



กรณีศึกษา: การพัฒนาระบบลงทะเบียนออนไลน์

วัตถุประสงค์

เพื่อให้ผู้เรียนสามารถ:

- เข้าใจกระบวนการวิเคราะห์ การออกแบบ การเขียนโปรแกรม และการพัฒนาระบบงานจริง

บทนำ

เนื้อหาในบทนี้ประกอบด้วยขั้นตอนการสร้างแบบจำลองต่างๆ ที่พัฒนาขึ้นในแต่ละช่วงของการวิเคราะห์และออกแบบระบบ รวมถึงการเขียนโปรแกรม การออกแบบฐานข้อมูลเพื่อจัดการกับข้อมูล ของระบบลงทะเบียนออนไลน์ ประกอบด้วยแบบจำลองและเนื้อหาดังต่อไปนี้

1. คำอธิบายระบบงาน
2. ตารางเหตุการณ์และแผนภาพแบบจำลอง Use Case (Event Table and Use Case Diagram)
3. ลำดับขั้นตอนของ Use case (Use Case Scenario)
4. แผนภาพกิจกรรม (Activity Diagram)
5. แบบจำลองลำดับเหตุการณ์ของระบบ (System Sequence Diagram)
6. แบบจำลองโดเมน (Domain Model)
7. แบบจำลองโต้ตอบ (Interaction Model)
 - แผนภาพลำดับเหตุการณ์ (System Diagram—SD)
 - SD ของการเริ่มต้นการลงทะเบียน

- SD ของการลงทะเบียนเรียน
 - SD ของการสิ้นสุดการลงทะเบียนและพิมพ์ใบแจ้งหนี้
 - แผนภาพความร่วมมือ (Collaboration Diagram—SD)
 - SD ของการเริ่มต้นการลงทะเบียน
 - SD ของการลงทะเบียนเรียน
 - SD ของการสิ้นสุดการลงทะเบียนและพิมพ์ใบแจ้งหนี้
8. แบบจำลองคลาส (Class Model)
- First-cut design class diagram
 - Design class diagram สำหรับ use case ลงทะเบียน
9. แบบจำลองแพ็คเกจ (Package Model)
10. การออกแบบส่วนติดต่อกับผู้ใช้งาน (User Interface (<<boundary class>>))
11. การออกแบบฐานข้อมูล
12. การออกแบบซอฟต์แวร์คลาส (Software Class) โดยใช้ภาษา JAVA
- Domain Class
 - Student Class
 - Course Class
 - Section Class
 - Enrollment Class
 - Data Access Class
 - StudentDA Class
 - CourseDA Class
 - SectionDA Class
 - EnrollmentDA Class
 - Controller Class
 - EnrollmentHandler Class
 - Boundary Class
 - LoginForm
 - EnrollmentForm

20.1. คำอธิบายระบบงาน

ระบบการลงทะเบียนออนไลน์ เป็นระบบการลงทะเบียนของสถาบันการศึกษาระดับอุดมศึกษา ที่อนุญาตให้นักศึกษาสามารถลงทะเบียนออนไลน์ผ่านเครือข่ายอินเทอร์เน็ตได้โดยตรง นักศึกษาแต่ละคน (ซึ่งสังกัดสาขาวิชาของแต่ละคณะในมหาวิทยาลัย) สามารถเลือกรายวิชาเพื่อลงทะเบียนได้ตามแผนการศึกษาของแต่ละหลักสูตร (ในที่นี้จะไม่กล่าวถึงรายละเอียดของหลักสูตร เพื่อไม่ให้ระบบงานมีความซับซ้อนมากเกินไป) ในแต่ละรายวิชาประกอบด้วยกลุ่มวิชา (Section) ซึ่งในแต่ละกลุ่มจะมีข้อกำหนดที่แตกต่างกันไป เช่น วันและเวลาของการเรียน จำนวนนักศึกษาต่อกลุ่ม และอาจารย์ผู้สอน เป็นต้น

หลังจากที่นักศึกษาทำการเลือกรายวิชาและกลุ่มวิชาที่ลงทะเบียนได้ครบจำนวนวิชาที่ต้องการแล้ว นักศึกษาจะทำการพิมพ์ใบแจ้งหนี้ เพื่อนำไปชำระเงินที่ธนาคารที่กำหนดไว้ เพื่อเป็นการยืนยันว่า นักศึกษาค้นนั้นได้ทำการลงทะเบียนเรียนเรียบร้อยแล้ว

อย่างไรก็ตาม การลงทะเบียนเรียน นักศึกษาสามารถเพิ่มรายวิชาในรายการลงทะเบียนได้ และในทางตรงกันข้าม นักศึกษาสามารถถอนรายวิชาเดิมที่เคยลงทะเบียนไว้แล้วออกได้เช่นกัน ในกรณีศึกษานี้จะแสดงให้เห็นถึงการพัฒนาแบบจำลองที่มีในกระบวนการวิเคราะห์และออกแบบระบบ เฉพาะในส่วนของการลงทะเบียนเท่านั้น

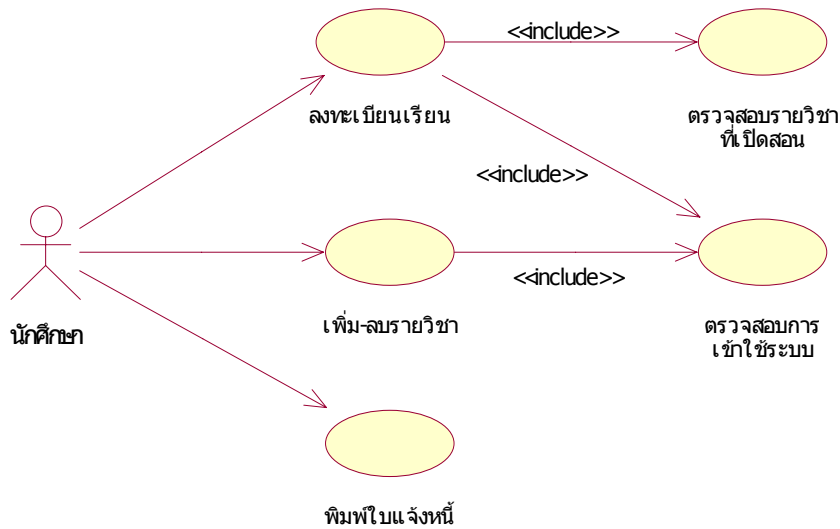
20.2. ตารางเหตุการณ์และแผนภาพแบบจำลอง Use Case

หลังจากที่นักวิเคราะห์ระบบ ได้ดำเนินการเก็บรวบรวมข้อมูลและวิเคราะห์ความต้องการแล้ว สามารถสร้างตารางเหตุการณ์ (Event Table) และ Use Case Diagram สำหรับระบบงานนี้ได้ดังนี้

Event	Trigger	Source	Use Case	Response	Destination
นักศึกษาลงทะเบียนเรียน	รายการวิชาที่จะลงทะเบียน	นักศึกษา	ลงทะเบียนเรียน	รายการวิชาที่ลงทะเบียน	นักศึกษา
นักศึกษาเพิ่ม-ลบรายวิชา	รายการวิชาที่ต้องการเพิ่ม-ถอน	นักศึกษา	เพิ่ม-ลบรายวิชา	รายการวิชาที่เพิ่ม-ถอน	นักศึกษา
นักศึกษาสั่งพิมพ์ใบแจ้งหนี้	ยอดเงินค่าหน่วยกิตที่ต้องชำระ	นักศึกษา	พิมพ์ใบแจ้งหนี้	ใบแจ้งหนี้	นักศึกษา

รูปที่ 20.1 แสดงข้อมูลในตารางเหตุการณ์

จากตารางจะเห็นว่า ระบบงานประกอบด้วย 3 Use Case คือ ลงทะเบียนเรียน เพิ่ม-[รายวิชา และพิมพ์ใบแจ้งหนี้ โดยแสดงเป็นแผนภาพได้ดังรูปที่ 20.2



รูปที่ 20.2 แผนภาพแสดงแบบจำลอง Use Case ของระบบลงทะเบียนออนไลน์

หัวข้อถัดไปจะแสดงรายละเอียดของ Use Case ลงทะเบียนเรียนทางเว็บ (Use Case Scenario) ซึ่งเป็นรูปแบบหนึ่งของการลงทะเบียน (Use Case)

20.3. ลำดับขั้นตอนของ Use Case

รูปที่ 20.3 แสดงลำดับขั้นตอนการลงทะเบียนทางเว็บ (ออนไลน์) (Use Case Scenario) โดยมีนักศึกษาเป็นผู้กระทำ

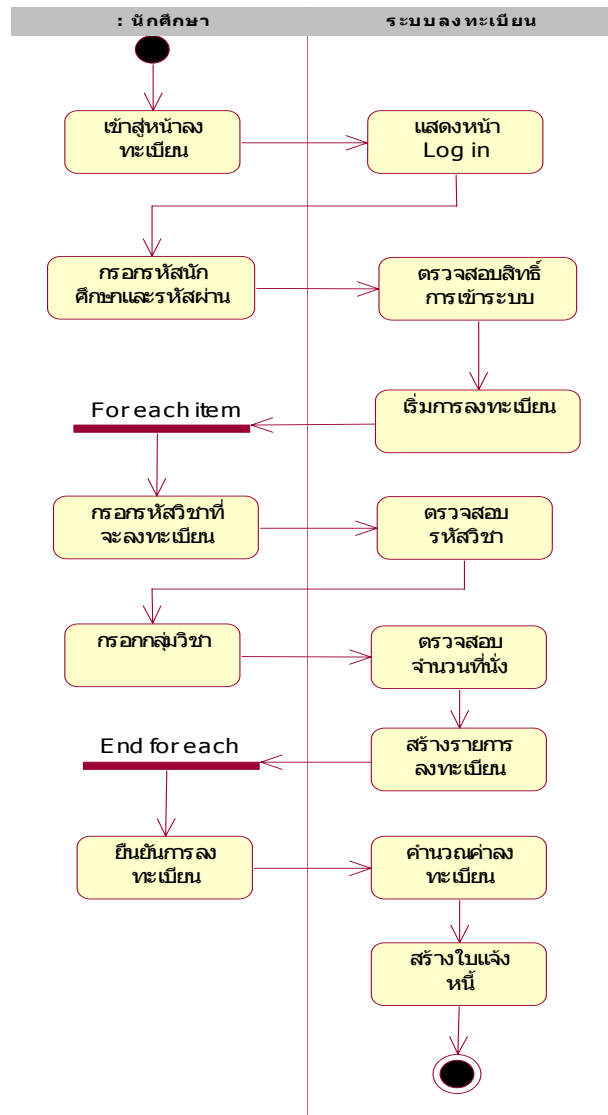
Use Case Name	ลงทะเบียนเรียน
Scenario	ลงทะเบียนเรียนทางเว็บ
Triggering Event	นักศึกษา(Actor) เข้าสู่ระบบลงทะเบียนออนไลน์ เป็นจุดเริ่มต้นที่ ระบบลงทะเบียนออนไลน์ จะเริ่มทำงาน เช่น การระบุปีและเทอมการศึกษา การเลือกรายวิชาที่ต้องการลงทะเบียน(หลายๆ วิชา) จนถึงการพิมพ์ใบแจ้งการชำระเงิน เป็นต้น
Brief Description	เมื่อนักศึกษาเข้าระบบเพื่อทำการลงทะเบียน ระบบจะตรวจสอบสิทธิ์การเข้าใช้ระบบของนักศึกษา และให้นักศึกษาระบุรายวิชาที่ต้องการลงทะเบียน ระบบตรวจสอบรายวิชา และให้นักศึกษาเลือกเลือกกลุ่ม เมื่อลงครบทุกวิชาแล้วระบบจะทำการออกใบแจ้งหนี้ให้กับนักศึกษา
Actors	นักศึกษา
Related Use Cases	- ตรวจสอบการเข้าใช้ระบบ - ตรวจสอบรายวิชาที่เปิดสอน
Stakeholders	<u>สำนักทะเบียน</u> : เป็นฝ่ายกำหนดรายละเอียดของรายวิชา <u>อาจารย์</u> : เป็นผู้แจ้งรายวิชาที่จะเปิดสอน
Preconditions	ข้อมูลต่อไปนี้อาจมีเรียบร้อยแล้ว คือ นักศึกษา, รายละเอียดรายวิชา, และ รายวิชาที่จะเปิดสอน
Postconditions	- การเข้าสู่ระบบ: วัตถุ (instance) ของนักศึกษาจะถูกสร้างขึ้น - การลงทะเบียน: วัตถุ (instance) ของการลงทะเบียน และรายวิชาจะถูกสร้างขึ้น - การพิมพ์ใบแจ้งหนี้: วัตถุ (instance) ของใบแจ้งหนี้จะถูกสร้างขึ้น - จำนวนที่ลงทะเบียนได้ในกลุ่มวิชานั้นต้องมีเพียงพอ - รายวิชาที่ลงทะเบียนจะสัมพันธ์กับนักศึกษา

Flow of Events	Actor	System
	1. นักศึกษาเข้าสู่ระบบ 2. นักศึกษาเตรียมการสำหรับการลงทะเบียน 3. นักศึกษากรอกรหัสวิชาที่ต้องการจะลงทะเบียน 4. นักศึกษาเลือกกลุ่มวิชา 5. ทำซ้ำขั้นตอนที่ 3 ถึง 4 จนกว่าจะครบทุกวิชาที่จะลงทะเบียน 6. นักศึกษายืนยันการลงทะเบียน 7. นักศึกษาสั่งพิมพ์ใบแจ้งหนี้	1.1 ระบบตรวจสอบสิทธิ์การเข้าใช้ระบบ 2.1 สร้างใบลงทะเบียนขึ้น 1 ใบ 3.1 ตรวจสอบการเปิดรายวิชา และกลุ่มของวิชา 4.1 ตรวจสอบจำนวนคนที่ลงทะเบียนแล้ว 6.1 เพิ่มรายวิชาที่นักศึกษาลงทะเบียนบนใบลงทะเบียน 6.2 สร้างรายการลงทะเบียน (Register Transaction) 6.3 เสร็จสิ้นการลงทะเบียน 7.1 คำนวณค่าลงทะเบียน 7.2 สร้างใบแจ้งหนี้
Exception Conditions	3.1 ในกรณีที่ไม่มีข้อมูลรายวิชาที่นักศึกษาต้องการในระบบ นักศึกษาจะต้องค้นหาข้อมูลใน Use case ตรวจสอบรายวิชาที่เปิดสอนก่อน และนำรหัสวิชานั้นมารอกอีกครั้ง 4.1 ถ้ากลุ่มวิชานั้นนั้นมีที่นั่งไม่เพียงพอจะทำการแจ้งข้อความเตือนให้นักศึกษาทราบ และเลือกกลุ่มวิชาใหม่ 6.1 ถ้าไม่สามารถยืนยันการลงทะเบียนได้ เนื่องจากยังมีที่นั่งเดิมอยู่ นักศึกษาสามารถเลือก <ul style="list-style-type: none"> a. ยกเลิกการลงทะเบียน b. ใบแจ้งหนี้จะถูกเก็บไว้จนกว่านักศึกษาจะได้รับการปลดล็อค 	

รูปที่ 20.3 แสดงลำดับขั้นตอน Use Case ลงทะเบียนทางเว็บ

20.4. แผนภาพกิจกรรม

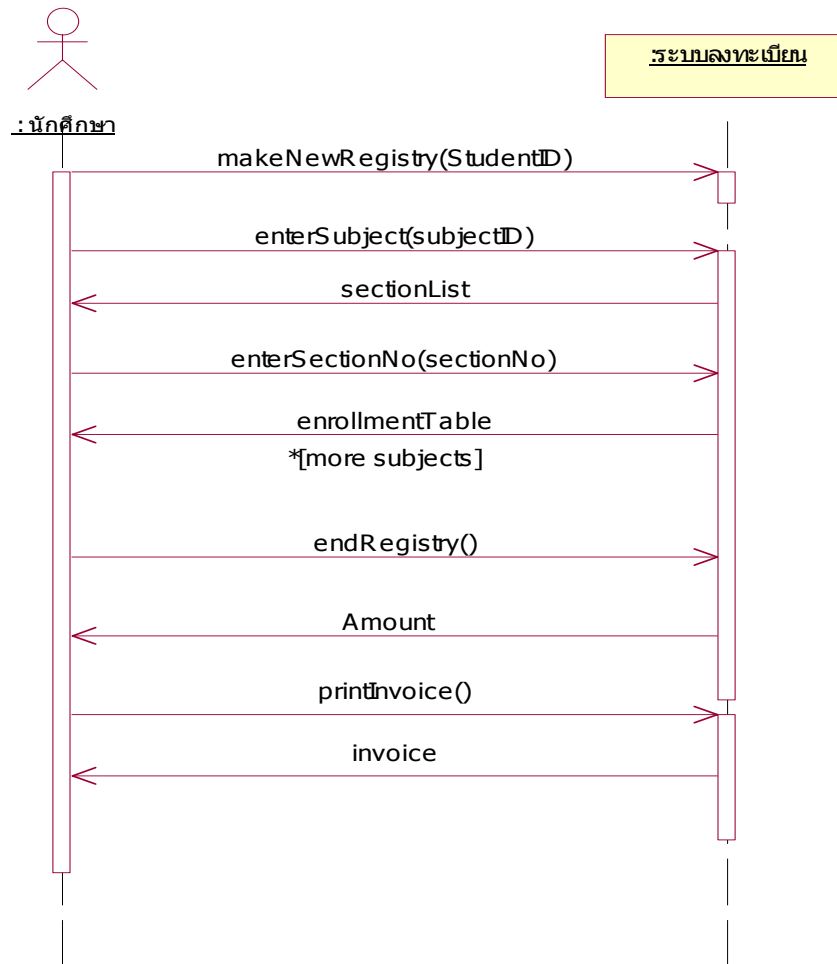
จากลำดับขั้นตอน Use Case ลงทะเบียนทางเว็บ ซึ่งมีส่วนที่สำคัญคือ การโต้ตอบระหว่างผู้กระทำ (Actor) และระบบ ลำดับการโต้ตอบดังกล่าวสามารถเขียนเป็นแผนภาพกิจกรรม (Activity Diagram) เพื่อให้เข้าใจได้ง่ายขึ้นได้ดังรูปที่ 20.4



รูปที่ 20.4 แสดงแผนภาพกิจกรรม Use Case ลงทะเบียนทางเว็บ

20.5. แบบจำลองลำดับเหตุการณ์ของระบบ

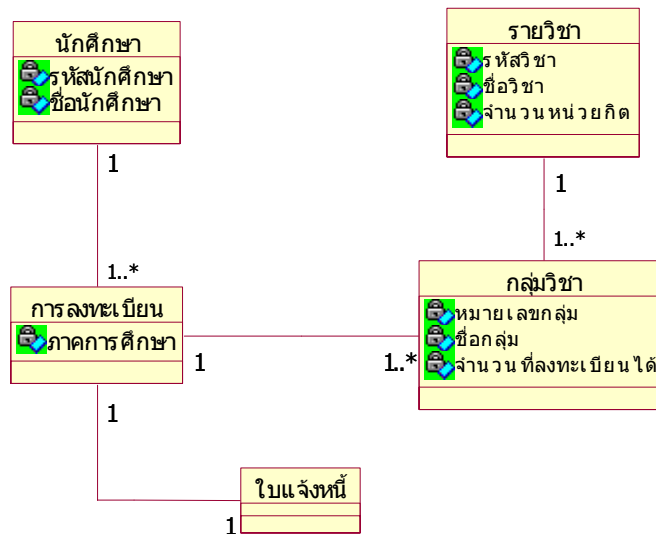
จากลำดับขั้นตอนของ Use Case รวมถึงแผนภาพกิจกรรม สามารถนำมาพัฒนาเป็นแผนภาพลำดับเหตุการณ์ของระบบ (System Sequence Diagram—SSD) ได้ดังรูปที่ 20.5



รูปที่ 20.5 แสดงแผนภาพแบบจำลองโดเมนของระบบการลงทะเบียนออนไลน์

20.6. แบบจำลองโดเมน

หลังจากการวิเคราะห์ โดยใช้ข้อมูลจากแบบจำลองที่ผ่านมา สามารถกำหนดเอนทิตีและความสัมพันธ์ระหว่างเอนทิตี และสามารถสร้างเป็นแบบจำลองโดเมน (Domain Model) ได้ดังรูปที่ 20.6



รูปที่ 20.6 แสดงแผนภาพแบบจำลองโดเมนของระบบการลงทะเบียนออนไลน์

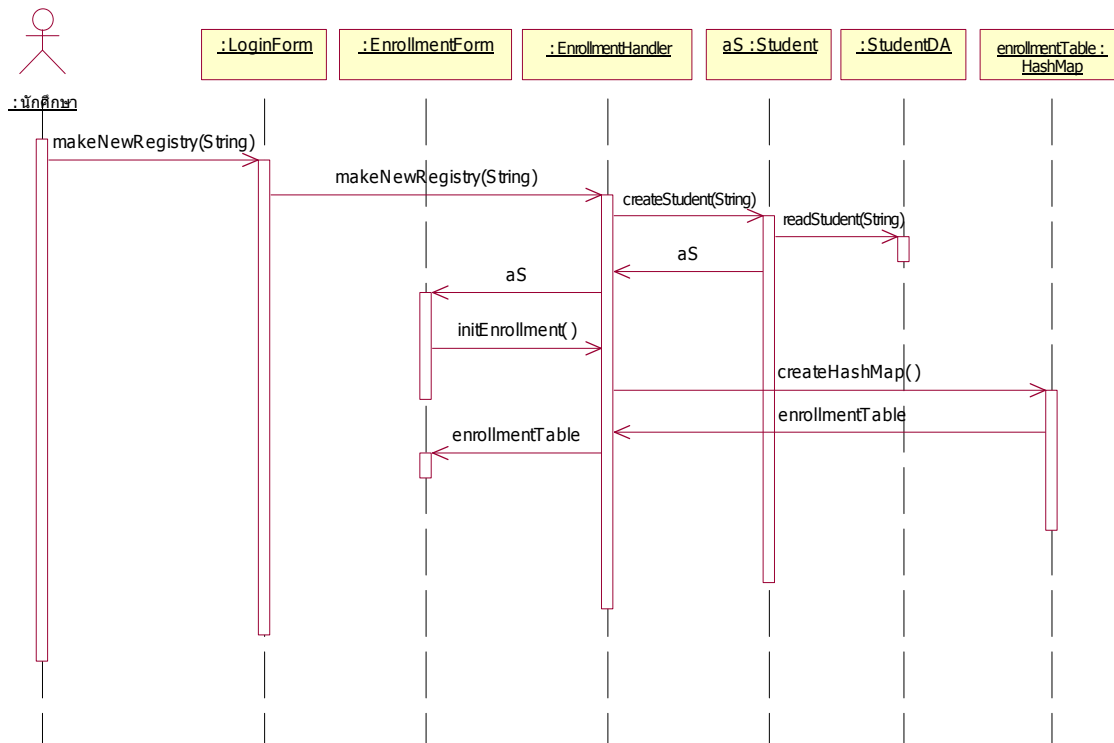
หลังจากที่สามารถพัฒนาแบบจำลอง Use Case (Use Case Scenario, Use Case Diagram, และ System Sequence diagram รวมถึง Activity Diagram) และ แบบจำลองโดเมนเรียบร้อยแล้ว ต่อไปเป็นการพัฒนาแบบจำลองในขั้นตอนของการวิเคราะห์ระบบ คือ แบบจำลองคลาส (Class Model) และแบบจำลองโต้ตอบ (Interaction Model) ซึ่งประกอบไปด้วย แบบจำลองลำดับเหตุการณ์ (Sequence Model) และแบบจำลองความร่วมมือ (Collaboration Model)

เนื่องจากขั้นตอนการพัฒนาแบบจำลองคลาสและแบบจำลองโต้ตอบ จะต้องดำเนินการควบคู่กันไป แบบจำลองทั้งสองจึงจะสมบูรณ์ ในที่นี้จะกล่าวถึงแบบจำลองโต้ตอบเป็นลำดับแรก แต่อย่างไรก็ตามเพื่อความเข้าใจผู้อ่านจะต้องพิจารณาแบบจำลองคลาสควบคู่ไปด้วยจึงจะเข้าใจยิ่งขึ้น

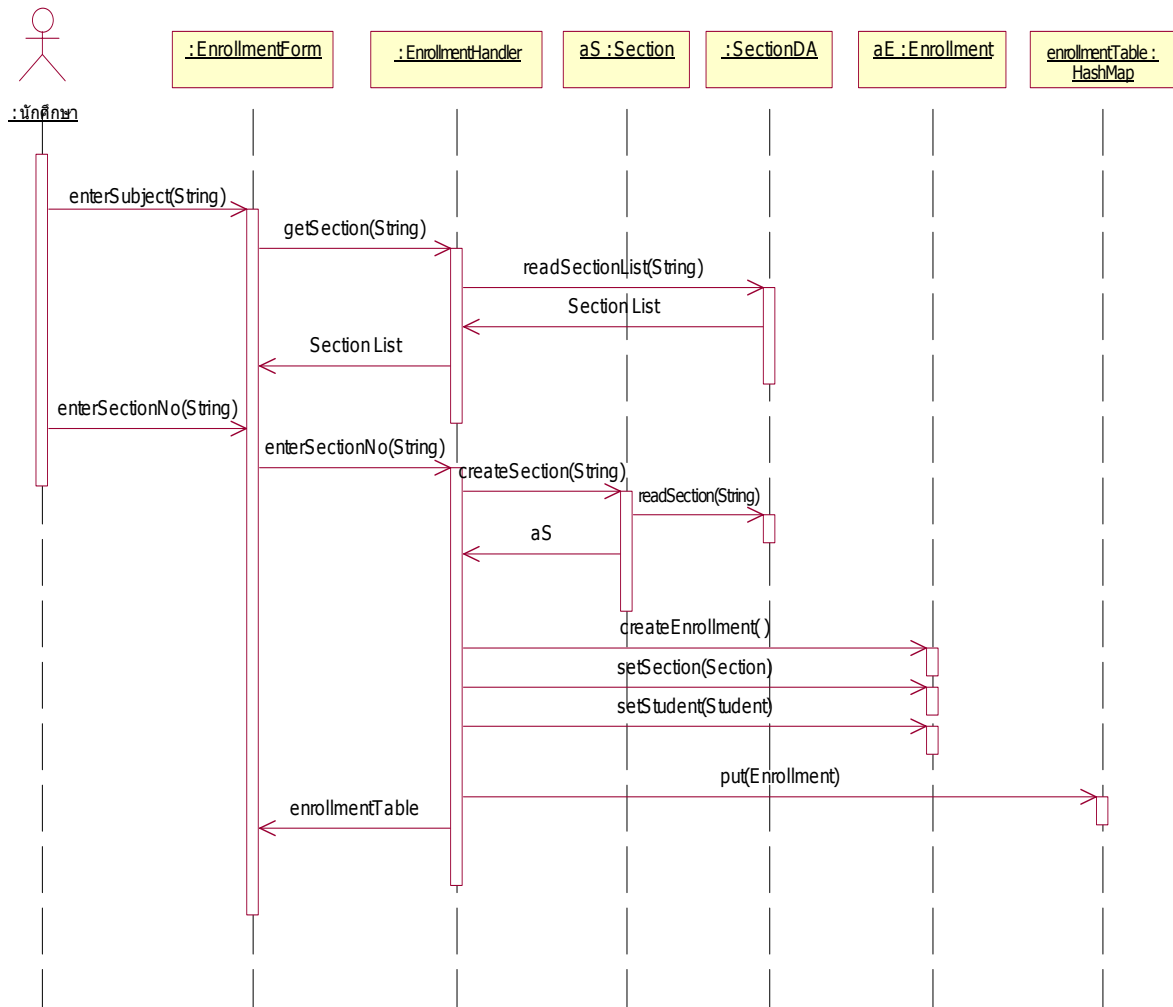
20.7. แบบจำลองโต้ตอบ

แบบจำลองโต้ตอบ สามารถนำเสนอได้ 2 รูปแบบ คือ แบบจำลองลำดับเหตุการณ์ และแบบจำลองความร่วมมือ โดยแต่ละแบบจำลองจะมีความเหมาะสมในการนำเสนอแตกต่างกันไป (ดูบทที่ 14) แต่แบบจำลองทั้งสองจะต้องมีความสอดคล้องกัน (Consistency) จากลำดับขั้นตอนของ Use Case ลงทะเบียนทางเว็บ หลังจากการวิเคราะห์และออกแบบแล้ว สามารถพัฒนาแบบจำลองลำดับเหตุการณ์ได้ดังรูปที่ 20.7 และแบบจำลองความร่วมมือได้ดังรูปที่ 20.8

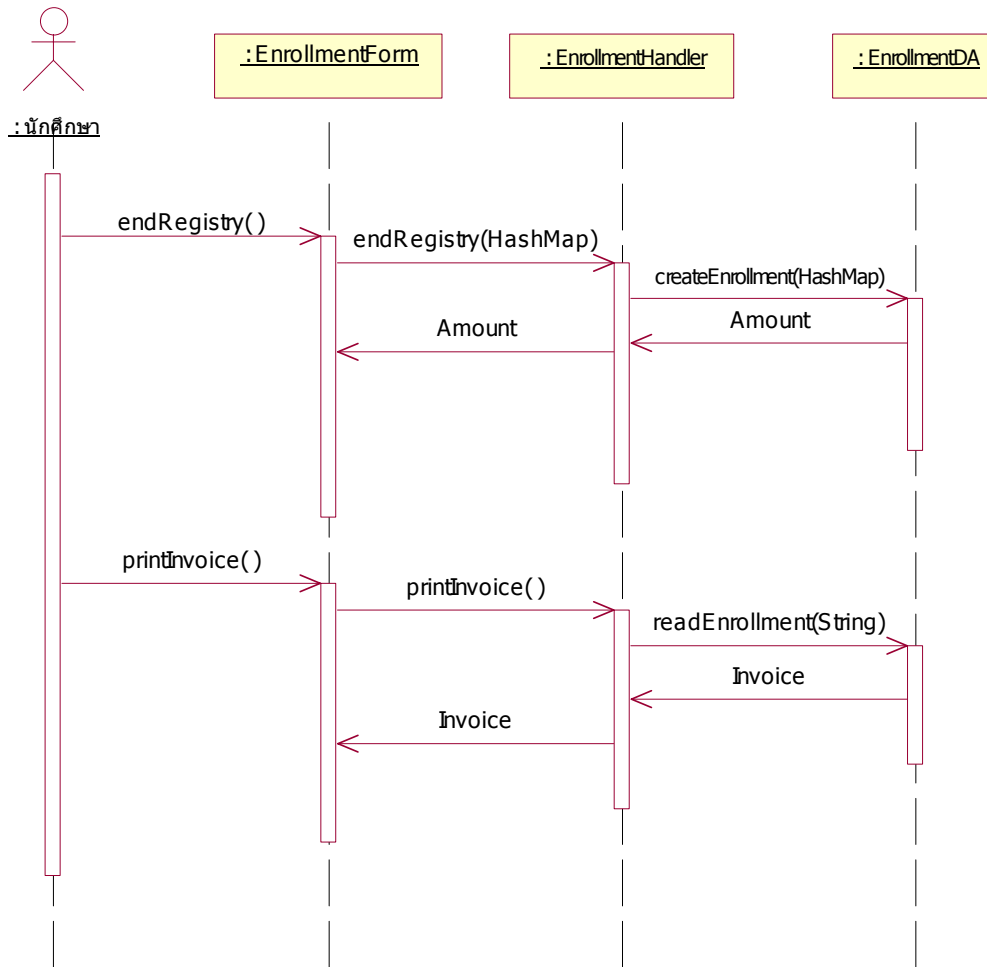
แบบจำลองโต้ตอบของ Use Case ลงทะเบียนทางเว็บ ประกอบด้วยแบบจำลองโต้ตอบย่อย 3 ส่วน คือ (a) เริ่มต้นการลงทะเบียน (b) ลงทะเบียนทางเว็บ และ (c) พิมพ์ใบแจ้งหนี้



รูปที่ 20.7 แสดงแผนภาพแบบจำลองลำดับเหตุการณ์ของ Use Case เริ่มต้นลงทะเบียนทางเว็บ (a)

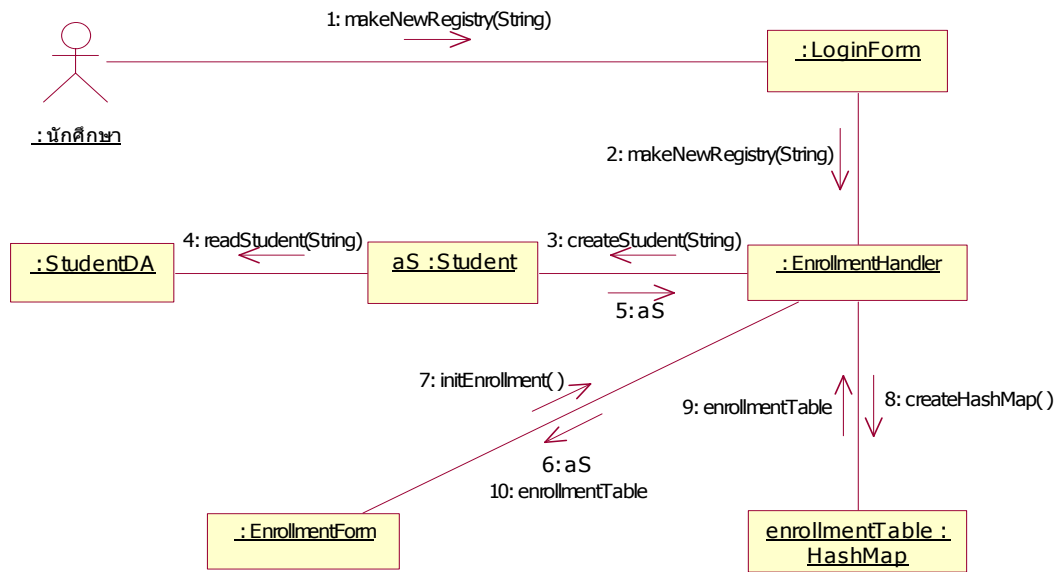


รูปที่ 20.7 แสดงแผนภาพแบบจำลองลำดับเหตุการณ์ของ Use Case ลงทะเบียนทางเว็บ (b)

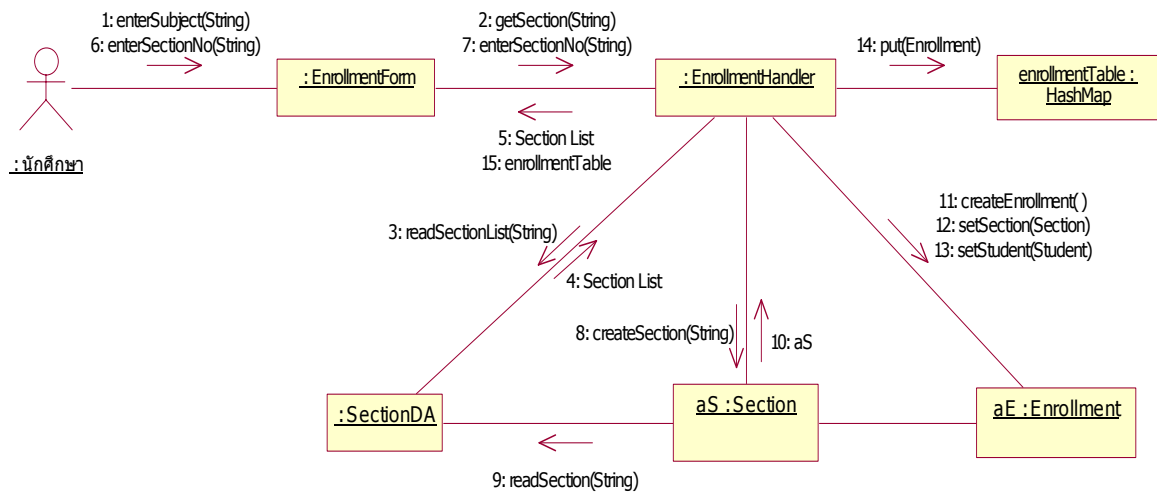


รูปที่ 20.7 แสดงแผนภาพแบบจำลองลำดับเหตุการณ์ของ Use Case พิมพ์ใบแจ้งหนี้ (c)

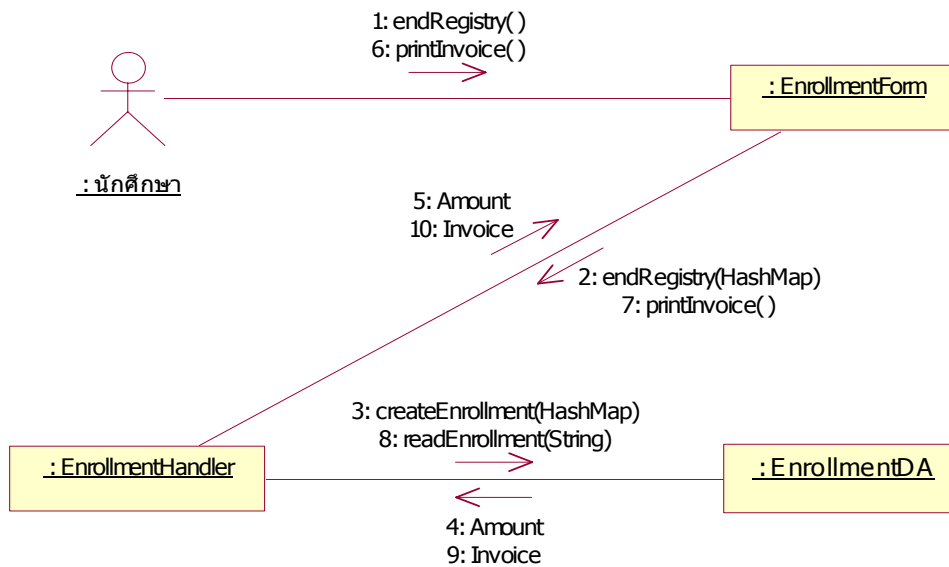
จากแผนภาพแสดงแบบจำลองลำดับเหตุการณ์ในรูปที่ 20.7 (a-c) สามารถนำมาสร้างเป็นแผนภาพความร่วมมือได้ดังรูปที่ 20.8 (a-c) ตามลำดับ



รูปที่ 20.8 แสดงแผนภาพแบบจำลองความร่วมมือของ Use Case เริ่มต้นลงทะเบียนทางเว็บ (a)



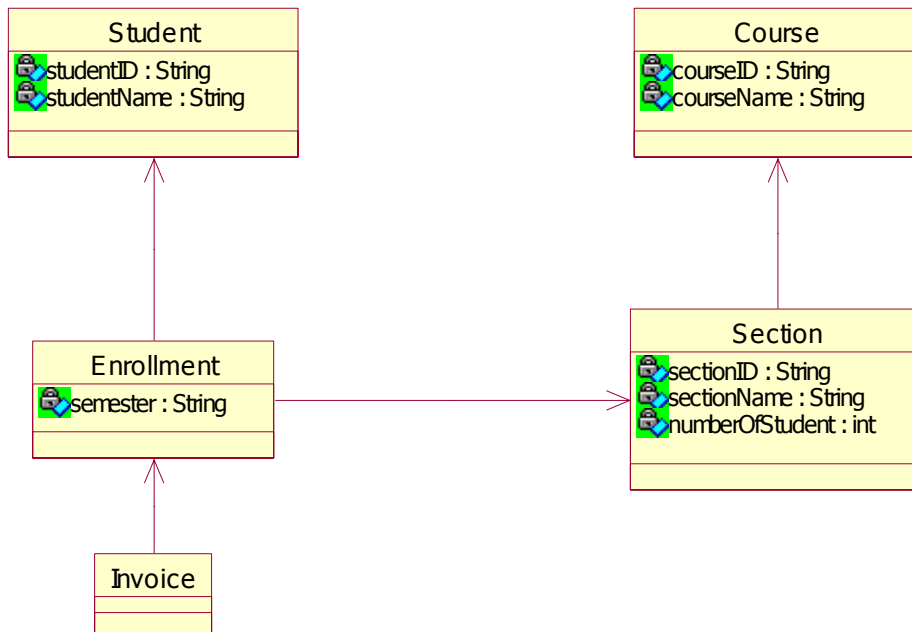
รูปที่ 20.8 แสดงแผนภาพแบบจำลองความร่วมมือของ Use Case ลงทะเบียนทางเว็บ (b)



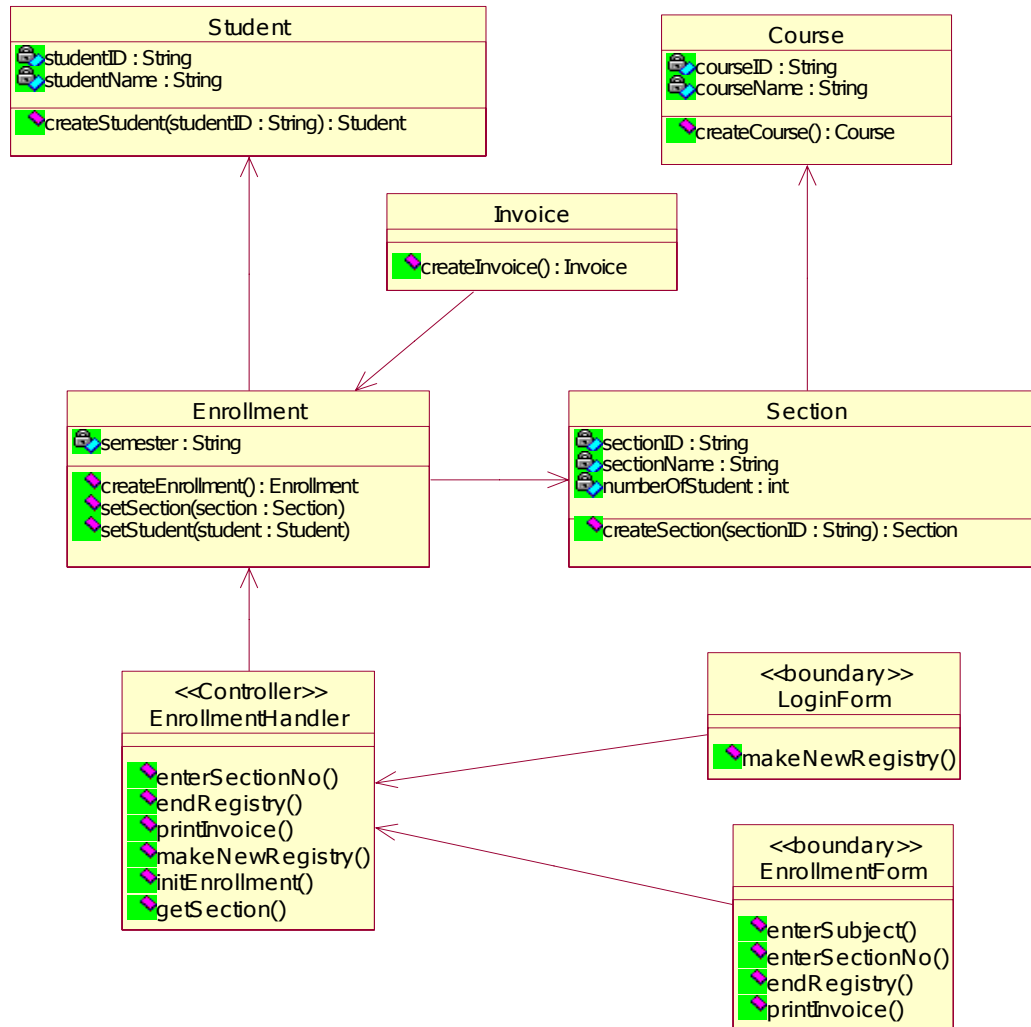
รูปที่ 20.8 แสดงแผนภาพแบบจำลองความร่วมมือของ Use Case พิมพ์ใบแจ้งหนี้ (c)

20.8. แบบจำลองคลาส

จากแบบจำลองโดเมน สามารถสร้างเป็นแบบจำลองคลาสเบื้องต้น ซึ่งประกอบด้วยชื่อคลาส แอททริบิวต์หลักๆ ได้ดังรูปที่ 20.9 (a) และเมื่อนำแบบจำลองคลาสเบื้องต้นไปทำการออกแบบร่วมกับแบบจำลองโต้ตอบ สามารถพัฒนาแบบจำลองคลาสที่สมบูรณ์ได้ดังรูปที่ 20.9 (b)



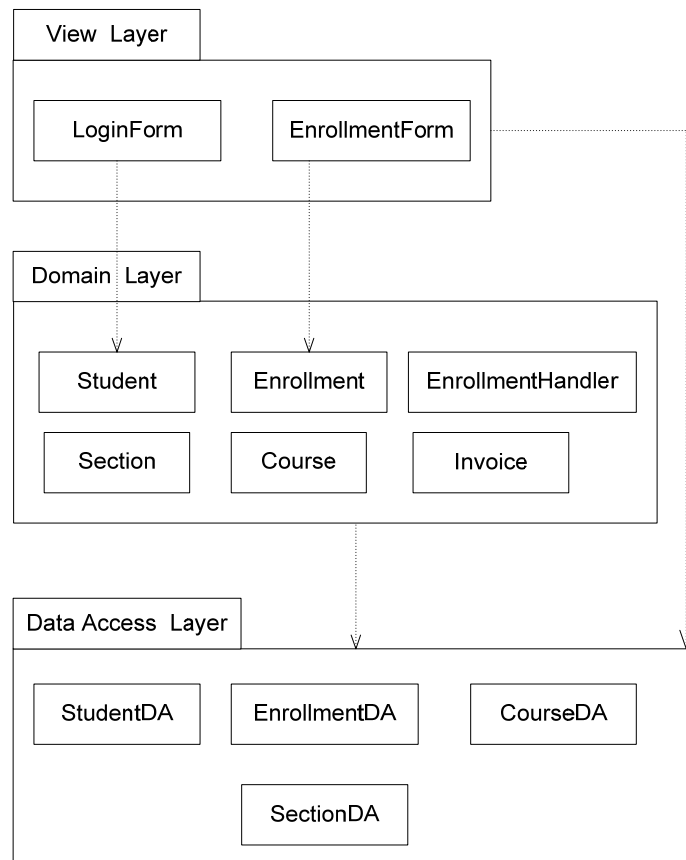
รูปที่ 20.9 แสดงแผนภาพแบบจำลองคลาสเบื้องต้นของระบบการลงทะเบียนเรียนออนไลน์ (a)



รูปที่ 20.9 แสดงแผนภาพแบบจำลองคลาสที่สมบูรณ์ของระบบการลงทะเบียนออนไลน์ (b)

20.9. แบบจำลองแพ็คเกจ

จากแบบจำลองคลาสและแบบจำลองโต้ตอบ ซึ่งออกแบบที่ตั้งอยู่บนสถาปัตยกรรมแบบลำดับชั้น สามารถนำมาสร้างเป็นแบบแพ็คเกจ ซึ่งประกอบด้วย 3 ชั้น คือ ชั้น การติดต่อกับผู้ใช้ (View/User Interface Layer) ชั้น โดเมน (Domain Layer) และ ชั้นการติดต่อกับข้อมูล (ชั้นการให้บริการ) (Data Access Layer) ได้ดังรูปที่ 20.10

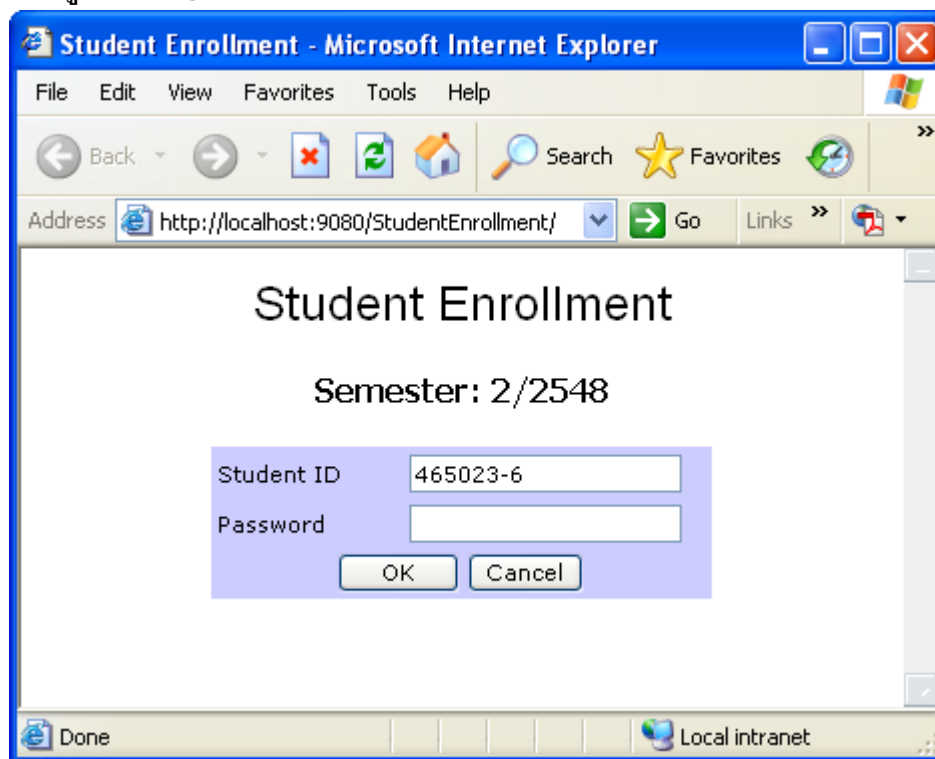


รูปที่ 20.10 แผนภาพแสดงแบบจำลองแพ็คเกจของระบบการลงทะเบียนเรียนออนไลน์

20.10. การออกแบบส่วนติดต่อกับผู้ใช้งาน

จากแบบจำลองคลาส ส่วนของการติดต่อกับผู้ใช้งาน (<<boundary>> คลาส) ซึ่งเป็นส่วนของการบันทึกข้อมูล ประกอบด้วย 2 ส่วน คือ (ดูรูปที่ 20.9 และ รูปที่ 20.10 ประกอบ) คือ LoginForm และ EnrollmentForm ซึ่งออกแบบได้ดังรูปที่ 20.11 และรูปที่ 20.12 ตามลำดับ รวมถึงการติดต่อกับผู้ใช้งานในรายละเอียดของการลงทะเบียน รูปที่ 20.13 (a-d)

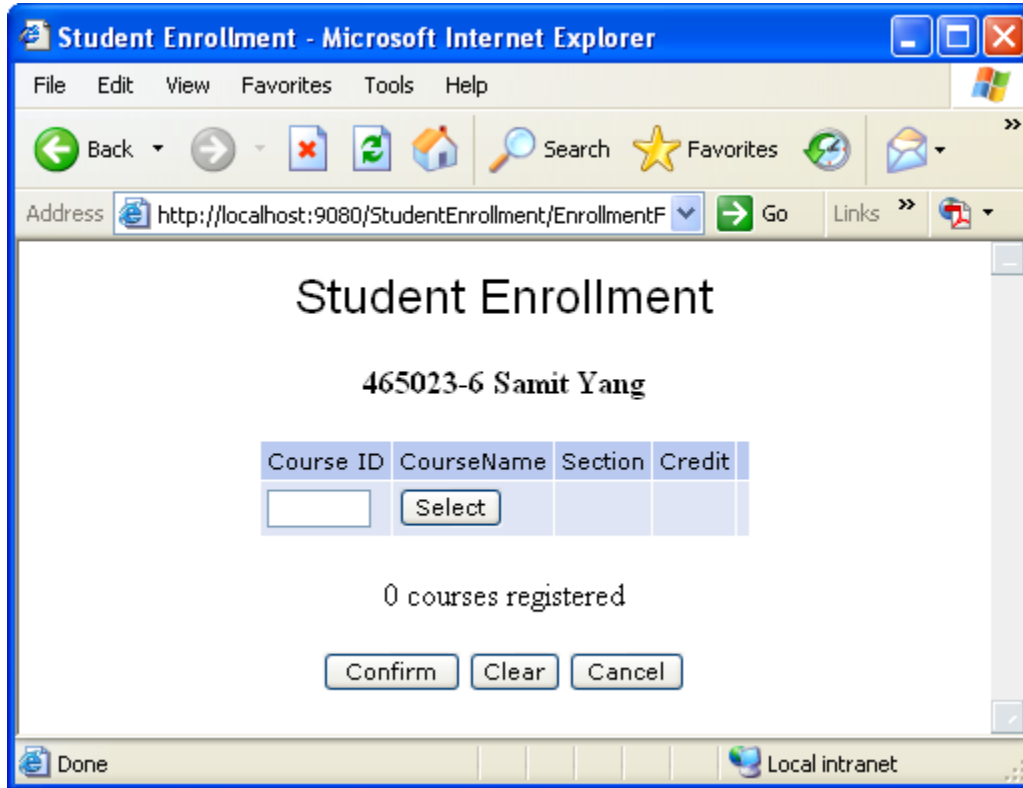
1. หน้าจอเข้าสู่ระบบ (Login Form)



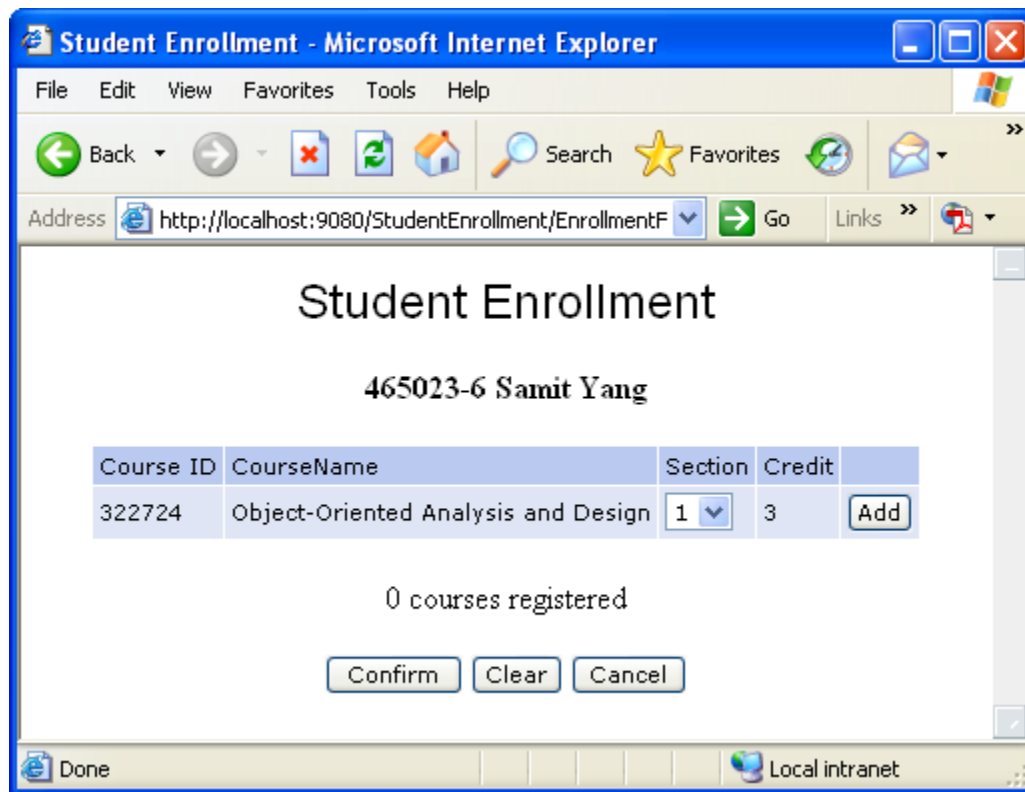
รูปที่ 20.11 แสดงหน้าจอการ Login เข้าสู่ระบบการลงทะเบียน

2. หน้าจอการลงทะเบียน (Enrollment Form)

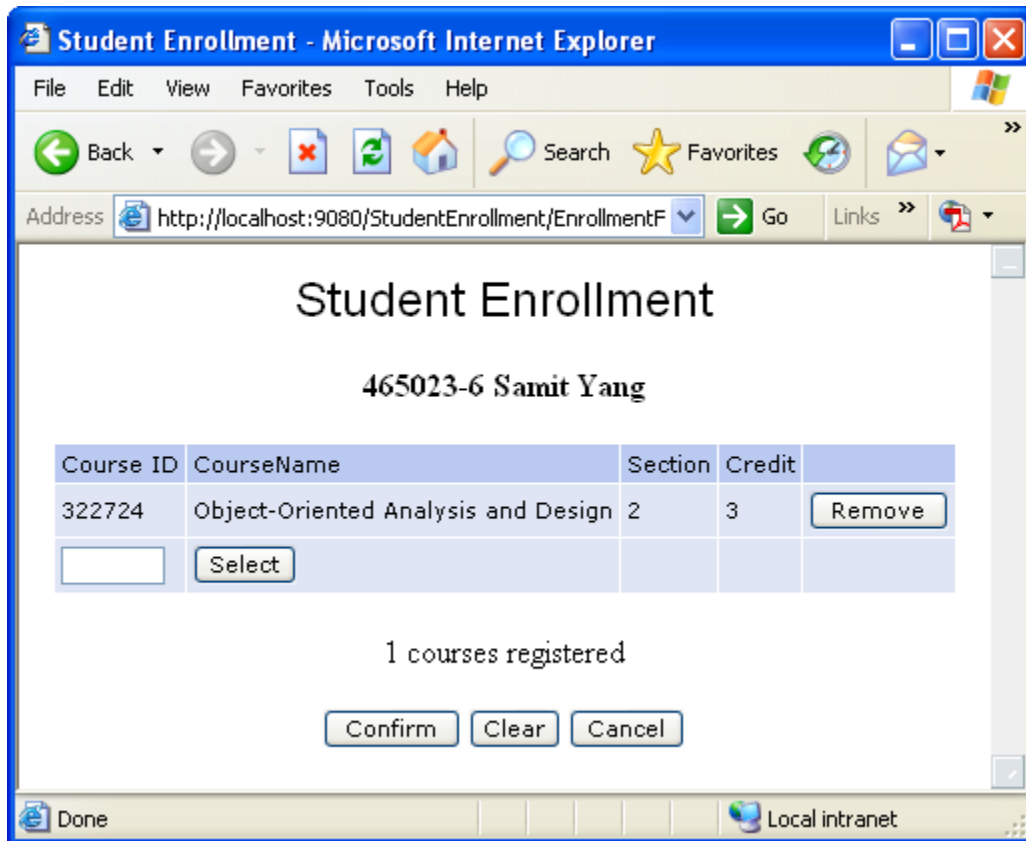
นักศึกษาจะกรอกรหัสวิชาที่ต้องการลงทะเบียนก่อน หลังจากนั้นจะแสดงรายการ Section



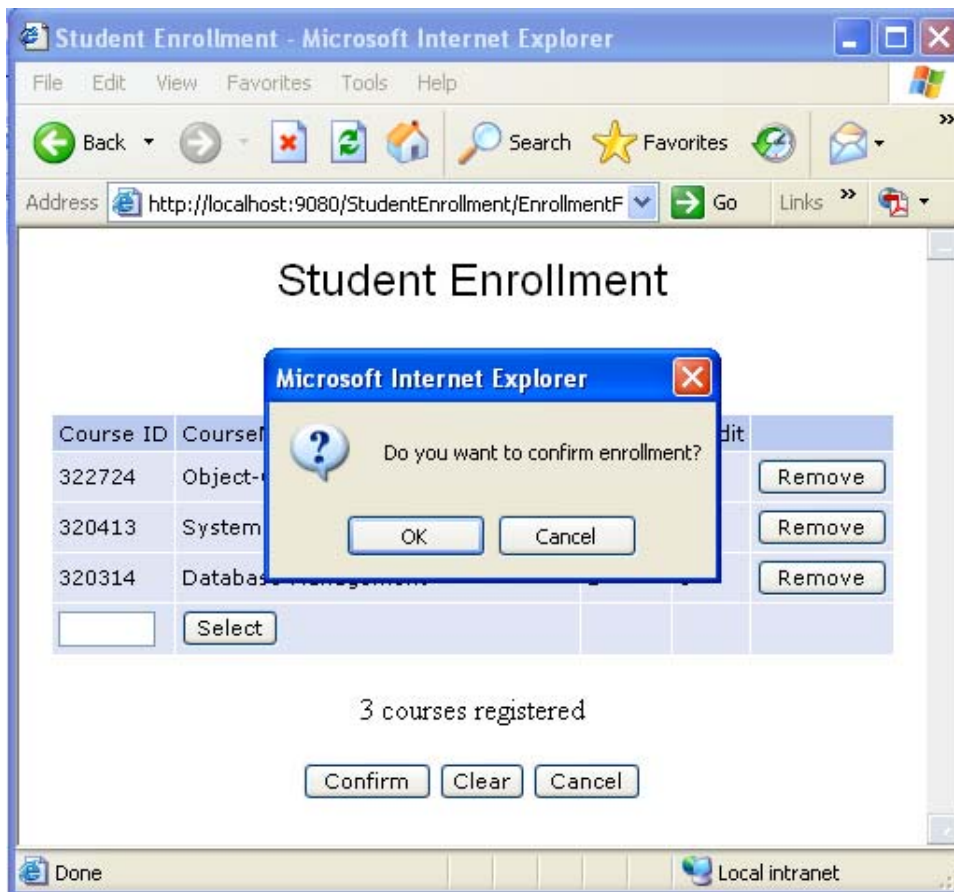
รูปที่ 20.12 แสดงหน้าจอการลงทะเบียน



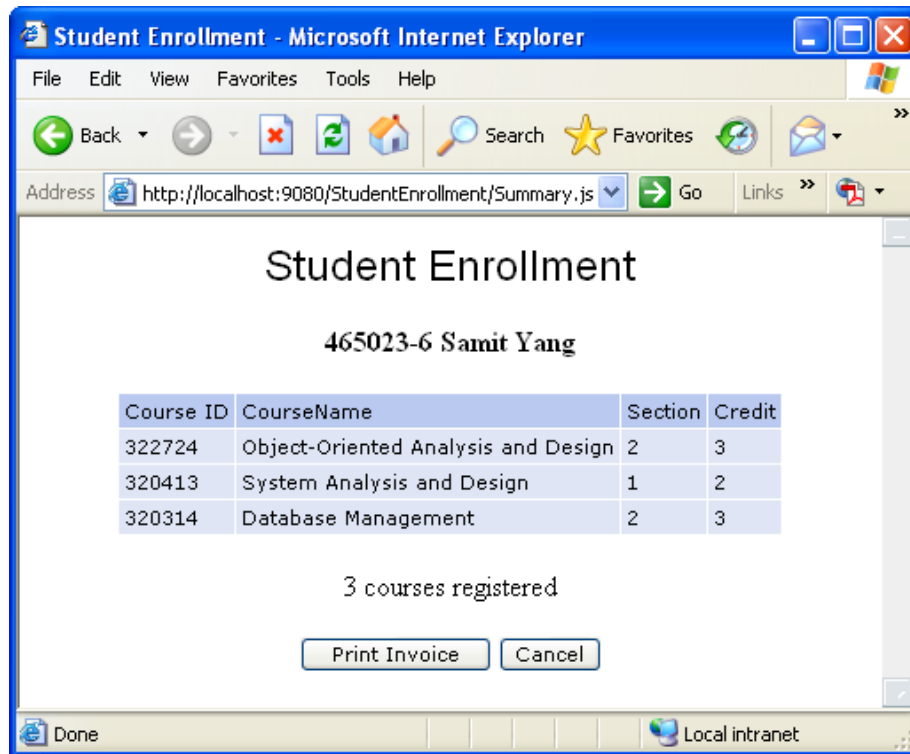
รูปที่ 20.13 แสดงหน้าจอการเลือกกลุ่มรายวิชาสำหรับการลงทะเบียน (a)



รูปที่ 20.13 แสดงหน้าจอการเลือกวิชาสำหรับการลงทะเบียน (b)



รูปที่ 20.13 แสดงหน้าจอการยืนยันการลงทะเบียน (c)



รูปที่ 20.13 แสดงหน้าจอสรุปผลการลงทะเบียน (d)

รายละเอียดของโปรแกรมในส่วนนี้ ปรากฏอยู่ในภาคผนวก ก ซึ่งเขียนโดยใช้ภาษา JSP เพื่อให้ผู้ใช้งานติดต่อกับระบบผ่านทางเว็บ

20.11. การออกแบบฐานข้อมูล

จากแบบจำลองคลาสและแบบจำลองแพ็คเกจ โดเมนคลาส (Entity Class) คือคลาสที่สร้างวัตถุถาวร (Persistence Objects) และจะต้องมีการจัดเก็บไว้ในฐานข้อมูล เพื่อการเรียกกลับมาใช้ในการประมวลผล ในที่นี้จะใช้แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model) สำหรับการจัดเก็บข้อมูล (วัตถุถาวร)

จากคลาสโดเมนในแบบจำลองคลาส (รูปที่ 20.9) สามารถออกแบบเป็นตารางรวมถึงแบบจำลองเชิงสัมพันธ์ได้ดังรูปที่ 20.14

Table	Attributes
Student (นักศึกษา)	StudentID , StudentName
Course (รายวิชา)	CourseID , CourseName, Credit
Section (กลุ่มวิชา)	SectionID , CourseID , SectionName, numberOfStudent
Enrollment (การลงทะเบียน)	StudentID , SectionID , Semester
ตัวหนา คือ PK และ ตัวหนาและเอียง คือ PK & FK	

รูปที่ 20.14 แสดงแบบจำลองฐานข้อมูลเชิงสัมพันธ์ของระบบการลงทะเบียนออนไลน์

20.12. การออกแบบคลาสโดยใช้ภาษา JAVA

การเขียนโปรแกรมประกอบด้วยกลุ่มของคลาส (โปรแกรม) 4 กลุ่ม (ตามประเภทของคลาส) คือ

- กลุ่มสำหรับติดต่อกับผู้ใช้งาน ประกอบด้วยโปรแกรมในส่วนของคลาสประเภท <<boundary>> ซึ่งในที่ประกอบด้วย 2 คลาส คือ LoginForm และ EnrollmentForm โดยมีรายละเอียดในภาคผนวก ก
- กลุ่มควบคุม ประกอบด้วยโปรแกรมในส่วนของคลาสประเภท <<controller>> ซึ่งในที่ประกอบด้วย 1 คลาส คือ EnrollmentHandler โดยมีรายละเอียดในภาคผนวก ข

- กลุ่มติดต่อกับฐานข้อมูล คือโปรแกรมในกลุ่มคลาสที่อยู่ในแพ็คเกจ ชั้น Data Access ประกอบด้วยคลาส StudentDA, คลาส CourseDA, คลาส SectionDA, และคลาส EnrollmentDA โดยมีรายละเอียดในภาคผนวก ก
- กลุ่มโดเมน คือโปรแกรมในกลุ่มคลาสที่อยู่ในแพ็คเกจ ชั้น Domain ซึ่งเป็นคลาสหลักของระบบ ประกอบด้วยคลาส Student, คลาส Course, คลาส Section, และคลาส Enrollment ดังรายละเอียดในรูปที่ 20.15 (a) ถึง รูปที่ 20.15 (d) ตามลำดับ

● Student

```

package domain;

import da.StudentDA;

public class Student {
    private String studentID;
    private String studentName;

    public Student() {
    }

    public Student(String studentID) {
        Student s = StudentDA.readStudent(studentID);
        this.studentID = s.studentID;
        this.studentName = s.studentName;
    }

    public String getStudentID() {
        return studentID;
    }

    public String getStudentName() {
        return studentName;
    }

    public void setStudentID(String studentID) {
        this.studentID = studentID;
    }

    public void setStudentName(String studentName) {
        this.studentName = studentName;
    }
}

```

รูปที่ 20.15 (a) โปรแกรมภาษา JAVA ของคลาส Student

● **Course**

```
package domain;

import da.CourseDA;

public class Course {
    private String courseID;
    private String courseName;
    private int credit;

    public Course() {
    }

    public Course(String courseID) {
        Course c = CourseDA.readCourse(courseID);
        this.courseID = c.courseID;
        this.courseName = c.courseName;
        this.credit = c.credit;
    }

    public String getCourseID() {
        return courseID;
    }

    public String getCourseName() {
        return courseName;
    }

    public int getCredit() {
        return credit;
    }

    public void setCourseID(String courseID) {
        this.courseID = courseID;
    }

    public void setCourseName(String courseName) {
        this.courseName = courseName;
    }

    public void setCredit(int credit) {
        this.credit = credit;
    }
}
```

รูปที่ 20.15 (b) โปรแกรมภาษา JAVA ของคลาส Course

- Section

```
package domain;

import da.SectionDA;
public class Section {
    private String sectionID;
    private Course course;
    private String sectionName;
    private int numberOfStudent;

    public Section() {
    }
    public Section(String sectionID) {
        Section s = SectionDA.readSection(sectionID);
        this.sectionID = s.getSectionID();
        this.course = s.getCourse();
        this.sectionName = s.getSectionName();
        this.numberOfStudent = s.getNumberOfStudent();
    }

    public Course getCourse() {
        return course;
    }

    public int getNumberOfStudent() {
        return numberOfStudent;
    }

    public String getSectionID() {
        return sectionID;
    }

    public String getSectionName() {
        return sectionName;
    }

    public void setCourse(Course course) {
        this.course = course;
    }

    public void setNumberOfStudent(int numberOfStudent) {
        this.numberOfStudent = numberOfStudent;
    }

    public void setSectionID(String sectionID) {
        this.sectionID = sectionID;
    }

    public void setSectionName(String sectionName) {
        this.sectionName = sectionName;
    }
}
```

รูปที่ 20.15 (c) โปรแกรมภาษา JAVA ของคลาส Section

• Enrollment

```
package domain;

public class Enrollment {
    private Student student;
    private Section section;
    private String semester;

    public Section getSection() {
        return section;
    }

    public String getSemester() {
        return semester;
    }

    public Student getStudent() {
        return student;
    }

    public void setSection(Section section) {
        this.section = section;
    }

    public void setSemester(String semester) {
        this.semester = semester;
    }

    public void setStudent(Student student) {
        this.student = student;
    }
}
```

รูปที่ 20.15 (d) โปรแกรมภาษา JAVA ของคลาส Enrollment