

# โครงงานวิชา 01418211 Software Construction

## ภาคต้น ปีการศึกษา 2563

### อ่านรายละเอียดอย่างระมัดระวัง

ให้นักศึกษาสร้าง Desktop Application สำหรับการบันทึกการรับ/ส่งจดหมาย เอกสาร หรือพัสดุ สำหรับเจ้าหน้าที่ส่วนกลางที่ดูแลห้องพัสดุในคอนโด ด้วย JavaFX (JavaSE 8u251 เท่านั้น) โดยต้องออกแบบและเขียนโปรแกรมที่ใช้หลักการโปรแกรมเชิงวัตถุ รายละเอียดและเกณฑ์การตรวจโปรแกรมมีดังนี้

### รายละเอียดโครงงาน

1. โปรแกรมนี้เป็นโปรแกรมสำหรับการบันทึกการรับ/ส่งจดหมาย เอกสาร หรือพัสดุ สำหรับผู้พัสดุในคอนโด\* แห่งหนึ่ง โดยจดหมายจะมีข้อมูลต่อไปนี้เป็นอย่างน้อย
  - ข้อมูลผู้รับที่เป็นผู้พัสดุ ได้แก่ ชื่อ หมายเลขห้องพัสดุ
  - ข้อมูลผู้ส่งจดหมาย
  - ขนาดของซองจดหมาย
  - ภาพซองจดหมายเอกสารจะมีข้อมูลต่อไปนี้เป็นอย่างน้อย
  - ข้อมูลผู้รับที่เป็นผู้พัสดุ ได้แก่ ชื่อ หมายเลขห้องพัสดุ
  - ข้อมูลผู้ส่งเอกสาร
  - ขนาดของซองเอกสาร
  - ภาพซองเอกสาร
  - ระดับความสำคัญของเอกสาร เช่นด่วนที่สุด ลับพัสดุจะมีข้อมูลต่อไปนี้เป็นอย่างน้อย
  - ข้อมูลผู้รับที่เป็นผู้พัสดุ ได้แก่ ชื่อ หมายเลขห้องพัสดุ
  - ข้อมูลผู้ส่งพัสดุ
  - ชื่อบริการที่ส่งพัสดุ เช่น ไปรษณีย์ไทย เคอร์รี่
  - หมายเลขติดตามการส่งพัสดุ (Tracking Number)
  - ขนาดของบรรจุภัณฑ์
  - ภาพบรรจุภัณฑ์\*คอนโดในโปรแกรมกำหนดให้มีคุณสมบัติดังนี้
  - มีอย่างน้อย 1 อาคาร
  - แต่ละอาคารมีอย่างน้อย 8 ชั้น
  - ประเภทของห้องมีอย่างน้อย 2 ประเภท เช่น 1 ห้องนอน หรือ 2 ห้องนอน
  - จำนวนห้องในแต่ละชั้นมีอย่างน้อย 10 ห้อง

2. จดหมาย เอกสาร หรือพัสดุ ที่ส่งมาที่คอนโด จะถูกเก็บอยู่ที่ส่วนกลาง  
โดยโปรแกรมจะต้องมีหน้าจอหรือส่วนสำหรับการกรอกจดหมาย เอกสาร หรือพัสดุที่เหมาะสม  
โดยผู้รับที่พักภายในห้องที่ระบุเท่านั้นที่จะเห็นว่ามียจดหมาย เอกสาร หรือพัสดุส่งมาหาตนเอง
3. โปรแกรมจะต้องมีหน้าจอหรือส่วนสำหรับการนำจดหมาย เอกสาร หรือพัสดุออกจากส่วนกลาง
4. โปรแกรมจะต้องมีหน้าจอหรือส่วนที่แสดงข้อมูลจดหมาย เอกสาร หรือพัสดุ อย่างเหมาะสม  
โดยเจ้าหน้าที่ส่วนกลางสามารถตรวจสอบวันที่จดหมาย เอกสาร หรือพัสดุเข้ามาส่วนกลาง  
และวันที่ผู้รับมารับจดหมาย เอกสาร หรือพัสดุดูออกไปจากส่วนกลาง
5. ต้องใช้หลักการโปรแกรมเชิงวัตถุในการเขียนโปรแกรม
6. มี User Interface ที่ง่ายต่อการใช้งาน มีการแสดงข้อมูลที่ชัดเจน  
ใช้ขนาดข้อความและขนาดภาพที่เหมาะสม สืบเสาะง่าย อ่านได้ง่าย เข้าใจง่าย

## เกณฑ์การตรวจโครงการ

นิสิตต้องทำโครงการทั้งหมดด้วยตนเอง หากพบหรือสงสัย/ส่อว่า

- (1) นิสิตคัดลอกงานมาจากอินเทอร์เน็ต ไม่ว่าส่วนใดส่วนหนึ่งหรือทั้งหมด

นิสิตจะได้คะแนนโครงการเป็น 0 และได้ระดับคะแนน I (ไม่สมบูรณ์)

- (2) นิสิตคัดลอกงานจากเพื่อน ไม่ว่าส่วนใดส่วนหนึ่งหรือทั้งหมด

ทั้งผู้ลอกและผู้ให้ลอกจะได้คะแนนโครงการเป็น 0 และได้ระดับคะแนน I (ไม่สมบูรณ์)

- (3) หากนิสิตนำโครงชุดคำสั่งของเพื่อนมาปรับใช้ในงานของตน

คะแนนจะถูกหารตามจำนวนนิสิตที่ใช้โครงเดียวกัน หรืออาจถูกพิจารณาว่าคัดลอกงานจากเพื่อน

กำหนดการตรวจโครงการทั้งสิ้น 3 รอบ

รอบที่ 1 โครงการของนิสิตจะต้องสำเร็จ ทดสอบการทำงานของโปรแกรมให้ทำงานในเครื่องอื่นได้  
ตรวจสอบเอกสารประกอบอย่างละเอียดถี่ถ้วน

รอบที่ 2 รอบแก้ไข หลังจากรอบที่ 1 นิสิตจะได้รับความเห็นในการแก้ไขโปรแกรม  
นิสิตมีโอกาสในการส่งงานแก้ไขเพียงครั้งเดียว โดยต้องระบุส่วนที่นิสิตแก้ไขอย่างชัดเจน

รอบที่ 3 รอบแก้ไขครั้งสุดท้าย หากรอบที่ 2 นิสิตยังมีข้อผิดพลาด  
นิสิตจะต้องแก้ไขโครงการให้สมบูรณ์ นิสิตมีโอกาสในการส่งงานแก้ไขเพียงครั้งเดียว  
โดยต้องระบุส่วนที่นิสิตแก้ไขอย่างชัดเจน

ในการตรวจทุกรอบ ผู้ตรวจจะตรวจการทำงานของโปรแกรมผ่าน executable jar file เป็นหลัก  
โดยรันโปรแกรมผ่าน JRE (version JavaSE 8u251) และตรวจสอบการออกแบบโปรแกรมจากชุดคำสั่ง  
โดยทั้ง jar file และ source code จะต้องอยู่ใน bitbucket repository ที่กำหนด

## กำหนดส่งแต่ละรอบ

- **รอบที่ 1 กำหนดส่งภายใน 16.30 น. ของวันอาทิตย์ที่ 1 พฤศจิกายน 2563** โดยจะต้องไม่มี commit ที่เกินเวลาดังกล่าว
- วันสอบไล่ 9 - 20 พฤศจิกายน 2563
- แจ้งความเห็นในการแก้ไขโปรแกรมและอาจารย์ผู้ตรวจ ภายในวันที่ 22 พฤศจิกายน 2563

- วันสุดท้ายส่งคะแนนสอบไล่ วันจันทร์ที่ 30 พฤศจิกายน  
โดยจะวัดผลคะแนนโครงงานจากการส่งรอบที่ 1 และไม่มีการปรับคะแนนเพิ่มจากรอบที่ 2, 3  
นิสิตที่ไม่ผ่านเกณฑ์การตรวจโครงงานในครั้งที่ 1 จะได้รับระดับคะแนน I
- **รอบที่ 2 เฉพาะนิสิตที่ไม่ผ่านเกณฑ์การตรวจโครงงาน กำหนดส่งภายใน 16.30 น. ของวันอังคารที่ 1 ธันวาคม 2563**  
นิสิตจะต้องแจ้งอาจารย์ผู้ตรวจตามช่องทางที่อาจารย์ผู้ตรวจกำหนด  
โดยให้นิสิตแจ้งแก้ไขโครงงานวิชา 01418211/217 พร้อมชื่อ รหัสนิสิต และ Link bitbucket  
เดิมที่แก้ไขแล้ว และระบุส่วนที่นิสิตแก้ไขอย่างชัดเจน
- **รอบที่ 3 เฉพาะนิสิตที่ต้องแก้ไขจากรอบที่ 2 กำหนดส่งภายใน 16.30 น. ของวันอาทิตย์ที่ 6 ธันวาคม 2563** นิสิตจะต้องแจ้งเหมือนรอบที่ 2
- เปิดเรียนภาคปลาย 2563 วันอังคารที่ 8 ธันวาคม
- กำหนดวันสุดท้ายแก้ระดับคะแนน I ของภาคต้น 2563 วันที่ 30 ธันวาคม  
แต่นิสิตจะต้องแก้ไขและส่งภายในเวลาที่กำหนดของแต่ละรอบ  
**อาจารย์ผู้ตรวจจะไม่ตรวจหากส่งพ้นกำหนดดังกล่าว**

\* นิสิตที่ไม่ผ่านเกณฑ์ที่กำหนด และไม่ส่งงานรอบใดก็ตาม (หรือส่งเลยกำหนดทุกกรณี) จะได้เกรด F

รายละเอียดด้านล่าง คือ เกณฑ์การตรวจโครงงาน **โดยนิสิตจะต้องทำข้อที่มีสีแดงให้ได้**  
หากนิสิตไม่สามารถทำข้อที่มีสีแดงได้ นิสิตจะได้รับระดับคะแนน I (ไม่สมบูรณ์) จนกว่านิสิตจะทำได้ ส่วนข้ออื่น  
นิสิตจะได้คะแนนตามผลงานที่นิสิตทำมา หากทำได้ นิสิตจะได้คะแนนในส่วนนั้น  
หากไม่ได้ทำหรือทำแต่ไม่สมบูรณ์ นิสิตจะไม่ได้คะแนนส่วนนั้นหรือได้คะแนนแค่บางส่วน  
**ข้อที่มีสีน้ำเงิน** เป็นข้อพึงระวัง นิสิตอาจไม่ได้รับการตรวจหรือถูกหักคะแนน  
คะแนนที่มีคำว่า extra นำหน้า เป็นคะแนนพิเศษ

คะแนนเต็ม 170 คะแนน (คิดเป็น 25%) คะแนนพิเศษ +34 คะแนน (คิดเป็น +5%)

1. นิสิตจะต้อง commit โครงงานอย่างน้อย 4 สัปดาห์ สัปดาห์ละอย่างน้อย 1 ครั้ง ไปไว้ที่ bitbucket  
(แต่ละครั้งที่ commit ต้องมีปริมาณโค้ดที่ต่างกัน ไม่ใช่ commit ใน 10 นาทีสุดท้ายก่อนส่ง)  
โดยแต่ละครั้งต้องมีอย่างน้อย 1 commit และนิสิตต้องใช้ชื่อ repository เป็นรหัสนิสิต และชื่อ  
workspace id เป็นรหัสนิสิต และกำหนดการเข้าถึงให้อาจารย์ผู้ตรวจแล้ว หากไม่ทำตามนี้  
จะไม่ได้รับการตรวจโครงงาน (ดูขั้นตอนการสร้าง repository ที่หน้าสุดท้าย)  
นิสิตที่มีจำนวนการ commit ไม่ถึง 4 สัปดาห์ จะถูกหักคะแนนตามจำนวนสัปดาห์ที่ไม่ได้ commit  
สัปดาห์ละ 20 คะแนน
2. ใน repository จะต้องมียูทิลิตี้โปรแกรม executable jar file (.jar) และไฟล์ README.md ที่เขียนบรรยาย  
แสดงถึงความตั้งใจเขียน และมีเนื้อหาที่อธิบาย วิธีการติดตั้งหรือรันโปรแกรม การวางโครงสร้างไฟล์  
(บอกว่า folder ใน repository เก็บอะไร) และสรุปสิ่งที่พัฒนาในแต่ละสัปดาห์  
หากผู้ตรวจไม่สามารถหา jar file ได้จาก repository จะไม่ได้รับการตรวจ

3. นิสิตต้องตั้งชื่อไฟล์ และ directory ให้ตรงกับบริบทของโครงงาน  
หากส่งงานด้วยชื่อ package เป็น sample ชื่อไฟล์เป็น sample.fxml, Controller.java, Main.java  
หรือชื่อ Title เป็น Hello World จะไม่ตรวจโครงงานให้
4. (5 คะแนน) ต้องมีไฟล์ jar ตั้งชื่อตามรหัสนิสิต เช่น 61XXXXXXX.jar ที่สามารถคลิกแล้วรันได้เลย  
หรือหากต้องอาศัย command ช่วยในการรัน ต้องระบุ command  
ในส่วนการติดตั้งโปรแกรมของไฟล์ README.md ด้วย
5. \*\*\* ต้องใช้หลักการโปรแกรมเชิงวัตถุพื้นฐาน
  - (10 คะแนน) ออกแบบคลาสให้สามารถสร้าง object ที่มีข้อมูลต่างกันผ่านการส่งค่าให้ parameter ใน constructor ได้ ดังนั้น นิสิตควรสร้าง object ที่เก็บข้อมูลต่างกันโดยใช้คลาสเดียวกันได้
  - (5 คะแนน) สร้าง attributes ได้ถูกต้อง ไม่ใช่ local variable มาเป็น attributes
6. ต้องใช้หลักการโปรแกรมเชิงวัตถุขั้นสูง รวมถึง การใช้ inheritance, interface, polymorphism ด้วย  
โดยนิสิตต้องออกแบบให้เหมาะสม  
เช่น ควรพิจารณาสถานการณ์ที่เหมาะสมในการใช้ inheritance ไม่ใช่สร้าง superclass  
แต่ไม่มีการใช้ประโยชน์จาก method ของ superclass หากนิสิตทำแบบนี้  
ก็จะไม่ได้คะแนนในส่วนการใช้ inheritance เป็นต้น นิสิตควรออกแบบให้ดังนี้
  - (5 คะแนน) มี superclass/subclass ที่เหมาะสม หมายความว่า ควรตรงกับคุณสมบัติ IS-A  
เช่น Car เป็น subclass ของ Vehicle นั้นเหมาะสม  
แต่ Book เป็น subclass ของ Library นั้นไม่เหมาะสม
  - อาจมีหรือไม่มี interface ก็ได้ ให้นิสิตพิจารณาตามความเหมาะสม
  - (10 คะแนน) superclass และ subclass ต้องมีการทำงานบางอย่างที่เหมือนกัน  
และบางอย่างที่แตกต่างกัน โดย  
subclass ควรระบุ method ทั้งหมดที่ subclass ใช้ร่วมกัน (เพื่อให้เกิด polymorphism ได้)  
และ subclass ควร override บาง method ของ superclass ให้ตรงกับความต้องการของ subclass  
ดังนั้น นิสิตควรออกแบบให้ดีเพื่อให้ตรงกับคุณสมบัติดังกล่าว  
หาก subclass ต่างจาก superclass แค่ constructor แสดงว่ายังออกแบบไม่ดี  
หรือหาก superclass ไม่มี method ใดเลย แต่ subclass มี method ที่แตกต่างกันไปหมด  
แสดงว่ายังออกแบบไม่ดี
  - (10 คะแนน) เขียนโค้ดให้มี polymorphism ที่เหมาะสม โดยควรออกแบบคลาสที่ใช้ object  
ของ super type/sub type ให้สามารถใช้ object ของ sub type/super type เหล่านี้แทนได้
7. (5 คะแนน) ระบุ access modifier ที่เหมาะสม โค้ดอยู่ใน package ที่เหมาะสม และไม่ใช่ static
8. (5 คะแนน) ควรตั้งชื่อ class, method, ตัวแปร ให้สื่อถึงบริบทหรือการใช้งาน  
ถูกต้องตามหลักการตั้งชื่อ
9. (5 คะแนน) ต้องสร้าง UML class diagram ที่แสดงถึงคลาสในโปรแกรม โดยเก็บเป็นไฟล์รูป เช่น  
.png, .jpg แล้ว push ขึ้น bitbucket ด้วย ระบุชื่อไฟล์เป็น รหัสนิสิต-UML.png หรือ รหัสนิสิต-  
UML.jpg เช่น 61XXXXXXX-UML.png
10. (5 คะแนน) ต้องมีไฟล์ pdf ตั้งชื่อไฟล์ตามรหัสนิสิต เช่น 61XXXXXXX.pdf โดยมีหัวข้อดังต่อไปนี้

- "วิธีการใช้งาน<ชื่อ application ของนิสิต>"  
เพื่ออธิบายวิธีการใช้งานโปรแกรมของนิสิต ซึ่งต้องมีการอธิบายอย่างเป็นขั้นตอน ใช้ภาษาที่เข้าใจง่าย และมีการแสดงรูปภาพประกอบ (นำเนื้อหาจาก README.md มาประกอบได้)
- "ตัวอย่างข้อมูลผู้ใช้ระบบ"  
เพื่อระบุตัวอย่างของ username และ password ในการเข้าใช้ระบบ ของผู้ใช้แต่ละ role
- "รายละเอียดของไฟล์ CSV"  
เพื่ออธิบายข้อมูลในไฟล์ .csv ว่า ไฟล์จะถูกสร้างไว้ที่ใด ข้อมูลที่เก็บในแต่ละ column/row ของไฟล์หมายถึงอะไร และ โปรแกรมของนิสิตอ่านข้อมูลจากไฟล์นี้เมื่อใด อ่านอย่างไร
- "Extra Features"  
หากนิสิตทำข้อ extra มาด้วย ให้ระบุด้วยว่าทำข้อใดบ้าง อยู่ส่วนใดของโปรแกรม หากผู้ตรวจอ่านวิธีการใช้งานที่นิสิตเขียนแล้วไม่เข้าใจได้โดยง่าย จะหักคะแนนอย่างน้อย 30 คะแนน

## Features ของ Application มีดังนี้

11. (5 คะแนน) ต้องมีส่วนของ GUI ที่แสดงถึงข้อมูลต่อไปนี้ โดยอาจเข้าถึงข้อมูลนี้ผ่านเมนู หรือมีปุ่มกดที่หน้าแรก
  - ข้อมูลนิสิตผู้จัดทำโปรแกรม ได้แก่ รูปที่แสดงหน้าตา (หน้ายิ้ม) ที่ชัดเจน ชื่อ นามสกุล ชื่อเล่น รหัสนิสิต
  - ข้อมูลคำสั่ง หรือคำแนะนำ ในการใช้โปรแกรมที่นิสิตสร้างขึ้น
12. ระบบจัดการข้อมูลเจ้าหน้าที่ส่วนกลาง ระบบนี้ถูกจำกัดสิทธิ์สำหรับผู้ดูแลระบบเท่านั้น ผู้ใช้ที่ไม่ใช่ผู้ดูแลระบบต้องไม่สามารถเข้าใช้งานส่วนนี้ได้
  - \* Application ไม่ต้องมีระบบการสร้างบัญชีสำหรับผู้ดูแลระบบ ให้รหัสระบุ username และ password ของผู้ดูแลระบบในไฟล์ pdf (ข้อ 10 หัวข้อ "ตัวอย่างข้อมูลผู้ใช้ระบบ")
- 12.1 (5 คะแนน) มีหน้าสำหรับการกรอกข้อมูลเพื่อสร้างบัญชีผู้ใช้ให้เจ้าหน้าที่ส่วนกลาง และมีการแจ้งข้อความที่เป็นประโยชน์ต่อผู้ใช้หากข้อมูลที่กรอกนั้นไม่ถูกต้อง ชื่อบัญชีผู้ใช้จะต้องไม่ซ้ำกัน
- 12.2 (5 คะแนน) ผู้ดูแลระบบสามารถเปลี่ยนรหัสผ่านของตนเองได้ และรหัสผ่านใหม่ต้องใช้ได้
- 12.3 (5 คะแนน) มีหน้าแสดงรายชื่อของเจ้าหน้าที่ส่วนกลาง โดยต้องแสดง username, ชื่อของเจ้าหน้าที่ และวันเวลาที่เข้าใช้ล่าสุดของเจ้าหน้าที่คนนั้น เรียงลำดับตามวันเวลาที่เจ้าหน้าที่เข้าใช้งานล่าสุดก่อน
- 12.4 (extra 2 คะแนน) แสดงภาพของเจ้าหน้าที่ในการแสดงรายชื่อ
- 12.5 (extra 4 คะแนน) ผู้ดูแลระบบสามารถระงับสิทธิ์การเข้าใช้งานของเจ้าหน้าที่ได้ผ่านโปรแกรม เจ้าหน้าที่ที่ถูกระงับสิทธิ์จะไม่สามารถเข้าใช้งานโปรแกรมได้ โดยมีข้อความแจ้งเตือนการถูกระงับสิทธิ์ และผู้ดูแลระบบสามารถทราบถึงจำนวนครั้งในการพยายามเข้าใช้โปรแกรมในขณะที่เจ้าหน้าที่ถูกระงับสิทธิ์ และผู้ดูแลระบบสามารถคืนสิทธิ์การเข้าใช้งานของเจ้าหน้าที่ได้
13. ระบบจัดการข้อมูลผู้พักอาศัย ระบบนี้ถูกจำกัดสิทธิ์สำหรับเจ้าหน้าที่ส่วนกลางเท่านั้น ผู้ใช้ที่ไม่ใช่เจ้าหน้าที่ส่วนกลางต้องไม่สามารถเข้าใช้งานส่วนนี้ได้

- 13.1 (5 คะแนน) มีหน้าสำหรับการแสดงรายการหมายเลขห้องพักในคอนโด  
โดยจะต้องทราบได้ว่าห้องพักนั้นเป็นห้องพักประเภทใด อยู่ในอาคารอะไร อยู่ี่ชั้นใด  
มีใครเป็นผู้พักอาศัยหรือไม่
- 13.2 (5 คะแนน) มีหน้าสำหรับการกรอกข้อมูลห้องพัก  
โดยต้องมีการตรวจสอบรูปแบบการกำหนดหมายเลขห้องพักให้สอดคล้องกับหมายเลขอาคารแ  
ละหมายเลขชั้น และมีการแจ้งข้อความที่เป็นประโยชน์ต่อผู้ใช้หากข้อมูลที่กรอกนั้นไม่ถูกต้อง  
หมายเลขห้องพักในอาคารเดียวกัน จะต้องไม่ซ้ำกัน
- 13.3 (5 คะแนน) มีหน้าสำหรับการกรอกข้อมูลเพื่อสร้างข้อมูลผู้เข้าพักอาศัย  
และมีการแจ้งข้อความที่เป็นประโยชน์ต่อผู้ใช้หากข้อมูลที่กรอกนั้นไม่ถูกต้อง  
ห้องพักแต่ละห้องจะมีจำนวนผู้เข้าพักสูงสุดขึ้นกับประเภทของห้องพัก
- 13.4 (extra 2 คะแนน) มีส่วนสำหรับการค้นหารายการห้องพักจากส่วนหนึ่งของชื่อของผู้เข้าพัก  
เพื่อไปแก้ไขข้อมูลผู้เข้าพัก
- 13.5 (extra 2 คะแนน) มีส่วนสำหรับการนำข้อมูลผู้เข้าพักออกจากห้องพัก  
โดยต้องมีการยืนยันก่อนที่จะนำออกจริง
- 13.6 (5 คะแนน) เจ้าหน้าที่ส่วนกลางสามารถเปลี่ยนรหัสผ่านของตนเองได้  
และรหัสผ่านใหม่ต้องใช้ได้
14. ระบบจัดการจดหมาย/เอกสาร/พัสดุ ระบบนี้ถูกจำกัดสิทธิ์สำหรับเจ้าหน้าที่ส่วนกลางเท่านั้น  
ผู้ใช้ที่ไม่ใช่เจ้าหน้าที่ส่วนกลางต้องไม่สามารถเข้าใช้งานส่วนนี้ได้
- 14.1 (10 คะแนน) มีหน้าสำหรับแสดงรายการจดหมาย/เอกสาร/พัสดุ ที่ส่วนกลางได้รับ  
ซึ่งประกอบด้วยข้อมูลของจดหมาย/เอกสาร/พัสดุ (ดูรายละเอียดโครงงานข้อ 1)  
ภาพจดหมาย/เอกสาร/พัสดุ และบอกวันเวลาที่ที่ส่วนกลางรับไว้  
เรียงลำดับตามวันเวลาที่รับล่าสุดก่อน
- 14.2 (5 คะแนน) สามารถเลือกเรียงลำดับการแสดงรายการจดหมาย/เอกสาร/พัสดุ ดังนี้
- เรียงลำดับตามวันเวลาที่ส่วนกลางรับไว้ล่าสุดก่อน หรือแรกสุดก่อน
  - เรียงลำดับตามหมายเลขห้องจากน้อยไปมาก หรือจากมากไปน้อย
- 14.3 (5 คะแนน) มีส่วนค้นหาจดหมาย/เอกสาร/พัสดุ จากหมายเลขห้องของผู้เข้าพัก
- 14.4 (5 คะแนน) เปลี่ยนสถานะเมื่อมีผู้เข้าพักมารับจดหมาย/พัสดุ/เอกสารแล้ว  
โดยไม่สามารถเปลี่ยนสถานะกลับเป็นยังไม่รับ  
จดหมาย/พัสดุ/เอกสารที่มีผู้เข้าพักมารับไปแล้วจะถูกนำออกจากรายการในข้อ 14.1, 14.2
- 14.5 (5 คะแนน) มีหน้าแสดงแบบฟอร์มเพื่อกรอกข้อมูลที่จดหมาย/เอกสาร/พัสดุที่ส่วนกลางได้รับ  
และมีการแจ้งข้อความที่เป็นประโยชน์ต่อผู้ใช้หากข้อมูลที่กรอกนั้นไม่ถูกต้อง  
เมื่อกรอกข้อมูลถูกต้อง จะต้องไปหน้าแสดงรายการในข้อ 14.1
- 14.6 (10 คะแนน) ระบบจะต้องบันทึกว่าเจ้าหน้าที่คนใด เป็นผู้รับจดหมาย/เอกสาร/พัสดุ  
และผู้เข้าพักคนใดมารับจดหมาย/เอกสาร/พัสดุจากเจ้าหน้าที่คนใด เมื่อวันและเวลาใด
- 14.7 (extra 2 คะแนน) ภาพจดหมาย/เอกสาร/พัสดุที่บันทึกและแสดงผลได้ถูกต้อง  
ถือเป็นคะแนนพิเศษ
- 14.8 (extra 4 คะแนน) มีส่วนที่แสดงจดหมาย/เอกสาร/พัสดุที่ผู้เข้าพักรับไปแล้ว  
และแสดงข้อมูลในข้อ 14.6 ประกอบด้วย



15. โปรแกรมต้องสามารถบันทึกสถานะของข้อมูลต่าง ๆ ในรูปแบบของไฟล์ csv (comma-separated values) และสามารถโหลดไฟล์ที่บันทึกไว้นั้นมาแสดงผลได้อย่างถูกต้อง
  - 15.1 ต้องมีการบันทึกสถานะและข้อมูลที่จำเป็นในรูปแบบไฟล์ csv
  - 15.2 (5 คะแนน) เมื่อเปิดโปรแกรมใหม่ หรือเมื่อโปรแกรมโหลดไฟล์ csv จะต้องแสดงข้อมูลที่ได้นั้นไว้อย่างถูกต้อง
16. (extra) การสร้างบัญชีผู้ใช้สำหรับผู้เข้าพัก
  - 16.1 (extra 4 คะแนน) มีระบบ register สำหรับผู้เข้าพักในคอนโด โดยผู้เข้าพักจะต้องกรอกข้อมูลที่ตรงกับข้อมูลในระบบเท่านั้นจึงจะสร้างเป็นบัญชีผู้ใช้ได้ และมีการแจ้งข้อความที่เป็นประโยชน์ต่อผู้ใช้หากข้อมูลที่กรอกนั้นไม่ถูกต้อง
  - 16.2 (extra 2 คะแนน) ผู้เข้าพักอาศัยสามารถเปลี่ยนรหัสผ่านของตนเองได้ และรหัสผ่านใหม่ต้องใช้ได้
  - 16.3 (extra 4 คะแนน) ผู้เข้าพักอาศัยสามารถดูรายการจดหมาย/เอกสาร/พัสดุ ทั้งที่อยู่ในส่วนกลาง หรือรับมาแล้ว มีเจ้าหน้าที่คนใดดูแลในชั้นตอนใดบ้าง และแสดงวันเวลาที่เกี่ยวข้อง
17. ความสวยงามและง่ายต่อการใช้
  - 17.1 (10 คะแนน) ใช้โทนสีและองค์ประกอบต่าง ๆ ของ GUI ที่ดูได้ชัดเจน นำใช้ ไม่แสบตา ไม่หลงตา โดยลองนึกถึงว่าหากเราทำโปรแกรมนี้นี้ให้คนทั่วไปใช้จริง ควรแสดงองค์ประกอบเหล่านี้อย่างไร
  - 17.2 โปรดตรวจสอบความถูกต้องของข้อความที่ปรากฏในส่วนต่าง ๆ ของโปรแกรม หรือเอกสารประกอบ หากพบว่ามีคำผิด **หักคำที่ผิดจุดละ 5 คะแนน**
  - 17.3 (10 คะแนน) Graphic User Interface (GUI) ควรมีรูปแบบที่เข้าใจง่าย มีการแสดงข้อมูลที่ชัดเจน มี Label กำกับข้อมูลที่ชัดเจน เพื่อให้เข้าใจว่าค่าที่แสดงนั้นคืออะไร มีการแสดงองค์ประกอบต่าง ๆ ในขนาดที่เหมาะสม ไม่เล็กหรือไม่ใหญ่จนเกินไป **ขนาดหน้าจอของโปรแกรมไม่ควรมีความสูงเกิน 1024 pixel**
  - 17.4 (extra 4 คะแนน) GUI มี effect ที่น่าสนใจ เมื่อมี action ต่าง ๆ เช่น มีการ highlight button เมื่อคลิกปุ่ม มี effect การ transition เมื่อเปลี่ยน scene เป็นต้น
18. (extra 4 คะแนน) ผู้ใช้โปรแกรมสามารถปรับเปลี่ยน Theme ของ Application ได้ เช่น เปลี่ยนโทนสี เปลี่ยนขนาดอักษร หรือเปลี่ยนฟอนต์ของตัวอักษรได้ โดยจะต้องเปลี่ยนให้สอดคล้องกันทุก Scene
19. งานทุกอย่างที่ต้องการให้ตรวจเพื่อให้คะแนน นิสิตจะต้อง push ไปที่ bitbucket รวมถึง jar file และ pdf file ด้วย โดยให้นิสิตสร้าง folder แยกจาก src เพื่อเก็บ jar file และ pdf file ให้ชัดเจน พร้อมอธิบายใน README.md ว่านิสิตจัดการโครงสร้างไฟล์อย่างไร

#### ตารางสรุปหน้าที่ของผู้ใช้ระบบ

ระดับของผู้ใช้ระบบ	หน้าที่
ผู้ดูแลระบบ	ข้อ 12
เจ้าหน้าที่ส่วนกลาง	ข้อ 13, 14
ผู้เข้าพัก	ข้อ 16

## ขั้นตอนการสร้าง repository ใน bitbucket สำหรับส่งโครงการ

1. สร้าง user account ที่เว็บ bitbucket.org (แนะนำให้ผูก account กับ mail @ku.th) หรือ login หากมี account อยู่แล้ว
2. สร้าง repository โดยตั้งชื่อตามรหัสสิตของตนเอง (เลขสิบหลัก ไม่มีตัวอักษรนำหน้า) หากมีอยู่แล้วให้ลบอันเก่าก่อน  
กำหนด access level เป็น private repository  
กำหนด Include a README? เป็น No  
กำหนด Version Control System เป็น Git
3. กำหนด workspace
  - 3.1 คลิกที่ Your profile and settings (มุมซ้ายล่างที่เป็นรูปวงกลมมีตัวย่อชื่อและนามสกุลของตนเอง)
  - 3.2 เลือก Bitbucket Settings
  - 3.3 ในส่วนของ GENERAL > Account settings ให้ดูหัวข้อ Bitbucket profile settings
  - 3.4 กำหนด Workspace ID เป็น รหัสสิตของตนเอง (เลขสิบหลัก ไม่มีตัวอักษรนำหน้า)
  - 3.5 ตรวจสอบ git repository ของนิสิต จะต้องมีรูปแบบเป็น รหัสสิต/รหัสสิต

### 4. เพิ่ม User สำหรับการตรวจ

- 4.1 ไปที่ repository ที่ชื่อเป็นรหัสสิตของตนเอง
- 4.2 เลือก Settings
- 4.3 ที่ส่วนของ GENERAL เลือก User and group access
- 4.4 ที่หัวข้อ Users ให้เพิ่ม java.cs.ku@gmail.com แล้วรอการยืนยัน
- 4.5\* นิสิตต้องได้รับการยืนยันการเพิ่ม User สำหรับการตรวจ ภายในคาบเรียน หากพ้นจากคาบเรียนหรือนิสิตสร้าง repository ใหม่ ให้ติดต่ออ.แซ็ก เพื่อยืนยันการเพิ่ม
- 4.6 หากลังการยืนยันการเพิ่ม user นั้นหมดอายุ นิสิตต้องเพิ่มใหม่และแจ้งใหม่

### 5. ทดสอบ repository

- 5.1 สร้าง java project ในเครื่อง แล้วทำให้เป็น git local repository
- 5.2 add remote repository ไปที่ bitbucket repository ที่ชื่อเป็นรหัสสิตตนเอง
- 5.3 create README.md file ใส่ชื่อและนามสกุลของตนเอง
- 5.4 add commit and push to repository

### 6. ทดสอบการ clone repository

\*\*\* ตลอดการทำงาน การส่งงาน การแก้ไขงาน ต้องทำงานกับ repository เดิมเท่านั้น หากมีการลบ repository แล้วสร้างใหม่ในระหว่างการตรวจ แล้วผู้ตรวจ pull งานของนิสิตไม่ได้ ซึ่งถือว่านิสิตไม่ได้ส่งงานแก้ไข และถูกพิจารณาให้ F