Assignment #4

Practice with System Design

CS 4354 Fall 2012
Instructor: Jill Seaman

Due: in class **Monday, 11/12/2012**
Submit a "hard copy" (hand-written, or computer-generated, or hybrid).

Problems over Chapters 6 and 7:

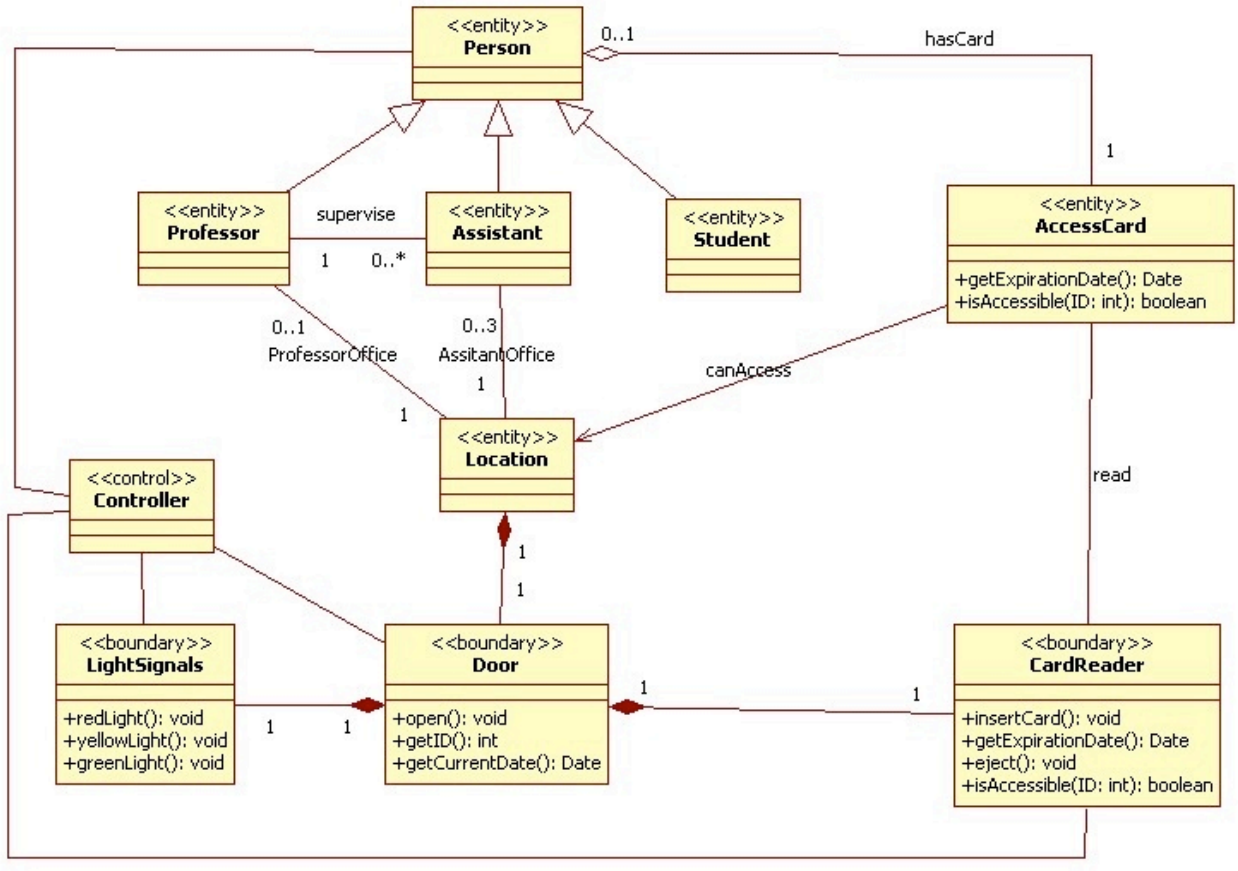
1 Given the following nonfunctional requirements for a system,
(A) identify (list) a set of design goals (see section 6.4.2). For each design goal,
(B) indicate which of the following category it belongs to: **Performance**, **Dependability**, **Cost**, **Maintenance**, or **End user** (note:the End user category was left out of the notes, but it is in the book).

These requirements are from a Postage-printing website application.

1. If the Postage Printing service goes down, it will be restored within one hour.
2. Users will be able to access only their own personal information and not that of other users.
3. The system should be able to accommodate new postage prices and new forms of package types (i.e. various flat rate boxes) and mail classes (first class mail, priority mail, etc).
4. [Not a non-functional requirement, but from the contract] The administration of the system must be kept under $10,000 per month.
5. System login/logout shall take less than 5 seconds.
   Print Postage function shall be processed within 10 seconds.
   Other requests shall be processed within 5 seconds.
6. The user interface must prevent the user from selecting an invalid combination of package type and mail class, and must prevent the user from selecting an international mail class for a domestic address (and vice versa).

2 Given the following object model (see next page), decompose it into a package model with at least 2 (but no more than 4) subsystems (Chapter 7, slide 39, Figure 6-29 in the book). Use the goals of low coupling and high cohesion (and any other heuristics that may apply) for deciding which objects should go in each subsystem.

You may print the page and draw on top of the diagram and submit that with the rest of your answers.



3 Draw a UML deployment diagram (Chapter 7 slide 6, Figure 7-2 in the book) representing the hardware/software mapping for an older version of Apple's iTunes software. The system includes a Web server (the iTunes Store) and a (separate) database server that contains all the music files (the servers are on Mac Pro machines). The user's Macintosh computer has an iTunes client app that accesses the web server, and downloads songs to its library. The user's iPhone (or iPod) connects to the Mac using a cable, and the iTunes app on the Mac syncs the iTunes library on the iPhone/ iPod with the one on the Mac.

4 Consider the iTunes system from question 3. The analysis object model (class diagram) contains Song objects. These objects are persisted separately by all three major components (the iTunesStore, the iTunesMacClient and the iTunesIPodClient). Draw a complete subsystem decomposition diagram (chapter 7 slide 10, Figure 7-6 in the book) for the iTunes system that includes components for the storage systems. For each storage system, indicate whether it should be implemented using flat files or a relational database.

5 Design the access control policies for the iTunes store. Customers access the store via the Web to browse songs (to get info and prices), input their address and payment

information (create and edit their information), and purchase songs (place orders). Suppliers (artists and music companies) can add new songs, update song information, and view orders. The store owner (Apple) sets the song prices, examines the customer accounts (to makes tailored offers based on their purchasing profiles), and fills orders. You have to deal with four actors: StoreAdministrator, Supplier, UnregisteredCustomers and Customers. Customers can create their own accounts via the Web, whereas Supplier accounts are created by theStoreAdministrator. Use these four objects: Song, CustomerAccount, SupplierAccount, and Order. Use these operations: create(), getInfo() getPrice(), updateInfo(), updatePrice(), view(), and fill(). Some operations apply to only one kind of object, some apply to all.

6 Which global control flow mechanism (chapter 7 slide 16) would you use in the following cases?

- A Web server (like the iTunes store) designed to sustain high loads: many people trying to access the server at once.
- The graphical user interface for the iTunes application on either the computer or iPhone/iPod.
- A compiler like g++ or javac, run from the command line, intended to process the source code in a sequence of non-overlapping steps.

7 When identifying boundary conditions (Chapter 17 slides 17-18, section 7.4.6) the main activity is adding new use cases. For example, for the iTunes application we would add use cases to startup the iTunes Store web server and to configure the database that contains the song information. Why are use cases that describe boundary conditions described during system design (as opposed to during requirements elicitation or analysis)?