Deliverable One: High Level Architecture

View/print courses on their roster

choose "schedule" to view a time sheet

View grades

View restrictions

Browse Courses

Register for Class

Add Class

Drop Course

Change Password

Check ID, password

Log In

Check Restriction

Register for lab

View/print course schedule and building/room assignments

View/print class rosters of registered students

Student

University Information System

<<Include>>

<<Include>>

<<Include>>

<<Include>>

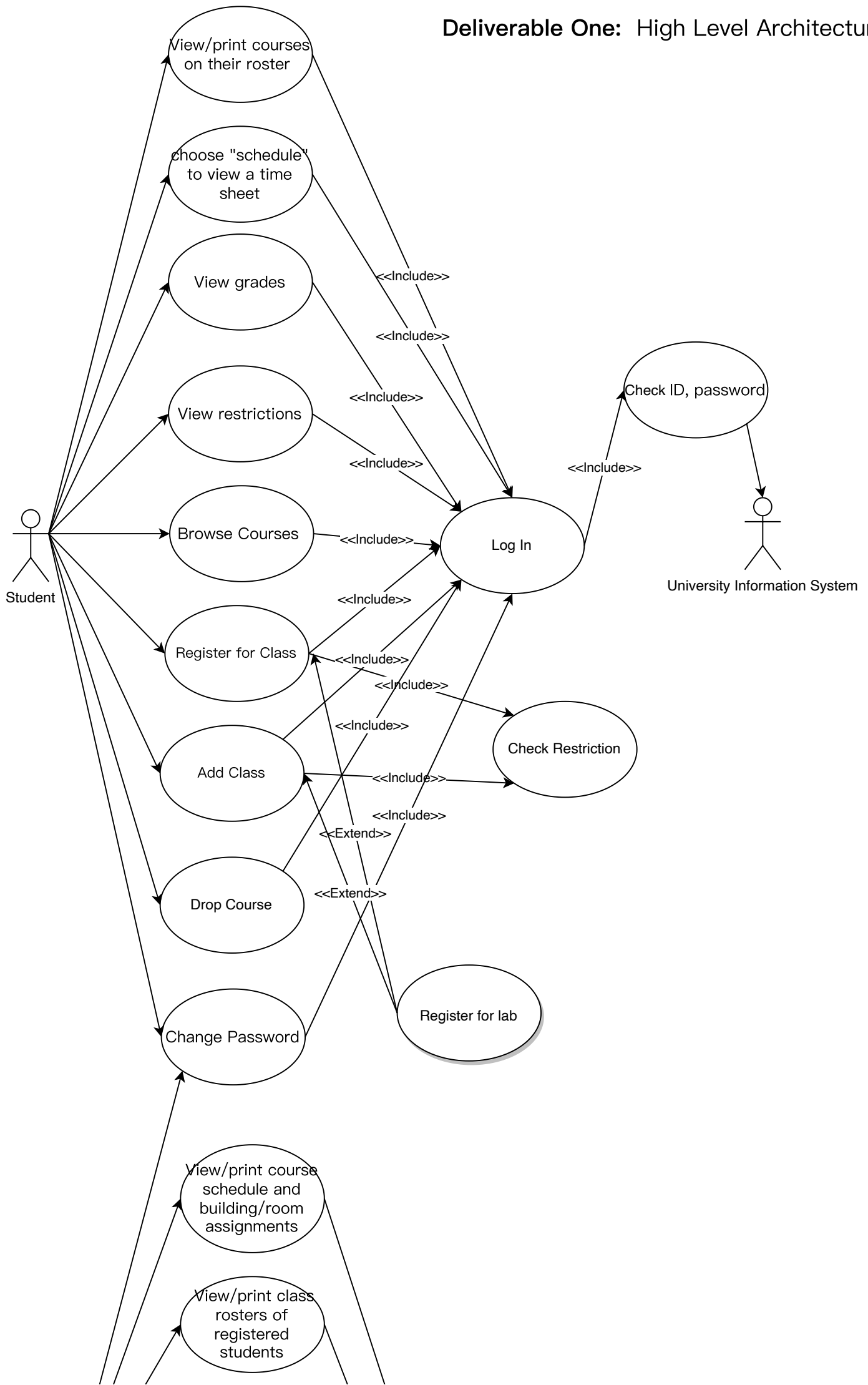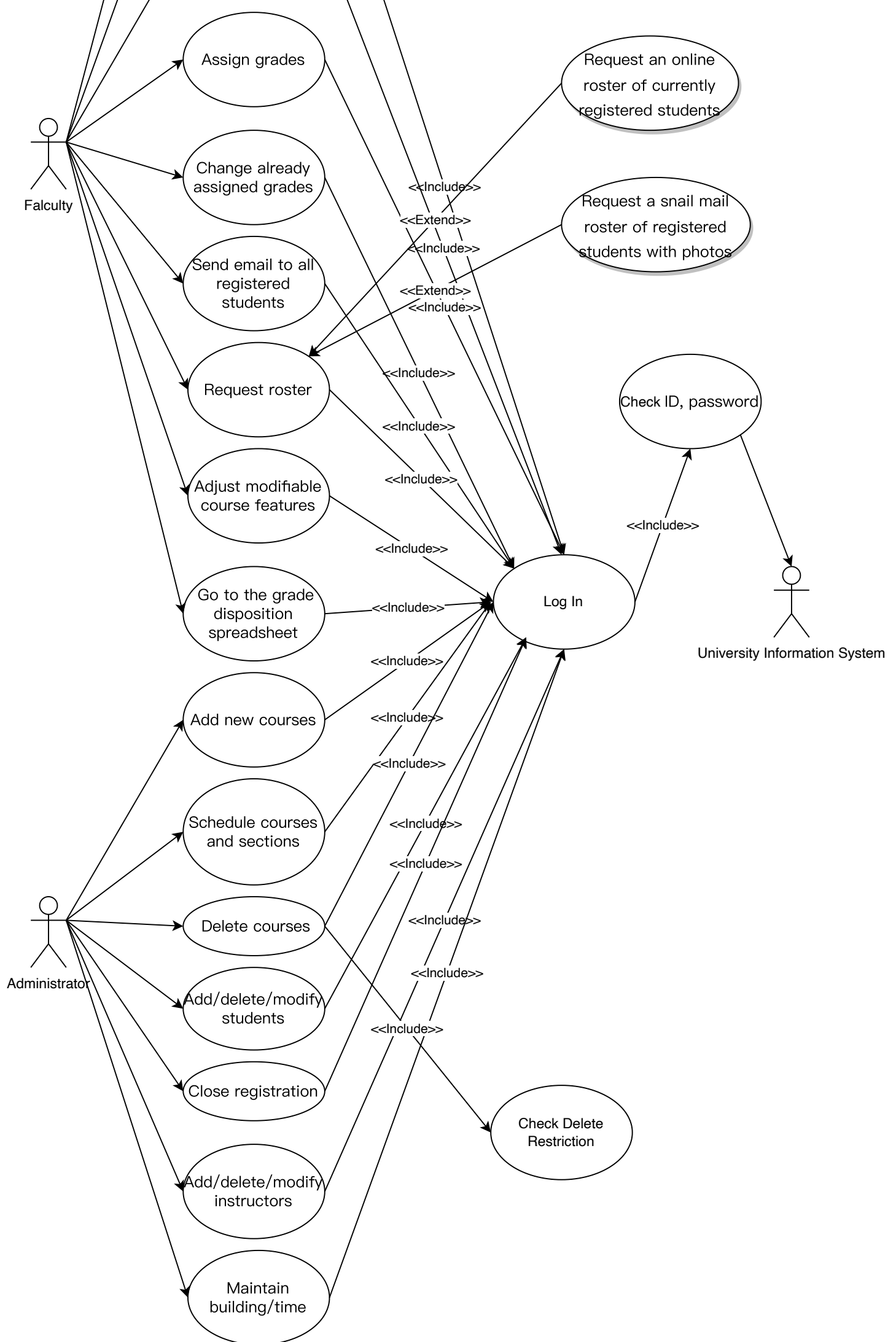<<Include>>

<<Include>>

<<Include>>

<<Include>>

<<Include>>

<<Include>>

<<Extend>>

<<Extend>>

Students register for courses:

Primary Flow: Student log in REGIE with their correct university ID with password successfully. Student meet all requirements of **register for courses** (e.g missing immunization records, failure to meet with academic advisor, failure to pay tuition bill, etc.) and they register successfully.

Alternative Flow: 1. Student enter wrong university ID or password, and can't log in REGIE. 2. Student doesn't meet requirements of **register for courses**. 3. Student log in **the system when the system is down for maintenance**

bounded contexts
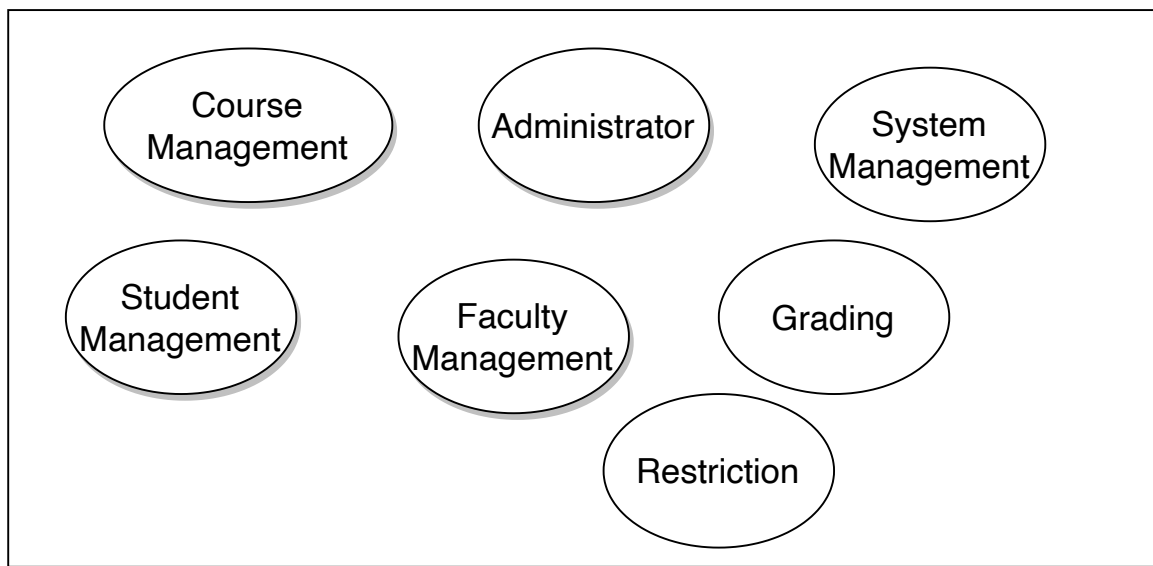
Deliverable Two: First pass analysis of the domain model within a single bounded context

Robustness diagram for Use case *student add a course* in Student Management bounded context

Student

Log in Page

Log in Check

Student Homepage

Course Display

Course display Page

Course

Selecte Course

Student

Restriction

Check Restriction

Get Result

Course Student

Result Page
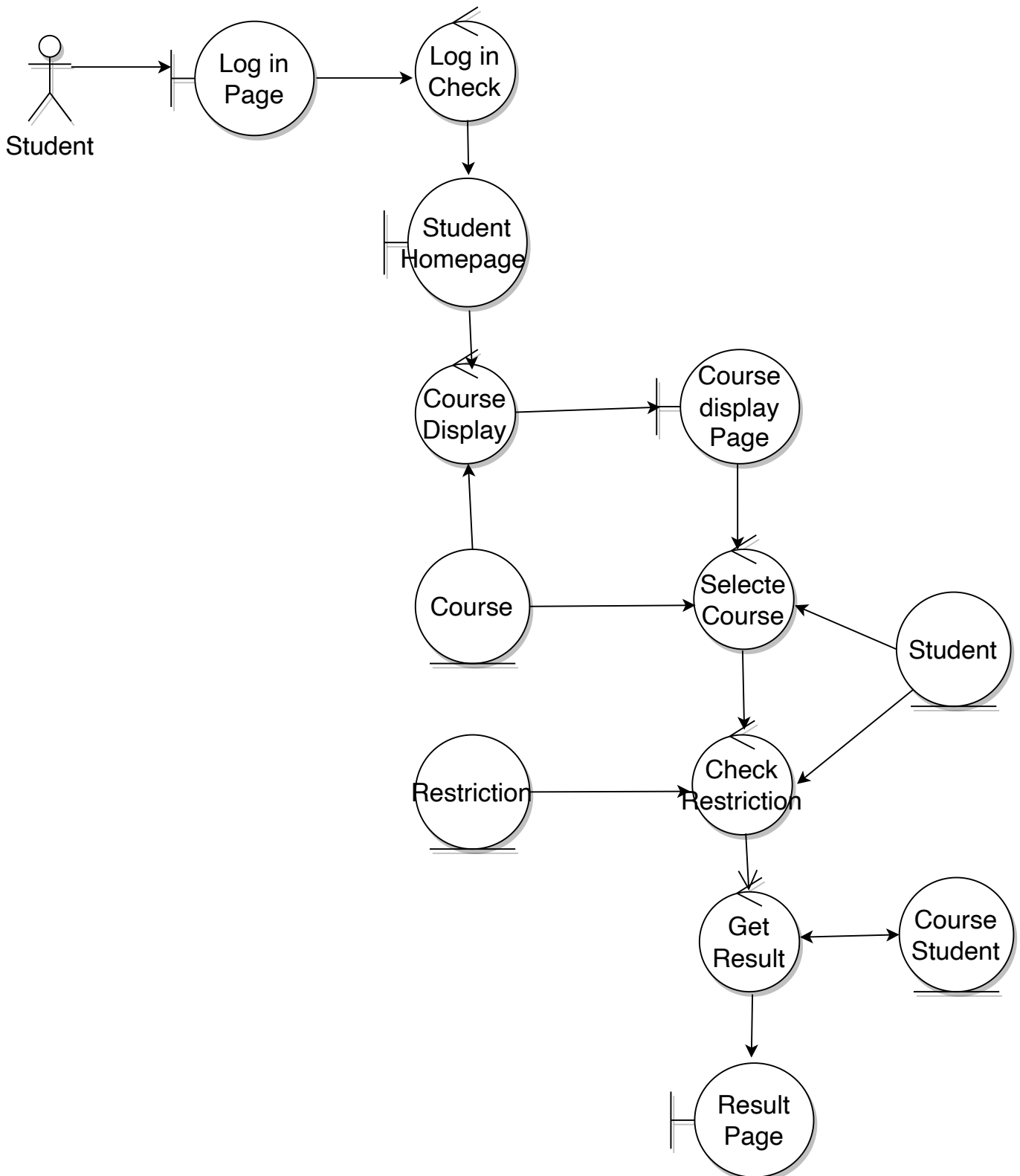
Deliverable Three

CRC Cards for Student Management bounded context

Student	
log in	
show student information	
view all course	
show course selected	Course
add course	StudentCourse
drop course	Restrction
view grade	
view restriction	
Change Password	

Course	
Display Course	
Get instructor	
Get Room	Building
Get time	
Get Prerequisite	

Restriction	
Show restriction	Course
Check restriction	Student

StudentCourse	
Get course Information	
Show status (in progress or completed)	Student
	Course
Show grade	

Transcript	
Get a student's transcript	StudentCourse

Building	
Get all available Building	Course
Get building for a course	

Schedule	
	Student
Show a student's schedule	Course
	StudentCour
	Building

LogIn	
Check ID	
Check Password	Student
Set password	

Sequence diagram for Student register for course

Student   Log in Page   Log in check   Course registration page   Restriction check   Database

login

check id and password

invalid

relogin

valid

check restriction

fail

modify dabase

result

result
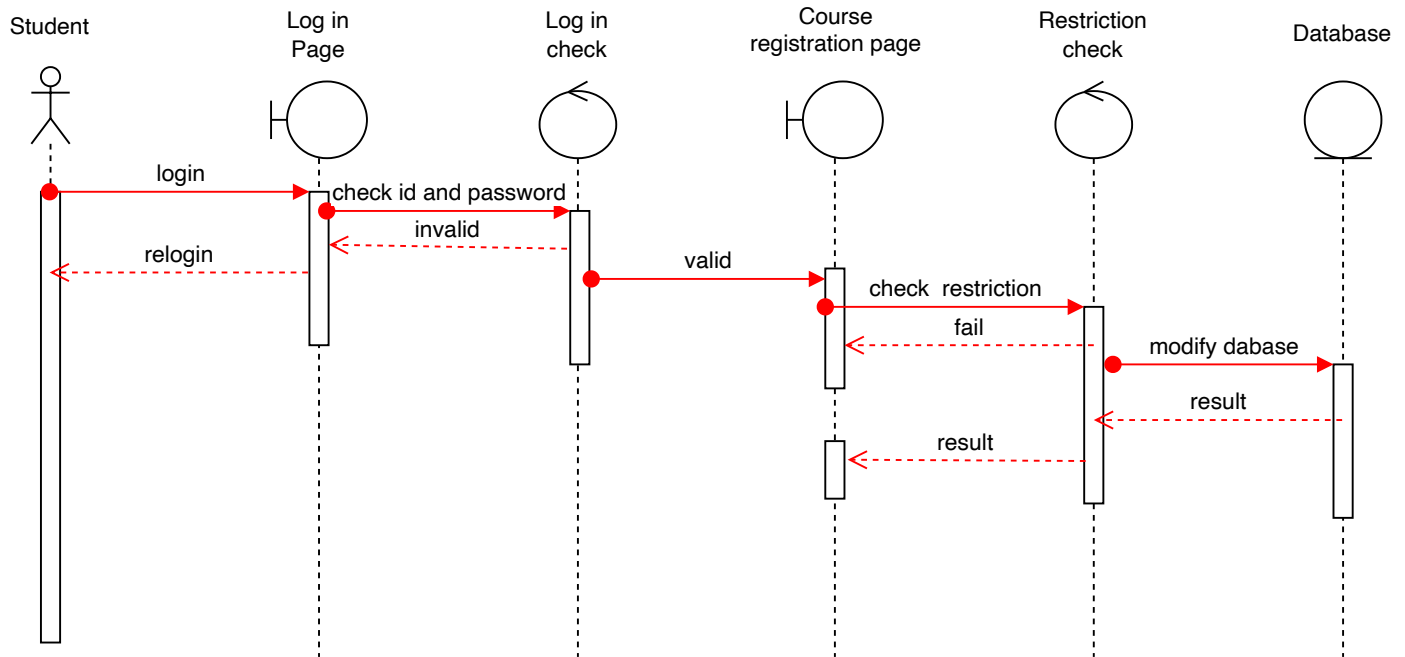
Activity diagram for Student

Log in → Check ID and Password → (correct) → Student Home page display

Check ID and Password → (incorrect) → Log in

Student Home page display → Choose function

Choose function branches to:
- Browse all course → All course display page → (end)
- View course → Student course display page → (end)
- View Schedule → Student schedule display page → (end)
- View grade → Student grade display page → (end)
- View transcript → Student transcript display page → (end)
- View Restriction → Student restriction display page → (end)
- Register for course → check restriction → (no) → Register for course; (yes) → modify database
- Drop course → modify database
- Change password → check old and new password → (no) → Change password; (yes) → modify database

modify database → (end)

UML for my design

«interface»
UserInterface

+ getCourseInfo(int): String
+ changePassword(String): boolean

allCourse: List<Course>

«interface»
RegisSystemInterface

+ changePassword(String): boolean
+ showCourseInfo(int): void
+ login(int, String): boolean
+ showAllCourses(): void

user: User

User

+ User(int, String, String, String):

- userId: int
- lastName: String
- firstName: String
- password: String

+ getCourseInfo(int): String
+ changePassword(String): boolean

password: String
lastName: String
firstName: String
userId: int
allCourse: List<Course>

«interface»
StudentInterface

+ getMyGrade(int): String
+ dropCourse(int): boolean
+ registerCourse(int): boolean

myCourse: List<Course>
myTranscript: List<String>
mySchedule: List<String>
myRestriction: List<Restriction>

«interface»
RegieStudentSystemInterface

+ showStudentRestriction(): void
+ showStudentGrade(int): void
+ showStudentSchedule(): void
+ registerStudentCourse(int): void
+ showStudentCourse(): void
+ showStudentTranscript(): void
+ dropStudentCourse(int): void
+ studentLogin(int, String): void
+ showStudentMenu(): void

RegieSystem

+ RegieSystem():

- user: User

+ login(int, String): boolean
+ showCourseInfo(int): void
+ showAllCourses(): void
+ changePassword(String): boolean

Student

+ Student(int, String, String, String):

+ initialize(): void
+ registerCourse(int): boolean
+ getMyGrade(int): String
+ dropCourse(int): boolean

myCourse: List<Course>
myTranscript: List<String>
mySchedule: List<String>
myRestriction: List<Restriction>

Instructor

+ Instructor(int, String, String, String):

RegieStudentSystem

+ RegieStudentSystem():

+ showStudentGrade(int): void
+ showStudentRestriction(): void
+ dropStudentCourse(int): void
+ showStudentCourse(): void
+ registerStudentCourse(int): void
+ studentLogin(int, String): void
+ showStudentTranscript(): void
+ showStudentMenu(): void
+ showStudentSchedule(): void

LoginEngine

+ LoginEngine():

+ login(int, String): User?

Course

+ Course(int, String, Instructor, Room, String, String, String):

- room: Room
- day: String
- time: String
- courseId: int
- courseName: String
- status: String
- instructor: Instructor

room: Room
instructor: Instructor
courseName: String
status: String
time: String
day: String
courseId: int

Factory

+ Factory():

+ createStudent(int, String, String, String): Student
+ createInstructor(int, String, String, String): Instructor
+ createBuilding(int, String, String, String): Building
+ createRestriction(int, String): Restriction
+ createRoom(int, String, Building, int): Room
+ createCourse(int, String, Instructor, Room, String, String, String): Course

---

Helper

+ Helper():

- getInstructorById(int, List<Instructor>): Instructor?
- getRoomById(int, List<Room>): Room?
- getBuildingById(int, List<Building>): Building?
+ getCourseById(int, List<Course>): Course?

roomList: List<Room>
buildingList: List<Building>
instructorList: List<Instructor>
courseList: List<Course>

Building

+ Building(int, String, String, String):

- buildingId: int
- address: String
- buildingName: String
- postCode: String

address: String
postCode: String
buildingName: String
buildingId: int

Room

+ Room(int, String, Building, int):

- roomBuilding: Building
- floor: int
- roomId: int
- roomName: String

roomBuilding: Building
roomName: String
floor: int
roomId: int

Restriction

+ Restriction(int, String):

- restrictionId: int
- restrictionName: String

restrictionName: String
restrictionId: int

RegieStudentSystemRunEngine

+ RegieStudentSystemRunEngine():

+ run(): void

Main

+ Main():

+ main(String[]): void

Extends
Use