

FINAL

Class: Object-oriented Analysis and Design

SEMESTER 1/2557 @ KMITL

GROUP NAME: STYLESCOUT

GITHUB REPOSITORY: STYLESCOUT

Project Name:

Style Scout App

Group Members:

1. ชนันทชัย ธาราศิริวัฒน์ รหัสนักศึกษา 55010528 sec.1
2. ญาณิศา ยิ้มสุวรรณ รหัสนักศึกษา 55010302 sec.1

สารบัญ

เรื่อง	หน้า
บทคัดย่อ.....	1
บทนำและรายละเอียดการวิเคราะห์หัวข้อ.....	2
งานที่เกี่ยวข้อง.....	3
ผลการวิเคราะห์ความต้องการของผู้ใช้ระบบ	4
- Functional Requirement.....	4
-Non-Functional Requirement.....	5
แนวทางการใช้งาน	6
-Use case diagram.....	6
-Use case specifications.....	7-8
-Activity Diagrams.....	10-14
สถาปัตยกรรมของระบบ.....	15
-Problem Analysis.....	15-16
-Application Architecture.....	17
-Subsystems / Components.....	19-24
-แผนภาพของคลาสหลัก.....	25
รายละเอียดการพัฒนาซอฟต์แวร์.....	26
-Deployment.....	26
-Implementation plan.....	27-28
ผลการทดสอบซอฟต์แวร์.....	29
-Unit Test	29-30

-Evaluation.....	31-32
บทสรุป.....	33
บรรณานุกรม.....	34

บทคัดย่อ

ABSTRACT: ถ้าพูดถึงเรื่อง Fashion หลายคนอาจจะรู้จักแค่เปลือกนอกของคาคานี้เท่านั้นว่าเป็นแค่การแต่งกายในรูปแบบต่างๆเท่านั้น แต่สำหรับหลายๆคนที่สนใจในด้านนี้จริงๆ ก็มีพื้นที่ในการแลกเปลี่ยนที่ค่อนข้างจำกัด และไม่เจาะจงพอ ด้วยเหตุนี้เราจึงตอบโจทย์ด้วยการสร้างสังคมที่เกี่ยวกับ Fashion ขึ้นมาอย่างเต็มตัว ที่เรียกว่า “StyleScout” ซึ่งเปรียบเสมือนพื้นที่ที่ให้แลกเปลี่ยนความคิดสร้างสรรค์ โดยผู้ใช้สามารถแบ่งปัน Style การแต่งตัวของตนได้ด้วยการอัปโหลดรูปเพื่อแชร์ Style ให้ผู้ใช้งานคนอื่น ๆ ได้ พร้อมทั้งระบุประเภทการแต่งตัว (Tag) และรายละเอียดของเครื่องแต่งกาย (details) พร้อมทั้งสามารถเข้าไปดู Style การแต่งตัวประเภทต่างๆได้ด้วยที่ผู้ใช้คนอื่น ๆ ได้ทำการอัปโหลดไว้ ด้วย Tag ที่แบ่งหมวดหมู่ไว้อย่างชัดเจน เช่น Street Fashion, Vintage เป็นต้น ดังนั้น Stylescout จึงเป็นสังคมคน Fashion ที่เปิดกว้างให้คนที่สนใจในเรื่อง Fashion ได้มาแลกเปลี่ยนแรงบันดาลใจ และ Style การแต่งตัวของตนเพื่อเป็นต้นแบบ

บทนำและรายละเอียดการวิเคราะห์หัวข้อ

INTRODUCTION & MOTIVATION: เมื่อก้าวถึง Fashion และ Street Style สำหรับบางคนคือเรื่องใกล้ตัวและไม่ได้ให้ความสนใจ จริงอยู่ที่เสื้อผ้าที่เราสวมใส่นั้นคือ 1 ใน ปัจจัย 4 ในชีวิต แต่เมื่อเราพูดถึงรายละเอียดปลีกย่อยลงไปผู้คนกลับเลือกที่จะให้ความสนใจน้อยลง ซึ่งไม่ใช่เรื่องที่ผิดเพราะแต่ละคนต่างมีความคิดเห็นอันแตกต่าง แต่เมื่อผู้คนให้ความสำคัญกับ Fashion & Style น้อยลงทำให้ในบางสังคมการเข้าถึงและเป็นส่วนหนึ่งของ Fashion นั้นยากขึ้น หากเราต้องการ Inspirations & Ideas สำหรับการแต่งตัวเราอาจจะต้องเสียเงินซื้อ Magazine ซึ่งจริงอยู่โลกของ Internet ปัจจุบันสามารถเข้าถึงและเสาะหาในสิ่งที่เราต้องการได้แต่มันไม่ Specificพอสำหรับผู้ที่ยื่นชอบ Fashion จริงๆ และ เขาคู่นั้นอาจไม่มีโอกาสในการเข้าร่วมจริงๆเลยก็ได้ ดังนั้น Style Scout App จึงจัดทำเพื่อบุคคลที่ชื่นชอบและหลงใหลใน Fashion & Style และอยากมีส่วนร่วมและได้ Expressed ตัวตนของตนเองออกมาอย่างถูกต้องโดยไม่มี Boundaries หรือ Judgment แ่ๆใดๆมาทำให้จินตนาการถูกล้อมกรอบได้ Style Scout App ทำให้ผู้ใช้งานสามารถ Upload รูป Outfit of the Day, Inspirations และสิ่งต่างๆที่ Related กับ Fashionและสามารถบอก Details ต่างๆให้ผู้อื่นสามารถรับทราบได้ และ ติด #tag เพื่อแยกประเภทของ Idea นั้นๆให้เกิดความชัดเจนมากยิ่งขึ้น โดยเราจะกำหนด #tag ให้ตามความเหมาะสมเพื่อไม่ให้เกิดความสับสนและการสร้าง #tag แบบไม่ควรอย่างที่เกิดขึ้นใน Instagram เช่น #likesforlikes , #idontwhattothinkbecauseimcraycray เป็นต้น ซึ่งอาจทำให้App ผิดจุดประสงค์ได้ #tag ที่เราจะกำหนดให้เช่น #Minimalist, #Preppy และ #HauteCorture เป็นต้น

งานที่เกี่ยวข้อง

RELATED WORKS: Weheartit.com – Style Scout App และ Weheartit เป็น source of inspirations ที่อนุญาตให้ผู้ใช้มีส่วนร่วมในการ Share Inspirations แต่ Style Scout App จะspecific กว่าเพราะเราเน้น Fashion Community เป็นหลัก Instagram – Style Scout App และ Instagram เป็นแหล่งรวมภาพถ่ายจากทุกมุมโลก (ตามผู้ใช้) ซึ่งทุกคนสามารถUpload รูปของตนเองหรือรูปที่ชื่นชอบและShare ต่อสาธารณะ แต่สำหรับ Instagram การติด Hashtag นั้นทำได้โดยอิสระ แล้วแต่Users นั้นต้องการ ซึ่งStyle Scout App ไม่อนุญาตให้ User สร้าง Hashtag ของตนเองขึ้นมาได้เนื่องจากเราต้องการความเป็นระเบียบและ Specific มากที่สุดเพื่อเกิดประโยชน์สูงสุดสำหรับผู้ใช้งาน

ผลการวิเคราะห์ความต้องการของผู้ใช้ระบบ

Functional Requirements

Requirement ID	Requirement Statement	*Must/Want
FR001	โปรแกรมประยุกต์บนเว็บ(Web Application) ควรมีหน้าแสดงข้อมูลส่วนตัว(Profile) ของผู้ใช้งาน เพื่อบอกข้อมูลส่วนตัวเล็กน้อยๆ และรวบรวมรูปภาพที่ผู้ใช้งานนั้นเคยทำการลงรูปภาพ (Post/Upload) ไว้แล้ว	Must
FR002	ผู้ที่เป็นผู้ลงรูปภาพ(Upload) สามารถลบรูปภาพ>Delete)นั้นๆได้ แต่สำหรับผู้ใช้งานอื่นที่ไม่ได้เป็นผู้ลงรูปภาพ(Upload)นั้นไม่สามารถทำได้	Must
FR003	ผู้ใช้งานที่เป็นสมาชิกสามารถเลือกหาประเภทแฮชแท็ก(Hashtag) ได้เมื่อต้องการในหน้ารายการรวมประเภทของแฮชแท็ก(Hashtag)	Must
FR004	สมาชิกต้องสามารถแก้ไขข้อมูลส่วนตัวได้เมื่อต้องการ	Must
FR005	ผู้ดูแลระบบต้องสามารถเพิ่มประเภทของแฮชแท็ก(Hashtag)ได้	Must

**Must = จำเป็นต้องมี*

Want = ต้องการมี

Non-Functional Requirements

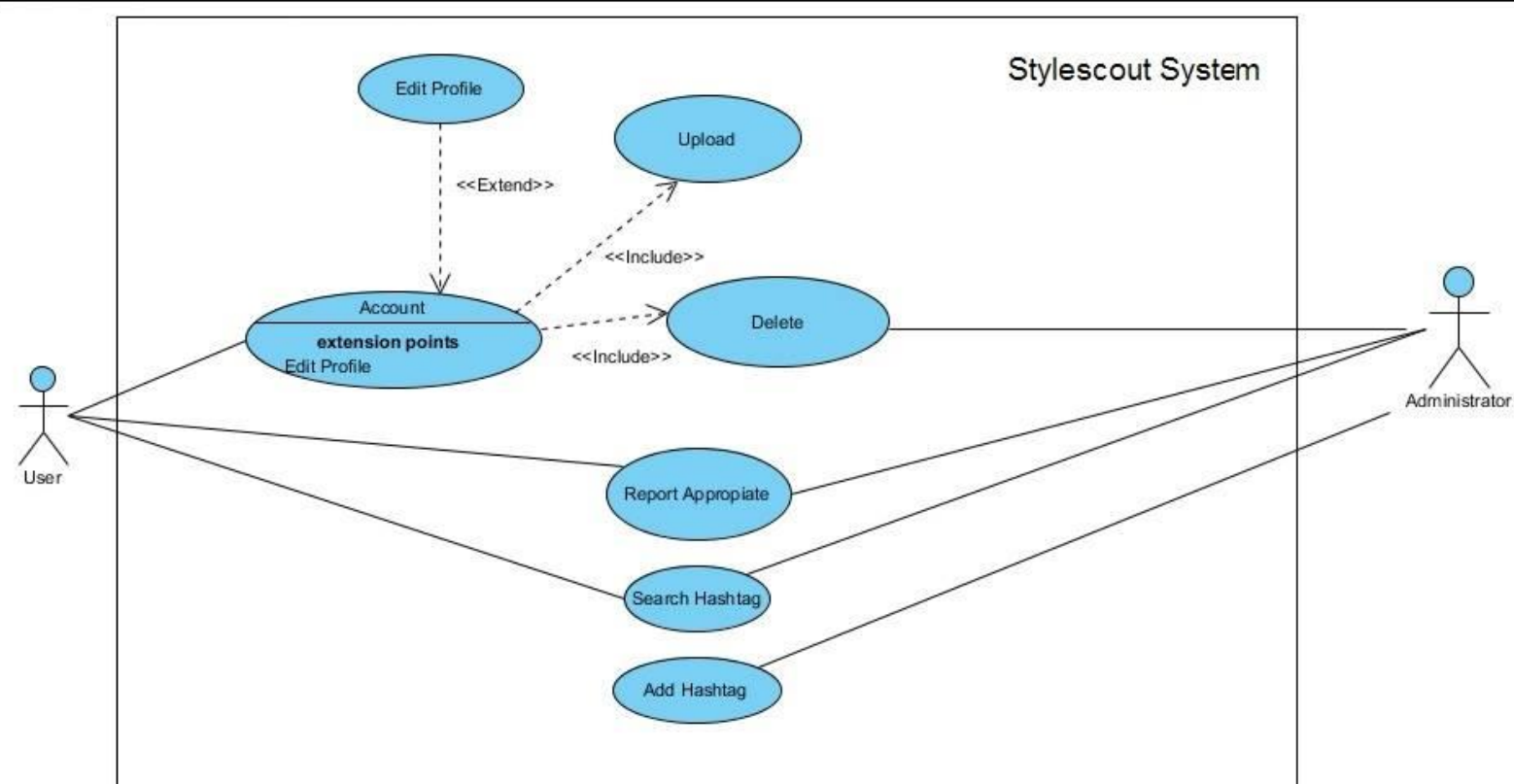
Requirement ID	Requirement Statement	*Must/Want
NR001	โปรแกรมประยุกต์บนเว็บ(Web Application) ควรมีความปลอดภัยจากการคุกคาม ปลอดภัยจากการขโมยข้อมูลจากกลุ่มผู้ที่ประสงค์ร้ายต้องการล้วงข้อมูลในระบบ(Hacker)	Must
NR002	โปรแกรมประยุกต์บนเว็บ(Web Application) ควรมีการสำรองข้อมูล(Back up) ทุกคืน และ หากเกิดเหตุการณ์สร้างความเสียหายควรจะสามารถกลับคืนสู่ ภาวะปกติได้ภายใน 1-2 ชั่วโมง	Must
NR003	เวลาที่ใช้ในการลงรูปภาพ(Upload) ควรมีความเร็วมากที่สุด เพื่อประสิทธิภาพที่ดีในการใช้งานของผู้ใช้บริการ	Want
NR004	มีความสามารถในการใช้งานบนระบบปฏิบัติการ (Operating System) ที่หลากหลายไม่เฉพาะเจาะจง	Must
NR005	ควรมีเหตุการณ์การล่มของระบบน้อยที่สุด อย่างมากที่สุดคือ 1 ครั้งต่อสัปดาห์และควรแก้ไขให้ได้เร็วที่สุด	Must
NR006	หน้าตาการใช้งาน(Display) ควรมีความทันสมัยและเหมาะสม	Want
NR007	การใช้งานระบบไม่ควรมีความซับซ้อนจนเกินไป เพื่อความเข้าใจง่ายและสะดวกในการใช้งานของผู้ใช้บริการ	Want
NR008	การเข้าใช้งานต้องสามารถเข้าใช้งานได้จากทุกพื้นที่จากทั่วโลก	Must
NR009	ระบบควรปราศจากปัญหาและข้อบกพร่องเมื่อผู้ใช้ได้ใช้บริการ	Must

**Must = จำเป็นต้องมี*

Want = ต้องการมี

แนวทางการใช้งาน

USE CASE DIAGRAM



Use Case Specification (1)

Use Case Name:	การลงรูปภาพ (Upload)		
Created By:	ญาณิศา ยิ้มสุวรรณ	Last Updated By:	ญาณิศา ยิ้มสุวรรณ
Date Created:	7/9/2014	Last Revision Date:	1/11/2014
Use Case Purpose:	เพื่อให้ผู้ใช้งานสามารถทำการลงรูปภาพ(Upload)ได้เมื่อต้องการ		
Actors:	ผู้ใช้บริการที่ลงทะเบียน(Sign up) เป็นสมาชิกแล้วและทำการเข้าใช้(Sign in)อยู่		
Pre-conditions:	<ol style="list-style-type: none"> 1. ผู้ใช้บริการมีระบบอินเทอร์เน็ต (Internet) 2. ผู้ใช้บริการทำการลงทะเบียน(Sign up)เรียบร้อยแล้ว 3. ผู้ใช้บริการทำการลงชื่อเข้าใช้(Sign in) แล้ว 4. ผู้ใช้บริการมีรูปภาพที่ตนเองต้องการนำเสนอหรือลงรูปภาพ(Upload) 		
Post-conditions:	<ol style="list-style-type: none"> 1. รูปภาพที่ผู้ใช้งานต้องการลง(Upload)สามารถบันทึกในรายการรูปภาพที่ผู้ใช้ทำการลงเอาไว้ได้ 		
Limitations:	การลงรูปภาพ(Upload) มีข้อจำกัดคือ ขนาดของรูปภาพต้องได้ขนาด(size)ตามกำหนดคือ 1:1 (Square) และความเร็วในการลงรูป(Upload) ซึ่งขึ้นอยู่กับความเร็วของอินเทอร์เน็ตของผู้ใช้หรือช่วงความถี่ของการส่งข้อมูล(Upstream)		
Assumptions:	ผู้ใช้สามารถแบ่งปันรูปภาพที่ต้องการหรือที่ชอบโดยการลงรูปภาพ(Upload)ผ่านทางระบบเรา ได้		
Primary Scenario:	<ol style="list-style-type: none"> 1. ผู้ใช้บริการ(User) ทำการเปิดระบบอินเทอร์เน็ตของตนเอง บนคอมพิวเตอร์, โทรศัพท์ เป็นต้น 2. ผู้ใช้บริการ(User) ทำการลงทะเบียน(Sign up) ในกรณีใช้งานครั้งแรก หรือทำการเข้าใช้บริการ(Sign in) ในกรณีลงทะเบียนแล้ว 3. ผู้ใช้บริการ(User) ทำการเลือกรูปภาพที่ตนต้องการ 4. ผู้ใช้บริการ(User) ทำการกดยืนยันการลงรูปภาพต่อสาธารณะชน(Share) 5. เป็นอันเสร็จสิ้น ผู้ใช้งานสามารถลงรูปภาพ(Upload)ได้ 		
Alternative Scenario:	<ol style="list-style-type: none"> 5. Condition triggering an alternate scenario: กรณีลงรูปภาพ(Upload) ไม่ได้เนื่องจากอินเทอร์เน็ตของผู้ใช้บริการ(User) เกิดเหตุขัดข้องไม่ว่ากรณีใดก็ตาม <ol style="list-style-type: none"> 5.1 ระบบไม่สามารถลงรูปภาพ(Upload) ของผู้ใช้งานได้ 5.2 ผู้ใช้ทราบผลว่าไม่สามารถลงรูปภาพ(Upload)ได้ 5.3 ผู้ใช้บริการ(User) ต้องทำการตรวจสอบข้อขัดข้องนั้นและทำการแก้ไขเพื่อที่จะสามารถลงรูปภาพ(Upload) ได้ 5.4 ผู้ใช้งานสามารถลงรูปภาพ(Upload)ได้ 		

	<p>4.Condition triggering an alternate scenario: ผู้ใช้งานต้องการเปลี่ยนรูปภาพที่ต้องการลง(Upload)</p> <p>4.1 ผู้ใช้งานไม่กดตกลงหรือยืนยันว่าจะทำการลงรูปภาพ(Upload)นั้นๆ</p> <p>4.2 กลับไปยัง Primary Scenario ขั้นตอนี่</p> <p>4.3 เลือกรูปภาพที่ต้องการแล้ว ก็ทำการกดตกลงหรือยืนยันว่าจะทำการลงรูปภาพ(Upload)นั้นๆ</p> <p>4.4 จบการทำงาน</p>
--	--

Use Case Specification (2)

Use Case Name:	การเรียกดูรูปตามประเภท (Hashtag)		
Created By:	ธนนท์ชัย ธาราศิริวัฒน์	Last Updated By:	ธนนท์ชัย ธาราศิริวัฒน์
Date Created:	7/9/2014	Last Revision Date:	7/9/2014
Use Case Purpose:	ผู้ใช้งานสามารถเลือกชมรูปการแต่งตัวแฟชั่นประเภทต่างๆ ผ่านตามประเภทที่กำหนด (Hashtag) ที่ผู้ใช้บริการทำการเลือกชมได้ตามความต้องการ		
Actors:	ผู้ใช้บริการที่เป็นสมาชิก(User) หรือ ผู้ใช้บริการทั่วไปที่ไม่ได้เป็นสมาชิก(Guest)		
Pre-conditions:	<ol style="list-style-type: none"> 1. ผู้ใช้บริการมีระบบอินเทอร์เน็ต (Internet) 2. ผู้ใช้บริการที่เป็นสมาชิก หรือผู้ให้บริการทั่วไปทำการกดคลิกแถบเมนูที่ชื่อว่า “List tags” 3. ผู้ใช้บริการที่เป็นสมาชิก หรือผู้ให้บริการทั่วไปทำการเลือกประเภทของภาพ (Hashtag) 		
Post-conditions:	1. ผู้ใช้บริการสามารถเข้าชมรูปภาพในประเภทแฮชแท็ก(Hashtag) นั้นๆได้		
Limitations:	การเลือกประเภทของภาพ(Hashtag) สามารถเข้าชมได้อย่างจำกัด กล่าวคือ มีประเภทของภาพที่กำหนดไว้(Hashtag) ให้เลือกไม่หลากหลาย จะมีประเภทของภาพ(Hashtag) ตามที่ผู้ดูแลระบบ (Admin) เป็นคนสร้างขึ้นมานั้นๆ เพื่อความเฉพาะเจาะจง		
Assumptions:	ผู้ใช้บริการเลือกชมรูปภาพต่างๆตามประเภทของแฮชแท็ก(Hashtag)ต่างๆที่กำหนดเอาไว้ได้		
Primary Scenario:	<ol style="list-style-type: none"> 1. ผู้ใช้บริการ(User) ทำการเปิดระบบอินเทอร์เน็ต(Internet) ของตนเอง แล้วทำการเปิดหน้าต่างโปรแกรมดูเว็บ(Web Browser) เพื่อเข้าสู่โปรแกรมประยุกต์บนเว็บ(Web Application) “Style Scout App” 2. ผู้ใช้บริการทำการเลือกเมนูประเภทหรือที่เรียกว่า “HashtagLists” ในแถบเมนู (Menu Bar) 3. ผู้ใช้บริการทำการเลือกประเภทของภาพที่กำหนด(Hashtag) ตามต้องการ 		
Alternative Scenario:	<p>3. Condition triggering an alternate scenario: ผู้ใช้บริการต้องการดูรูปภาพของเจ้าของรูปภาพนั้นๆเพิ่ม</p> <ol style="list-style-type: none"> 3.1 ผู้ใช้ทำการเลือกที่ชื่อของเจ้าของรูปภาพนั้นๆ เพื่อเข้าชมรูปภาพของเขา/เธอ 3.2 จบการทำงาน 		

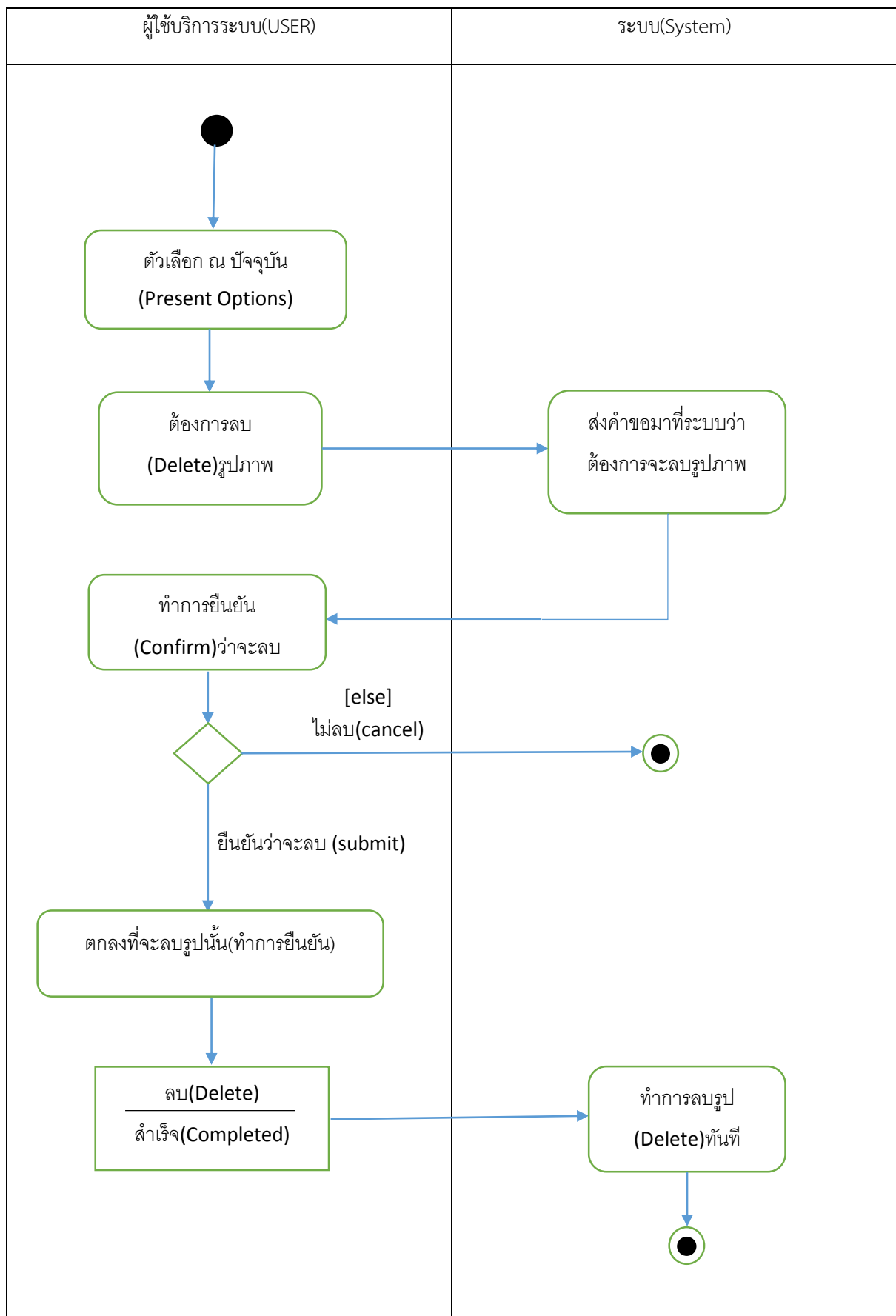
ACTIVITY DIAGRAM

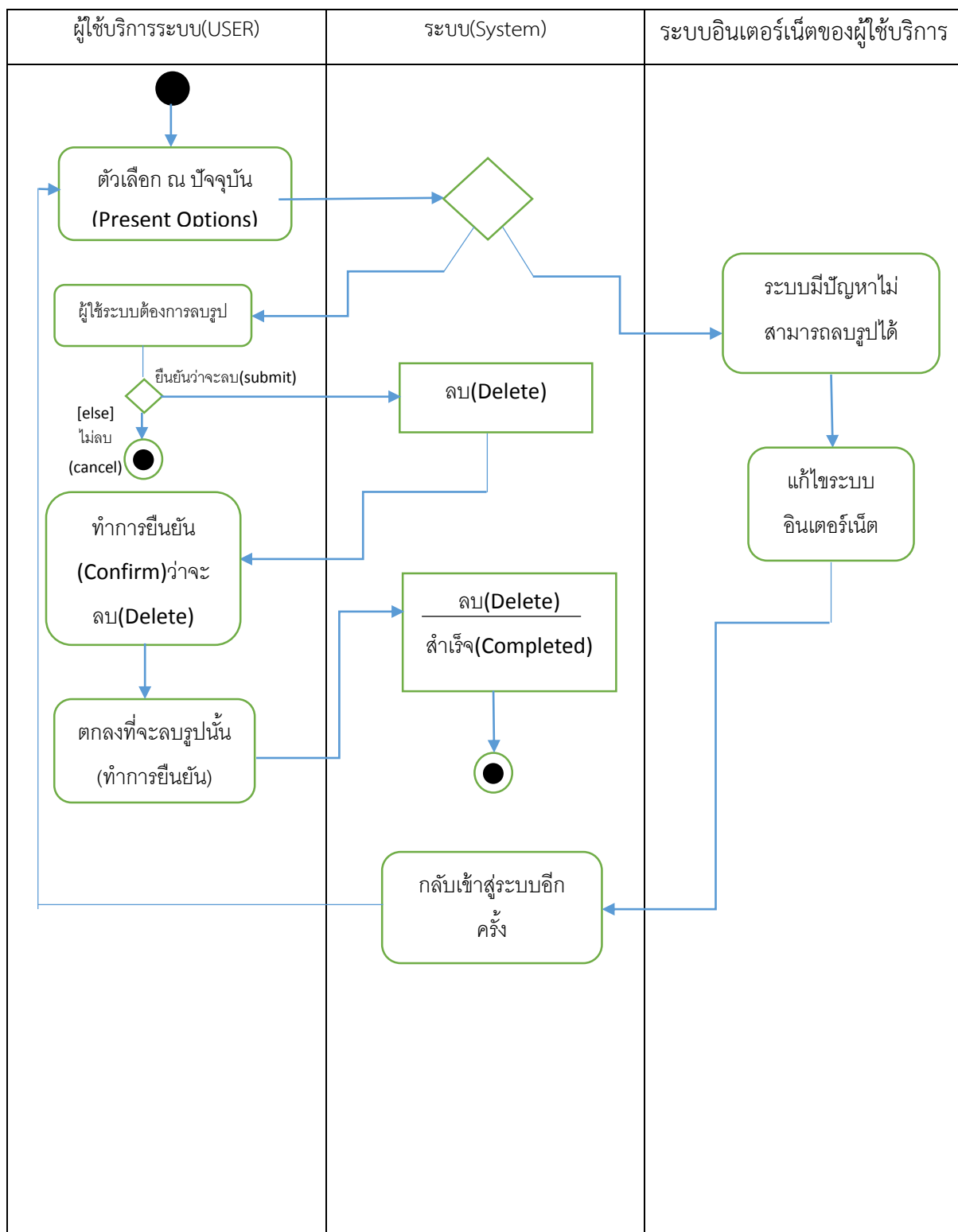
Primary Scenario Activity Diagram (Use case: Delete)

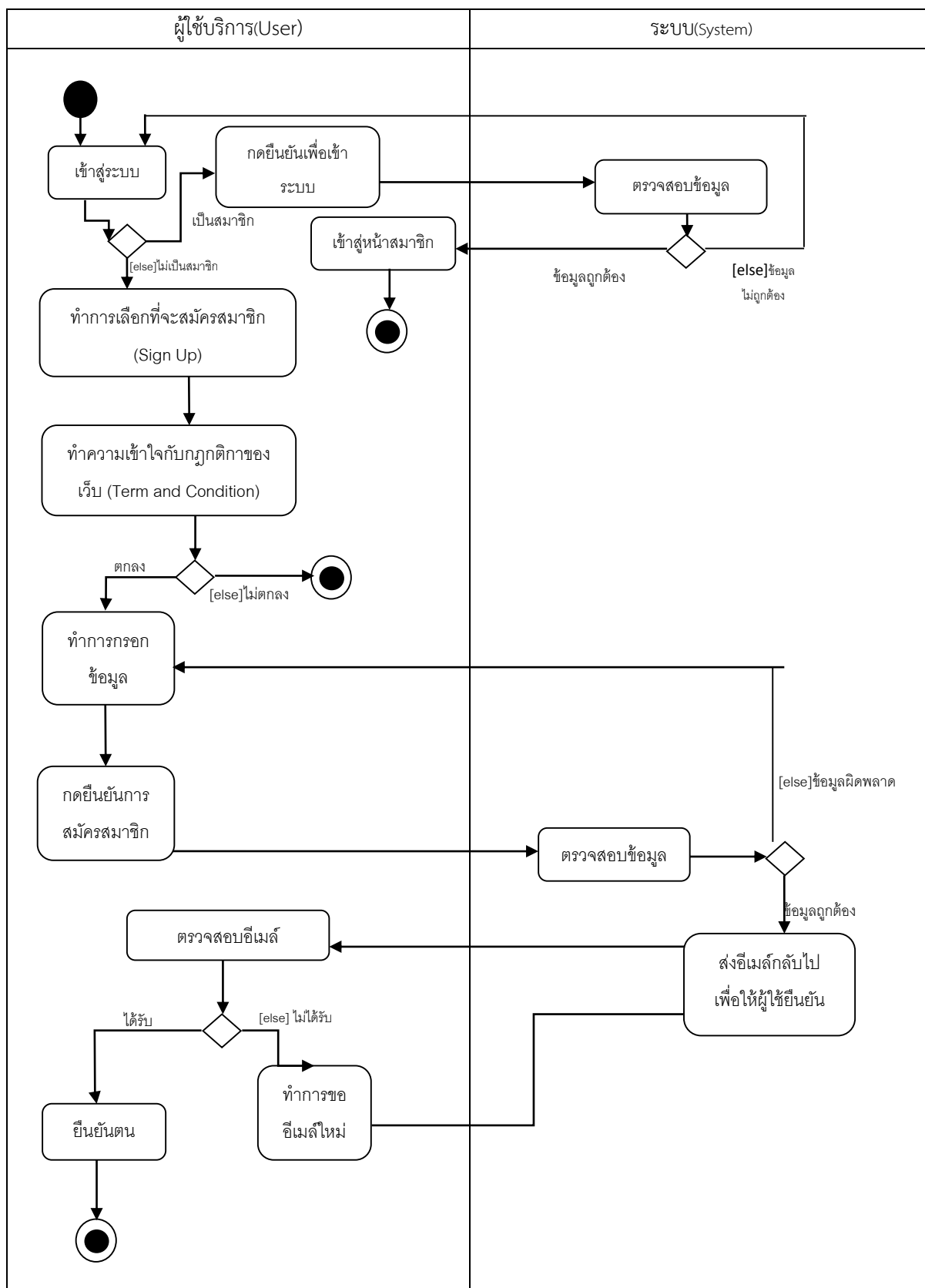
Alternate Scenario Activity Diagram (Use case: Delete)

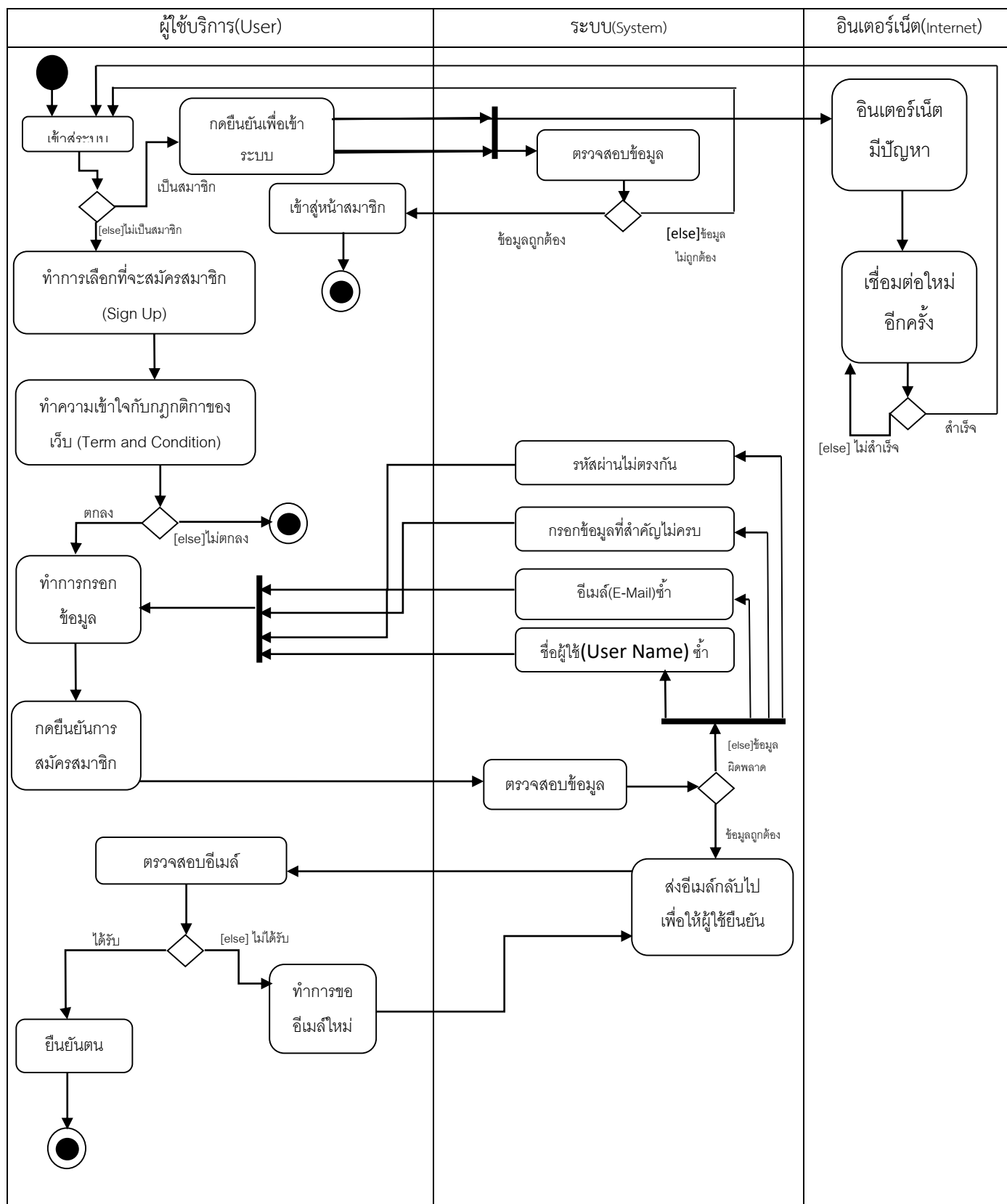
Primary Scenario Activity Diagram (Use case: Account)

Alternate Scenario Activity Diagram (Use case: Account)









สถาปัตยกรรมของระบบ

Problem Analysis

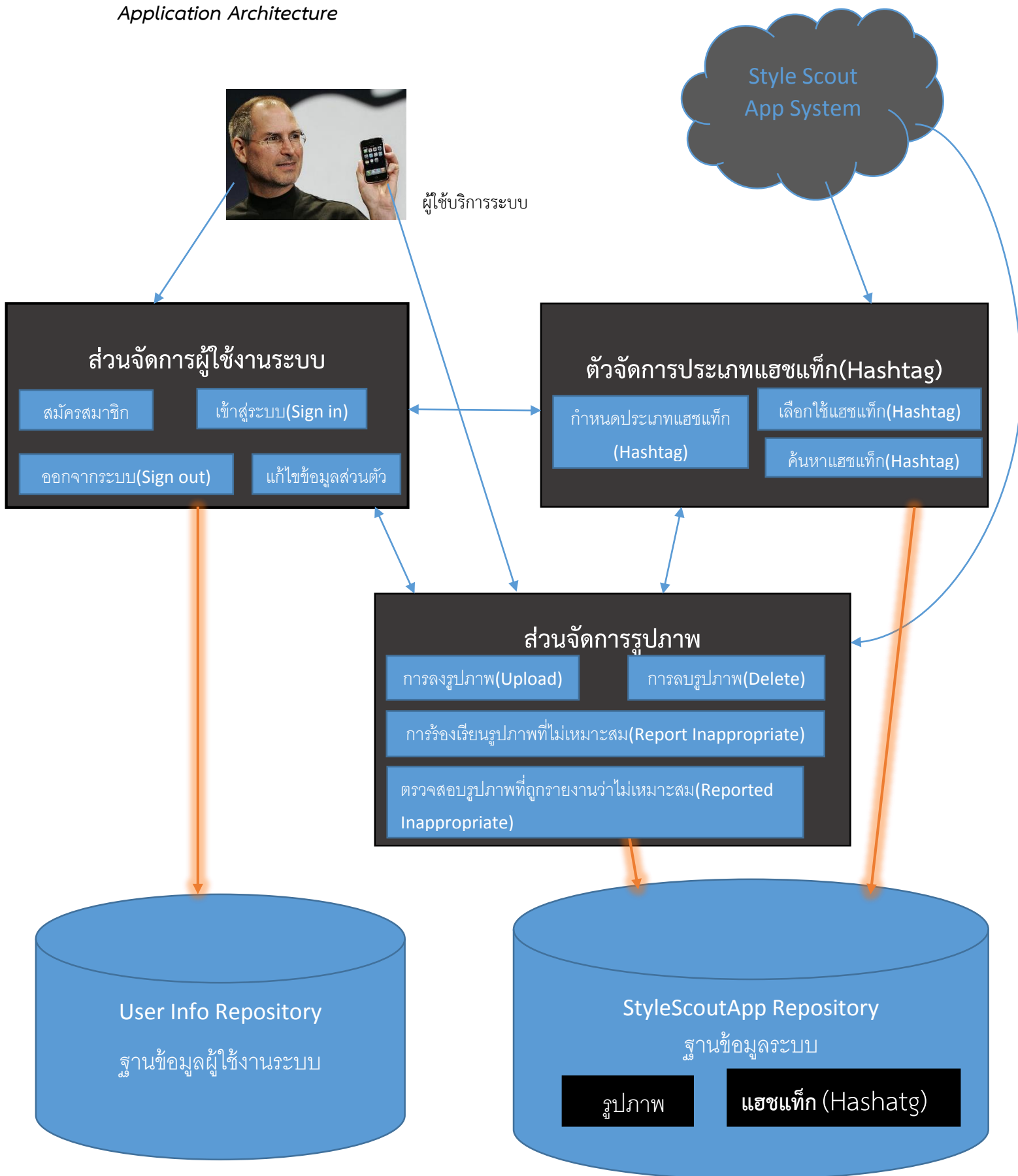
Abstraction:

1. รูปภาพ(Pictures)	ประกอบไปด้วย - ชื่อของเจ้าของภาพหรือผู้ลงภาพ(Upload) นั้นๆ - รายละเอียด(Details) ของภาพนั้นๆ - ประเภทของภาพนั้นๆ หรือที่เรียกว่า “แฮชแท็ก(Hashtag)”
2. ประเภทของรูปภาพที่ถูกกำหนดไว้ (Hashtag)	คือตัวที่บ่งชี้ประเภท(Hashtag) ของภาพนั้นๆซึ่งถูกกำหนดเอาไว้แล้วเช่น ประเภทมินิมัลลิสต์ (#Minimalist) เป็นต้น
3. บัญชีผู้ใช้(User Account)	บัญชีของผู้ที่ใช้บริการระบบ StyleScoutApp ซึ่งสามารถใช้บริการภายในระบบได้
4. ผู้ดูแลระบบ(Administrator)	ทำหน้าที่ให้บริการแก่ผู้ใช้ระบบ

Components:

1. ระบบจัดการการสมัครสมาชิก (Registration)	จัดการข้อมูลของสมาชิกที่มาลงทะเบียนใช้บริการ Style Scout App ของเราเพื่อดูแลข้อมูลส่วนตัวของบุคคลที่ทำการสมัครลงทะเบียนเป็นสมาชิก
2. ระบบจัดการการเข้าสู่ระบบ(Sign in)และออกจากระบบ(Sign out)	ระบบจัดการการเข้าสู่ระบบ(Sign in)และออกจากระบบ(Sign out) จัดทำเพื่อประโยชน์สูงสุดแก่สมาชิกในด้านความปลอดภัยในการใช้บริการ เพื่อทุกครั้งที่ใช้บริการทำการเข้าสู่ระบบ(Sign in)แล้วต้องการเลิกใช้Style Scout App ชั่วคราว สมาชิกสามารถทำการออกจากระบบ(Sign out)เพื่อทำการยกเลิกข้อมูลการเข้าสู่ระบบ(Sign in)ออกไป เพื่อความปลอดภัยในการป้องกันผู้อื่นสวมรอยทับ
3. ระบบการจัดประเภทของรูปภาพ (Hashtag)	เพื่อสร้างความเป็นระเบียบในการใช้งาน เพื่อป้องกันการตั้งประเภทรูปที่ไม่เหมาะสมและไม่เกิดความเข้าใจอันเป็นอันหนึ่งอันเดียวกัน
4. ระบบการจัดการสำหรับการลงรูปภาพ (Upload)	เพื่อให้ระบบจะได้ทำการเก็บข้อมูลรูปภาพที่สมาชิกลง(Upload) เอาไว้และไม่สูญหาย
5. ระบบการจัดการลบรูปภาพ (Delete)	เพื่อให้ระบบจะได้ทำการทราบว่าสมาชิกต้องการลบรูป แล้วทางระบบจะได้ทำการลบรูปนั้นออกจากหน้าประวัติของสมาชิก
6. ระบบการแก้ไขข้อมูลส่วนตัวสมาชิก(Edit Profile)	เมื่อสมาชิกต้องการแก้ไขข้อมูล และได้ทำการแก้ไขข้อมูลแล้ว ระบบของเราจะได้ทราบถึงการเปลี่ยนแปลงข้อมูลนั้นๆ เพื่อทำการแก้ไขในฐานข้อมูลของระบบให้มีความทันต่อเหตุการณ์ (Update) ตลอดเวลา
7. ระบบการจัดการรูปภาพที่ถูกแจ้งว่าไม่เหมาะสม(Report Inappropriate)	เพื่อทุกครั้งที่มีสมาชิกทำการแจ้งความไม่เหมาะสมในรูปภาพได้ ระบบจะได้เก็บข้อมูลไว้และผู้ดูแลระบบจะได้ทำการพิจารณาว่าสมควรแก่การลบทิ้งหรือไม่

Application Architecture



- **ส่วนจัดการผู้ใช้งานระบบ:** ในส่วนนี้จะประกอบด้วย ตัวสมัครสมาชิก(Registration) การเข้าใช้ระบบ(Sign in) ออกจากระบบ(Sign out) และการแก้ไขข้อมูลส่วนตัว(Edit Profile) โดยจะติดต่อกับฐานข้อมูลผู้ใช้งานระบบ และติดต่อกับส่วนจัดการรูปภาพในการลงรูปภาพ ลบรูปภาพ และรายงานรูปภาพที่ไม่เหมาะสม รวมทั้งส่วนจัดการผู้ใช้งานระบบนั้นจะติดต่อกับส่วนจัดการแฮชแท็ก(Hashtag)ด้วย เพื่อทำการเลือกใช้แฮชแท็ก(Hashtag)และค้นหารูปภาพจากแฮชแท็ก(Hashtag)
- **ส่วนจัดการรูปภาพ:** ประกอบด้วยการลงรูปภาพ(Upload) การลบรูปภาพ>Delete) การร้องเรียนรูปภาพที่ไม่เหมาะสม(Report Inappropriate)โดยสมาชิกหรือผู้ใช้งานระบบ และตรวจสอบรูปภาพที่ถูกรายงานว่าไม่เหมาะสม(Reported Inappropriate) โดยตัวระบบ
- **ส่วนจัดการประเภทรูปภาพ(Hashtag):** ประกอบด้วย กำหนดประเภทแฮชแท็ก(Hashtag)โดยระบบ การเลือกใช้แฮชแท็ก(Hashtag) และค้นหาแฮชแท็ก(Hashtag) โดยผู้ใช้งานระบบ
- **ฐานข้อมูลผู้ใช้งาน:** ประกอบไปด้วยข้อมูลส่วนตัวของผู้ใช้งานระบบ ชื่อบัญชีผู้ใช้(Username) รหัสผ่าน>Password) อีเมล>Email) ของผู้ใช้งาน
- **ฐานข้อมูลระบบ:** ประกอบไปด้วยรูปภาพที่สมาชิกระบบทำการลงไว้(Uploaded) ภาพที่สมาชิกต้องการลบ>Delete) ประเภทแฮชแท็ก(Hashtag)ที่กำหนดไว้โดยระบบ แฮชแท็ก(Hashtag)ที่สมาชิกเลือกใช้กับรูปภาพของตน ภาพที่ถูกร้องเรียนว่าเป็นรูปภาพที่ไม่เหมาะสม(Report Inappropriate)โดยสมาชิกหรือผู้ใช้งานระบบ และภาพที่ต้องตรวจสอบเนื่องจากถูกรายงานว่าไม่เหมาะสม(Reported Inappropriate)

Subsystem/Components

<p>1. การกำหนดประเภทของแฮชแท็ก (Hashtag)</p>	<p>รายละเอียด(Description):</p> <ul style="list-style-type: none"> - ประกอบด้วย ประเภทของแฮชแท็ก (Hashtag) ที่จะกำหนดเช่น มินิมัลลิสต์ (#Minimalist) <p>การติดต่อกันของส่วนที่เกี่ยวข้อง (Cooperation):</p> <ul style="list-style-type: none"> - เมื่อกำหนดประเภทของแฮชแท็ก (Hahstag) แล้วผู้ใช้บริการในระบบที่ต้องการทำการลงรูป(Upload)จะสามารถเลือกใช้ประเภทของแฮชแท็ก(Hahstag) นั้นๆได้
<p>2. ตรวจสอบรูปที่รายงานว่าไม่เหมาะสม (Reported Inappropriate)</p>	<p>รายละเอียด(Description):</p> <ul style="list-style-type: none"> - ประกอบด้วยลิงค์ของลบรูปภาพที่ถูก รายงานว่าไม่เหมาะสมโดยผู้ใช้บริการที่เป็นสมาชิกของระบบ <p>การติดต่อกันของส่วนที่เกี่ยวข้อง (Cooperation):</p> <ul style="list-style-type: none"> - รูปภาพที่ถูกรายนั้นเป็นรูปภาพที่ถูกนำเสนอโดยผู้ใช้บริการอื่นที่ไม่ใช่ผู้รายงานความไม่เหมาะสม(Report Inappropriate)

<p>3. การลงรูปภาพ(Upload)</p>	<p>รายละเอียด(Description):</p> <ul style="list-style-type: none"> - การลงรูปภาพ(Upload) จะถูกนำเสนอโดยสมาชิกผู้ใช้บริการของระบบซึ่งทำการเลือกรูปภาพที่ต้องการนำเสนอ โดยที่สมาชิกสามารถกำหนดประเภทของรูปภาพของตน (#Hashtag) ได้ตามความต้องการและความเหมาะสม - นอกจากนั้นผู้ใช้งานยังสามารถบอกรายละเอียดของรูปภาพนั้นได้อีกด้วยเช่นกัน <p>การติดต่อกันของส่วนที่เกี่ยวข้อง (Cooperation):</p> <ul style="list-style-type: none"> - ซึ่งแฮชแท็ก(Hashtag) จะถูกกำหนดโดยระบบ(System) - สมาชิกต้องทำการเข้าสู่ระบบ(Sign in)ก่อนทุกครั้งจึงจะสามารถลงรูปภาพ(Upload)ได้
<p>4. การลบรูปภาพ>Delete)</p>	<p>รายละเอียด(Description):</p> <ul style="list-style-type: none"> - การลบรูปภาพ>Delete) โดยสมาชิกจะสามารถทำการลบรูปภาพของตนได้เมื่อต้องการ <p>การติดต่อกันของส่วนที่เกี่ยวข้อง (Cooperation):</p> <ul style="list-style-type: none"> - โดยสมาชิกผู้นั้นจะต้องได้ทำการลงรูปภาพ(Upload)ไว้เรียบร้อยแล้วจึงจะสามารถทำการเลือกลบรูปภาพที่ต้องการลบ>Delete) ได้

	<ul style="list-style-type: none"> - สมาชิกต้องทำการเข้าสู่ระบบ(Sign in)ก่อนทุกครั้งจึงจะสามารถลบภาพ>Delete)ได้
5. การค้นหาแฮชแท็ก(Search Hashtag)	<p>รายละเอียด(Description):</p> <ul style="list-style-type: none"> - สมาชิกของระบบที่ทำการเข้าสู่ระบบเรียบร้อยแล้วสามารถเลือกชมรูปภาพของสมาชิกอื่นๆในระบบได้โดยการเลือกประเภทของรูปภาพ หรือการค้นหาแฮชแท็ก(Search Hashtag) ที่ถูกกำหนดไว้ <p>การติดต่อกันของส่วนที่เกี่ยวข้อง</p> <p>(Cooperation):</p> <ul style="list-style-type: none"> - ซึ่งแฮชแท็ก(Hashtag) จะถูกกำหนดโดยระบบ(System) - โดยรูปภาพที่อยู่ในลิสต์(List) ของการค้นหาแฮชแท็ก(Search Hashtag) นั้นจะถูกทำการลงรูปไว้(Uploaded) โดยสมาชิกในระบบ - สมาชิกต้องทำการเข้าสู่ระบบ(Sign in)ก่อนทุกครั้งจึงจะสามารถค้นหาประเภทรูปภาพ(Search Hashtag)ได้
6. การแก้ไขบัญชีส่วนตัว(Edit Profile)	<p>รายละเอียด(Description):</p> <ul style="list-style-type: none"> - ผู้ใช้งานในระบบสามารถแก้ไขข้อมูลของตนเองได้ ตามความต้องการ นอกจากนี้ยังสามารถทำการเปลี่ยนรูปของภาพบัญชีผู้ใช้ของตนเองได้(Profile Picture) อีกด้วย <p>การติดต่อกันของส่วนที่เกี่ยวข้อง</p> <p>(Cooperation):</p> <ul style="list-style-type: none"> - โดยผู้ที่จะสามารถทำการแก้ไขบัญชีส่วนตัวนั้นต้องทำการลงทะเบียนเป็นสมาชิก(Register) แล้ว และยังต้องทำการเข้าสู่ระบบ(Sign in) แล้วด้วยเช่นกัน

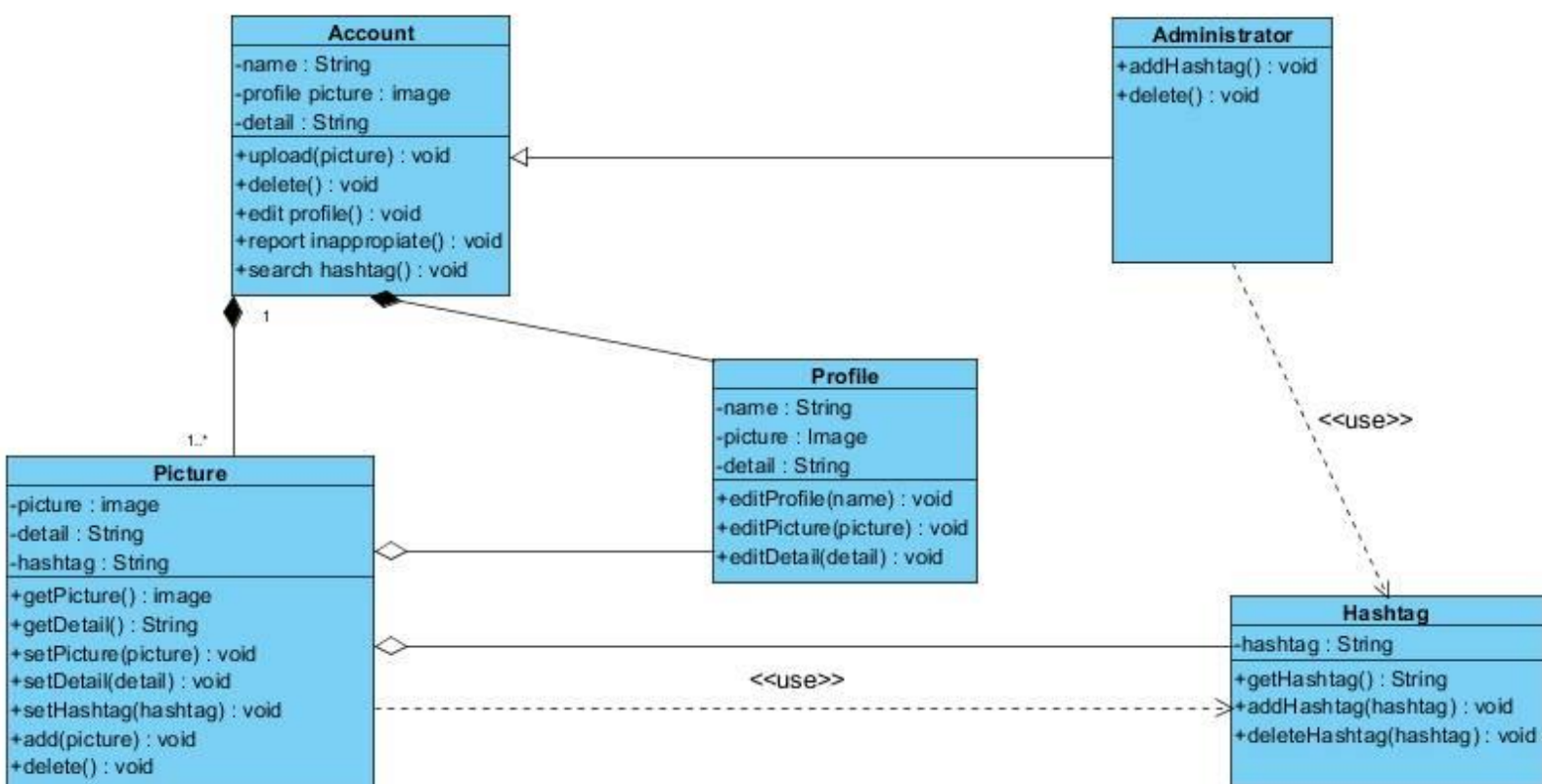
	<ul style="list-style-type: none"> - หรือกรณีผู้ใช้งานระบบได้เป็นสมาชิกระบบอยู่แล้วก็ทำการเข้าสู่ระบบ(Sign in) เพียงอย่างเดียว
7. การรายงานความไม่เหมาะสมของภาพ (Report Inappropriate)	<p>รายละเอียด(Description):</p> <ul style="list-style-type: none"> - ผู้ใช้งานระบบจะสามารถทำการแจ้งร้องเรียนกับระบบได้เมื่อพบเห็นภาพที่มีความไม่เหมาะสมโดยทำการร้องเรียนรูปภาพนั้นๆ(Report Inappropriate) โดยสามารถร้องเรียนรูปภาพของสมาชิกคนอื่นเท่านั้นไม่ใช่ของตนเอง <p>การติดต่อกันของส่วนที่เกี่ยวข้อง (Cooperation):</p> <ul style="list-style-type: none"> - โดยผู้ใช้งานจะสามารถร้องเรียนรูปภาพที่ไม่เหมาะสม(Report Inappropriate) ได้ก็ต่อเมื่อผู้ใช้บริการอื่นได้ทำการลงรูปภาพ (Upload) แล้ว
8. การเลือกใช้แฮชแท็ก(Hashtag)	<p>รายละเอียด(Description):</p> <ul style="list-style-type: none"> - สมาชิกเมื่อทำการลงรูปภาพ(Upload)ที่ต้องการแล้วต้องการการกำหนดประเภทของรูปภาพนั้นซึ่งคือแฮชแท็ก(Hashtag)ตามที่ระบบได้กำหนดเอาไว้ <p>การติดต่อกันของส่วนที่เกี่ยวข้อง (Cooperation):</p> <ul style="list-style-type: none"> - เมื่อผู้ใช้งานทำการลงรูปภาพ(Upload)แล้วจึงจะสามารถเลือกกำหนดแฮชแท็ก (Hashtag) ได้ - ระบบจะต้องทำการกำหนดแฮชแท็ก (Hashtag) ไว้แล้ว

9. การสมัครเป็นสมาชิกระบบ (Registration)	<p>รายละเอียด(Description):</p> <ul style="list-style-type: none"> - บุคคลที่ต้องการเป็นส่วนรวมและใช้บริการระบบของเราต้องทำการลงทะเบียนเป็นสมาชิกก่อนถึงจะสามารถเข้าใช้บริการได้อย่างเต็มรูปแบบ <p>การติดต่อกันของส่วนที่เกี่ยวข้อง (Cooperation):</p> <ul style="list-style-type: none"> - เมื่อทำการลงทะเบียนเป็นสมาชิกแล้วถึงจะสามารถทำการเข้าใช้ระบบ(Sign in) หรือออกจากระบบ(Sign out) - เมื่อทำการลงทะเบียนเป็นสมาชิกแล้วจึงจะสามารถใช้บริการที่ระบบมีให้คือ ลงรูปภาพ(Upload), ลบรูปภาพ>Delete), แก้ไขข้อมูลส่วนตัว(Edit Profile), ร้องเรียนภาพที่ไม่เหมาะสม(Report Inappropriate), ค้นหาแฮชแท็ก(Search Hashtag) ได้
10. การเข้าสู่ระบบ(Sign in)	<p>รายละเอียด(Description):</p> <ul style="list-style-type: none"> - เมื่อบุคคลเป็นสมาชิกระบบเรียบร้อยแล้วจะสามารถเข้าใช้บริการโดยทำการเข้าสู่ระบบ(Sign in) ได้ทุกเมื่อตามต้องการ <p>การติดต่อกันของส่วนที่เกี่ยวข้อง (Cooperation):</p> <ul style="list-style-type: none"> - บุคคลต้องเป็นสมาชิกระบบก่อนจึงจะสามารถเข้าสู่ระบบ(Sign in)ได้ - เมื่อต้องการทำการลงรูปภาพ(Upload), ลบรูปภาพ>Delete), แก้ไขข้อมูลส่วนตัว(Edit Profile), ร้องเรียนภาพที่ไม่เหมาะสม(Report Inappropriate), ค้นหาแฮชแท็ก

	(Search Hashtag) สมาชิกต้องทำการเข้าสู่ระบบก่อนทุกครั้ง
11. การออกจากระบบ(Sign out)	<p>รายละเอียด(Description):</p> <ul style="list-style-type: none"> - เมื่อสมาชิกต้องการเลิกใช้บริการชั่วคราว สมาชิกสามารถทำการออกจากระบบ(Sign out) ได้เพื่อป้องกันการสวมรอยใช้บัญชีของตัวเอง <p>การติดต่อกันของส่วนที่เกี่ยวข้อง</p> <p>(Cooperation):</p> <ul style="list-style-type: none"> - เมื่อสมาชิกได้เข้าสู่ระบบ(Sign in)แล้วจะสามารถออกจากระบบ(Sign out)ได้ตามต้องการ

แผนภาพของคลาสหลัก

Class Diagram



รายละเอียดการพัฒนาซอฟต์แวร์

Deployment

Front End:

1. HTML5 – โครงสร้างเบื้องต้นของ User Interface
2. CSS3 – ใช้ออกแบบ Interface ให้มีความสวยงามมากขึ้น
3. Bootstrap – ทำให้เว็บไซต์(Website) รองรับการแสดงผล ทั้งบนโทรศัพท์เคลื่อนที่(Mobile) และ คอมพิวเตอร์เดสก์ทอป (Desktop)

Back End:

1. Laravel (PHP Framework) – เพื่อทำให้ง่ายต่อการพัฒนาเว็บแอปพลิเคชัน (Web Application)
2. AZURE – ใช้เป็น Server ของระบบ มี VM 1ตัว IP 23.101.12.224
3. AZURE SQL – ใช้สำหรับเก็บฐานข้อมูลของระบบ

พัฒนาระบบแจ้ง ผู้ใช้งานไม่เหมาะสม (Report Inappropriate)										
ทำการตรวจสอบหา ข้อผิดพลาด(Bug) และ แก้ไขครั้งที่ 2										
ทำการรวมส่วนต่างๆ เข้าด้วยกัน										
ทดลองระบบ										



สีเหลี่ยมสีเหลือง = รับผิดชอบทั้งคู่



สีเหลี่ยมสีส้มอ่อน = นางสาวญาณิศา ยิ้มสุวรรณ เป็นคนรับผิดชอบ



สีเหลี่ยมสีส้มเข้ม = นายธนันท์ชัย ธาราศิริวัฒน์ เป็นคนรับผิดชอบ

ผลการทดสอบซอฟต์แวร์

Test Unit: Test อยู่ใน Repository Stylescout/myphp_test

Test : Upload

```

1  <?php
2
3  require_once 'ImageController.php';
4  require_once 'Image.php';
5  include_once '../upload.html';
6
7  class uploadtest extends PHPUnit_Framework_TestCase
8  {
9      public static function mockImage($hashtag,$caption,$images) {
10
11          $image= new Image();
12          $image->images($images);
13          $image->hashtag($hashtag);
14          $image->caption($caption);
15          return $image;
16      }
17      public function testAdd(){
18          $mock = uploadtest::mockImage('testHashtag','testCaption','testImage');
19          $this->assertTrue('testImage',$result->images);
20          $this->assertEquals('testHashtag',$result->hahtag);
21          $this->assertEquals('testCaption',$result->caption);
22      }
23  }
24  }
25  ?>

```

PHPUnit 4.0.17 by Sebastian Bergmann.

You have installed PHPUnit via PEAR. This installation method is no longer supported and <http://pear.phpunit.de/> will be shut down no later than December, 31 2014.

Please read <http://phpunit.de/manual/current/en/installation.html> and learn how to use PHPUnit from a PHAR or install it via Composer.

D:\OOAD ASSIGNMENT\StyleScout\myphp_test>phpunit uploadtest

Time: 154 ms, Memory: 3.75Mb

OK(2 tests, 3 assertions)

Test: Hashtag

```

1  <?php
2  require_once 'HashtagController.php';
3  require_once 'ImageController.php';
4  require_once 'Image.php';
5  require_once 'Hashtag';
6  require_once 'User';
7  include_once '../HashtagList.html';
8
9  class SeeEachImageInHashtagList extends PHPUnit_Framework_TestCase
10 {
11     public static function mockHashtag($tag,$image) {
12
13         $tag= new Hashtag();
14         $tag->pic($images);
15         $tag->hashtag($hahtag);
16         $tag->caption($caption);
17         return $image;
18     }
19     public function testTag() {
20         $mock = SeeEachImageInHashtagList::mockHashtag('testHashtag','testCaption','testImage');
21         $showHashtagPic = new repository\HashtagList();
22         $HashtagPic = new hashtagPic();
23         $HashtagPic->id = bossboss;
24         $user = new HashtagList();
25         $user->showHashtagPic= 'showHashtag';
26         $result = $showHashtagPic->findAndShowHashtag($HashtagPic,$mock,$user);
27         $this->assertTrue('testImage',$result->images);
28         $this->assertEquals('testHashtag',$result->hahtag);
29         $this->assertEquals('testCaption',$result->caption);
30     }
31 }
32 ?>

```

PHPUnit 4.0.17 by Sebastian Bergmann.

You have installed PHPUnit via PEAR. This installation method is no longer supported and <http://pear.phpunit.de/> will be shut down no later than December, 31 2014.

Please read <http://phpunit.de/manual/current/en/installation.html> and learn how to use PHPUnit from a PHAR or install it via Composer.

D:\OOAD ASSIGNMENT\StyleScout\myphp_test>phpunit SeeEachImageInHashtagList

Time: 67ms, Memory: 3.25Mb

OK<2 tests, 3 assertions>

Evaluation

“การทดลองที่1: Non-Functional Requirements: NR007 การใช้งานระบบไม่ควรมีความซับซ้อนจนเกินไป”

1. จุดประสงค์ของการทดลอง (Goal of the experiment)	เพื่อเป็นการสร้างความมั่นใจว่าแอปพลิเคชันของเรานั้นง่ายต่อการใช้งานสำหรับผู้ใช้งานทุกคนจริงๆ และเป็นการทดลองเพื่อหาจุดด้อยของอินเตอร์เฟซของระบบเราด้วยเช่นกัน
2. สิ่งที่จะวัด(Measurement metrics)	ผู้ให้บริการสามารถเข้าใจการใช้งานแอปพลิเคชัน (Application) ได้อย่างง่ายและรวดเร็ว
3. วิธีทำการทดลองและสิ่งที่ต้องใช้ในการทดลอง (Setup and methods of your experiment)	<ul style="list-style-type: none"> - ให้บุคคลที่เคยใช้งานแอปพลิเคชันหรือมีประสบการณ์การใช้งานระบบที่ใกล้เคียงมาทดลองใช้งานและดูผลตอบรับจากเขา - ให้บุคคลที่ไม่เคยใช้งานแอปพลิเคชันหรือไม่มีประสบการณ์การใช้งานระบบที่ใกล้เคียงมาทดลองใช้งานและดูว่าบุคคลนั้นสามารถทดลองใช้งานได้ อย่างสะดวกหรือไม่
4. ผลที่ได้จากการทดลอง (Experimental results)	ผู้ใช้งานทุกคนควรสามารถใช้งานอย่างสะดวกและไม่ติดขัดในการใช้งานในเรื่องการทำความเข้าใจการใช้ระบบ
5. สรุปและสิ่งที่ได้จากการทดลองนี้ (Conclusion)	หน้าการใช้งาน(Interface) สามารถรองรับการใช้งานจากบุคคลทั้งที่มีประสบการณ์ หรือไม่มีเลยก็ตามซึ่งผู้ใช้งานสามารถใช้งานอย่างง่ายที่สุด

“การทดลองที่2: Functional Requirements: FR004 สมาชิกต้องสามารถแก้ไขข้อมูลส่วนตัวได้”

1. จุดประสงค์ของการทดลอง (Goal of the experiment)	เพื่อทดสอบการใช้งานในมุมมองของผู้ใช้ระบบว่าบุคคลนั้นสามารถใช้งานในการแก้ไขข้อมูลส่วนตัวได้จริงหรือไม่ เพื่อประโยชน์สูงสุดของผู้ใช้งาน
2. สิ่งที่จะวัด (Measurement metrics)	บุคคลที่ใช้บริการระบบสามารถแก้ไขข้อมูลส่วนตัวได้เมื่อต้องการ
3. วิธีการทดลองและสิ่งที่ต้องใช้ในการทดลอง (Setup and methods of your experiment)	ทดลองใช้ระบบโดยการเข้าสู่ระบบแล้วทำการแก้ไขข้อมูลส่วนตัว แล้วบันทึก จากนั้นตรวจสอบในหน้าโปรไฟล์(Profile) ว่าข้อมูลนั้นมีการบันทึกหรือปรับปรุง (Update) แล้วหรือยัง
4. ผลที่ได้จากการทดลอง (Experimental results)	สามารถแก้ไขข้อมูลได้
5. สรุปและสิ่งที่ได้จากการทดลองนี้ (Conclusion)	ผู้ให้บริการสามารถแก้ไขข้อมูลได้ตามต้องการ

บทสรุป

“Style Scout App” การแสดงความคิดและตัวตนในโลกปัจจุบันสามารถนำเสนอให้ผู้คนทั่วโลกเห็นได้ ไม่ว่าเราจะรู้จักกันหรือไม่ก็ตาม “Style Scout App” ต้องการเป็นสื่อกลางและเครื่องมือที่จะนำเสนอแนวคิดส่วนบุคคลที่แตกต่างแต่สวยงามเป็นตัวของตนเองมารวมกันภายใต้ “Style Scout App” ผู้ใช้งานทำการนำเสนอรูปภาพของตนโดยการ “Upload” ขึ้นไปบน “Style Scout App” และเมื่อทำการนำเสนอรูปภาพไปแล้ว ผู้ใช้งานอื่นๆก็จะสามารถสัมผัสแนวคิดของเราได้ โดยการแยกเป็นประเภท “Hashtag” ซึ่งถูกกำหนดเอาไว้แล้วโดย “Style Scout App” และเมื่อต้องการลบรูปนั้นซึ่งก็สามารถทำได้เช่นเดียวกัน

ในการเขียนเว็บแอปพลิเคชัน (Web Application) “Style Scout App” เราใช้ภาษา PHP ในการพัฒนาระบบ และ ใช้ Bootstrap ในการทำหน้าเว็บไซต์(Website or Interface) เพื่อให้เกิดความสวยงามและเหมาะสมต่อแนวคิดของ “Style Scout App”

บรรณานุกรม

<http://www.ofnisystems.com/services/validation/functional-requirements/> ,Ofni System

<http://www.uml-diagrams.org/> ,UML

http://www.jot.fm/issues/issue_2008_01/article2/ ,Carlo Caloro and Luca Vetti

Tagliati in *Journal of Object Technology*, vol. 7, no. 1, January – February 2008, pp. 85-

101