



1. ระบบสารสนเทศนักศึกษา(Aegis) ประกอบไปด้วย บัญชีนักศึกษา(StudentMembers), รายวิชา(courseList), กิจกรรมของนักศึกษา(activityList), เลขประจำตัวนักศึกษา(Id), บัญชีของผู้จัดการฐานข้อมูล(DatabaseManagement), ตัวนับจำนวนสมาชิก(countMember), ฐานข้อมูลบัญชี(acddb), ฐานข้อมูลกิจกรรม(actdb) ฐานข้อมูลรายวิชา(cdb), ตัวรับข้อมูลกับทางแป้นพิมพ์(scan)

1.1 ส่วนกลาง(static void main)เป็นหน้าแรกของระบบซึ่งจะให้ผู้ใช้บริการเลือกใช้บริการต่างๆที่มีในระบบสารสนเทศนักศึกษา ซึ่งผู้ใช้บริการจะต้องป้อนตัวเลขและข้อมูลบางส่วนเพื่อโต้ตอบกับระบบ และระบบจะทำการส่งข้อมูลที่ท่านได้เลือกออกมา ซึ่งจะมีเมนูขึ้นมา 4 ตัวเลือกด้วยกัน มีดังนี้

1.1.1 กด 1 : ผู้ดูแลระบบเข้ารหัส (Department Login) ใส่รหัสประจำตัว(id) และรหัสผ่าน(password) และจะเข้าไปใช้งานระบบให้บริการของผู้ดูแล ซึ่งมี 11 ตัวเลือก

1.1.1.1 กด 1 : ตั้งจำนวนผู้ใช้บริการ (Set MAX Student) เรียกใช้ method ตั้งจำนวนของนักศึกษาที่สามารถมีได้(setMaxMember) โดยการพิมพ์ใส่จำนวนลงไป

1.1.1.2 กด 2 : เพิ่มรายวิชาลงไปให้หน. เลือกลงทะเบียน(Add Courses) เรียกใช้ฐานข้อมูลรายวิชา(cdb) ให้เพิ่มรายวิชาลงไป โดยระบบจะให้กรอกชื่อวิชา(subject), รหัสรายวิชา(courseCode), หน่วยกิต(credits)

1.1.1.3 กด 3 : เพิ่มกิจกรรมหน.(Add Activity) เรียกใช้ฐานข้อมูลกิจกรรม(actdb) ให้เพิ่มกิจกรรมลงไป โดยระบบจะให้กรอกช.กิจกรรม(hour), ชื่อกิจกรรม(nameOfActivity) , รหัสกิจกรรม(codeAct)

1.1.1.4 กด 4 : เพิ่มสมาชิกหน.(Add Member) เรียกใช้ method เพิ่มผู้ใช้บริการหน(addMember) โดยระบบจะให้กรอกรหัสผู้ใช้งาน(id), รหัสผ่าน(password), ชื่อ(firstname) และนามสกุล(lastname)

1.1.1.5 กด 5 : เพิ่มชั่วโมงกิจกรรมหน. (Add Activity Hour) โดยระบบจะให้กรอกชื่อกิจกรรม(nameOfActivity) , รหัสกิจกรรม(codeAct), ชม.กิจกรรม(hour) และรหัสของหน.ที่ต้องการจะเพิ่มชั่วโมงกิจกรรมให้(id)

1.1.1.6 กด 6 : ลบรายวิชาที่ให้หน.เลือกลงทะเบียนออก(Remove Course) เรียกใช้ฐานข้อมูลรายวิชา(cdb) ให้ลบรายวิชาลงไปโดยระบบจะให้กรอกชื่อวิชา(subject), รหัสรายวิชา(courseCode), หน่วยกิต(credits)

1.1.1.7 กด 7 : ลบกิจกรรมหน.(Remove Activity) เรียกใช้ฐานข้อมูลกิจกรรม(actdb) ให้ลบกิจกรรมลงไป โดยระบบจะให้กรอกชื่อกิจกรรม(nameOfActivity) , รหัสกิจกรรม(codeAct), ชม.กิจกรรม(hour)

1.1.1.8 กด 8 : เปลี่ยนแปลงสถานะการเงินของหน.(Change PaymentStatus of student) โดยระบบจะให้กรอกรหัสหน.(id) และสถานะที่ต้องการ(PaymentStatus)

1.1.1.9 กด 9 : แสดงรายวิชาที่มี(Check Courses list) ระบบเรียกใช้เรียกใช้ฐานข้อมูลรายวิชา(cdb) ให้แสดงข้อมูลในฐานข้อมูลออกมาทั้งหมด

1.1.1.10 กด 10 : แสดงรายการกิจกรรมหน.ที่มี(Check Activity list) ระบบเรียกใช้เรียกใช้ฐานข้อมูลกิจกรรม(actdb) ให้แสดงข้อมูลในฐานข้อมูลออกมาทั้งหมด

1.1.1.11 กด 0 : ออกจากระบบผู้ควบคุมระบบไปหน้าแรก(Exit)

1.1.2 กด 2 : สร้างรหัสผู้ควบคุมดูแลระบบ(Create Department Account Menu) ระบบจะ set รหัสผู้ควบคุมระบบโดยเรียกใช้ method ตั้งบัญชีของผู้จัดการฐานข้อมูล(setDatabaseManagement) และระบบจะ

เรียกใช้ฐานข้อมูลบัญชี(acddb) เพื่อใช้ method insert และ delete เพื่อเพิ่มบัญชีและลบรหัสผู้ใช้งานเก่าที่ซ้ำกับของเก่าลงในฐานข้อมูล โดยระบบจะให้กรอกรหัสผู้ใช้งาน(id), รหัสผ่าน(password), ชื่อ(firstname) และนามสกุล(lastname)

1.1.3 กด 3 : หน้าเข้าสู่ระบบของนศ.(Student Login) ใส่รหัสประจำตัว(id) และรหัสผ่าน(password) และจะเข้าไปใช้งานระบบให้บริการของนศ. ซึ่งมี 12 ตัวเลือก ดังนี้

1.1.3.1 กด 1 : ตรวจสอบค่าลงทะเบียนเรียนทั้งหมดว่ามีค่าเท่าไร(Check total cost to pay)

1.1.3.2 กด 2 : แสดงรายวิชาที่นศ.คนนั้นๆได้ลงทะเบียนไป(My courses taken list) รับ

เรียกใช้ method

1.1.3.3 กด 3 : จ่ายเงินค่าลงทะเบียน(Pay) ระบบให้เรากดรอกจำนวนเงินที่จะจ่ายแล้วเมื่อดูแล้วถูกต้องจะส่งจำนวนที่เรากรอกเรียกใช้ method จ่ายเงิน(pay)

1.1.3.4 กด 4 : แสดงรายการกิจกรรมนศ.ที่มี(Check Activity list) ระบบเรียกใช้เรียกใช้ฐานข้อมูลกิจกรรม(actdb) ให้แสดงข้อมูลในฐานข้อมูลออกมาทั้งหมด

1.1.3.5 กด 5 : แสดงรายการกิจกรรมนศ.ที่มี(Check Activity list) ระบบเรียกใช้เรียกใช้ฐานข้อมูลกิจกรรม(actdb) ให้แสดงข้อมูลในฐานข้อมูลออกมาทั้งหมด

1.1.3.6 กด 6 : ลงทะเบียนเรียน(Register Course) ระบบแสดงรายวิชาออกมาทั้งหมดแล้วในนศ. กรอกรหัสรายวิชา(courseCode) และ รหัสนศ.(id) จากนั้นเรียกใช้ method ลงทะเบียนเรียน(registerCourses)

1.1.3.7 กด 7 : เปลี่ยนรหัสผ่านนศ.(Reset password) ระบบให้กรอกเลขประจำตัวนศ(id), รหัสผ่านเก่า(oldPassword) และรหัสผ่านใหม่(newPassword) เพื่อไปเรียกใช้ method เปลี่ยนรหัสผ่านนศ.(changePassword)

1.1.3.8 กด 8 : ตรวจสอบจำนวนวิชาที่จำเป็นต้องลงทะเบียนอีก(Check necessary total course to register) ระบบแสดง out put ออกมาให้ทราบ

1.1.3.9 กด 9 : ตรวจสอบชั่วโมงกิจกรรมของนศ.(Check my Activity hour) ระบบแสดง out put ว่ามีชั่วโมงกิจกรรมเท่าใดแล้ว

1.1.3.10 กด 10 : ข้อมูลส่วนตัวของนศ.(My profile) ระบบแสดงหน้าต่างข้อมูลของนศ. ทั้งหมด

1.1.3.11 กด 11 : ปริ้นใบเสร็จ(Payment Bill) ไปเรียกใช้ method ปริ้นใบเสร็จ(printBill)

1.1.3.12 กด 0 : ออกจากระบบผู้ควบคุมระบบไปหน้าแรก(Exit)

1.2 ระบบสารสนเทศนักศึกษา(Aegis)เป็น method เปล่า

1.3 นักศึกษาเข้าสู่ระบบ(Login) ผู้ให้บริการป้อนเลขประจำตัว(Id) และรหัสผ่าน(password) เพื่อเข้าสู่ระบบและทำการใช้บริการอื่นๆต่อไป

1.4 ลงทะเบียนเรียน(registerCourses) ให้นักศึกษาเลือกรายวิชาที่จะลงทะเบียนเรียนแล้วนำไปคำนวณหาค่าลงทะเบียนเรียนออกมา แล้วเปลี่ยนสถานะ(paymentStatus) ของนศ.เป็น PENDING

1.5 แสดงรายวิชาที่นศ.ได้ลงทะเบียนไปแล้ว(getMyCourseTakenList) โดยจะใช้บัญชีนักศึกษา(StudentAccount) ไปวิ่งหาวิชาที่บัญชีนี้ได้ลงทะเบียนไปทั้งหมด

1.6 แสดงรายวิชา(getCourses) โดยจะแสดงรายวิชา(Courses) ที่มีให้เลือกทั้งหมด

1.7 แสดงกิจกรรม(getActivity) โดยจะแสดงกิจกรรม(Activity) ทั้งหมดที่มีในตอนนั้น

1.8 เพิ่มชั่วโมงกิจกรรม(addActivityHour) โดยบัญชีของผู้จัดการฐานข้อมูล(DatabaseManagement) จะเพิ่มชั่วโมงกิจกรรมให้กับบัญชีนศ.(StudentMembers) นั่นๆที่ได้ร่วมกิจกรรม

1.9 จ่ายเงิน(pay) นักศึกษาจ่ายเงิน(money)ตามจำนวนราคาค่าลงทะเบียนเรียนและถ้าจ่ายถูกต้องจะลบหนี้(deptOfRegisterCourses)ออกและเปลี่ยน(paymentStatus)เป็นจ่ายแล้ว(PAID)

ลงทะเบียนเรียน

1.10 เปลี่ยนสถานะการเงินของนศ.(changePaymentStatus) ซึ่งผู้ที่เปลี่ยนสถานะการเงิน (PaymentStatus) ได้จำเป็นต้องมีบัญชีของผู้จัดการฐานข้อมูล(DatabaseManagement) ใช้ในกรณีทีนศ. จ่ายเงิน ไม่ทันตามที่กำหนดและกรณีอื่นๆ

1.11 เปลี่ยนรหัสผ่านนศ.(changePassword)

1.12 เพิ่มผู้ใช้บริการนศ.(addMember) นำผู้ใช้บริการเข้าไปในฐานข้อมูลบัญชี(accdb) และ บัญชีนักศึกษา(StudentMembers)ลงใน array

1.13 แสดงบัญชีนักศึกษา(getStudentAccount) ค้นหาและ return บัญชีนักศึกษา(StudentAccount) จากการเทียบเลขประจำตัว(Id)

1.14 เรียกบัญชีของผู้จัดการฐานข้อมูล(getDatabaseManagement)

1.15 ตั้งบัญชีของผู้จัดการฐานข้อมูล(setDatabaseManagement)

1.16 ตั้งจำนวนของนักศึกษาที่สามารถมีได้(setMaxMember) โดยจะกำหนดจำนวนได้จากการใช้ array ซึ่งผู้ที่ทำได้จำเป็นต้องมีบัญชีของผู้จัดการฐานข้อมูล(DatabaseManagement)

1.17 แสดงรายวิชาทีนศ.คนนั้นๆได้ลงทะเบียนไปแล้วก่อนหน้านี้(getMyCourseTakenList) โดยจะแสดงออกมาทั้งหมดและมีข้อมูลโดยละเอียดของรายวิชานั้น

1.18 ปรีนใบเสร็จ(printBill) เมื่อใช้แล้วเราจะได้ file.txt และ แสดงข้อมูลใบเสร็จตามที่มิใน txt.file ออกมา

2. บัญชีผู้ใช้(Account) ประกอบไปด้วย รหัสผู้ใช้งาน(id), รหัสผ่าน(password), ตำแหน่ง(position), ผู้ใช้งาน(person) ในคลาสนี้สามารถเรียกข้อมูล รหัสผู้ใช้งาน(getId), รหัสผ่าน(getPassword), ตำแหน่ง(getPosition), ผู้ใช้งาน(getPerson),ตั้งรหัสผ่าน (setPassword), ยังสามารถเช็คการกรอกของนักศึกษาได้ว่าข้อมูลตรงกันมั้ย (equals) และแสดงข้อมูลของบัญชีผู้ใช้ (toString) เอาไว้แสดงข้อมูลออกมาในรูปแบบที่สวยงาม

3. ประเภทของผู้ใช้(Postion) ได้แก่ ผู้บริหารจัดการระบบ(DEPARTMENT), นักศึกษา(STUDENT)

4. ผู้ใช้บริการ(Person) ประกอบไปด้วย รหัสผู้ใช้งาน(id), ชื่อ(firstname) และนามสกุล(lastname) โดยในคลาสนี้ สามารถทำการตั้ง(set) และทำการเรียกดู(get) รหัสผู้ใช้งาน(id), ชื่อ(firstname) และนามสกุล(lastname) และแสดงข้อมูลของผู้ใช้บริการ (toString) เอาไว้แสดงข้อมูลออกมาในรูปแบบที่สวยงาม

5. บัญชีของนักศึกษา(StudentAccount) ประกอบด้วย การลงทะเบียนเรียน (RegisteredCourses,registerCoursesList), จำนวนรายวิชาที่ลงทะเบียน(CourseCount),จำนวนรายกิจกรรม (countActivity), รายชื่อกิจกรรม(Activity), พิมพ์ใบเสร็จชำระเงิน(PaymentBill), สถานะการชำระเงิน (PaymentStatus), ชั่วโมงกิจกรรมของตนเอง(MyActivityHour), จำนวนเงินจากการลงทะเบียนที่ต้องจ่าย (deptOfRegisterCourses), โดยทั้งหมดนี้เป็นส่วนสนับสนุนในการดำเนินงาน ดังนี้

5.1 แสดงรายวิชาที่ลงทะเบียน (getMyRegisteredCourses,getRegisterCoursesList) เรียกดูรายวิชา ทีนศ.ได้ลงทะเบียนไป

5.2 ตรวจสอบจำนวนรายวิชาที่ลงทะเบียน (checkCountOfCourseForRegister) แสดงจำนวนรายวิชาที่
นศ.ได้ลงทะเบียนไป

5.3 เรียกดูใบเสร็จชำระเงิน (getMyPaymentBill) แสดงใบเสร็จที่นศ.ต้องชำระเงิน

5.4 ตรวจสอบสถานะการชำระเงิน(checkPaymentStatus) แสดงสถานะการชำระเงินของนศ.

5.5 แสดงชั่วโมงกิจกรรม (getMyActivityHour) เอาไว้ตรวจสอบชม.กิจกรรมของนศ. ที่ได้ทำไปแล้ว

5.6 แสดงจำนวนเงินจากการลงทะเบียนที่ต้องจ่าย(deptOfRegisterCourses)

5.7 ตั้งค่าสถานะการจ่ายเงิน (setPaymentStatus)

5.8 ตั้งค่าจำนวนเงินจากการลงทะเบียนเรียน (setDeptOfRegisterCoures)

5.9 ล้างยอดชำระ (clearDept)

5.10 ตั้งค่ารายวิชาที่ลงทะเบียน (setRegisterCourses)

5.11 ลบรายวิชาที่ลงทะเบียน (removeRegisterCourse)

5.12 toString

6. กิจกรรม(Activity) ประกอบด้วย ชม.กิจกรรม(hour), ชื่อกิจกรรม(nameOfActivity) , รหัสกิจกรรม (codeAct)โดยมีเมธอดสามารถเรียกดู ชื่อกิจกรรม(getNameOfActivity), ชม.กิจกรรม(getHours), รหัสกิจกรรม (getCodeAct)

6.1 กิจกรรม(Activity) เป็น constuctor รับชื่อกิจกรรม(nameOfActivity), รหัสกิจกรรม (codeAct), ชม.กิจกรรม(hour)

6.2 แสดงข้อมูลของกิจกรรม (toString) เอาไว้แสดงข้อมูลออกมาในรูปแบบที่สวยงาม

6.3 เปรียบเทียบ(equals) เอาไว้ตรวจสอบข้อมูลที่ซ้ำ

7. ใบเสร็จชำระเงิน(PaymentBill) ประกอบด้วย จำนวนเงินทั้งหมดที่ต้องชำระ(totalPay), รายวิชาที่ลงทะเบียน (registerCourses) สามารถเรียกดูยอดที่ต้องจ่าย(getTotalPay), รายวิชาที่ลงทะเบียน(registerCourses) และแสดงข้อมูลของใบเสร็จชำระเงิน (toString) เอาไว้แสดงข้อมูลออกมาในรูปแบบที่สวยงาม

8. สถานะการชำระเงิน(PaymentStatus) ได้แก่ จ่ายแล้ว(PAID), ระหว่างดำเนินการ(PENDING) และ เลยกําหนดการ(EXPIRES)

9. รายวิชา(Courses) ประกอบไปด้วย ชื่อวิชา(subject), รหัสรายวิชา(courseCode), หน่วยกิต(credits) และ ราคาต่อ 1 หน่วยกิต 1200 บาท(cost), ราคาเมื่อคำนวณสำเร็จ(calCost) โดยสามารถเรียกดูชื่อวิชา(getSubject),รหัสรายวิชา(getCourseCode), หน่วยกิต(getCredits)ราคา(getCalCost)

9.1 วิชา(Courses) เป็น constuctor รับชื่อวิชา(subject), รหัสรายวิชา(courseCode), หน่วยกิต (credits)

9.2 คำนวณจากราคาหน่วยกิต(calculateCost) รับหน่วยกิตเข้าไปแล้วคำนวณจากการเอาหน่วยกิตคูณราคาต่อ 1 หน่วยกิต 1200 บาท return ค่าออกมาเป็นราคาที่แท้จริง(calCost)

9.3 แสดงข้อมูลของกิจกรรม (toString) เอาไว้แสดงข้อมูลออกมาในรูปแบบที่สวยงาม

10. ลงทะเบียนเรียน(RegisterCourses) ประกอบด้วย บัญชีนักศึกษา(StudentAccount), รายวิชา(Courses), นับจำนวนรายวิชา(count) และจำนวนเงินทั้งหมดที่ต้องชำระ(TotalPay) โดยสามารถเรียกดูบัญชีนักศึกษา(StudentAccount), รายวิชา(getCourses), จำนวนที่ลงทะเบียน(getCount) ยอดรวมทั้งสิ้นจากการคำนวณ(getTotalPay) และ toString เอาไว้ return ค่าข้อมูลกลับมา

11. นโยบาย(Policy) เอาไว้กำหนดจำนวนขั้นต่ำ และสูงสุดในการลงทะเบียนเรียนของนศ. ได้แก่ (MAX_COURSES = 7) คือจำนวนรายวิชาที่ลงต้องไม่เกิน 7 รายวิชา, (MIN_COURSES = 3) คือจำนวนรายวิชาขั้นต่ำที่นศ.ต้องลงเกิน 3 รายวิชา

12.การจัดการบริการแก่นศ.(StudentService) ประกอบด้วย เรียกดูจำนวนคอร์สที่ลงทะเบียนไป(getMyCourseTakenList), การชำระเงิน(pay), ลงทะเบียนเรียน(registerCourse), เปลี่ยนรหัสผ่าน(changePassword)

13.การจัดการบริหารคณะ(DepartmentService) ประกอบด้วย เพิ่มรายวิชา(addCourses), เพิ่มสมาชิก(addMember), เพิ่มกิจกรรม(addActivity), เพิ่มชม.กิจกรรม(addActivityHour), ลบกิจกรรม(removeActivity), ลบรายวิชา(removeCourses) และ เปลี่ยนสถานะการชำระเงิน(changePaymentsStatus)

14. บัญชีของผู้บัญชีของผู้จัดการฐานข้อมูล(DepartmentAccount) เป็น class เปล่าที่ดึงเอา(extend) account มาใช้

15. คลาสเชื่อมต่อดาต้าเบส (ConnectDB) ซึ่งมี Attribute ประกอบไปด้วย DRIVER, URL, USERNAME, PASSWORD แต่ละแอตทริบิวต์ก็จะนำมาใส่ในเมธอด getConnection ภายในคลาส

16. คลาสหลักดาต้าเบส(LibraryDao) ซึ่งภายในก็จะมีเมธอด

- 16.1 insert(dbd obj) รับข้อมูลเป็นชนิด object
- 16.2 delete(dbd obj)รับข้อมูลเป็นชนิด object
- 16.3 update(dbd obj)รับข้อมูลเป็นชนิด object
- 16.4 findById(String id)รับข้อมูลเป็นชนิดString
- 16.5 ArrayList<dbd> getAll()รับข้อมูลเป็นชนิด ArrayList
- 16.6 delete2(int id)รับข้อมูลเป็นชนิด int ที่ชื่อว่าId

โดยทุกเมธอดภายในคลาสนี้จะถูก implements โดยคลาสดังต่อไปนี้ AccountDaoImp, AcctivityDaoImp และ CoursesDaoImp แล้วนำไปประยุกต์ให้เหมาะสมกับประเภทของ parameter ในแต่ละคลาสที่รับเข้ามา