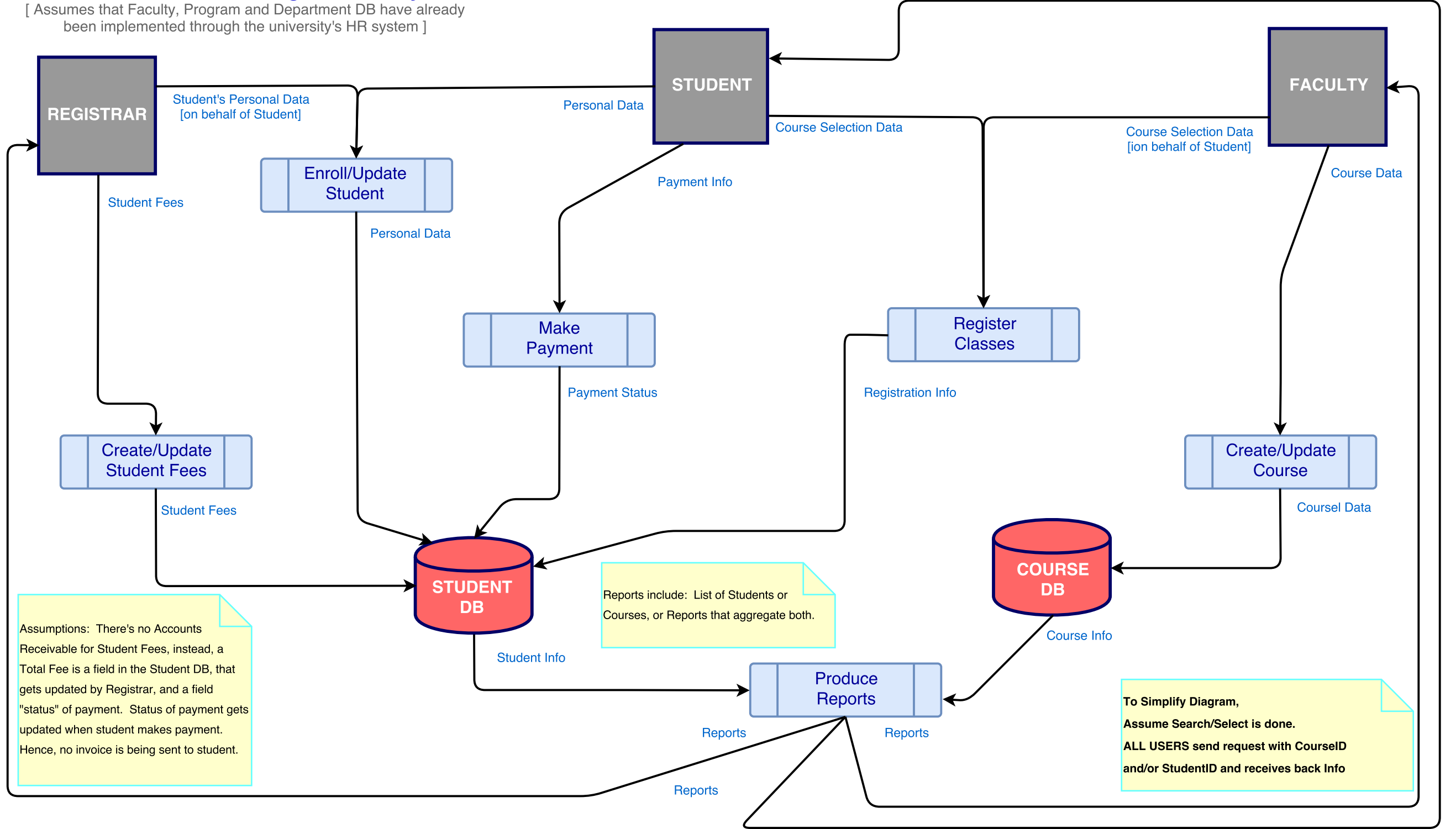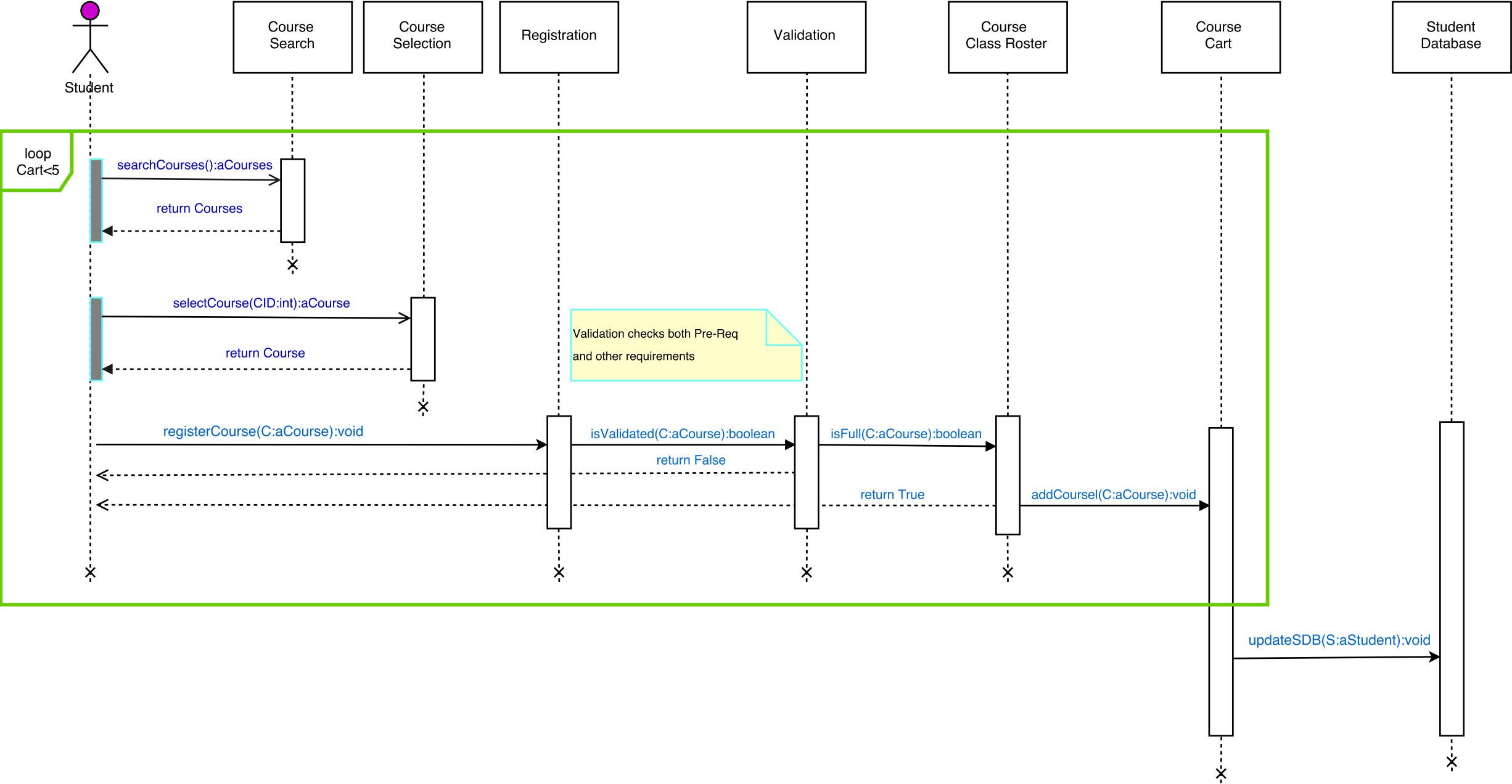# DATA FLOW DIAGRAM: Registration System
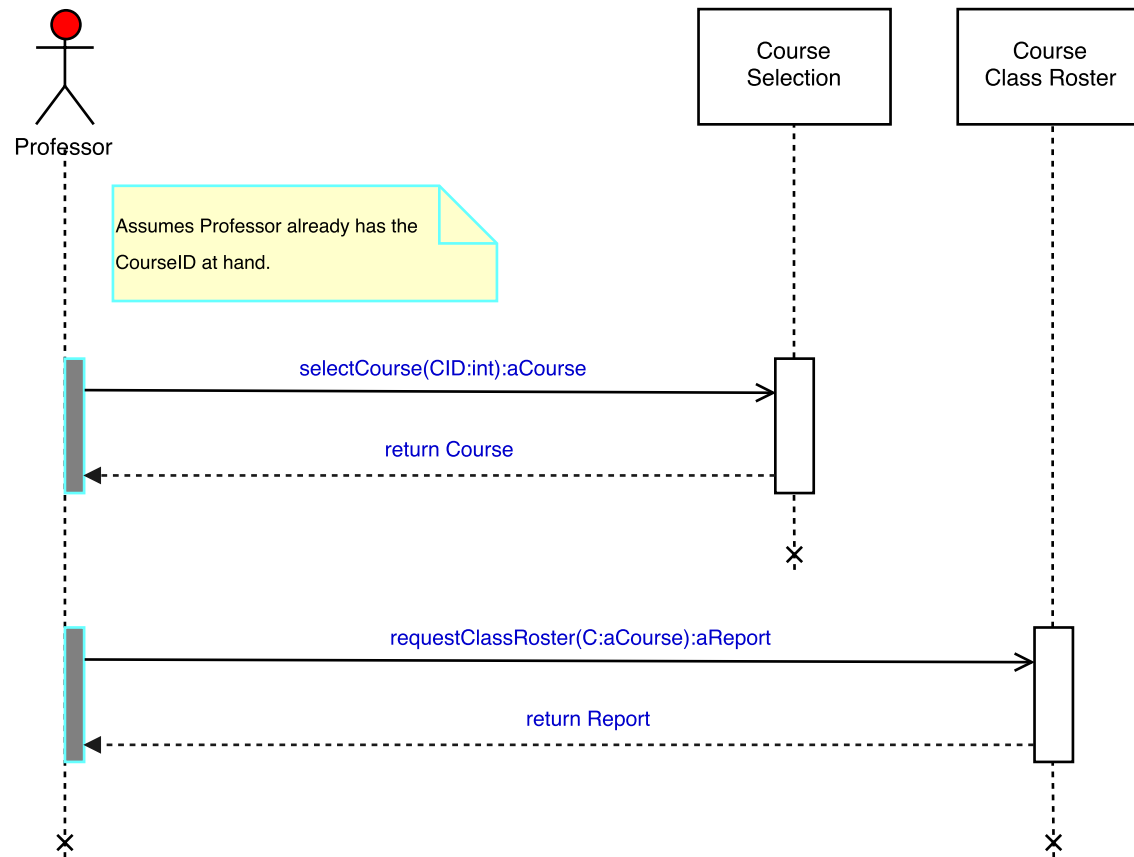
[Assumes that Faculty, Program and Department DB have already
been implemented through the university's HR system]

REGISTRAR

STUDENT

FACULTY

Student's Personal Data
[on behalf of Student]

Personal Data

Course Selection Data

Course Selection Data
[ion behalf of Student]

Course Data

Student Fees

Enroll/Update
Student

Personal Data

Payment Info

Make
Payment

Register
Classes

Create/Update
Course

Payment Status

Registration Info

Coursel Data

Create/Update
Student Fees

Student Fees

STUDENT
DB

Reports include: List of Students or
Courses, or Reports that aggregate both.

COURSE
DB

Course Info

Assumptions: There's no Accounts
Receivable for Student Fees, instead, a
Total Fee is a field in the Student DB, that
gets updated by Registrar, and a field
"status" of payment. Status of payment gets
updated when student makes payment.
Hence, no invoice is being sent to student.

Student Info

Produce
Reports

Reports

Reports

Reports

**To Simplify Diagram,
Assume Search/Select is done.
ALL USERS send request with CourseID
and/or StudentID and receives back Info**

SEQUENCE DIAGRAM: STUDENT REGISTERING COURSES

SEQUENCE DIAGRAM: PROFESSOR VIEWING CLASS ROSTER

Professor

Course
Selection

Course
Class Roster

Assumes Professor already has the
CourseID at hand.

selectCourse(CID:int):aCourse

return Course

requestClassRoster(C:aCourse):aReport

return Report

implementation: ALL COURSE
FUNCTIONS directly access Course DB

SEQUENCE DIAGRAM: REGISTRAR UPDATING COURSE

Registrar

Course Selection

Student Selection

Course Database

Course Requirements

Student Database

Assumes Registar already has
the CourseID and StudentID at hand.

selectCourse(CID:int):aCourse

return Course

selectStudent(SID:int):aStudent

return Student

Assumes that there is a
Student/Course relationship and that it can be
accessed thru Course DB by Course Info and
can remove Student here.

The assignment suggest that this can be done
through Course.

Alternately, a course can be removed from
Student DB.

removeStudentFromCourse(C:aCourse, S:aStudent):void

updateSDB(S:aStudent):void

loop
till exit

updateCoursePre-Requisites(C:aCourse, W:aWaiver):void

swapProfessorsForCourse(ProfA:aProf, ProfB:aProf):void

updateCourseDB(C:aCourse):void

implementation: ALL COURSE
FUNCTIONS directly access Course DB