

ชื่อกลุ่ม Destiny's Gift

ชื่อโครงการ DatingAgency

CE-KMITL-OOAD-2014/Dating-Agency-

รายชื่อสมาชิก

นางสาว	กนกพร	ช่วงฉ่ำ	รหัสนักศึกษา	55010006
นางสาว	ทิฆัมพร	แสงสุริยศิลป์	รหัสนักศึกษา	55010462

วิชา Object Oriented Analysis and Design

ภาคการศึกษาที่ 1/2557

บทคัดย่อ

เนื่องจากในชีวิตประจำวันของคนเรามักคลุกคลีอยู่กับการทำงานมากจนเกินไป ทำให้ไม่มีโอกาสได้พบปะหรือสร้างมิตรภาพกับผู้คนใหม่ๆ ซึ่งเป็นแรงบันดาลใจในการจัดทำโปรเจกต์นี้ขึ้นเพื่อเป็นอีกทางเลือกหนึ่งให้สำหรับคนที่มีความคิดตรงกัน คืออยากจะสร้างมิตรภาพกับผู้คนใหม่ๆ ได้พบปะพูดคุยและสร้างมิตรภาพซึ่งกันและกัน

บทนำ

ในปัจจุบันคนเรามักมีการดำเนินชีวิตแบบซ้ำไปซ้ำมาในทุกๆวัน ตัวอย่างเช่น พนักงานบริษัทที่ทำงานประจำ ก็จะมีชีวิตประจำวันที่ซ้ำๆกันในทุกวัน คือ ตื่นเช้ามาก็เร่งรีบไปทำงาน เลิกงานมาก็ติดแล้ว ทำให้พบเจอแต่แค่ผู้คนในที่ทำงานซึ่งเป็นผู้คนเดิมๆ ซึ่งอาจไม่ถูกใจตามที่เราต้องการ โดยโปรเจกต์ที่เราจัดทำขึ้นนี้ จึงมีประโยชน์ในการนำเสนอประวัติส่วนตัวของบุคคลต่างๆที่เข้ามาสมัครเป็นสมาชิกให้ผู้ใช้ได้เลือกสรรตามความพอใจ โดยหากชื่นชอบคนไหนก็สามารถกดถูกใจได้ และอาจมีโอกาสดูติดต่อกันภายหลังต่อไป

รายละเอียดการวิเคราะห์หัวข้อ

ในการดำเนินงานในส่วนของการพัฒนาเว็บแอปพลิเคชันจะแบ่งการทำงานออกเป็นส่วนๆ ดังนี้

1. ผู้ใช้ ซึ่งประกอบด้วย ชื่อผู้ใช้งาน (user name) รหัสผ่าน ประวัติส่วนตัว ข้อมูลการติดต่อ และรูปประจำตัว
2. รูป เพื่อใช้ในการแทนรูปประจำตัวของผู้ใช้
2. รายการประวัติส่วนตัว เป็นรายการที่เก็บข้อมูลประวัติส่วนตัว ได้แก่ ชื่อ นามสกุล อายุ เพศ การทำงาน และความสนใจ
3. รายการข้อมูลการติดต่อ เป็นรายการที่เก็บข้อมูลการติดต่อของผู้ใช้ ได้แก่ E-mail Facebook และ Line ID (รหัส line ส่วนบุคคล)
4. รายการการกดถูกใจ เป็นรายการที่เก็บข้อมูลการกดถูกใจของผู้ใช้ว่าได้กดถูกใจใครบ้าง
5. หน่วยควบคุมการแสดงผลข้อมูลส่วนบุคคล ช่วยในการแสดงผลข้อมูลส่วนบุคคลของผู้ใช้ที่เข้าระบบผู้นั้น
6. หน่วยควบคุมการแสดงผลข้อมูลส่วนบุคคลของผู้ใช้คนอื่น ช่วยในการแสดงผลข้อมูลส่วนบุคคลของผู้ใช้คนอื่นให้ผู้ใช้ที่ทำการเข้าระบบผู้นั้นได้ทราบ
7. หน่วยควบคุมการจัดการข้อมูลส่วนบุคคล ช่วยในการจัดการข้อมูลส่วนบุคคลของผู้ใช้ทุกคนในระบบ
8. หน่วยควบคุมการจัดการการกดถูกใจ ช่วยในการกดถูกใจของผู้ใช้แต่ละบุคคล
9. หน่วยควบคุมการจัดการการรับส่งข้อความ ช่วยในการจัดการการติดต่อสื่อสารกันระหว่างผู้ใช้ที่กดถูกใจตรงกัน
7. หน่วยควบคุมการจัดการการส่ง virtual items (รูปเสมือนจริง) ช่วยในการจัดการการส่ง virtual items (รูปเสมือนจริง)) เป็นของขวัญให้กับคนที่ตนเองถูกใจ

งานที่เกี่ยวข้อง

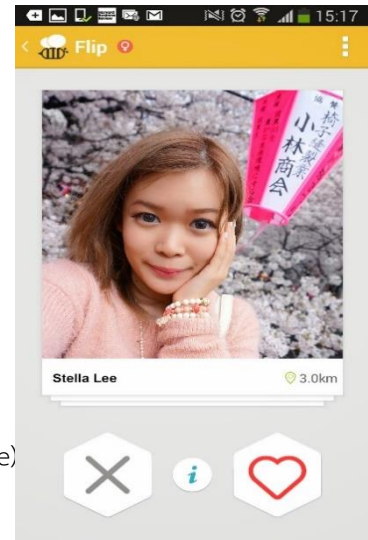
- BeeTalk

ความเหมือนของ BeeTalk กับโปรเจ็ค

- 1.) มีการแสดงรูปประจำตัว
- 2.) มีการระบุอายุและเพศ
- 3.) แสดงความประทับใจด้วยการกดถูกใจ (like)

ความต่างของ BeeTalk กับโปรเจ็ค

- 1.) BeeTalk จะสุ่มจากแค่คนในระยะใกล้ แต่โปรเจ็คจะสุ่มจากสมาชิกทั้งระบบ
- 2.) BeeTalk ถ้ามีการถูกใจ (Like) ตรงกันจะกลายเป็นเพื่อนกัน
แต่โปรเจ็คจะระบุข้อมูลการติดต่อให้ทราบ
- 3.) BeeTalk สามารถส่งรูปเสมือนจริง (virtual items) ได้หลังจากที่มีการกดถูกใจ (like)
แล้ว แต่โปรเจ็คไม่จำเป็นต้องกดถูกใจ (like) ตรงกันก่อน



- eHarmony

ความเหมือนของ eHarmony กับโปรเจ็ค

- 1.) มีการสร้างประวัติส่วนตัวและมีการนำรูปประจำตัวขึ้นหน้าประวัติส่วนตัว
- 2.) มีการรับ-ส่งข้อความ

ความต่างของ eHarmony กับโปรเจ็ค

- 1.) eHarmony จะมีการจับคู่ให้โดยมีการพิจารณาตามหลักวิทยาศาสตร์
แต่โปรเจ็คจะมีการจับคู่ให้ตามความพึงพอใจของผู้ใช้เองโดยพิจารณาตาม
การกดถูกใจ (like)
- 2.) eHarmony จะมีการจับคู่ให้เพียง1คนต่อ1ผู้ใช้นั้น แต่โปรเจ็ค
สามารถจับคู่ได้หลายคนต่อ1ผู้ใช้ พิจารณาตามการกดถูกใจ (like)
- 3.) ผู้ใช้ eHarmony จะไม่สามารถมองเห็นหน้าประวัติส่วนตัวของผู้ใช้คนอื่นได้
แต่ผู้ใช้โปรเจ็คจะสามารถมองเห็นหน้าประวัติส่วนตัวของผู้ใช้คนอื่นได้



ผลการวิเคราะห์ความต้องการของผู้ใช้ระบบ

Functional Requirement

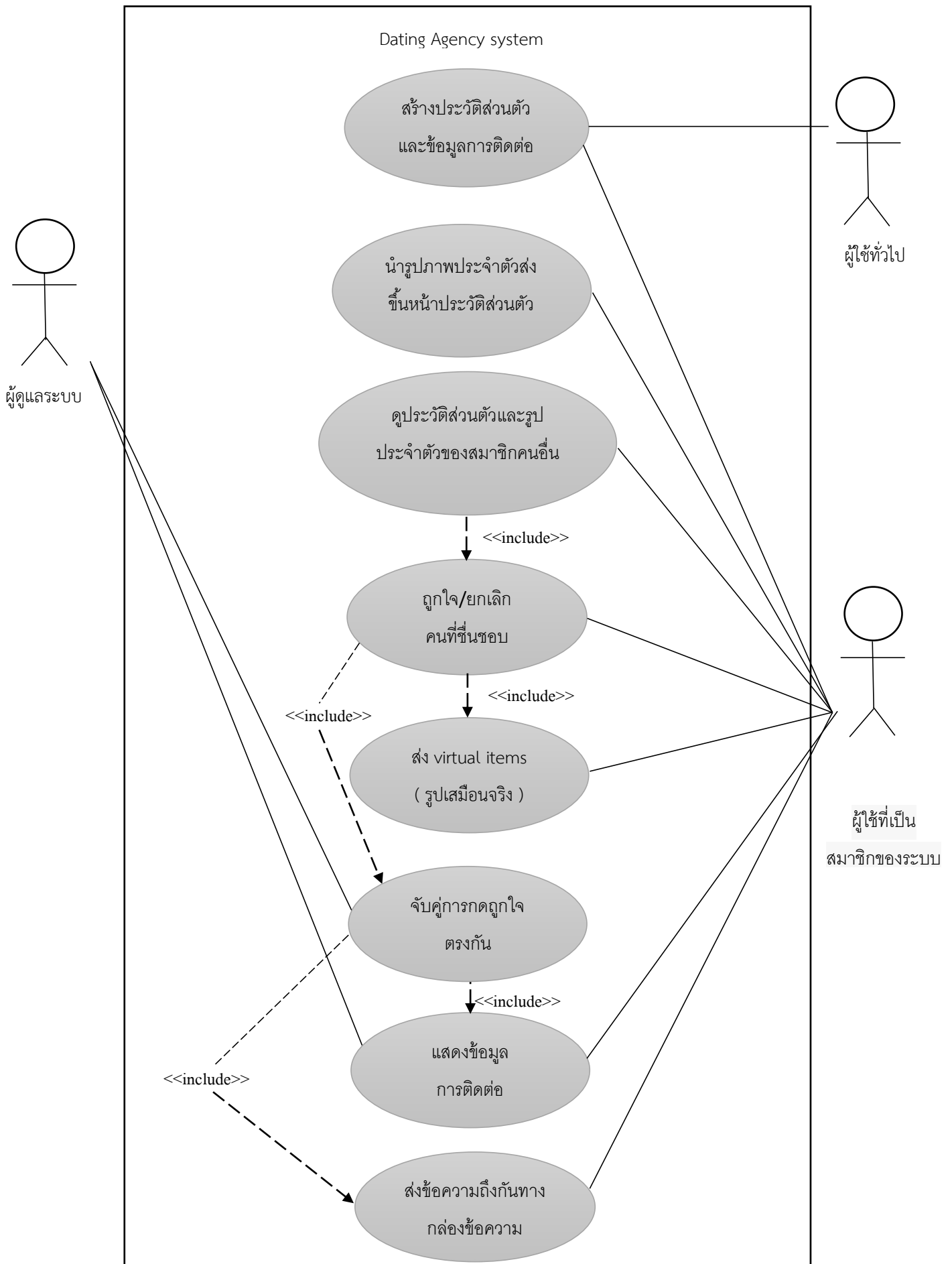
- มีการสมัครสมาชิก , การเข้าระบบและออกจากระบบสมาชิกเพื่อยืนยันตัวตน
- มีข้อมูลประวัติส่วนตัว , ข้อมูลการติดต่อ และรูปประจำตัว
- มีการกดถูกใจ / ยกเลิกถูกใจ
- มีการส่ง virtual items (รูปเสมือนจริง)
- มีการส่งข้อความถึงกัน

Non-Functional Requirement

- มีการตอบสนองที่รวดเร็ว
- สามารถใช้งานได้ง่าย มีการแนะนำลำดับขั้นตอนให้ผู้ใช้งานเรียนรู้การใช้งานได้ง่าย
- สามารถใช้งานได้มีประสิทธิภาพ
- มีความสวยงาม
- สามารถใช้งานได้ทั้งในคอมพิวเตอร์และโทรศัพท์สมาร์ทโฟน

แนวทางการใช้งาน

Use case diagram



Use case specification

Use case : สร้างข้อมูลประวัติส่วนตัวและข้อมูลการติดต่อ

- **Use case name** : สร้างข้อมูลประวัติส่วนตัวและข้อมูลการติดต่อ
- **Use case purpose** : เพื่อสร้าง แก้ไข หรือ เข้าดู ข้อมูลประวัติส่วนตัวและข้อมูลการติดต่อของตนเอง
- **Date modifier** : 10 / 9 / 57
- **Preconditions** : ผู้ใช้ทั่วไปต้องการเป็นสมาชิกของระบบหรือต้องการจะสร้างประวัติส่วนตัวและข้อมูลการติดต่อของตนเอง และผู้ใช้ที่เป็นสมาชิกของระบบต้องการแก้ไขหรือเข้าดูหน้าประวัติส่วนตัวและข้อมูลการติดต่อของตนเอง
- **Postconditions** : ผู้ใช้ที่เป็นสมาชิกของระบบทุกคนมีหน้าประวัติส่วนตัวและข้อมูลการติดต่อของตนเอง
- **Limitations** : ผู้ใช้ทั่วไปต้องการเป็นสมาชิกของระบบและผู้ใช้ที่เป็นสมาชิกของระบบทุกคนต้องกรอกประวัติส่วนตัวและข้อมูลการติดต่อของตนเองให้ครบและตรงตามเงื่อนไข
- **Assumptions** : ผู้ใช้ที่เป็นสมาชิกของระบบมีหน้าประวัติส่วนตัวและข้อมูลการติดต่อของตนเอง แสดงต่อผู้ใช้ที่เป็นสมาชิกของระบบคนอื่นดูได้
- **Primary Scenario** :
 - A. ผู้ใช้ทั่วไปหรือผู้ใช้ที่เป็นสมาชิกของระบบต้องการจะเข้าดูหน้าประวัติส่วนตัวและข้อมูลการติดต่อ
 - B. ผู้ใช้ทั่วไปหรือผู้ใช้ที่เป็นสมาชิกของระบบทำการเข้าสู่ระบบ
 - C. ถ้าเป็นผู้ใช้ทั่วไปที่ต้องการเป็นสมาชิกใหม่ที่ไม่มีข้อมูลประวัติส่วนตัวและข้อมูลการติดต่อ
 1. กรอกประวัติส่วนตัว
 2. กรอกข้อมูลการติดต่อ
 - D. เข้าดูหน้าประวัติส่วนตัวและข้อมูลการติดต่อ
- **Alternate Scenarios** :
 - **Condition triggering an alternate scenario:**
 - **condition 1** : กรอกข้อมูลไม่ครบ หรือกรอกข้อมูลไม่ตรงตามเงื่อนไข
 - C1. แสดงข้อผิดพลาดให้ผู้ใช้ทราบ
 - C2. แสดงผลขึ้นตอน B

- condition 2 : ต้องการแก้ไขข้อมูลประวัติส่วนตัวและข้อมูลการติดต่อ

D1. เข้าสู่หน้าแก้ไขข้อมูลประวัติส่วนตัวและข้อมูลการติดต่อ

D2. ถ้ากรอกข้อมูลไม่ครบ จะแสดงข้อผิดพลาดให้ผู้ใช้ทราบและแสดงผลขั้นตอน D1

D3. ถ้ากรอกข้อมูลประวัติส่วนตัวและข้อมูลการติดต่อ จะแสดงผลขั้นตอน D

Use case specification

Use case : นำรูปภาพประจำตัวส่งขึ้นหน้าประวัติส่วนตัว

- Use case name : นำรูปภาพประจำตัวส่งขึ้นหน้าประวัติส่วนตัว
- Use case purpose : เพื่อให้ผู้ใช้ที่เป็นสมาชิกของระบบนำรูปภาพประจำตัวแสดงต่อผู้ใช้ที่เป็นสมาชิกของระบบคนอื่นและสามารถแก้ไขรูปประจำตัวได้
- Date modifier : 10 / 9 / 57
- Preconditions : ผู้ใช้ต้องการจะนำรูปภาพประจำตัวส่งขึ้นหน้าประวัติส่วนตัวของตนเอง หรือผู้ใช้ที่เป็นสมาชิกของระบบอยู่แล้วต้องการแก้ไขรูปภาพประจำตัวส่งขึ้นหน้าประวัติส่วนตัว
- Postconditions : รูปประจำตัวของผู้ใช้ที่เป็นสมาชิกของระบบจะแสดงต่อผู้ใช้ที่เป็นสมาชิกในระบบคนอื่น
- Limitations : ผู้ใช้ที่เป็นสมาชิกของระบบทุกคนต้องนำรูปภาพประจำตัวส่งขึ้นหน้าประวัติส่วนตัวของตนเอง
- Assumptions : ผู้ใช้ที่เป็นสมาชิกของระบบมีรูปภาพประจำตัวที่ต้องการ แสดงต่อผู้ใช้ที่เป็นสมาชิกของระบบคนอื่นดูได้
- Primary Scenario :
 - A. ถ้าเป็นสมาชิกใหม่ของระบบที่ไม่มีรูปภาพประจำตัว
 - 1. นำรูปภาพขึ้นแสดงต่อผู้ใช้ที่เป็นสมาชิกของระบบคนอื่น
 - B. ถ้าผู้ใช้ที่มีรูปประจำตัวอยู่แล้วต้องการแก้ไขรูปประจำตัวเก่า
 - 1. นำรูปภาพประจำตัวที่ต้องการขึ้นแทนที่รูปประจำตัวเก่า
- Alternate Scenarios :
 - Condition triggering an alternate scenario :
 - condition 1 : ไม่นำรูปภาพขึ้นแสดงหน้าประวัติส่วนตัว
 - A1. แสดงข้อผิดพลาดให้ผู้ใช้ทราบ
 - A2. แสดงผลขั้นตอน A

○ **condition 2** : ต้องการลบรูปประจำตัว

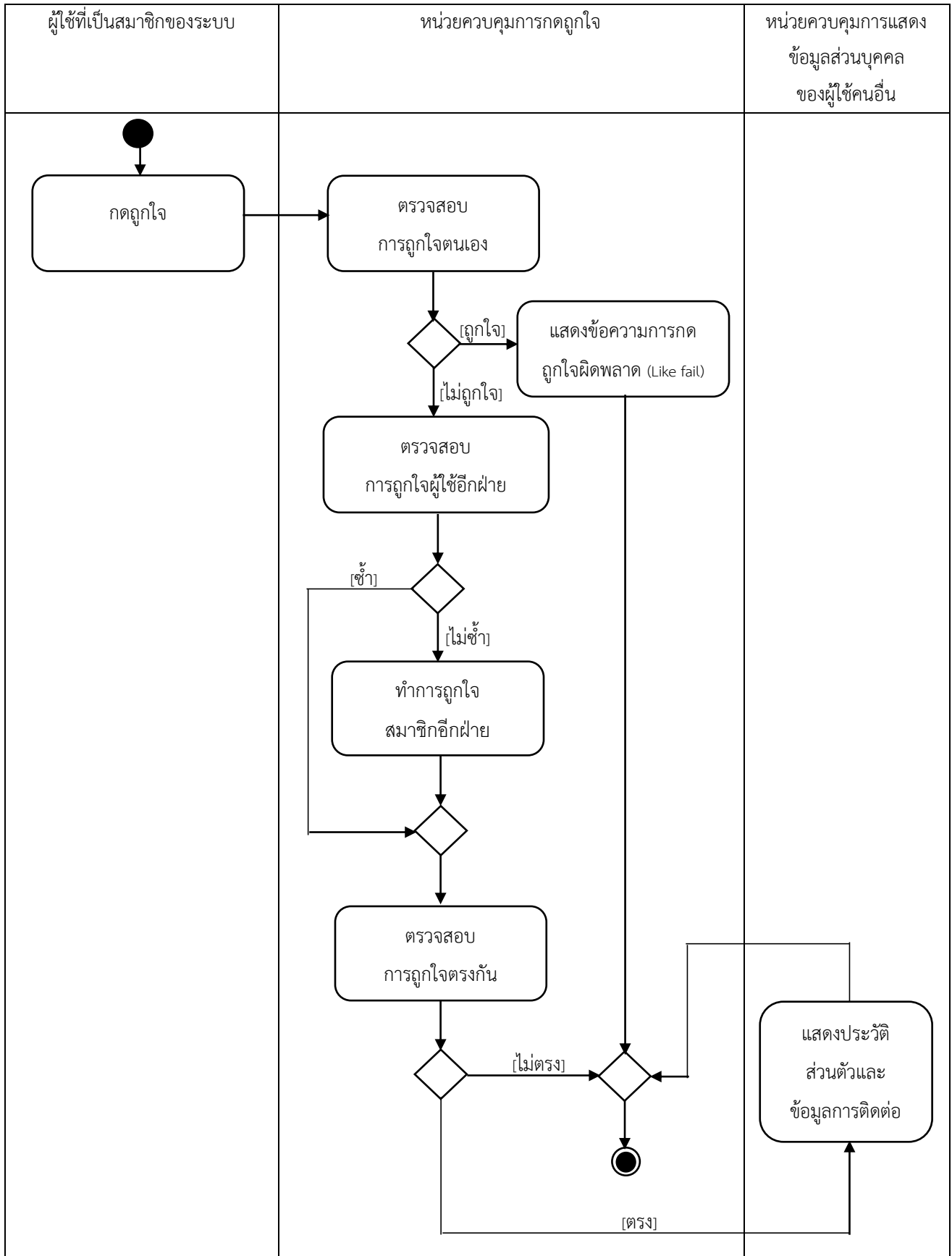
B1. แสดงข้อผิดพลาดให้ผู้ใช้งานทราบว่าไม่สามารถลบรูปประจำตัวได้

B2. แสดงผลขั้นตอน B

Activity diagram

Use case : ถูกใจ/ยกเลิก คนที่ชื่นชอบ

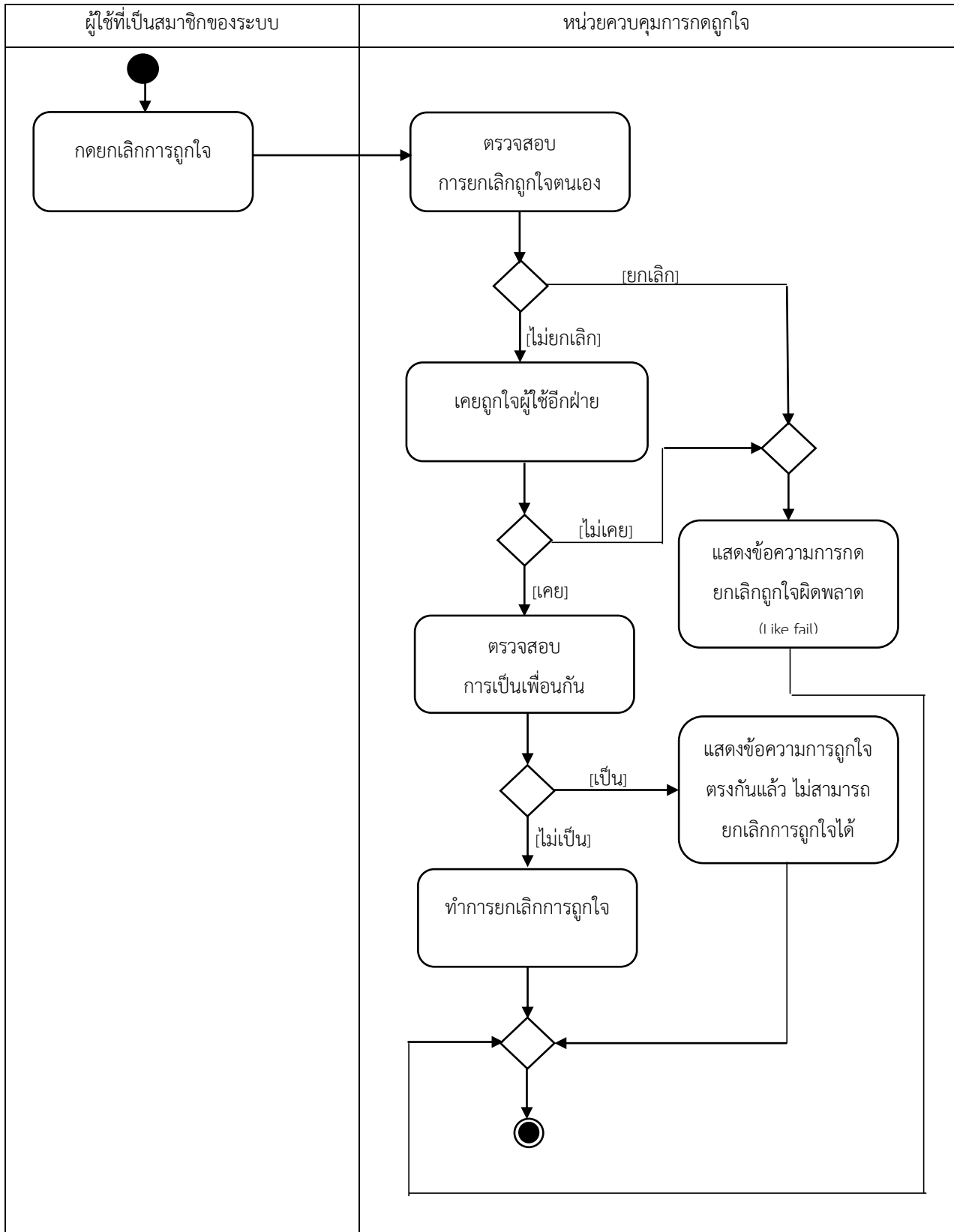
ถูกใจคนที่ชื่นชอบ



Activity diagram

Use case : ถูกใจ/ยกเลิก คนที่ชื่นชอบ

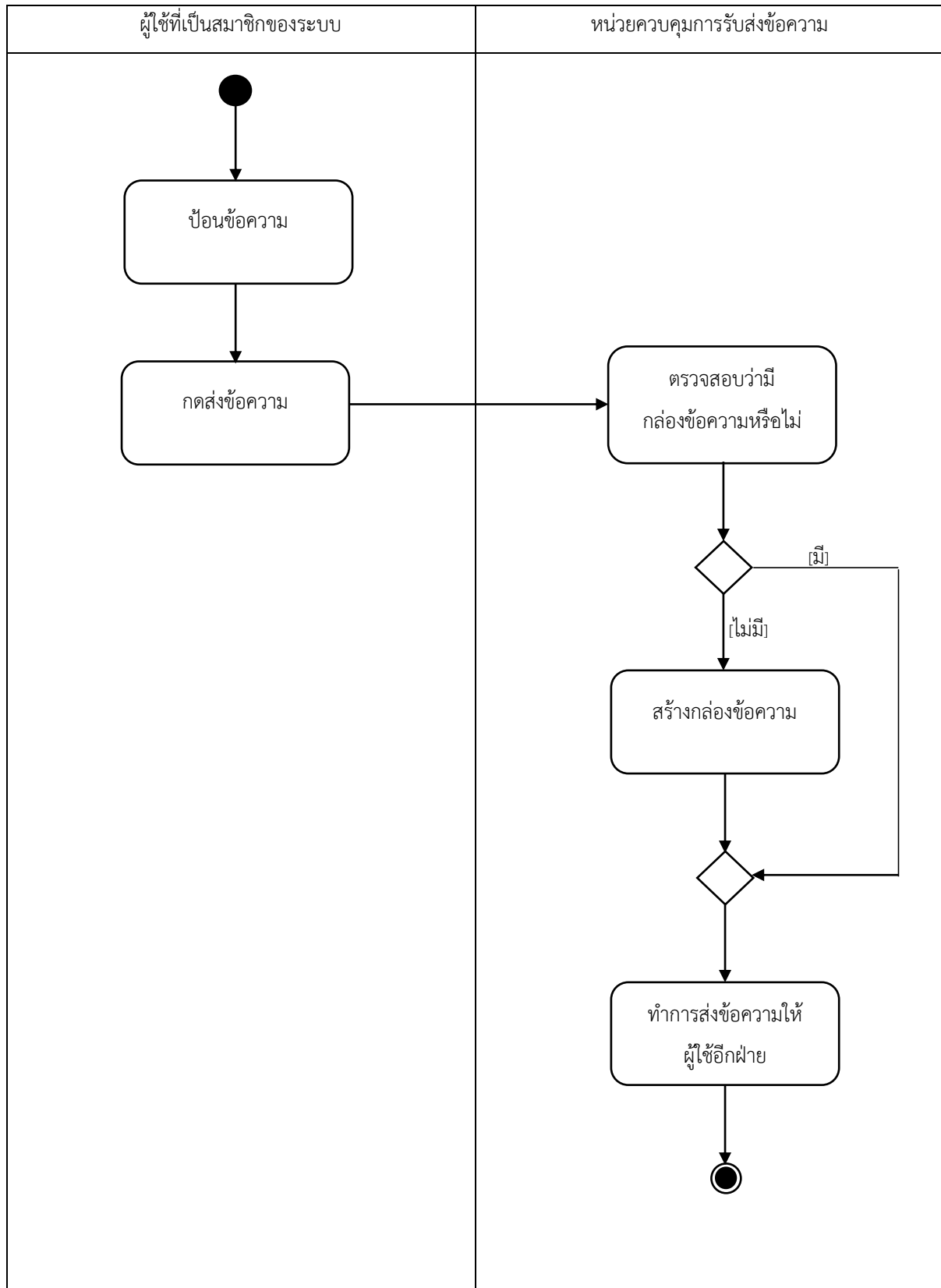
ยกเลิกถูกใจคนที่ชื่นชอบ



Activity diagram

Use case : ส่งข้อความถึงกันผ่านกล่องข้อความ

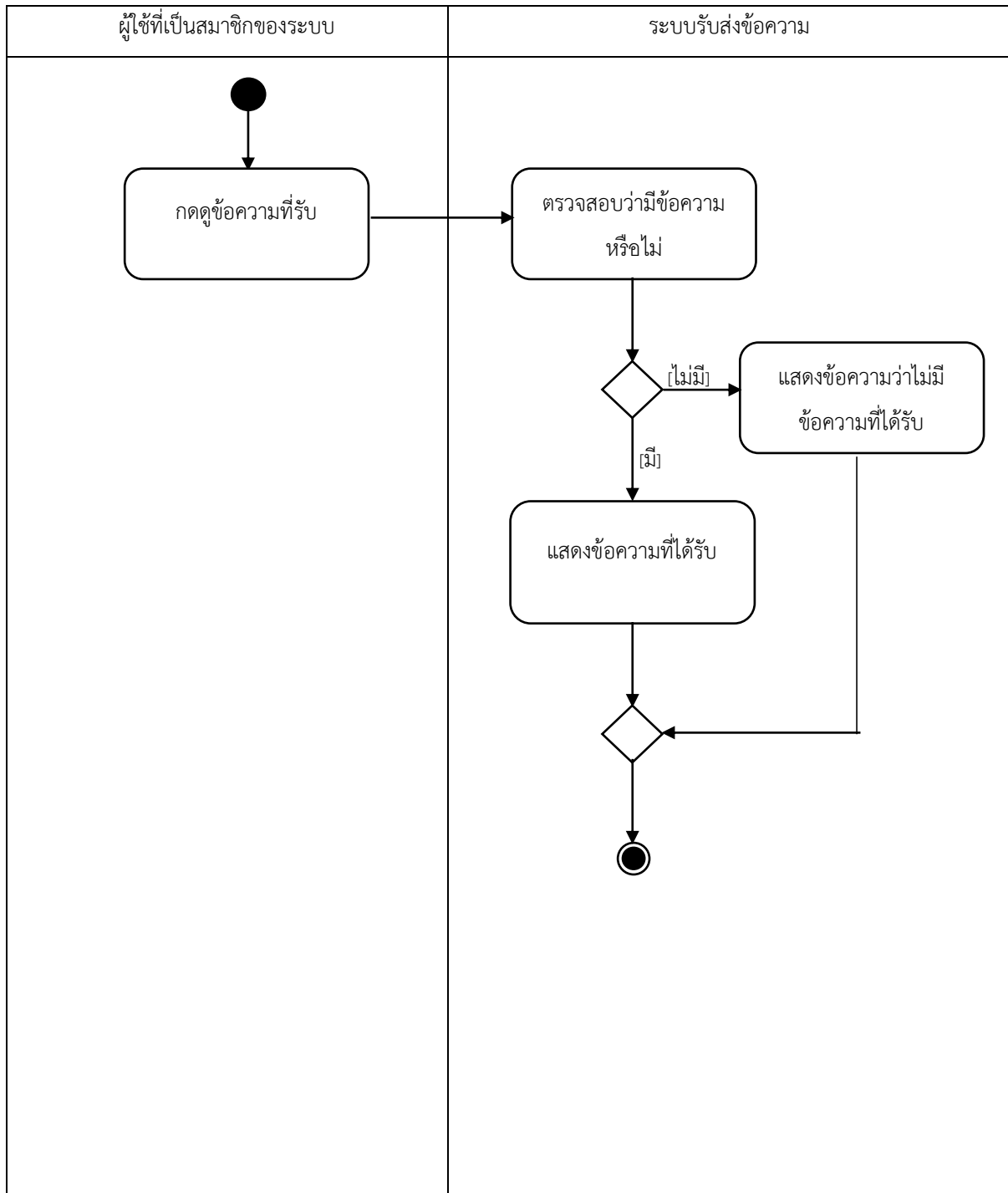
ส่งข้อความผ่านกล่องข้อความ



Activity diagram

Use case : ส่งข้อความถึงกันผ่านกล่องข้อความ

รับข้อความผ่านกล่องข้อความ



สถาปัตยกรรมของระบบ

Problem Analysis

ในการดำเนินงานในส่วนของการพัฒนาเว็บแอปพลิเคชันจะแบ่งการทำงานออกเป็นส่วนๆ ดังนี้

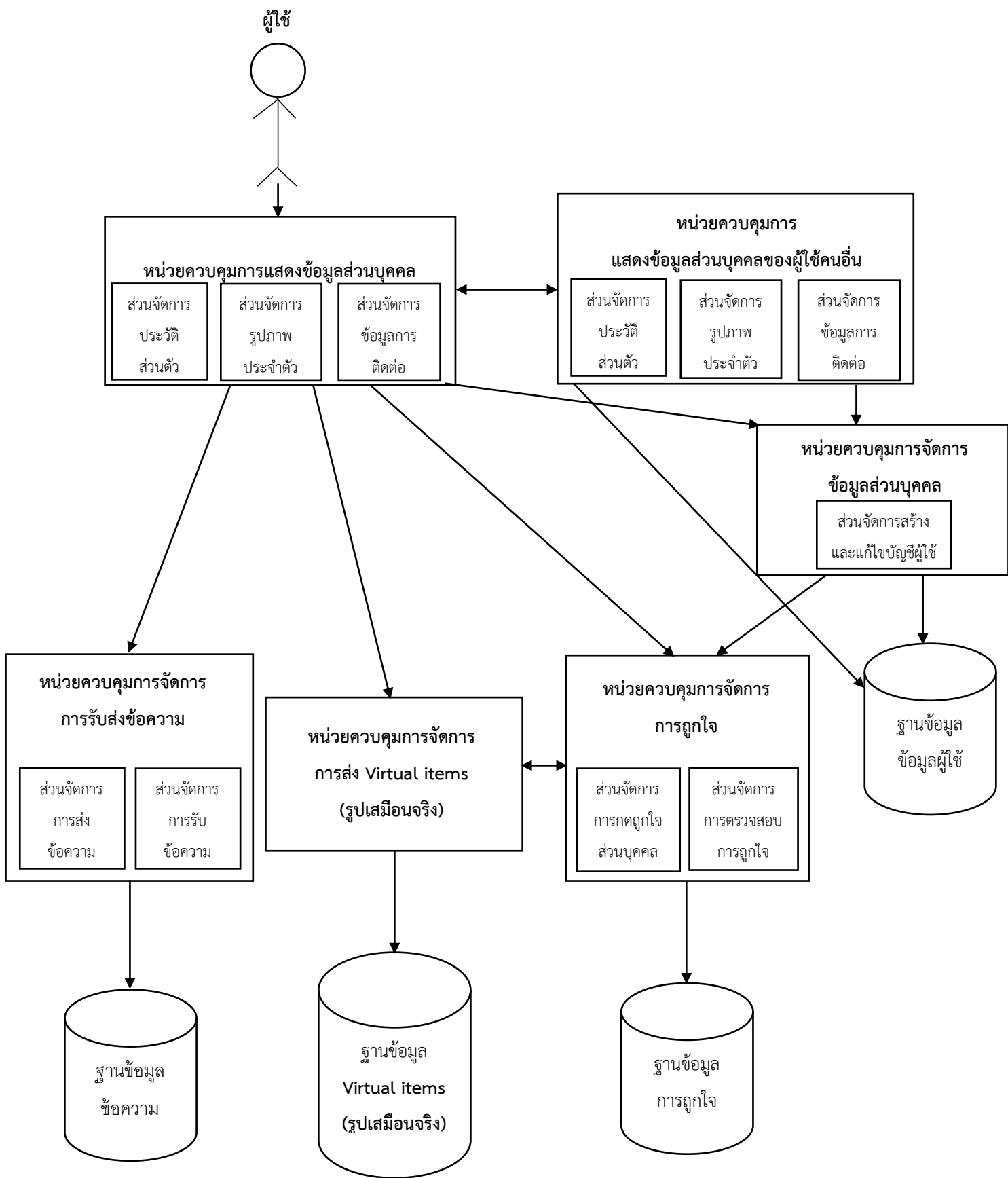
Abstraction

1. **ผู้ใช้** ซึ่งประกอบด้วย ชื่อผู้ใช้งาน(user name) รหัสผ่าน ประวัติส่วนตัว ข้อมูลการติดต่อ และรูปประจำตัว
2. **รายการประวัติส่วนตัว** เป็นรายการที่เก็บข้อมูลประวัติส่วนตัว ได้แก่ ชื่อ นามสกุล อายุ เพศ การทำงาน และความสนใจ
3. **รายการข้อมูลการติดต่อ** เป็นรายการที่เก็บข้อมูลการติดต่อของผู้ใช้ ได้แก่ E-mail Facebook และ Line ID (รหัส line ส่วนบุคคล)
4. **รายการการกดถูกใจ** เป็นรายการที่เก็บข้อมูลการกดถูกใจของผู้ใช้ว่าได้กดถูกใจใครบ้าง

Component

1. **หน่วยควบคุมการแสดงผลข้อมูลส่วนบุคคล** ช่วยในการแสดงผลข้อมูลส่วนบุคคลของผู้ใช้ที่เข้าระบบผู้นั้น
2. **หน่วยควบคุมการแสดงผลข้อมูลส่วนบุคคลของผู้ใช้คนอื่น** ช่วยในการแสดงผลข้อมูลส่วนบุคคลของผู้ใช้คนอื่นให้ผู้ใช้ที่ทำการเข้าระบบผู้นั้นได้ทราบ
3. **หน่วยควบคุมการจัดการข้อมูลส่วนบุคคล** ช่วยในการจัดการข้อมูลส่วนบุคคลของผู้ใช้ทุกคนในระบบ
4. **หน่วยควบคุมการจัดการการกดถูกใจ** ช่วยในการกดถูกใจของผู้ใช้แต่ละบุคคล
5. **หน่วยควบคุมการจัดการการรับส่งข้อความ** ช่วยในการจัดการการติดต่อสื่อสารกันระหว่างผู้ใช้ที่กดถูกใจตรงกัน
6. **หน่วยควบคุมการจัดการการส่ง virtual items (รูปเสมือนจริง)** ช่วยในการจัดการการส่ง virtual items (รูปเสมือนจริง) เป็นของขวัญให้กับคนที่ตนเองถูกใจ

Application Architecture



Subsystems / Components

1. **หน่วยควบคุมการแสดงผลข้อมูลส่วนบุคคล** จะประกอบด้วยข้อมูล 3 ส่วน ได้แก่ รูปประจำตัว, ประวัติส่วนตัว และข้อมูลการติดต่อ โดยในส่วนของประวัติส่วนตัวนั้นจะมีข้อมูลต่างๆ ซึ่งเป็นข้อมูลพื้นฐานที่ใช้ในการตัดสินใจที่จะกดถูกใจ ได้แก่ ชื่อ-นามสกุล อายุ เพศ อาชีพ ความสนใจ เป็นต้น และในส่วนของข้อมูลการติดต่อจะเป็นข้อมูลที่จะใช้ในการติดต่อสื่อสารกับคนที่กดถูกใจตรงกัน ได้แก่ หมายเลขโทรศัพท์ E-mail Facebook เป็นต้น

2. **หน่วยควบคุมการแสดงผลข้อมูลส่วนบุคคลของผู้ใช้คนอื่น** จะประกอบด้วยข้อมูล 3 ส่วนเหมือนกับระบบแสดงผลข้อมูลส่วนบุคคล คือ รูปประจำตัว, ประวัติส่วนตัว และข้อมูลการติดต่อ แต่หน่วยควบคุมการแสดงผลข้อมูลส่วนบุคคลของผู้ใช้คนอื่นจะต่างจากระบบแสดงผลข้อมูลส่วนบุคคลตรงที่ในระบบแสดงผลข้อมูลส่วนบุคคลของผู้ใช้คนอื่นจะแสดงผลเพียง 2 ส่วน คือ รูปประจำตัว และประวัติส่วนตัว โดยข้อมูลการติดต่อจะแสดงหลังจากที่มีการกดถูกใจตรงกันแล้ว ซึ่งเป็นเหตุให้ต้องเชื่อมโยงข้อมูลกับหน่วยควบคุมการจัดการการถูกใจ

3. **หน่วยควบคุมการจัดการข้อมูลส่วนบุคคล** เป็นการกระทำต่อหน่วยควบคุมการแสดงผลข้อมูลส่วนบุคคล ซึ่งแบ่งแยกออกเป็น 3 ส่วน คือ

3.1. ส่วนจัดการการสร้างและแก้ไขบัญชีผู้ใช้ ได้แก่

- นำรูปภาพประจำตัวส่งขึ้นหน้าประวัติส่วนตัวและแก้ไขรูปประจำตัว
- สร้างและแก้ไขข้อมูลประวัติส่วนตัว
- สร้างและแก้ไขข้อมูลการติดต่อ

3.2. ส่วนจัดการลำดับบัญชีผู้ใช้ ได้แก่

- นับจำนวนของผู้ใช้ทั้งหมด
- แสดงข้อมูลผู้ใช้ทั้งหมดโดยเรียงลำดับตามที่ใช้สมัครเป็นสมาชิกของระบบ

4. **ฐานข้อมูลข้อมูลผู้ใช้** เป็นที่จัดเก็บข้อมูลข้อมูลส่วนบุคคลทั้งประวัติส่วนตัว ข้อมูลการติดต่อ และรูปประจำตัวของผู้ใช้ทุกคน

5. **หน่วยควบคุมการจัดการการถูกใจ**

แบ่งการทำงานออกเป็น 2 ส่วน คือ

1. ส่วนจัดการการกดถูกใจส่วนบุคคล เป็นการจัดการเก็บข้อมูลการกดถูกใจของตนเองว่าได้กดถูกใจประวัติส่วนตัวของใครบ้าง โดยจะมีการจัดเก็บข้อมูลส่วนนี้ลงใน ฐานข้อมูลข้อมูลผู้ใช้

2. ส่วนจัดการการตรวจสอบการถูกใจ เป็นการวิเคราะห์หาผู้ใช้ที่มีการกดถูกใจตรงกัน โดยจะนำเข้าข้อมูลจากฐานข้อมูลข้อมูลของผู้ใช้มาใช้ในการวิเคราะห์ โดยหลังจากวิเคราะห์หาบุคคลที่กดถูกใจตรงกันเจอแล้ว จะมีการแสดงข้อมูลการติดต่อของแต่ละฝ่ายให้อีกฝ่ายทราบ

6. ฐานข้อมูลการกดถูกใจ เป็นที่จัดเก็บข้อมูลการกดถูกใจของทุกคน

7. หน่วยควบคุมการจัดการการส่ง Virtual items (รูปเสมือนจริง)

ในส่วนนี้จะมีการอ้างอิงข้อมูลมาจาก ฐานข้อมูล Virtual items ซึ่งจัดเก็บ virtual items (รูปเสมือนจริง) ไว้ โดยการส่ง Virtual items (รูปเสมือนจริง) นั้นจะส่งได้ก็ต่อเมื่อได้กดถูกใจประวัติส่วนตัวของอีกฝ่ายแล้ว จึงจำเป็นต้องอาศัยข้อมูลจากหน่วยควบคุมการจัดการการถูกใจในส่วนของการกดถูกใจส่วนบุคคล

8. ฐานข้อมูล Virtual Items(รูปเสมือนจริง) เป็นที่จัดเก็บ Virtual Items (รูปเสมือนจริง) รูปแบบต่างๆ

9. หน่วยควบคุมการจัดการการรับส่งข้อความ

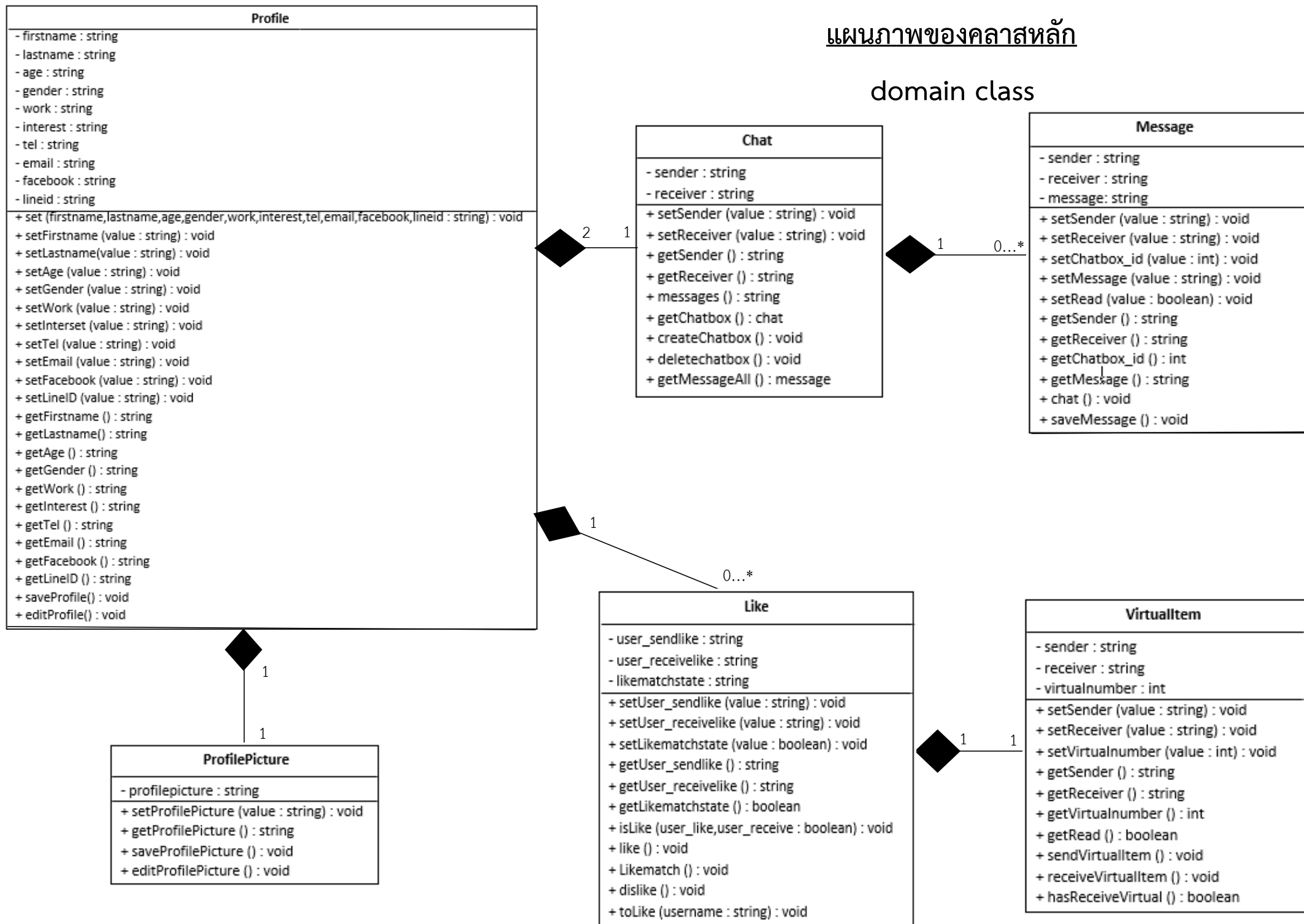
มีการแบ่งการทำงานออกเป็น 2 ส่วน คือ

1. ส่วนจัดการการส่งข้อความ คือ การที่ผู้ใช้ส่งข้อความหาบุคคลอื่น
2. ส่วนจัดการการรับข้อความ คือ การที่บุคคลอื่นส่งข้อความมายังผู้ใช้

10. ฐานข้อมูลข้อความ เป็นฐานข้อมูลที่เก็บข้อความการสนทนาของผู้ใช้ทั้ง 2 ฝ่าย

แผนภาพของคลาสหลัก

domain class



รายละเอียดการพัฒนาซอฟต์แวร์

Deployment

- มี virtual machine (VM) 1 ตัว โดยใช้ Ubuntu Server 14.04 LTS ซึ่งเป็นระบบปฏิบัติการ Linux จำลองการทำงานบน virtual machine (VM)
- **Server Technologies**
 - ใช้ Apache2 เป็นเว็บเซิร์ฟเวอร์
 - ใช้ PHP: php5 libapache2-mod-php5 php5-mcrypt เป็นแอปพลิเคชันเซิร์ฟเวอร์
 - ใช้ MySQL เป็นฐานข้อมูล
 - ใช้ phpmyadmin ในการบริหารจัดการฐานข้อมูล MySQL
 - ใช้ Laravel ซึ่งเป็น PHP Framework ในการพัฒนาเว็บแอปพลิเคชัน
 - ใช้ Eloquent ของ Laravel ในการติดต่อฐานข้อมูล
- **Client-side (Browser)**
 - ใช้ Bootstrap เป็น Front-end Framework (UI)
- ระบบทุกระบบทำงานอยู่บน virtual machine (VM) หมายเลข Public IP address : 23.101.28.55

Implementation plan

Dating-Agency	Name	Start Date	End Date
ศึกษา		26/9/14	17/10/14
ศึกษา Laravel framework (โครงสร้าง)	กนกพร,ทิชัมพร	26/09/14	17/10/14
ศึกษาภาษา php	กนกพร,ทิชัมพร	26/09/14	17/10/14
จัดเตรียมเครื่องมือและอุปกรณ์		26/09/14	29/09/14
จัดเตรียม virtual machine (เครื่องเสมือน)	กนกพร,ทิชัมพร	26/09/14	29/09/14
งานเอกสาร		19/09/14	21/11/14
M2 : Architectural Design, UML diagram & Implementation Plan (ออกแบบภาพรวมของระบบ และวางแผนการดำเนินการดำเนินงาน)	กนกพร,ทิชัมพร	19/09/14	25/09/14
จัดเตรียม Design Presentation (นำเสนอการออกแบบ)	กนกพร,ทิชัมพร	11/10/14	17/10/14
M3 : Implementation Progress & Test Plan (ความคืบหน้าของงานและแผนการทดสอบ)	กนกพร,ทิชัมพร	26/10/14	30/10/14
จัดเตรียมการนำเสนอ	กนกพร,ทิชัมพร	14/11/14	21/11/14
ดำเนินการ		11/10/14	20/11/14
คลาสข้อมูลผู้ใช้		11/10/14	17/10/14
สร้างฐานข้อมูล	กนกพร,ทิชัมพร	11/10/14	14/10/14
ทำระบบสมัครสมาชิก และการเข้าสู่ระบบ	กนกพร,ทิชัมพร	11/10/14	14/10/14
สร้างหน้าประวัติส่วนตัว, ข้อมูลการติดต่อ	กนกพร,ทิชัมพร	15/10/14	17/10/14
นำรูปประจำตัวขึ้นหน้าประวัติส่วนตัว	กนกพร	18/10/14	2/10/14
การกดถูกใจ	ทิชัมพร	3/11/14	9/11/14
การส่ง Virtual Items (รูปเสมือนจริง)	ทิชัมพร	5/11/14	10/11/14
การส่งข้อความถึงกันผ่านทางกล่องข้อความ	กนกพร	7/11/14	16/11/14
ทดสอบโปรแกรม และแก้ไขในส่วนที่ผิดพลาด	กนกพร,ทิชัมพร	17/11/14	20/11/14

ผลการทดสอบซอฟต์แวร์

Test

Directory ของ test file ใน repository อยู่ที่ Dating-Agency- / app / tests /

1. Test การกดถูกใจของผู้ใช้ ว่าผู้ใช้ถูกใจผู้ใช้อีกฝ่ายแล้วหรือยัง (LikeTest.php)

- Code ที่ใช้ในการเขียน Test

```
<?php

class LikeTest extends TestCase {

    public function testlike()
    {
        $user1=new Like;
        $user1->setUser_sendlike("user1");
        $user1->toLike("user2");
        $like = LikeRepository::find(DB::table('likes')->max('id'));
        $this->assertEquals('user1',$like->user_sendlike);
        $this->assertEquals('user2',$like->user_receivelike);
        $this->assertEquals('0',$like->likematchstate);
    }

    public function testIslike()
    {
        $user1 = new Like();
        $test=$user1 ->Islike("user1","user2");
    }

}
```

- ผลลัพธ์ของการ Test

```
bownan@datingagency:/var/www/html/Dating-Agency-$ phpunit
PHPUnit 3.7.28 by Sebastian Bergmann.

Configuration read from /var/www/html/Dating-Agency-/phpunit.xml

..

Time: 260 ms, Memory: 10.75Mb

OK (2 tests, 3 assertions)
```

2. Test การกดถูกใจตรงกันของผู้ใช้ เพื่อตรวจสอบว่าผู้ใช้ได้ถูกใจตรงกัน (LikematchTest.php)

- Code ที่ใช้ในการเขียน Test

```
<?php

class LikematchTest extends TestCase {
    public function testLikematch()
    {
        $user3=new Like;
        $user3->setUser_sendlike("user3");
        $user3->setID((DB::table('likes')->max('id'))+1);
        $user3->toLike("user4");
        $like = LikeRepository::find(DB::table('likes')->max('id'));
        $this->assertEquals('user3',$like->user_sendlike);
        $this->assertEquals('user4',$like->user_receivelike);
        $this->assertEquals('0',$like->likematchstate);

        $user4=new Like;
        $user4->setUser_sendlike("user4");
        $user4->setID((DB::table('likes')->max('id'))+1);
        $user4->toLike("user3");
        $like = LikeRepository::find(DB::table('likes')->max('id'));
        $this->assertEquals('user4',$like->user_sendlike);
        $this->assertEquals('user3',$like->user_receivelike);
        $this->assertEquals('0',$like->likematchstate);

        $user3->Likematch();
        $user4->Likematch();

        $like = LikeRepository::find((DB::table('likes')->max('id'))-1);
        $this->assertEquals('user3',$like->user_sendlike);
        $this->assertEquals('user4',$like->user_receivelike);
        $this->assertEquals('1',$like->likematchstate);
        $like = LikeRepository::find(DB::table('likes')->max('id'));
        $this->assertEquals('user4',$like->user_sendlike);
        $this->assertEquals('user3',$like->user_receivelike);
        $this->assertEquals('1',$like->likematchstate);
    }
}
```

- ผลลัพธ์ของการ Test

```
bownan@datingagency:/var/www/html/Dating-Agency-$ phpunit
PHPUnit 3.7.28 by Sebastian Bergmann.

Configuration read from /var/www/html/Dating-Agency-/phpunit.xml

.

Time: 305 ms, Memory: 10.25Mb

OK (1 test, 12 assertions)
```

Evaluation

การทดลองที่ 1 การส่ง virtual items (รูปเสมือนจริง)

- จุดประสงค์ของการทดลอง (Goal of the experiment)
เพื่อตรวจสอบว่าระบบสามารถส่ง virtual items (รูปเสมือนจริง) ของผู้ใช้ได้สำเร็จ
- สิ่งที่จะวัด (Measurement metrics)
ระบบทำการส่ง virtual items (รูปเสมือนจริง) ถึงผู้ใช้อีกฝ่ายได้จริงและระบบตอบรับว่าการส่งสำเร็จ
- วิธีการทดลองและสิ่งที่ต้องใช้ในการทดลอง (Setup and methods of your experiment)
 1. ผู้ใช้ทำการเข้าสู่ระบบ
 3. เลือกกดถูกใจประวัติส่วนตัวผู้ที่ชื่นชอบ
 4. ทำการส่ง virtual items (รูปเสมือนจริง)
 5. ระบบตอบรับว่าการส่งสำเร็จ
 6. เข้าฐานข้อมูลแล้วดูว่าระบบจัดเก็บ virtual items (รูปเสมือนจริง) ได้ถูกต้องตามที่ออกแบบหรือไม่
- ผลที่ได้จากการทดลอง (Experimental results)
ผู้ใช้อีกฝ่ายที่ถูกชื่นชอบจะได้รับ virtual items (รูปเสมือนจริง) โดยจะไปแสดงในหน้า Receive Virtual items
- สรุปและสิ่งที่ได้จากการทดลองนี้ (Conclusion)
ผู้ใช้สามารถทำการส่ง virtual items (รูปเสมือนจริง) ให้ผู้ใช้อีกฝ่ายที่ชื่นชอบได้จริง

การทดลองที่ 2 การส่งข้อความทางกล่องข้อความ

- จุดประสงค์ของการทดลอง (Goal of the experiment)
เพื่อตรวจสอบว่าระบบมีการส่งข้อความหากันได้จริงและแสดงผลข้อความได้ถูกต้องตามที่ป้อนหรือไม่
- สิ่งที่จะวัด (Measurement metrics)
ระบบสามารถส่งข้อความหากันระหว่างผู้ใช้ได้จริงและแสดงข้อความของผู้ใช้ที่ป้อนออกมาได้ถูกต้อง
- วิธีการทดลองและสิ่งที่ต้องใช้ในการทดลอง (Setup and methods of your experiment)
 1. ผู้ใช้ทำการเข้าสู่ระบบ
 2. เปรียบเทียบหากผู้ใช้มีการกดถูกใจตรงกัน ก็สามารถทำการส่งข้อความได้
 3. ทดลองพิมพ์ข้อความและทำการส่งข้อความ
 4. มีการแสดงข้อความที่ส่งไปออกมา
- ผลที่ได้จากการทดลอง (Experimental results)
ผู้ใช้สามารถส่งข้อความหากันและสามารถแสดงข้อความของทั้งสองฝ่ายที่ป้อนได้ โดยแสดงที่หน้า Chatbox และสามารถดูข้อความที่ผู้ใช้คนอื่นส่งมาให้ โดยแสดงที่หน้า Receive Message ได้ถูกต้อง
- สรุปและสิ่งที่ได้จากการทดลองนี้ (Conclusion)
ผู้ใช้สามารถทำการส่งข้อความหากันและแสดงผลข้อความถูกต้องตามที่ป้อนได้จริง

บทสรุป

โปรเจ็ค DatingAgency นี้ เป็นโปรเจ็คของรายวิชา Object-Oriented Analysis and Design เป็นการสร้างเว็บแอปพลิเคชันซึ่งเกี่ยวกับการจัดหาคู่ให้กับผู้ใช้บริการเว็บไซต์ กลุ่มของข้าพเจ้าหวังว่าสิ่งที่ผู้ใช้บริการจะได้รับ คือ จะได้พบปะกับผู้คนที่หลากหลายรูปแบบ ได้คู่รักตามแบบที่ตนเองถูกใจและต้องการ ถ้าหากไม่ได้สานสัมพันธ์ถึงขั้นคู่รัก ผู้ใช้ก็เหมือนเป็นการได้เพื่อนใหม่เพิ่มขึ้นอีกด้วย

คุณสมบัติของของเว็บแอปพลิเคชันนี้ ผู้ใช้นั้นสมัครเป็นสมาชิกของระบบ มีการกรอกประวัติส่วนตัว รูปภาพประจำตัว และข้อมูลการติดต่อของตนเอง การกรอกประวัติส่วนตัวกับรูปภาพประจำตัวนั้นเพื่อเป็นพื้นฐานในการตัดสินใจในการที่จะชื่นชอบและถูกใจ ส่วนข้อมูลการติดต่อจะแสดงก็ต่อเมื่อผู้ใช้ทั้ง 2 ฝ่าย มีการถูกใจตรงซึ่งกันและกัน และผู้ใช้อย่างสามารถดยยกเลิกการถูกใจผู้ใช้ที่เคยถูกใจไปแล้วก็ยังไม่ได้มีการถูกใจตรงกันอีกด้วย ทั้งยังมีการส่ง virtual items (รูปเสมือนจริง) หลังจากทำการถูกใจ ผู้ใช้สามารถเลือกที่จะส่ง virtual items (รูปเสมือนจริง) ให้กับผู้ใช้อีกฝ่าย ซึ่งทางเว็บแอปพลิเคชันมีให้เลือกมากมายหลากหลายรูปแบบ นอกจากนี้ยังมีการส่งข้อความหากันผ่านกล่องข้อความ เป็นการเปิดโอกาสผู้ใช้ทั้ง 2 ฝ่าย ที่ถูกใจตรงกันแล้วได้ทำความรู้จักพูดคุยเพื่อสานสัมพันธ์ซึ่งกันและกันต่อไป

ในการทำโปรเจ็คครั้งนี้ ทำให้กลุ่มของข้าพเจ้าได้รับความรู้ไม่ใช่จากแค่เนื้อหาที่เรียนในห้องเรียน แต่การทำโปรเจ็คถือเป็นประสบการณ์โดยตรงอีกอย่างหนึ่ง ได้วิเคราะห์และออกแบบเชิงวัตถุเพื่อใช้งานเหมือนเป็นการทำงานในชีวิตจริง และยังเกิดความภาคภูมิใจที่จากผู้ที่ไม่เคยมีประสบการณ์ในการเขียนเว็บแอปพลิเคชันนั้น สามารถเขียนเว็บแอปพลิเคชันขึ้นมาใช้งานได้จริง

บรรณานุกรม

Rees, Dayle. CodeBright. พิมพ์ครั้งที่ 4 : Leanpub, 2014.

“Stella Lee X Bee Talk Beauty Consultation” 2014.

[ระบบออนไลน์]. แหล่งที่มา บรรณานุกรม <http://www.stellalee.net/2014/04/stella-lee-x-bee-talk-beauty.html> (1 กันยายน 2557)

“วิธีการหาคู่ทางวิทยาศาสตร์ — eHarmony” 2557.

[ระบบออนไลน์]. แหล่งที่มา บรรณานุกรม <http://www.angelspace.net/วิธีการหาคู่ทางวิทยาศาสตร์.html> (7 พฤศจิกายน 2557)

“How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu” 2012.

[ระบบออนไลน์]. แหล่งที่มา บรรณานุกรม <https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu> (2 ตุลาคม 2557)

“Design elements, playground and code snippets for Bootstrap HTML/CSS/JS framework” 2014.

[ระบบออนไลน์]. แหล่งที่มา บรรณานุกรม <http://bootsnipp.com/> (5 ตุลาคม 2557)

“Getting started with testing Laravel 4 Models” 2013.

[ระบบออนไลน์]. แหล่งที่มา บรรณานุกรม <http://culttt.com/2013/05/20/getting-started-with-testing-laravel-4-models/> (13 พฤศจิกายน 2557)