

Final Project Document & Deliverables

กลุ่ม : 2PM

ระบบจัดการห้องสมุด

(Library Management System)

GitHub repository : `Library_Management_System`

รายชื่อสมาชิก

1.นาย	พีรพล	ตั้งขวัญพาณิชย์	รหัสประจำตัว 55010877	กลุ่มเรียนที่ 2
2.นาย	ภาควัต	วงศ์แหวน	รหัสประจำตัว 55010904	กลุ่มเรียนที่ 2

Object-oriented Analysis and Design

ภาคการศึกษาที่ 1 ปีการศึกษา 2557

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	1
บทนำและรายละเอียดการวิเคราะห์หัวข้อ	1
งานที่เกี่ยวข้อง	2
ผลการวิเคราะห์ความต้องการของผู้ใช้ระบบ	
• Functional Requirements	2
• Non - Functional Requirements	2
แนวทางการใช้งาน	
• Use case diagram	3
• Use case specification	4
• Activity diagram	6
สถาปัตยกรรมของระบบ	
• Problem analysis	8
• Application Architecture	9
• Subsystems/Components	10
แผนภาพของคลาสหลัก	12
รายละเอียดการพัฒนาซอฟต์แวร์	13
ผลการทดสอบซอฟต์แวร์	15
บทสรุป	20
บรรณานุกรม	21

บทคัดย่อ

เนื่องจากห้องสมุดแต่ละแห่งนั้น จะมีบรรณารักษ์เป็นผู้ที่คอยจัดการและดูแลห้องสมุด ซึ่งในบางครั้ง การทำหน้าที่ของบรรณารักษ์ที่ให้บริการแก่ผู้ที่มาใช้บริการห้องสมุดนั้นอาจทำให้บริการได้ไม่ทั่วถึง ไม่เต็ม ประสิทธิภาพ ซึ่งอาจทำให้สิ่งเหล่านี้ถูกมองเป็นภาพลักษณ์ที่ไม่ดีแก่ห้องสมุดนั้นได้ จึงเป็นที่มาของระบบ จัดการห้องสมุด ซึ่งระบบนี้สามารถตอบสนองดูแลได้อย่างทั่วถึงและมีประสิทธิภาพแก่ผู้ที่มาใช้บริการ ห้องสมุด และรวมไปถึงบรรณารักษ์ด้วย ระบบจัดการห้องสมุดนั้นเป็น Web Application ที่มีการให้บริการ แก่ผู้ใช้งานด้านต่างๆที่เกี่ยวกับห้องสมุด เสมือนว่ามีบรรณารักษ์นั้นมาให้บริการแก่ผู้ใช้งานเอง ทำให้เกิดความ สะดวกสบายแก่ผู้ใช้งาน

บทนำและรายละเอียดการวิเคราะห์หัวข้อ

ในอดีตบรรณารักษ์จะเป็นบุคคลที่ต้องทำการให้บริการแก่ผู้ที่มาใช้งานในห้องสมุดตลอดเวลา ไม่ว่าจะเป็น การสืบค้นหนังสือ การบริการยืมหรือคืนหนังสือ เป็นต้น ซึ่งบางครั้งการทำงานของบรรณารักษ์นั้นก็จะมี ประสิทธิภาพหากมีผู้ให้บริการจำนวนมาก หรือผู้ให้บริการห้องสมุดอยากที่จะทำการยืมหนังสือสักเล่ม แต่พอ มาถึงที่ห้องสมุดปรากฏว่าไม่มีหนังสือเล่มนี้อยู่ในห้องสมุดหรือหนังสือเล่มนี้ได้ถูกผู้ให้บริการรายอื่นทำการยืม ไปแล้ว ก็จะทำให้ผู้ที่มาใช้บริการเสียเวลาโดยเปล่าประโยชน์ ดังนั้นจึงเป็นเหตุผลให้พวกเราได้เลือกที่จะทำ ระบบจัดการห้องสมุดขึ้นมา เพื่อที่จะมาแก้ปัญหาดังที่ได้กล่าวมาเบื้องต้น และให้สอดคล้องกับเทคโนโลยีที่ใช้ ในปัจจุบัน

ระบบจัดการห้องสมุด เป็น Web Application ที่ช่วยให้ผู้ที่ใช้บริการห้องสมุดนั้นมีความสะดวกสบายใน การที่จะใช้งานห้องสมุดมากขึ้น ไม่ว่าจะเป็น

1. สามารถสืบค้นหนังสือที่ต้องการหาได้ ช่วยลดในการติดต่อให้บรรณารักษ์สืบค้นให้
2. สามารถรู้สถานะของหนังสือเล่มที่ต้องการว่า มีจำนวนกี่เล่ม ถูกยืมหรือมีการจองเพื่อใช้งานหรือยัง
3. สามารถทำการจองหนังสือ หรือยืมต่อหนังสือได้ หากผู้ใช้นั้นเกิดความต้องการ ไม่ต้องเสียเวลา ในการที่ต้องมาติดต่อบรรณารักษ์ที่ห้องสมุด

ระบบจัดการห้องสมุด มีหลักการทำงานดังนี้

1. ระบบจัดการหนังสือ โดยบรรณารักษ์ต้องทำการลงทะเบียนหนังสือ พร้อมทั้งปรับปรุงข้อมูลที่มีอยู่ลง ในระบบ
2. ระบบสืบค้นหนังสือ โดยจะให้สมาชิกหรือบรรณารักษ์ค้นหาหนังสือ และตรวจสอบหนังสือว่ามีอยู่ใน ห้องสมุดหรือไม่ มีสถานะเป็นอย่างไร
3. ระบบสมาชิก โดยบุคคลทั่วไปต้องทำการลงทะเบียนเพื่อใช้ในการจอง/ยืมต่อหนังสือ รวมไปถึง สามารถตรวจสอบประวัติในการยืมของตัวเองได้

ระบบจอง/ยืมต่อหนังสือ โดยสมาชิกนั้นสามารถจอง ยืมต่อหนังสือ ได้ด้วยตนเองผ่านระบบสมาชิก อีกทั้งยังสามารถยกเลิกการจองหนังสือของตัวเองได้อีกด้วย

งานที่เกี่ยวข้อง

ระบบจัดการห้องสมุดนี้จะมีลักษณะคล้ายกับ WebOpac (161.246.37.11:80) ของสำนักหอสมุดกลาง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่สามารถสืบค้นหนังสือ การจอง/ยืมต่อหนังสือ การตรวจสอบสถานะของหนังสือ เป็นต้น แต่แตกต่างกันที่ มีระบบสมาชิก ซึ่ง WebOpac ของสำนักหอสมุดกลางนั้น จะให้เรากรอกชื่อ-สกุล และรหัสนักศึกษาในการคืน ซึ่งบางครั้งการกรอกชื่อ-สกุลนี้อาจทำให้เกิดความยุ่งยากเสียเวลาต่อผู้ใช้งาน แต่ระบบจัดการห้องสมุดนี้จะเป็นการกรอก Username และ Password ที่ได้ทำการลงทะเบียนไว้ต่อบรรณารักษ์ประจำห้องสมุดนั้น

ผลการวิเคราะห์ความต้องการของผู้ใช้ระบบ

Functional Requirements

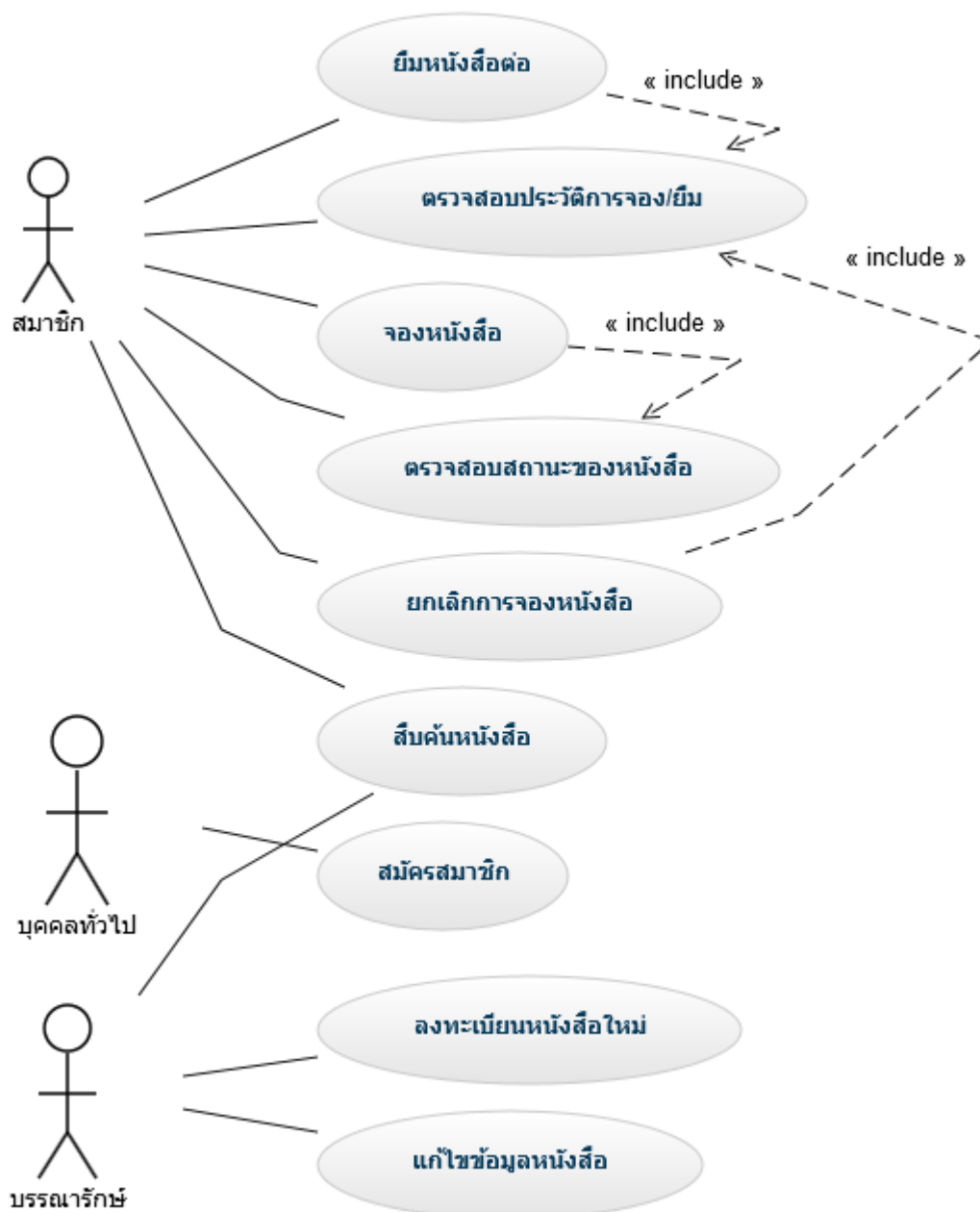
- สืบค้นหนังสือได้
- ลงทะเบียนหนังสือใหม่ได้
- บุคคลทั่วไปสามารถสมัครสมาชิกได้
- แก้ไขข้อมูลของหนังสือและข้อมูลของสมาชิกหลังจากลงทะเบียนได้
- ตรวจสอบสถานะของหนังสือได้ (หนังสือเล่มนี้มีการจอง/ยืมหรือเปล่า)
- จองหนังสือที่ต้องการใช้งานได้
- ต่อเวลาให้กับหนังสือที่ยืมมาได้
- ตรวจสอบประวัติข้อมูลการยืมหรือจองหนังสือของตัวเองได้
- สามารถยกเลิกการจองหนังสือได้

Non-Functional Requirements

- สามารถใช้งานได้ทั้งเว็บเบราว์เซอร์บนเครื่องคอมพิวเตอร์หรือสมาร์ทโฟนได้
- สามารถใช้งานได้ง่าย ไม่มีความซับซ้อนจนเกินไป
- สามารถรองรับการใช้งานของผู้ใช้อย่างน้อย 10 คนพร้อมกัน
- มีการสำรองฐานข้อมูลของผู้ใช้และข้อมูลหนังสือเพื่อป้องกันไม่ให้ข้อมูลนั้นสูญหาย
- รับรองความปลอดภัยในข้อมูลส่วนตัวของสมาชิกได้

แนวทางการใช้งาน

- Use case Diagram



● Use case Specifications

- Use Case Name:** ยืมหนังสือต่อ
- Use Case Purpose:** เป็นการยืมหนังสือต่อจากระยะเวลาเดิมที่ได้กำหนดคืนไว้ก่อนหน้านี้แล้ว โดยที่สมาชิกนั้นไม่ต้องเดินทางไปถึงที่ห้องสมุด
- Actors:** ผู้ใช้บริการที่มีบัญชีผู้ใช้งาน
- Pre-conditions:**
1. ผู้ใช้งานจะต้องเป็นสมาชิก
 2. ผู้ใช้งานจะต้องล็อกอินเข้าสู่ระบบ
 3. ผู้ใช้งานจะต้องมีหนังสือที่ได้ยืมมาก่อนหน้านี้แล้ว
 4. ระยะเวลาการยืมหนังสือที่ต้องการยืมต่อ นั้น จะต้องยังไม่หมดเวลา
- Post-conditions:** ผู้ใช้งานทำการยืมหนังสือต่อจากระยะเวลาเดิมได้สำเร็จ
- Limitations:**
1. จะไม่สามารถยืมหนังสือต่อได้หากมีเงินค้างชำระในระบบเป็นจำนวนเงิน 50 บาทขึ้นไป
 2. จะไม่สามารถยืมหนังสือต่อได้หากมีผู้อื่นทำการจองหนังสือนั้นแล้ว
 3. สมาชิกจะสามารถยืมต่อได้เพียง 1 ครั้งต่อการยืมหนังสือ 1 เล่ม
- Assumptions:** ผู้ใช้มีความต้องการยืมหนังสือต่อจริงๆ
- Primary Scenario:**
- A. ผู้ใช้งานทำการเลือกหัวข้อการยืมหนังสือต่อ
 - B. ผู้ใช้งานเลือกหนังสือที่ต้องการที่จะยืมต่อ
 - C. ผู้ใช้งานทำการยืนยันการยืมหนังสือต่อ
 - D. ระบบจะทำการแสดงผลรายการหนังสือที่ทำการยืมต่อ
 - E. เสร็จสิ้นการยืมหนังสือต่อ
- Alternative Scenario:**
- Condition triggering an alternate scenario:**
- เงื่อนไขที่ 1:** ระบบไม่อนุญาตให้ผู้ใช้ทำการยืมหนังสือต่อ
- C1. ระยะเวลาการยืมหนังสือของผู้ใช้สิ้นสุดแล้ว
 - C2. ระบบแจ้งแสดงผลให้กับผู้ใช้ทราบ
 - C3. กลับไปทำขั้นตอน B อีกครั้ง

● Use case Specifications (2)

Use Case Name: ลงทะเบียนหนังสือ

Use Case Purpose: เป็นการลงทะเบียนหนังสือใหม่ของบริษัท เพื่อบรรจุหนังสือลงฐานข้อมูลระบบ

Actors: บริษัทที่มีบัญชีผู้ใช้งาน

Pre-conditions:

1. ผู้ใช้งานทำการล็อกอินเพื่อเข้าสู่ระบบ
2. ผู้ใช้งานต้องมีข้อมูลต่างๆของหนังสือที่จะทำการลงทะเบียน

Post-conditions:

1. ผู้ใช้งานทำการลงทะเบียนหนังสือสำเร็จ
2. มีข้อมูลของหนังสือที่ได้ทำการลงทะเบียนไว้แล้วในระบบ

Limitations: ลงทะเบียนหนังสือได้ครั้งละ 1 เล่มเท่านั้น

Assumptions: ผู้ใช้ตรวจสอบข้อมูลของหนังสือเรียบร้อยแล้ว

Primary Scenario:

- A. ผู้ใช้งานทำการเลือกหัวข้อลงทะเบียนหนังสือใหม่
- B. ผู้ใช้งานทำการกรอกข้อมูลของหนังสือที่จะทำการลงทะเบียน
- C. ผู้ใช้งานทำการยืนยันในการลงทะเบียนหนังสือ
- D. เสร็จสิ้นการลงทะเบียนหนังสือ

Alternative Scenario:

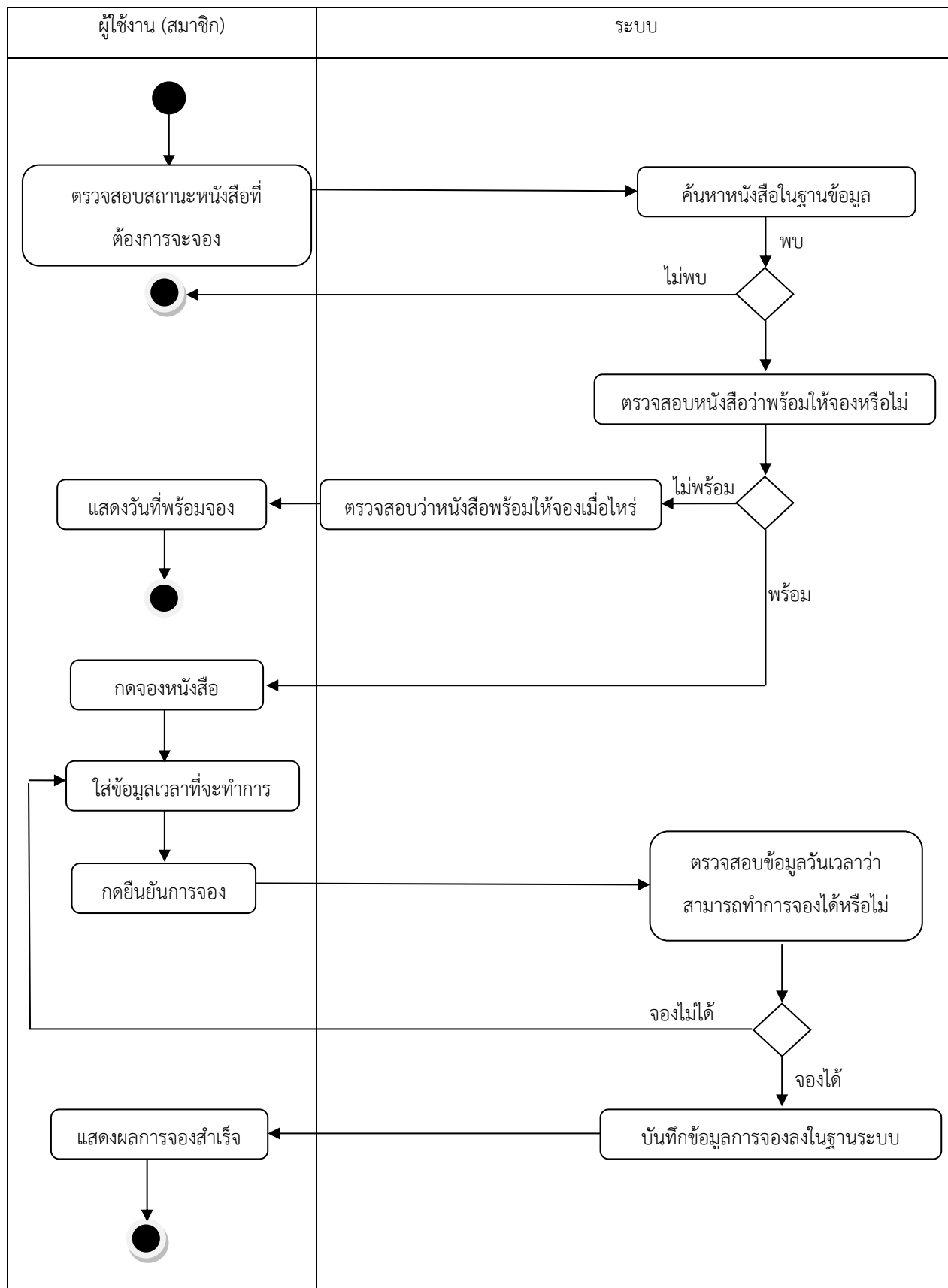
Condition triggering an alternate scenario:

เงื่อนไขที่ 1: ระบบไม่อนุญาตให้ผู้ใช้ทำการลงทะเบียนหนังสือ

- C1. ผู้ใช้กรอกข้อมูลของหนังสือที่จะลงทะเบียนไม่ครบถ้วน
- C2. ระบบแจ้งแสดงผลให้กับผู้ใช้ทราบ
- C3. กลับไปทำขั้นตอน B อีกครั้ง

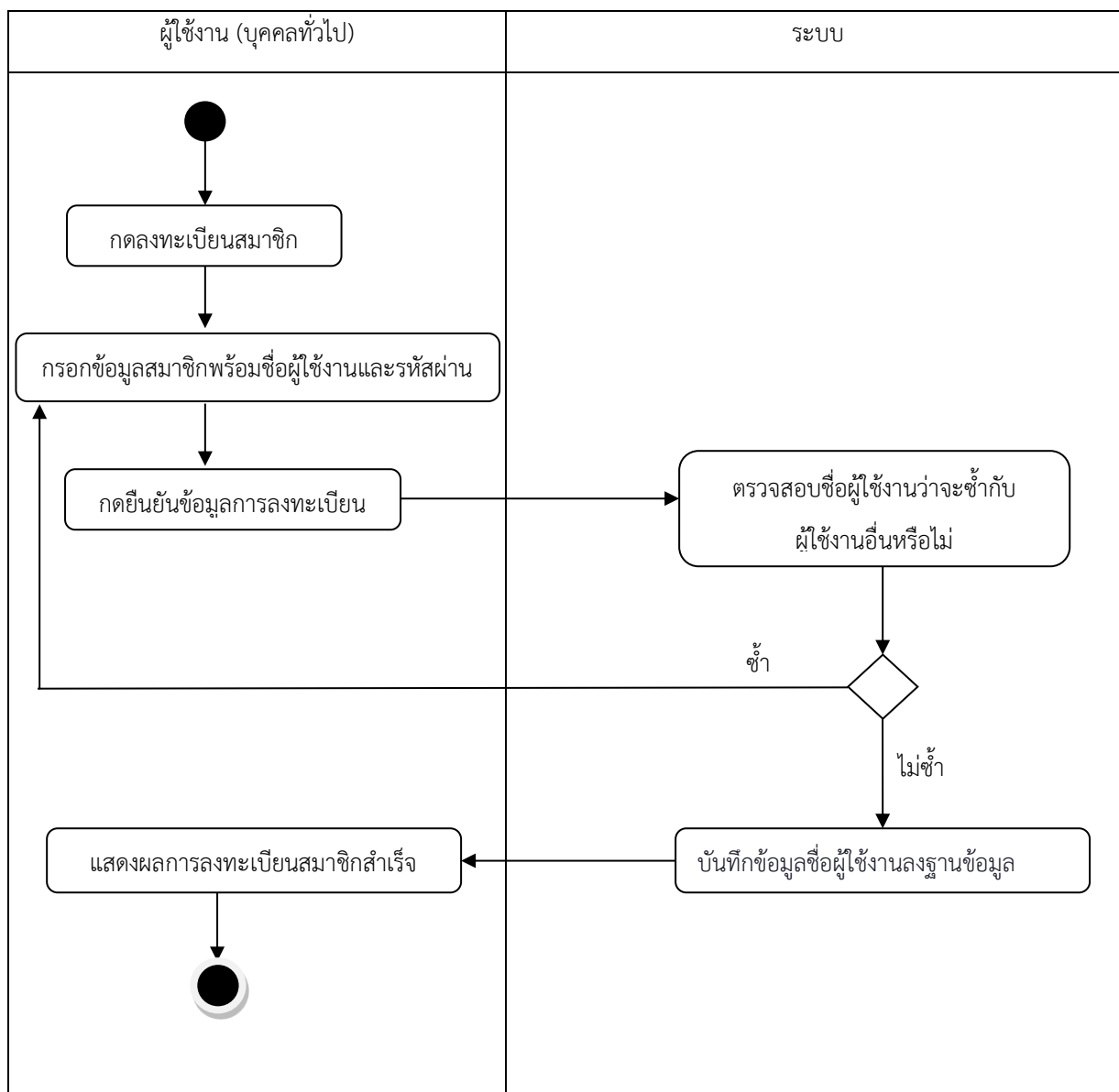
- Activity Diagram

○ (Use case: จองหนังสือ)



● Activity Diagram (2)

○ (Use case: ลงทะเบียนสมาชิก)



สถาปัตยกรรมของระบบ

● Problem Analysis

ระบบของ Library Management System นั้น มีความเกี่ยวข้องกับการใช้งานภายในห้องสมุด ซึ่งผู้ใช้งานได้แก่บรรณารักษ์กับสมาชิก และสิ่งที่ผู้ใช้งานต้องการใช้งานก็คือหนังสือ ซึ่งสามารถวิเคราะห์ส่วนของปัญหาได้ดังนี้

○ Abstraction

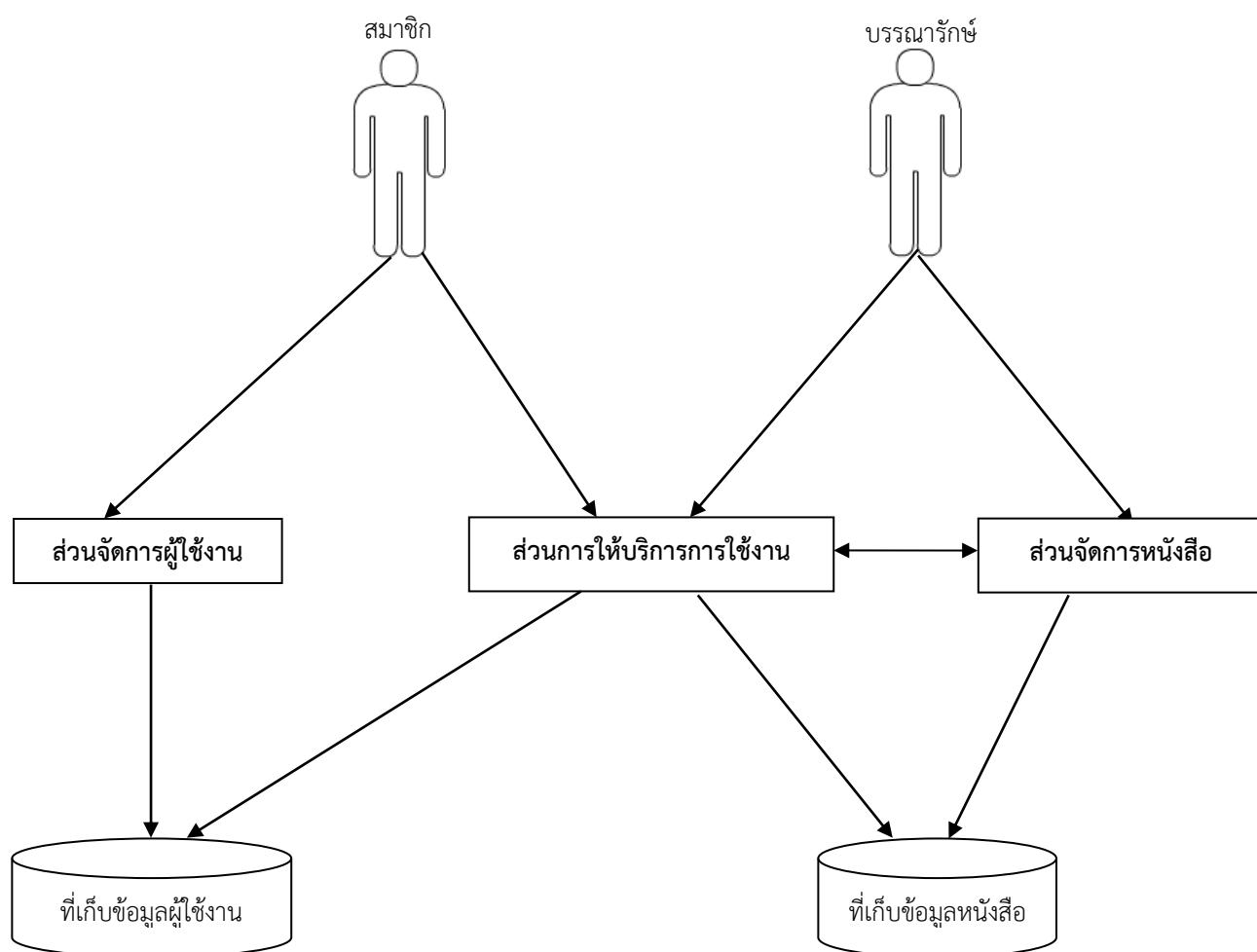
- ดัชนีหนังสือ เป็นสิ่งที่เก็บรวบรวมรายชื่อหนังสือที่มีอยู่ในห้องสมุดทั้งหมด ซึ่งจะประกอบไปด้วย บัตรรายการที่มีข้อมูลต่างๆของหนังสือแต่ละเล่ม
- หนังสือ ประกอบไปด้วย ชื่อหนังสือ ชื่อผู้แต่ง รหัสหนังสือสากล (ISBN) ฯลฯ รวมไปถึงข้อมูลการถูกยืมและคืนในแต่ละครั้ง
- สมาชิก ในสมาชิกแต่ละคนนั้นจะประกอบไปด้วย ข้อมูลส่วนตัวของสมาชิก ประวัติในการยืม จอง หรือคืนหนังสือ
- วันที่ ซึ่งจะใช้ประกอบการพิจารณาในการยืม จอง หรือคืนของหนังสือ ในแต่ละครั้ง (เช่น วันที่นี้มีหนังสือเล่มนี้ว่างสามารถที่จะทำการยืมได้ หรือมีการกำหนดส่งคืนหนังสือยืมภายในวันนี้)

○ Component

- ส่วนจัดการผู้ใช้งาน
 - ส่วนย่อยสมัครผู้ใช้งาน
 - ส่วนย่อยแก้ไขข้อมูลผู้ใช้งาน
 - ส่วนย่อยยกเลิกผู้ใช้งาน
- ส่วนการให้บริการการใช้งาน
 - ส่วนการจองหนังสือ
 - ส่วนยืมหนังสือ
 - ส่วนการต่ออายุการยืมหนังสือ (ยืมต่อ)
 - ส่วนคืนหนังสือ
 - ส่วนยกเลิกการจองหนังสือ
 - ส่วนค้นหาหนังสือ
 - ส่วนย่อยตรวจสอบประวัติการจอง/ยืม
 - ส่วนตรวจสอบค่าปรับ

- ส่วนจัดการหนังสือ
 - ส่วนลงทะเบียนหนังสือใหม่
 - ส่วนแก้ไขข้อมูลหนังสือ
 - ส่วนจำหน่ายหนังสือ (ลบหนังสือ)

- Application Architecture



- ส่วนจัดการผู้ใช้งาน

- ส่วนนี้จะประกอบไปด้วย ส่วนย่อยสมัครผู้ใช้งาน ส่วนย่อยแก้ไขข้อมูลผู้ใช้งาน และส่วนย่อยลบผู้ใช้งาน โดยส่วนจัดการผู้ใช้งานนี้จะทำการเชื่อมต่อกับที่เก็บข้อมูลผู้ใช้งาน ซึ่งผู้ใช้งาน และบรรณารักษ์จะสามารถเข้าถึงส่วนนี้ได้

○ ส่วนการให้บริการการใช้งาน

- ส่วนนี้จะประกอบไปด้วย ส่วนย่อยการจองหนังสือ ส่วนย่อยการยืมหนังสือ ส่วนย่อยการต่ออายุการยืมหนังสือ (ยืมต่อ) ส่วนย่อยการยกเลิกการจองหนังสือ ส่วนย่อยคืนหนังสือ ส่วนย่อยการค้นหาหนังสือ และส่วนย่อยตรวจสอบประวัติการจอง/ยืม โดยส่วนการให้บริการการใช้งาน จะทำการเชื่อมต่อกับที่เก็บข้อมูลหนังสือ และที่เก็บข้อมูลผู้ใช้งาน ซึ่งส่วนย่อยการจองหนังสือ ส่วนย่อยการต่ออายุการยืมหนังสือ (ยืมต่อ) ส่วนย่อยการยกเลิกการจองหนังสือ ส่วนย่อยการค้นหาหนังสือ ส่วนย่อยตรวจสอบค่าปรับ และส่วนย่อยตรวจสอบประวัติการจอง/ยืม สมาชิกจะเข้าถึงส่วนนี้ได้ และส่วนย่อยการยืมหนังสือ ส่วนย่อยคืนหนังสือ ส่วนย่อยตรวจสอบค่าปรับ และส่วนย่อยการค้นหาหนังสือ บรรณารักษ์จะเข้าถึงส่วนนี้ได้

○ ส่วนจัดการหนังสือ

- ส่วนนี้จะประกอบไปด้วย ส่วนย่อยลงทะเบียน ส่วนย่อยแก้ไขข้อมูล และส่วนย่อยจำหน่ายออก (ลบหนังสือ) โดยส่วนจัดการหนังสือนี้ จะทำการเชื่อมต่อกับที่เก็บข้อมูลหนังสือ ซึ่งบรรณารักษ์เท่านั้นที่จะเข้าถึงส่วนนี้ได้

● Subsystems / Components

○ ส่วนจัดการผู้ใช้งาน

- ส่วนย่อยสมัครผู้ใช้งาน มีหน้าที่ในการจัดเก็บข้อมูลต่างๆของผู้ใช้งาน ลงในที่เก็บข้อมูลผู้ใช้งาน
- ส่วนย่อยแก้ไขข้อมูลผู้ใช้งาน มีหน้าที่ในการเปลี่ยนแปลงหรือแก้ไขข้อมูลทั่วไปของผู้ใช้งาน ลงในที่เก็บข้อมูลผู้ใช้งาน
- ส่วนย่อยยกเลิกผู้ใช้งาน มีหน้าที่ในการลบข้อมูลของผู้ใช้งานนั้น จากที่เก็บข้อมูลผู้ใช้งาน

○ ส่วนจัดการหนังสือ

- ส่วนย่อยลงทะเบียนหนังสือ มีหน้าที่ในการจัดเก็บข้อมูลจำเพาะของหนังสือ ลงในที่เก็บข้อมูลหนังสือ
- ส่วนย่อยแก้ไขข้อมูลหนังสือ มีหน้าที่ในการเปลี่ยนแปลงข้อมูลหรือแก้ไขข้อมูลจำเพาะของหนังสือ ลงในที่เก็บข้อมูลหนังสือ
- ส่วนย่อยจำหน่ายหนังสือ มีหน้าที่ในการลบข้อมูลของหนังสือเล่มนั้น ซึ่งจะทำการลบออกจากที่เก็บข้อมูลหนังสือ

○ ส่วนการให้บริการการใช้งาน

- ส่วนย่อยการจองหนังสือ มีหน้าที่ในการจัดเก็บข้อมูลการจองของสมาชิก ลงในที่เก็บข้อมูลหนังสือ โดยก่อนที่จะทำการจองได้ จะมีการตรวจในที่เก็บข้อมูลหนังสือ ว่า มีหนังสือที่ต้องการจองนั้นถูกยืมในช่วงเวลานั้นหรือเปล่า หากมีสมาชิกจะไม่สามารถทำการจองได้
- ส่วนย่อยหนังสือ มีหน้าที่ในการจัดเก็บข้อมูลการยืมของสมาชิก ลงในที่เก็บข้อมูลหนังสือ โดยก่อนที่จะทำการยืมได้ จะมีการตรวจในที่เก็บข้อมูลหนังสือ ว่า หนังสือที่ต้องการยืมนั้น สมาชิกที่ทำการยืมได้จองไว้ หรือว่าไม่มีผู้ใช้งานหรือในช่วงเวลานั้นหรือเปล่า หากไม่ว่าง สมาชิกจะไม่สามารถทำการยืมได้
- ส่วนย่อยการต่ออายุการยืมหนังสือ มีหน้าที่ในการต่อเวลาในการยืมหนังสือที่เราได้เคยยืมมา โดยจะทำการแก้ไขเวลาในการยืมในที่เก็บข้อมูลหนังสือ โดยก่อนที่จะทำการยืมต่อได้นั้น จะมีการตรวจในที่เก็บข้อมูลหนังสือว่า เวลาที่ทำการยืมต่อ่นั้นเป็นเวลาที่ว่า (ไม่มีคนจอง) หรือเปล่า หากมีสมาชิกจะไม่สามารถทำการยืมต่อได้
- ส่วนย่อยการยกเลิกการจองหนังสือ มีหน้าที่ในการยกเลิกการจองหนังสือของสมาชิก ซึ่งจะทำการลบข้อมูลการจองของสมาชิกในที่เก็บข้อมูลหนังสือ
- ส่วนย่อยตรวจสอบประวัติการจอง/ยืม มีหน้าที่ในการตรวจสอบประวัติของสมาชิกที่ได้ทำการการจอง/ยืมในระยะเวลาก่อนคืน โดยจะตรวจสอบจากที่เก็บข้อมูลหนังสือ
- ส่วนย่อยสืบค้นหนังสือ มีหน้าที่ในการหาหนังสือตามที่สมาชิกนั้นต้องการ ซึ่งจะทำการค้นหาในที่เก็บข้อมูลหนังสือ
- ส่วนย่อยตรวจสอบค่าปรับ มีหน้าที่ในการตรวจสอบค่าปรับ ซึ่งเกิดจากการยืมหนังสือเกินเวลาที่กำหนด ซึ่งจะทำการตรวจสอบในที่เก็บข้อมูลผู้ใช้งาน
- ส่วนย่อยคืนหนังสือ มีหน้าที่ในการยกเลิกการยืมหนังสือของสมาชิก ซึ่งจะทำการลบข้อมูลการยืมของสมาชิกในที่เก็บข้อมูลหนังสือ

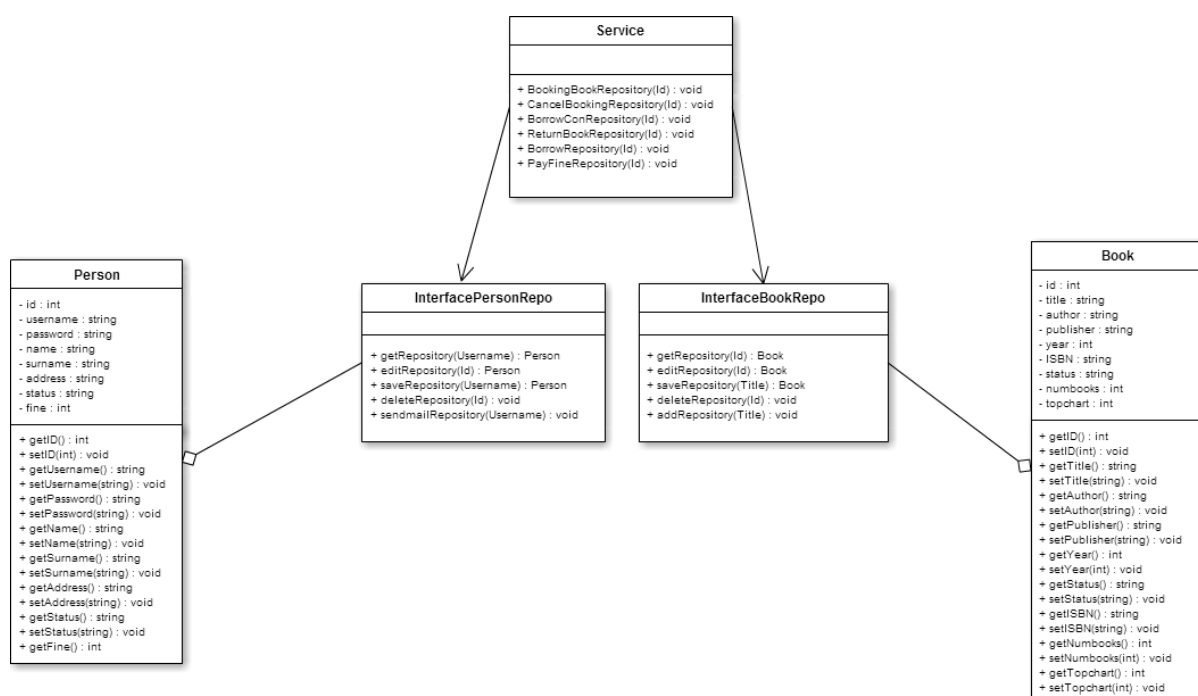
○ ที่เก็บข้อมูลผู้ใช้งาน

- ซึ่งจะประกอบไปด้วย เลข id ซึ่งเป็น Primary Key ,ชื่อผู้ใช้งาน ,รหัสผ่าน ,ข้อมูลส่วนตัวที่จำเป็นของสมาชิก รวมไปถึงค่าปรับที่ค้างไว้ที่ยังไม่ได้ชำระ

○ ที่เก็บข้อมูลหนังสือ

- ซึ่งจะประกอบไปด้วย เลข id ซึ่งเป็น Primary Key ,ข้อมูลจำเพาะที่เกี่ยวข้องกับหนังสือ ,สถานะของหนังสือเล่มนี้ (เช่น ถูกยืมแล้ว ,อยู่บนชั้น ฯลฯ) , id ของสมาชิกที่ได้ทำการจอง ,เลข id ของหนังสือที่ถูกจอง ,วันที่และเวลาที่สมาชิกนั้นจะมารับหนังสือ ,เลข id ของหนังสือที่ถูกยืม ,วันที่และเวลาที่สมาชิกเริ่มยืมหนังสือ และกำหนดคืนหนังสือ

แผนภาพคลาสหลัก



คำอธิบาย Domain Class นี้

- ระบบการจัดการห้องสมุดนี้ จะประกอบไปด้วย Class หลักๆคือ Class Person และ Class Book โดยจะมี Class ที่เป็น Aggregation ของ Class Person และ Class Book คือ Class InterfacePersonRepo และ Class InterfaceBookRepo ตามลำดับ
- Class Service จะเป็น Class ที่เรียกใช้งาน Class InterfacePersonRepo และ Class InterfaceBookRepo เป็น Class ที่จัดการส่วนให้บริการเกี่ยวกับหนังสือทั้งหมด
- Class InterfacePersonRepo จะเป็น Class ที่ใช้จัดการเกี่ยวกับส่วนต่างๆ ของ ผู้ใช้ (Person) ไม่ว่าจะเป็น การบันทึกข้อมูล การเรียกใช้ข้อมูล การลบข้อมูล เป็นต้น
- Class InterfaceBookRepo จะเป็น Class ที่ใช้จัดการเกี่ยวกับส่วนต่างๆ ของ หนังสือ (Book) ไม่ว่าจะเป็น การบันทึกข้อมูล การเรียกใช้ข้อมูล การลบข้อมูล การเพิ่มข้อมูล เป็นต้น
- Class Person จะเป็น Class ที่เป็นตัวแทนของผู้ใช้งานแต่ละคน ซึ่งจะมีตัวแปรเก็บข้อมูลต่างๆที่เกี่ยวข้องกับผู้ใช้งาน
- Class Book จะเป็น Class ที่เป็นตัวแทนของหนังสือ ซึ่งจะมีข้อมูลเกี่ยวกับหนังสือ รวมไปถึงมีสถานะประจำแต่ละเล่ม (เช่น ไม่ว่าง, กำลังซ่อมแซม ฯลฯ)

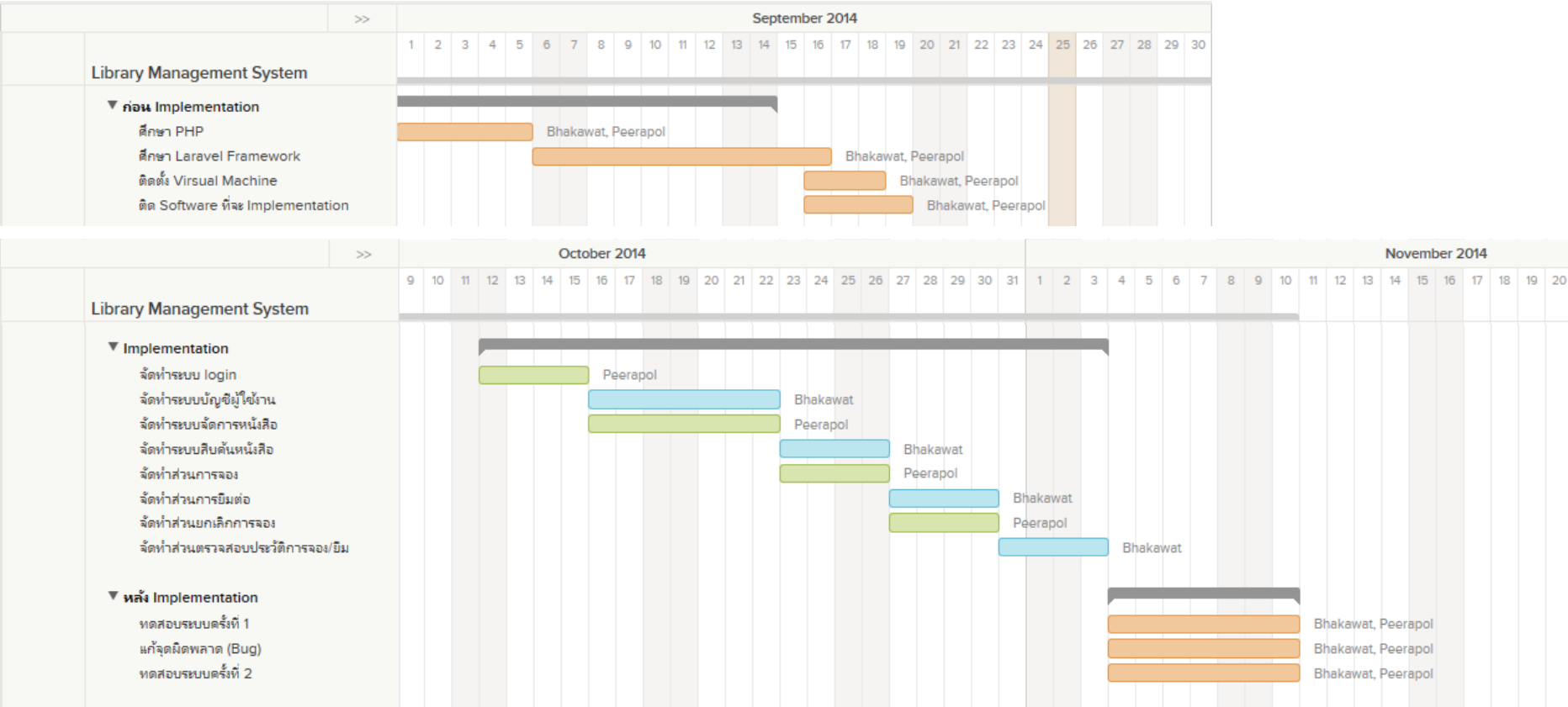
รายละเอียดการพัฒนาซอฟต์แวร์

● Deployment

ระบบทั้งหมดนี้จะอยู่ลงใน Microsoft Azure Pass ซึ่งเป็น Virtual Machine จำนวน 1 ตัว โดยมี Domain name ชื่อ library2pm.cloudapp.net (หมายเลข IP: 23.101.27.110) ซึ่งใช้ระบบปฏิบัติการ Ubuntu Server 14.04 LTS ซึ่ง Virtual Machine นี้จะใช้ Software และ Technology ดังต่อไปนี้

- ใช้ Apache2 เป็น Web application server ของระบบ
- ใช้ MySQL เป็นฐานข้อมูลของระบบทั้งหมด
- ใช้ phpmyadmin เป็นตัวจัดการเกี่ยวกับฐานข้อมูลทั้งหมด
- ใช้ PHP เป็นภาษาที่ใช้สำหรับการพัฒนา Web application
- ใช้ laravel ซึ่งเป็น PHP framework ใช้ในการพัฒนาระบบ
- ใช้ phpunit ซึ่งเป็น PHP testing framework ใช้ในทดสอบการทำงานส่วนต่างๆของระบบ
- ใช้ bootstrap เป็น Frontend Framework ของระบบ
- ใช้ HTML ,CSS และ Javascript ในการตกแต่ง User Interface ของระบบ

● Implementation Plan



ผลการทดสอบซอฟต์แวร์

การทดสอบ (Test)

Directory ของ Test file ใน Repository คือ app/tests โดยใน Directory นี้จะมี Test file จำนวน 3 files คือ

Test file ที่ 1: การ Test การลงทะเบียนสมาชิก (app/tests/registUserTest.php)

- Code ที่เขียนเพื่อทำการ Test มีดังนี้

```
<?php
class registUserTest extends TestCase {

    public static function mockRegisUser($inUsername,$inPassword,$inName,$inSurname,$inAddress,$inEmail,$inPhone,$inStatus){
        $newPerson = new Person;
        $newPerson->setUsername($inUsername);
        $newPerson->setPassword($inPassword);
        $newPerson->setName($inName);
        $newPerson->setSurname($inSurname);
        $newPerson->setAddress($inAddress);
        $newPerson->setEmail($inEmail);
        $newPerson->setPhone($inPhone);
        $newPerson->setStatus($inStatus);
        return $newPerson;
    }

    public static function mockServiceUser(){
        $newService = new InterfacePersonRepo;
        return $newService;
    }

    public function testRegisUser(){
        $Username = 'TestGuest';
        $Password = 'guest';
        $Name = 'Phaholyothin';
        $Surname = 'Chaengwatthana';
        $Address = 'Donmuaeng Bangkok';
        $Email = 'test@kmitl.ac.th';
        $Phone = '02-534-1000';
        $Status = 'ADMIN';

        $test = registUserTest::mockRegisUser($Username,$Password,$Name,$Surname,$Address,$Email,$Phone,$Status);
        $service = registUserTest::mockServiceUser();
        $result = $service->saveRepository($test);

        $this->assertEquals($Username,$result->Username);
        $this->assertEquals($Password,$result->Password);
        $this->assertEquals($Name,$result->Name);
        $this->assertEquals($Surname,$result->Surname);
        $this->assertEquals($Address,$result->Address);
        $this->assertEquals($Email,$result->Email);
        $this->assertEquals($Phone,$result->Phone);
        $this->assertEquals($Status,$result->Status);
    }
}
```

- ผลของการ test ได้ดังนี้

```
web@ubuntu:/var/www/html/laravel$ phpunit
PHPUnit 3.7.28 by Sebastian Bergmann.

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

.

Time: 161 ms, Memory: 10.25Mb

OK (1 test, 8 assertions)
web@ubuntu:/var/www/html/laravel$
```

Test file ที่ 2: การ Test การจัดการหนังสือ (แก้ไขหนังสือ) (app/tests/editBookTest.php)

- Code ที่เขียนเพื่อทำการ Test มีดังนี้

```
<?php
class editBookTest extends TestCase {

    public static function mockEditBook($inBookID,$inTitle1,$inTitle2,$inSubject,$inAuthor,$inPublishing,$inEdition,$inYear,$inDetail,$inISBN
,$inCallNum,$inType,$inStatus)
    {
        $editBook = new Book;
        $editBook->setBookIDold($inBookID);
        $editBook->setBookID($inBookID);
        $editBook->setTitle1($inTitle1);
        $editBook->setTitle2($inTitle2);
        $editBook->setSubject($inSubject);
        $editBook->setAuthor($inAuthor);
        $editBook->setPublishing($inPublishing);
        $editBook->setEdition($inEdition);
        $editBook->setYear($inYear);
        $editBook->setDetail($inDetail);
        $editBook->setISBN($inISBN);
        $editBook->setCallNum($inCallNum);
        $editBook->setType($inType);
        $editBook->setStatus($inStatus);
        return $editBook;
    }

    public static function mockServiceBook(){
        $newService = new InterfaceBookRepo;
        return $newService;
    }

    public function testEditBook(){

        $BookID = 39;
        $Title1 = 'ศตวรรษภาพเหนือขอบฟ้า'; //Title(Thai)
        $Title2 = 'Power Beyond the Horizon'; //Title(English)
        $Subject = 'การขุดค้น';
        $Author = 'Prajin Jayema';
        $Publishing = 'RTAF';
        $Edition = 3;
        $Year = 2557;
        $Detail = 'การขุดค้น';
        $ISBN = '936-534-002';
        $CallNum = '500 ส 49';
        $Type = 600;
        $Status = 'Borrow';

        $test = editBookTest::mockEditBook($BookID,$Title1,$Title2,$Subject,$Author,$Publishing,$Edition,$Year,$Detail,$ISBN,$CallNum,$Type,$Status);
        $service = editBookTest::mockServiceBook();
        $result = $service->editBookRepository($test);

        $this->assertEquals($BookID,$result->id);
        $this->assertEquals($Title1,$result->Title1);
        $this->assertEquals($Title2,$result->Title2);
        $this->assertEquals($Subject,$result->Subject);
        $this->assertEquals($Author,$result->Author);
        $this->assertEquals($Publishing,$result->Publishing);
        $this->assertEquals($Edition,$result->Edition);
        $this->assertEquals($Year,$result->Year);
        $this->assertEquals($Detail,$result->Detail);
        $this->assertEquals($ISBN,$result->ISBN);
        $this->assertEquals($CallNum,$result->CallNum);
        $this->assertEquals($Type,$result->Type);
        $this->assertEquals($Status,$result->Status);
    }
}
```

- ผลของการ test ได้ดังนี้

```
web@ubuntu:/var/www/html/laravel$ phpunit
PHPUnit 3.7.28 by Sebastian Bergmann.

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

.

Time: 117 ms, Memory: 10.00Mb

OK (1 test, 13 assertions)
web@ubuntu:/var/www/html/laravel$
```

Test file ที่ 3: การ Test การจองหนังสือ (app/tests/bookingTest.php)

- Code ที่เขียนเพื่อทำการ Test มีดังนี้

```
<?php
class bookingTest extends TestCase {

    public static function mockBooking($idBook,$idUser,$day,$month,$year){

        $BookingBook = new Book;
        $toRepository = new Service;
        $BookingBook->setBookID($idBook);
        $BookingBook->setUserBooking($idUser);

        $dateNow=date("d-m-Y");
        $dateBooking = $day."-".$month."-".$year;

        $BookingBook->setDateBooking($dateNow);
        $BookingBook->setDateBorrow($dateBooking);
        $returnValue = $toRepository->BookingBookRepository($BookingBook);
        return $returnValue;
    }

    public static function mockServiceBook(){
        $newService = new InterfaceBookRepo;
        return $newService;
    }

    public function testBooking(){
        $day = 16;
        $month = 11;
        $year = 2014;
        $idBook = 39;
        $idUser = 2;

        $bookingTest::mockBooking($idBook,$idUser,$day,$month,$year);
        $service = $bookingTest::mockServiceBook();
        $result = $service->getRepositoryByID($idBook);

        $dateBooking = $day."-".$month."-".$year;
        $this->assertEquals($idBook,$result->id);
        $this->assertEquals($idUser,$result->UserBooking);
        $this->assertEquals($dateBooking,$result->DateBorrow);
    }
}
?>
```

- ผลของการ test ได้ดังนี้

```
web@ubuntu:/var/www/html/laravel$ phpunit
PHPUnit 3.7.28 by Sebastian Bergmann.

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

.

Time: 122 ms, Memory: 10.00Mb

OK (1 test, 3 assertions)
web@ubuntu:/var/www/html/laravel$
```

การประเมินผล (Evaluation)

การทดลองที่ 1: การจองหนังสือของสมาชิก

- จุดประสงค์ของการทดลอง (Goal of the experiment)

สมาชิกสามารถจองหนังสือที่ต้องการจะยืมได้เพื่อที่จะไม่พลาดโอกาสในการที่จะยืมหนังสือเล่มที่ต้องการ และไม่เสียเวลาในการไม่พบหนังสือเล่มนั้นเมื่อสมาชิกไปที่ห้องสมุด

- สิ่งที่จะวัด (Measurement metrics)

สมาชิกสามารถจองหนังสือที่ต้องการจะยืมได้ และหลังจากทำการจองหนังสือแล้วก็จะมีการแสดงประวัติการจองในระบบ และสถานะของหนังสือที่สมาชิกได้ทำการจองต้องถูกเปลี่ยนว่า “กำลังถูกจอง”

- วิธีการทดลองและสิ่งที่ต้องใช้ในการทดลอง (Setup and methods of your experiment)

- วิธีการทดลอง

- สมาชิกต้องทำการเข้าสู่ระบบเพื่อใช้งานก่อน
- เสร็จแล้วก็ไปที่หัวข้อเกี่ยวกับการสืบทอดหนังสือ และสืบทอดหนังสือที่ต้องการ
- หลังจากได้หนังสือที่ต้องการแล้ว ก็ทำการจองหนังสือนั้น
- เลือกวันที่ที่ต้องการไปรับหนังสือ หากเรียกร้อยแล้วก็ยืนยันการทำรายการ

- สิ่งที่ต้องการใช้ในการทดลอง

- บัญชีผู้ใช้งานของสมาชิก
- ชื่อเรื่องของหนังสือที่ต้องการจะทำการจองหนังสือ
- วันเดือนปีที่จะใช้ในการจองหนังสือ

- ผลที่ได้จากการทดลอง (Experimental results)

หนังสือเล่มที่สมาชิกได้ทำการจองไว้นั้นจะถูกเปลี่ยนสถานะจากเดิม “อยู่บนชั้นหนังสือ” เป็นสถานะ “ถูกจอง” แทน และสมาชิกคนอื่นๆ จะไม่สามารถจองหรือยืมได้หลังจากนี้

- สรุปและสิ่งที่ได้จากการทดลองนี้ (Conclusion)

สมาชิกสามารถทำการจองหนังสือที่ต้องการจะยืมได้

การทดลองที่ 2: การจัดการหนังสือ (แก้ไขข้อมูลหนังสือ)

- จุดประสงค์ของการทดลอง (Goal of the experiment)

บรรณารักษ์สามารถแก้ไขข้อมูลของหนังสือที่ต้องการได้ เพื่อที่จะได้แก้ไขข้อมูลของหนังสือที่ได้มีความผิดพลาดในตอนลงทะเบียน หรือมีการเพิ่มเติมข้อมูลที่เกี่ยวข้องกับหนังสือข้อมูลที่เป็นปัจจุบัน

- สิ่งที่จะวัด (Measurement metrics)

บรรณารักษ์สามารถแก้ไขข้อมูลของหนังสือที่ต้องการได้ และหลังจากแก้ไขแล้วก็จะสามารถเห็นข้อมูลที่แก้ไขได้ในรายละเอียดของหนังสือเล่มนั้น

- วิธีการทดลองและสิ่งที่ต้องใช้ในการทดลอง (Setup and methods of your experiment)

- วิธีการทดลอง

- บรรณารักษ์ต้องทำการเข้าสู่ระบบเพื่อใช้งานก่อน
- เสร็จแล้วก็ไปที่หัวข้อเกี่ยวกับการสืบค้นหนังสือ และสืบค้นหนังสือที่ต้องการ
- หลังจากได้หนังสือที่ต้องการแล้ว ก็ทำการเลือกการแก้ไขหนังสือ
- ทำการเปลี่ยนแปลงข้อมูลตามที่ต้องการ หากเรียบร้อยแล้วก็ยืนยันการทำรายการ

- สิ่งที่ต้องการใช้ในการทดลอง

- บัญชีผู้ใช้งานของบรรณารักษ์
- หมายเลขเรียกของหนังสือที่ต้องการจะทำการแก้ไข
- ข้อมูลที่จะต้องการแก้ไขของหนังสือเล่มนั้น

- ผลที่ได้จากการทดลอง (Experimental results)

ข้อมูลที่ถูกแก้ไขของหนังสือเล่มนั้นที่บรรณารักษ์ได้ทำการแก้ไขไว้แล้ว จะแสดงผลในหน้าของรายละเอียดของหนังสือเล่มนั้น ตรงตามข้อมูลที่บรรณารักษ์นั้นได้ทำการแก้ไขไว้

- สรุปและสิ่งที่คุณได้จากการทดลองนี้ (Conclusion)

บรรณารักษ์สามารถทำแก้ไขข้อมูลของหนังสือที่ต้องการได้

บทสรุป

สิ่งที่ทางผู้พัฒนา Web Application : Library Management System คาดหวังจากผู้เข้ามาใช้งาน Application นี้จะได้รับคือ อำนวยความสะดวกให้แก่สมาชิกที่มาใช้บริการ ไม่ว่าจะเป็นการอำนวยความสะดวกในด้านการตรวจสอบว่าหนังสือเล่มนี้มีอยู่ในห้องสมุดหรือเปล่า แทนการที่สมาชิกนั้นจะต้องเดินทางมาที่ห้องสมุดและสอบถามกับทางบรรณารักษ์หรือค้นหภายในห้องสมุดแทน ถ้าเกิดหนังสือที่สมาชิกนั้นไม่มีอยู่ในห้องสมุดหรือ ก็อาจทำให้สมาชิกนั้นเสียเวลา เสียค่าใช้จ่ายในการเดินทางมาที่ห้องสมุด หรือการอำนวยความสะดวกให้แก่สมาชิกนั้นสามารถจองหนังสือที่ต้องการจะยืมผ่านทางเว็บ เพื่อที่สมาชิกนั้นจะได้ไปรับหนังสือที่ห้องสมุดได้ในวันที่สมาชิกนั้นกำหนดไว้ในการจอง รวมไปถึงการอำนวยความสะดวกให้แก่สมาชิกให้สามารถต่ออายุการยืมให้แก่หนังสือเองได้ (กรณีที่ไม่มีสมาชิกคนอื่นทำการจองไว้) เพื่อที่จะได้ไม่ต้องเดินทางมาที่ห้องสมุดและติดต่อกับบรรณารักษ์ในเรื่องดังกล่าว อีกทั้งการช่วยลดภาระงานด้านต่างๆให้แก่บรรณารักษ์ ไม่ว่าจะเป็นการจัดเก็บรวบรวมข้อมูลหนังสือต่างๆให้มีความเป็นระเบียบเรียบร้อย การจัดเก็บค่าปรับตามการคำนวณของเว็บเองโดยที่ทางบรรณารักษ์ไม่ต้องมาคำนวณค่าปรับเองให้เสียเวลา และรวมไปถึงลดการใช้ทรัพยากรกระดาษของทางห้องสมุดเองที่จะใช้จัดเก็บข้อมูลของหนังสือ สมาชิก ฯลฯ ซึ่งจะทางผู้พัฒนา Web Application : Library Management System หวังว่า Web Application จะทำให้เกิดความพึงพอใจในการใช้งาน และช่วยให้เกิดการกระจายโอกาสให้แก่ห้องสมุดที่ยังขาดแคลนทางงบประมาณอีกด้วย

สิ่งที่ทางผู้พัฒนา Web Application : Library Management System ได้รับจากการทำ Web Application นั้น ประการแรก คือ การได้เรียนรู้การวางแผนในการทำงานเป็นทีม ไม่ว่าจะเป็นการช่วยการคิด Requirement ให้ตรงตามกับระบบตามที่เราจะพัฒนา การช่วยการพัฒนา Web Application ให้ตรงตามแผนงานตามที่ได้วางไว้ ประการที่ 2 คือ การได้เรียนรู้จากการพัฒนา Application ให้ตรงตามหลักการออกแบบเชิงวัตถุ (Object-Oriented) เพื่อให้เกิดการเสริมประสบการณ์จากการที่ได้เรียนรู้ในห้องเรียน ประการที่ 3 คือ การได้ฝึกฝนการพัฒนา Application โดยใช้ภาษา PHP เพื่อทำให้เกิดความหลากหลายในการใช้ใช้ภาษาต่างๆ ที่ไม่ได้เรียนรู้ในห้องเรียน ซึ่งเดิมนั้นผู้พัฒนามีความรู้ไม่มากนักในทางด้านภาษา PHP ก็เลยต้องทำการศึกษาภาษา PHP มาใช้ในการพัฒนา Application ซึ่งทำให้เกิดการเรียนรู้จากตัวเองไปด้วย ประการที่ 4 คือ การได้ฝึกฝนความอดทนในการทำงาน เพราะว่าการพัฒนา Application นี้มีการมีการกำหนดระยะเวลาในการจัดทำประมาณ 8 สัปดาห์ ซึ่งในช่วงระหว่างในการจัดทำนี้ก็มีการเกิดปัญหาต่างๆจากการพัฒนา Application หลายอย่าง ไม่ว่าจะเป็นการไม่ชำนาญในการเขียนภาษา PHP หรือการเกิด Bug บางอย่างโดยไม่ทราบสาเหตุ ซึ่งต้องใช้ความอดทนอย่างมากในการแก้ไข จนทำให้สามารถผ่านไปได้ ประการสุดท้ายคือ การรู้จักแบ่งเวลาในการทำงาน เพราะว่า ไม่ได้มีเวลาที่ต้องทำการพัฒนา Application เพียงอย่างเดียว แต่มีเวลาที่เรียนหนังสือ หรือเวลาต่างๆด้วย ซึ่งต้องมีการแบ่งเวลาให้มีความคุ้มค่าในทุกวินาทีด้วย เพื่อให้เกิดประสิทธิภาพสูงที่สุดให้แก่ตัวเองด้วย

บรรณานุกรม

- Dayle Rees. Laravel: Code Bright. พิมพ์ครั้งที่ 4. : Leanpub, 2014.
- getbootstrap.com
- www.w3schools.com
- laravel.com
- stackoverflow.com
- bootstrapzero.com
- laravel.io/forum