง

- ชื่อ และ รหัสผ่าน เพื่อเข้าสู่ระบบ

- ผลที่ได้จากการทดลอง

- (ยังไม่สามารถทำการทดลองได้)



## บทสรุป

สิ่งที่กลุ่มผู้พัฒนาคาดหวังจากผู้ใช้งานเว็บไซต์คือ ความสะดวกสบายที่ได้รับหลังจากใช้งานแอปพลิเคชัน และมีความพึงพอใจกับลักษณะของการจัดการและการใช้งานที่สะดวกของเว็บไซต์Daily-exercise-log ดังนั้นกลุ่มผู้พัฒนาเว็บไซต์จึงหวังว่าผู้ใช้จะพึงพอใจและใช้บริการเว็บไซต์ของเราต่อไป

สำหรับสิ่งที่ผู้พัฒนาได้รับจากการจัดทำเว็บแอปพลิเคชันนี้คือ การพัฒนาเว็บแอปพลิเคชันจะเกิดขึ้นได้นั้น จำเป็นต้องมีการวางแผนในหลายส่วน เริ่มตั้งแต่การศึกษา requirement ความต้องการของผู้ใช้การใช้งาน พร้อมทั้งมองหาด้านแบบที่คล้ายคลึงกับแอปพลิเคชันของเรา เพื่อนำมาพัฒนาให้ดีขึ้นจากการมองหาจุดบอดของงานต้นแบบ และต่อมาคือการวางแผนการออกแบบการเขียนโปรแกรมให้ออกมาใช้งานได้และสามารถนำไปพัฒนาต่อได้ โดยการเลือกใช้การออกแบบแบบ object oriented ทำให้เราได้ฝึกฝนและเรียนรู้การเขียนเว็บแอปพลิเคชันเพื่อการใช้งานโดยใช้ PHP และ Laravel Framework ซึ่งเป็นสิ่งที่ผู้พัฒนาไม่เคยได้เรียนมาก่อนจึงทำให้ได้ประสบการณ์ใหม่ๆเพิ่มมากขึ้น และการได้เรียนรู้การใช้งานของ Git ในรูปแบบ Git workflows ทำให้เกิดการเรียนรู้ในการใช้ version control สำหรับการทำงานร่วมกันเป็นกลุ่ม

## บรรณานุกรม

Jeffrey Way. Laravel Testing Decoded. : Leanpub, 2013

<http://stackoverflow.com>

<http://laravel.com>

<https://getcomposer.org>

<http://bootsnipp.com/>

<http://getbootstrapไทย.com/>

