Recap on last lecture

Detailed Design

Detailed Design Outline:

I. Software Design and Implementation
II. Design Processes
    1.Functional Decomposition
    2.Relational Database Design
    3.Object-oriented Design and UML
III. Design Characteristics and Metrics
    1.McCabe's Cyclomatic Complexity
    2.Coupling and Cohesion

Nice, but how do I get started – step by step ?

Task Set for Software Design

1) Examine the information domain model and design appropriate data structures for data objects and their attributes
2) Using the analysis model, select an architectural style (and design patterns) that are appropriate for the software
3) Partition the analysis model into design subsystems and allocate these subsystems within the architecture
   a) Design the subsystem interfaces
   b) Allocate analysis classes or functions to each subsystem
4) Create a set of design classes or components
   a) Translate each analysis class description into a design class
   b) Check each design class against design criteria; consider inheritance issues
   c) Define methods associated with each design class
   d) Evaluate and select design patterns for a design class or subsystem

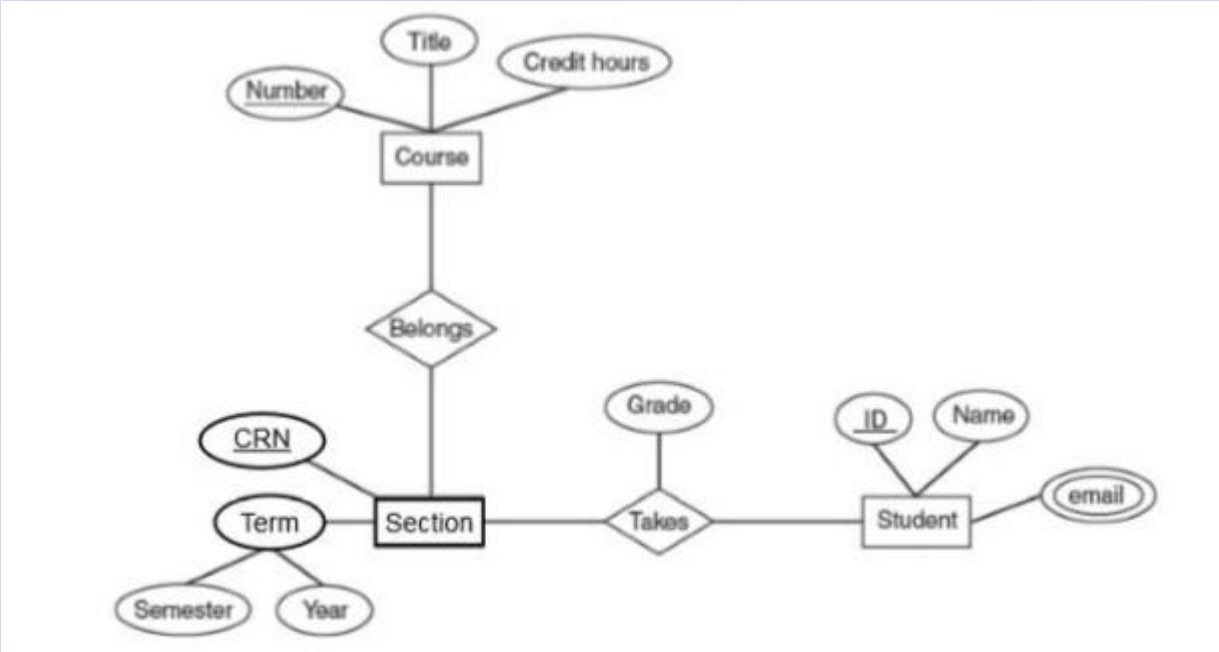
Task Set for Software Design (continued)

5) Design any interface required with external systems or devices
6) Design the user interface
7) Conduct component-level design
   a) Specify all algorithms at a relatively low level of abstraction
   b) Refine the interface of each component
   c) Define component-level data structures
   d) Review each component and correct all errors uncovered
8) Develop a deployment model
   - Show a physical layout of the system, revealing which components will be located where in the physical computing environment

Task Set for Software Design

1) Examine the information domain model and design appropriate data structures for data objects and their attributes

- Data design translates data objects defined as part of the analysis model into
  - Data structures at the software component level
  - A possible database architecture at the application level



Task Set for Software Design

2) Using the analysis model, select an architectural style (and design patterns) that are appropriate for the software

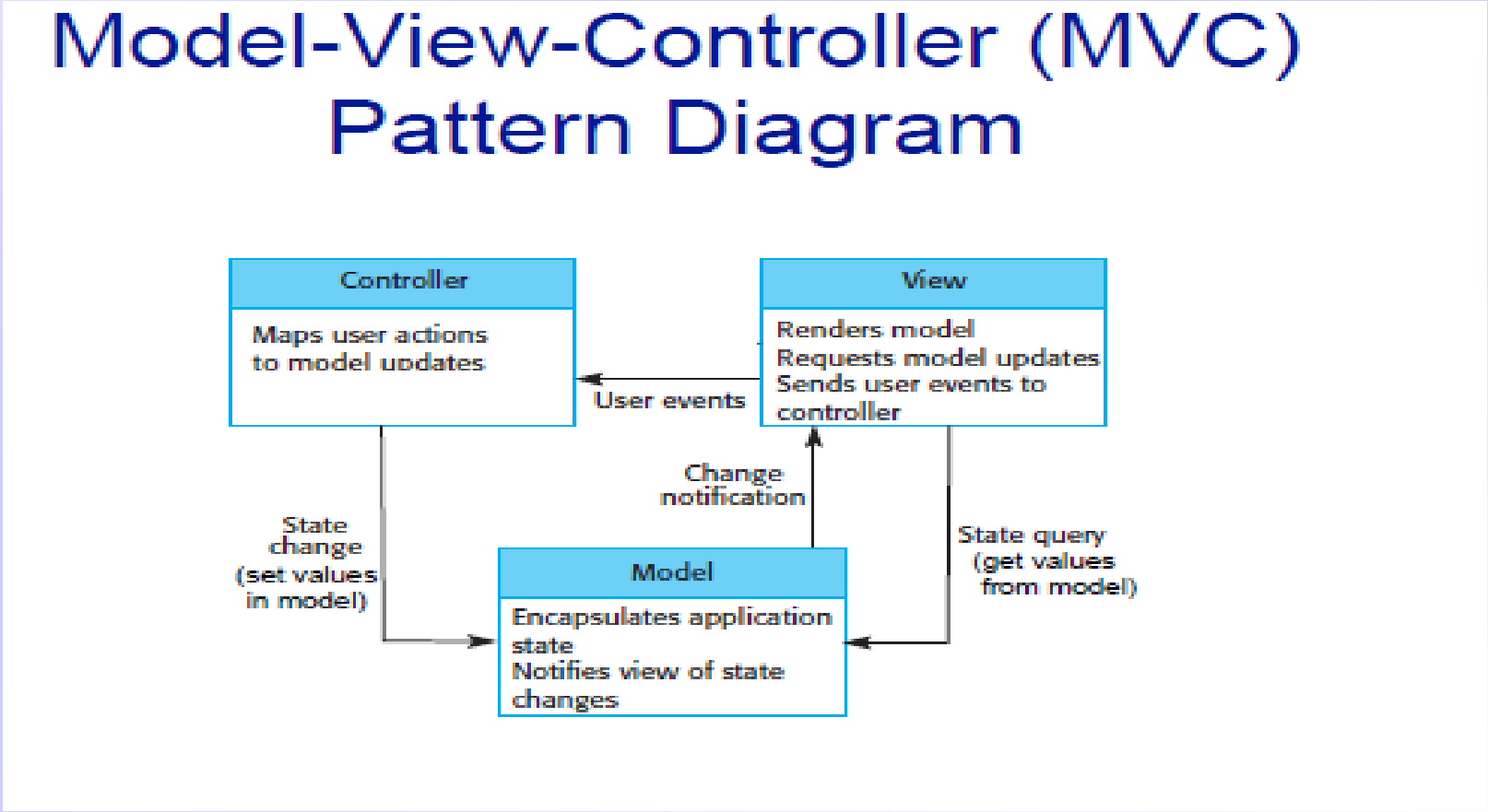
Consider using Model View Controller pattern:

1. Model-View-Controller (MVC) Pattern

- Commonly used in desktop, mobile phone, and web applications.
- Used to separate the data (the model) from the way it is presented to the user (the views)
- Model objects encapsulate the data.
- View objects present data to and receive actions from the user.
- Controller Responds to user actions (from View) by updating Model (and View).

Task Set for Software Design

2) Using the analysis model, select an architectural style (and design patterns) that are appropriate for the software

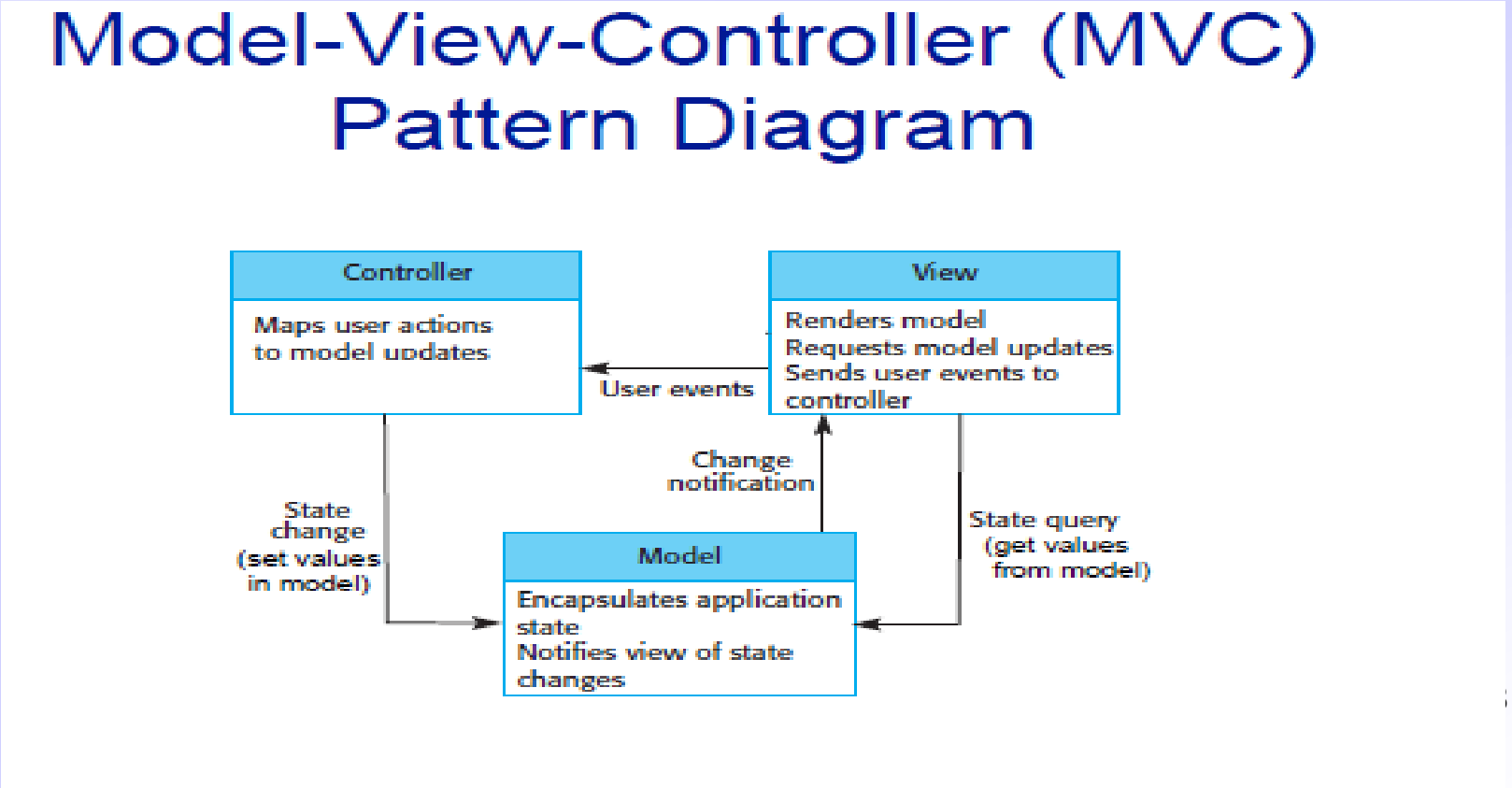


Task Set for Software Design

3) Partition the analysis model into design subsystems and allocate these subsystems within the architecture

a) Design the subsystem interfaces

b) Allocate analysis classes or functions to each subsystem



Task Set for Software Design

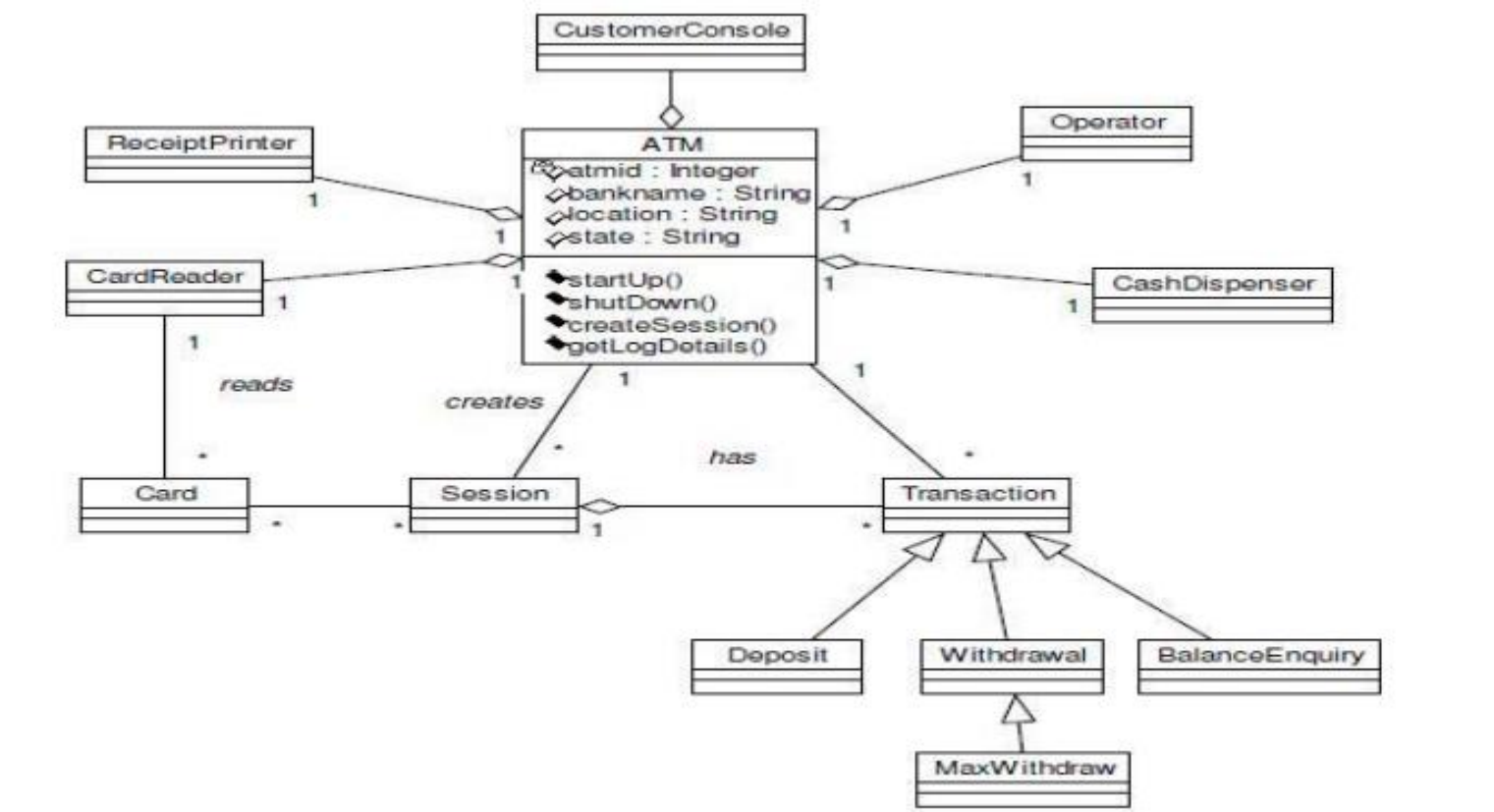
4) Create a set of design classes or components (see last week class base modelling)

a) Translate each analysis class description into a design class

b) Check each design class against design criteria; consider inheritance issues

c) Define methods associated with each design class

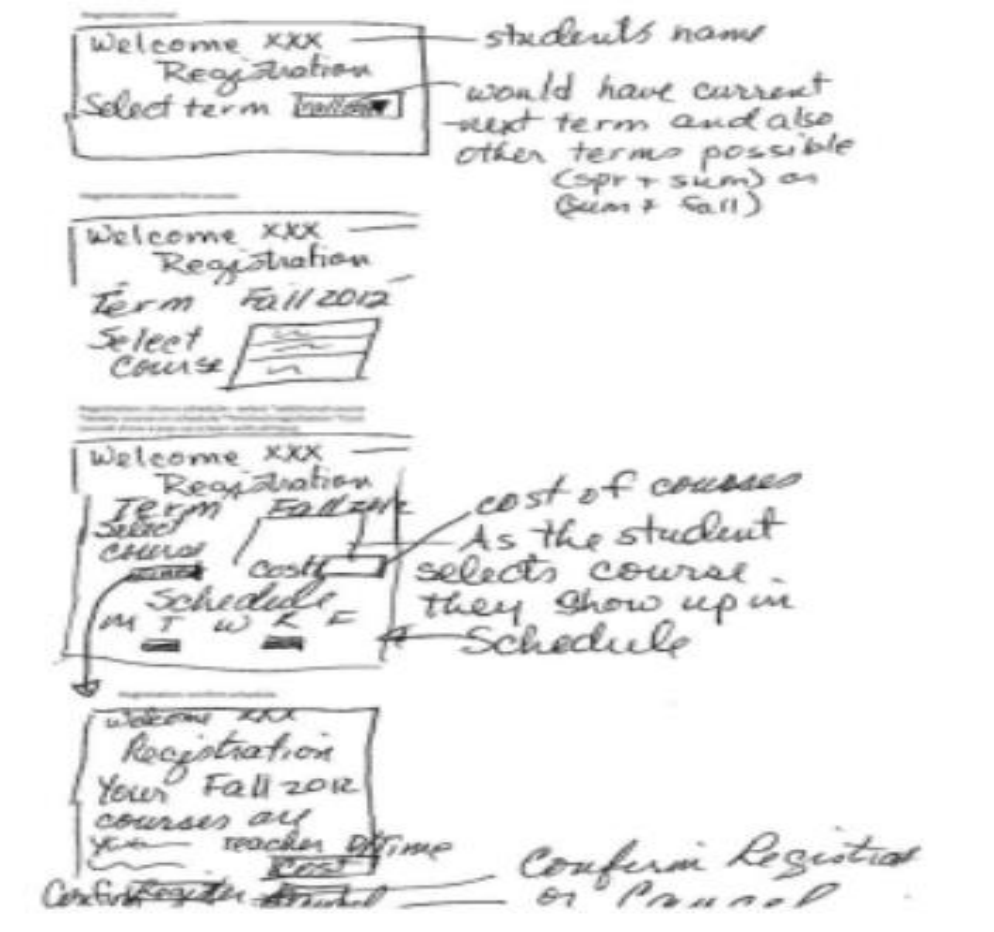
d) Evaluate and select design patterns for a design class or subsystem



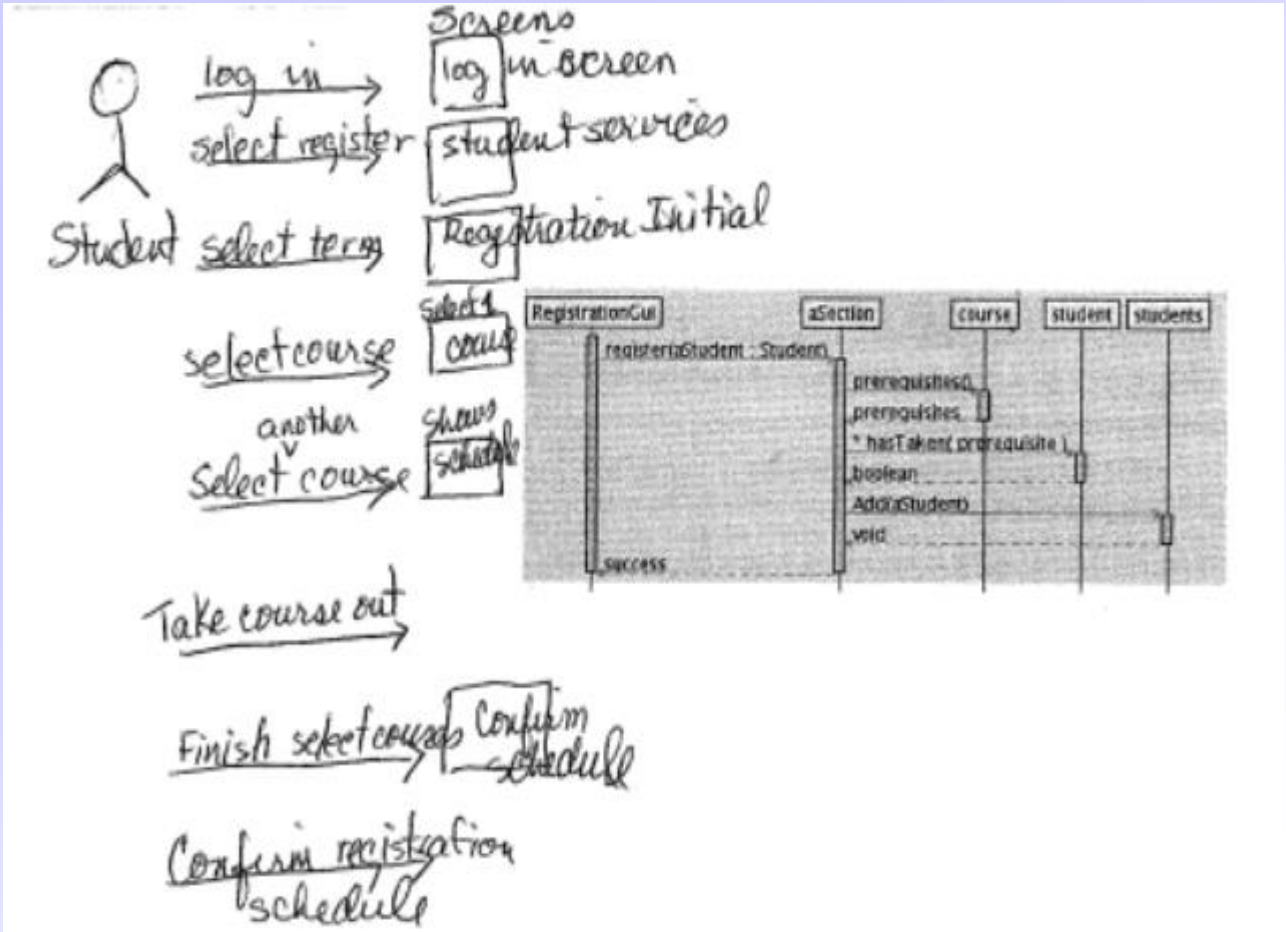
Task Set for Software Design

5) Design any interface required with external systems or devices
6) Design the user interface

Prototype screens



User Interaction added to sequence diagram



Task Set for Software Design

7) Conduct component-level design
   a) Specify all algorithms at a relatively low level of abstraction
   b) Refine the interface of each component
   c) Define component-level data structures
   d) Review each component and correct all errors uncovered

Fine-tuning lower level class/methods: interface, data structure, special algorithm needs to be specified.

Design Deliverables
(Date to be announced in SacCT)

- Database design diagram (ERD), tables where applicable
- Architectural diagram with all interfaces specified.
- Class diagram with complete class names, attributes, methods, and relations.
- Prototype screens, user interactions vs. sequence diagrams
- Any external interface specification where applicable