I’d like to build the employee class, the log in system and hook both up to a database since I feel I can reuse a great deal of the code I had for the banking app. For that reason I’d like to take on the log in system as a natural extension of those two.

I’ll use an Enum to declare an employee type unless I need some extra functionality from them.

Public class Employee

Subclass BENEFITS\_COORDINATOR

//variables

Long totalRemainingTuition; //will reset this to 1000 each year

EnumEmployeeType employeeType;

Employee directSupervisor;

//check from on database on load

//if no subordinates don’t load those things

List<Employee> subordinates;

Can department head approve themselves or not?

Benefits Coordinater approves department head

BC can have any including themselves approve.

//login

//view accounts (Personal and For Those Responsible) <- Home page

//Possibly tab this down

//approving reimbersemnt subordinates

///tuition reimbursement Form

// Grade/Presentation Upload

//approving Grad/presentation upload

//benefits coordinator special functionality

login, home, view all accounts, view/approve request details, request, document upload, approve grade, benCo page

Going to be working Quisi independently for one week until the 7th. We’ll consider these two models and Combine the best elements of each.

Then we’ll combine them next week on the 8th-10th debugging features

11th through 13th we’ll split what’s left and assign those items.

//employee tables

Id

Firstname

Lastname

Supervisor id

Department id

//department table

Id

Name

Manager