


University of Southern California  
Center for Software Engineering

---

# Object-Oriented Analysis & Design (OOAD) II Use-Case Driven Design CS577a, Fall 2006

Developed by  
David Klappholz, Stevens Institute of Technology  
&  
Ed Colbert, USC Center for Software Engineering

Introduction to Use-Case Driven Design, CS577a, Fall 2006 1 10/5/2006




University of Southern California  
Center for Software Engineering

---

## Goal of Lecture

- Complete the discussion of the “simple version” of OOA&D that we started discussing in OOA&D Lecture 1
- You should understand the simple OOA&D process
  - Lecture 1 emphasized *object-oriented analysis*
  - This lecture will emphasize *object-oriented design* using a *use-case driven approach*

Introduction to Use-Case Driven Design, CS577a, Fall 2006 2 10/5/2006




University of Southern California  
Center for Software Engineering

---

## Pre-Class Exercise

- ❑ This lecture has a two-part pre-class exercise
  - You’ll find later in these slides.
- ❑ The exercise relies on a relatively solid understanding of the material in the lecture, so read the lecture carefully
  - More than once if necessary
  - Before you do the exercise

Introduction to Use-Case Driven Design, CS577a, Fall 2006 3 10/5/2006




University of Southern California  
Center for Software Engineering

---

## “Simple Version” OOA&D

- ❑ Technically
  - Analysis means “figuring out what we want to accomplish”
  - Design means “figuring out how we want to accomplish it.”
  - We’ll get further into the distinction in OOA&D Lecture 3
- ❑ Analysis consists of
  - Defining a simple domain model
    - Creating a class model to capture the information in the domain that the system needs to know
    - Creating a use-case model to capture the behavior in the domain that the system needs to support
  - If we want to be really precise, we did some “design” in lecture 1, since our use-case descriptions refer to specific implementation (GUI) features

Introduction to Use-Case Driven Design, CS577a, Fall 2006 4 10/5/2006




University of Southern California  
Center for Software Engineering

---

## “Simple Version” OOA&D (cont.)

- ❑ In OOA&D Lecture 3, we’ll go into further detail on why our current approach, which is OK for simple projects with very low-level L.O.S. (quality) requirements, isn’t an approach that can be used for more complex projects
  - e.g.:
    - Most embedded software projects
    - All safety-critical and life-critical software projects
    - Even an Internet Bookstore if it’s to be on the scale of, say, Amazon, which:
      - Must run on a large server farm
      - Must sustain very high transaction rates

Introduction to Use-Case Driven Design, CS577a, Fall 2006 5 10/5/2006




University of Southern California  
Center for Software Engineering

---

## “Simple Version” OOA&D (cont.)

- ❑ In this lecture, we’ll look at the simple *object-oriented design process*
- ❑ We’ll
  - For each use-case, describe how a set of *objects interact* to implement the behavior defined textually for that use-case
    - Each use-case will be described by 1 or more Sequence Diagram
  - Refine our class model based on what we’ve learned by describing each use-case

Introduction to Use-Case Driven Design, CS577a, Fall 2006 6 10/5/2006




University of Southern California  
Center for Software Engineering

---

## Use-Case Driven Design

- ❑ The purpose of Use-Case Driven Design is to discover the necessary:
  - *Interface Objects* & their classes  
e.g. pages, dialog boxes, representations of entities
  - *Entity Objects* & their classes
    - Both temporary & persistent,  
e.g., data structures, DB tables, etc.
  - *Control Objects* & their classes
    - We'll see examples on later slides
      - You know what an *Interface Class* is & what an *Entity Class* is, so don't worry if you don't yet know what a Control Class is (you'll find out soon)
  - The *messages* they exchange

Introduction to Use-Case Driven Design, CS577a, Fall 2006 7 10/5/2006




University of Southern California  
Center for Software Engineering

---

## Use-Case Driven Design (cont.)

- ❑ Use-Case Driven Design is a process that is performed one Use-Case at a time
  - Its result is one *Collaboration Diagram* per Use-Case
- ❑ The first Use-Case for which we'll do Use-Case Driven Design is the Login Use-Case
- ❑ The text of the Login Use-Case is repeated on the next two slides
- ❑ The slide after that contains the Login Collaboration Diagram, some of whose symbols will likely be unfamiliar to you, but most of which you'll be able to figure out if you spend a bit of time looking at the diagram
- ❑ Explanations of all the new symbols are given on slides immediately following the diagram itself

Introduction to Use-Case Driven Design, CS577a, Fall 2006 8 10/5/2006



University of Southern California  
Center for Software Engineering


---

## Login Use-Case

□ Basic Course

- The Customer clicks the Login button on the Home Page. The system displays the Login Page. The Customer enters his or her user ID and password and then clicks the Login button. The system validates the login information against the persistent Account data and then returns the Customer to the Home Page.

Introduction to Use-Case Driven Design, CS577a, Fall 2006 9 10/5/2006



University of Southern California  
Center for Software Engineering

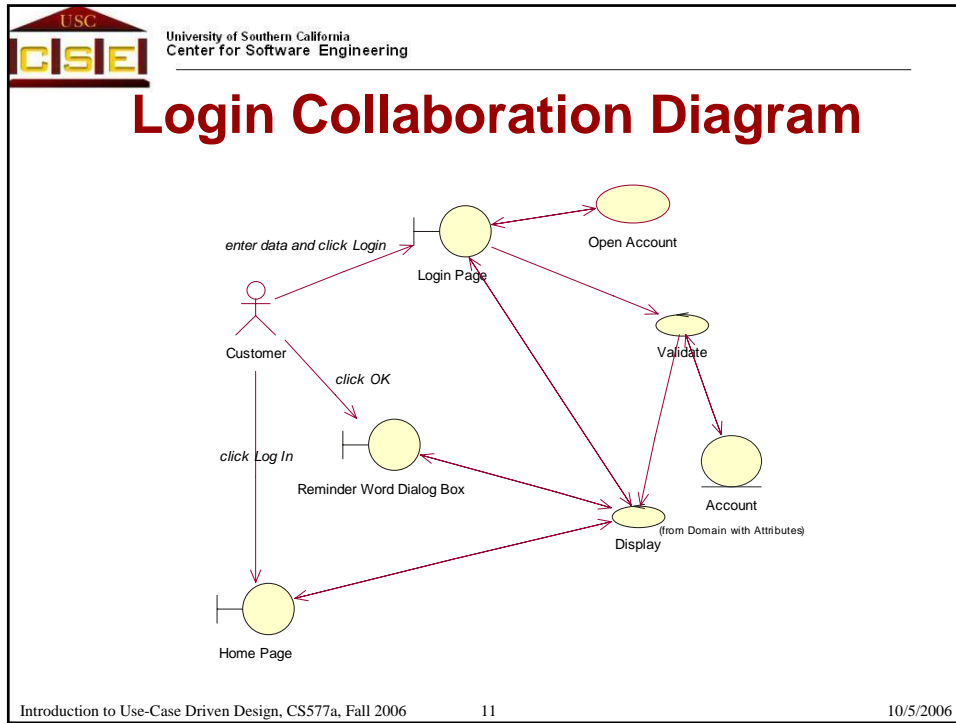
---

## Login Use-Case

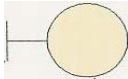
□ Alternate Courses

- If the Customer clicks the New Account button on the Login Page, the system invokes the Open Account Use-Case.
- If the Customer clicks the Reminder Word button on the Login Page, the system displays the reminder word stored for that Customer, in a separate dialog box. When the Customer clicks the OK button, the system returns the Customer to the Login Page.
- If the Customer enters a user ID that the system does not recognize, the system displays a message to that effect and prompts the Customer to either enter a different ID or click the New Account button.
- If the Customer enters an incorrect password, the system displays a message to that effect and prompts the Customer to reenter his or her password.
- If the Customer enters an incorrect password three times, the system displays a page telling the Customer that he or she should contact customer service, and also freezes the Login Page.

Introduction to Use-Case Driven Design, CS577a, Fall 2006 10 10/5/2006

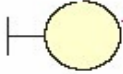


## New Symbols: Interface Class

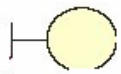
The symbol  represents an "Interface Class."

It's object interact with an actor(s)

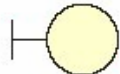
Some examples from the Login Collaboration Diagram are:




Home Page



Login Page



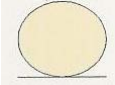
Reminder Word Dialog Box




University of Southern California  
Center for Software Engineering

---

## New Symbols: Entity Class


The symbol  represents an “Entity Class.”

The only example from the Login Collaboration Diagram is:



which represents customer account data that the Login Use-Case needs to refer to.


Introduction to Use-Case Driven Design, CS577a, Fall 2006
13
10/5/2006




University of Southern California  
Center for Software Engineering

---


## New Symbols: Control Class

The standard symbol for a “Control Class” is:  i.e., a circle with a left-pointing arrow head on top. We will see instances of this symbol, later, in the robustness diagram for the Open New Account Use-Case. An alternative symbol for a “Control Class,” and the one used in the Login Use-Case Collaboration Diagram is:



The small arrow head on the oval is a bit hard to see, but it’s there – pointed to by the red arrow.

Introduction to Use-Case Driven Design, CS577a, Fall 2006
14
10/5/2006




University of Southern California  
Center for Software Engineering

---

## New Symbols: Control Class (cont.)

The two instances of Control Classes in the Login Use-Case Collaboration Diagram are named “Display” and “Validate.”

Introduction to Use-Case Driven Design, CS577a, Fall 2006 15 10/5/2006




University of Southern California  
Center for Software Engineering

---

## Constructing the Collaboration Diagram

- Remember that there will be one Collaboration Diagram for each Use-Case.
- That said, the Collaboration Diagram for a Use-Case is created (sort of) sentence-by-sentence from the text of the Use-Case, starting with the text of the basic course.
- What we mean by “sort of” is illustrated, on the next slide, by the break-down of Login’s basic course into “(sort of) sentences.”

Introduction to Use-Case Driven Design, CS577a, Fall 2006 16 10/5/2006




University of Southern California  
Center for Software Engineering

## Constructing the Collaboration Diagram (cont.)

The Customer clicks the Login button on the Home Page. The system displays the Login Page. The Customer enters his or her user ID and password and then clicks the Login button. The system validates the login information against the persistent Account data and then returns the Customer to the Home Page.

*That is, for the purpose of constructing a Collaboration Diagram, a sentence with no “and” or “or” is treated as an individual unit, but a sentence with an “and” or an “or” is broken into its constituent parts. NOTE THAT THIS MEANS THAT Use-Case TEXT MUST BE WRITTEN AS COHERENT, UNAMBIGUOUS, SENTENCES!!!*

Introduction to Use-Case Driven Design, CS577a, Fall 2006 17 10/5/2006




University of Southern California  
Center for Software Engineering

## Constructing the Collaboration Diagram (cont.)

- If you look carefully at Login’s basic course’s decomposition, on the previous slide, into five components or “sentences,” you’ll find two apparent violations of the “break at an ‘and’ or an ‘or’ “ rule
- See if you can find the two apparent violations before you go to the next slide

Introduction to Use-Case Driven Design, CS577a, Fall 2006 18 10/5/2006




University of Southern California  
Center for Software Engineering

## Constructing the Collaboration Diagram (cont.)

- ❑ The two apparent violations are the two “and”s and one “or” in the following “sentence” that didn’t cause us to break it up into separate parts as we said we’d do when we saw instances of “and” and “or”: **The Customer enters his or her user ID and password and then clicks the Login button.**
- ❑ The reason we didn’t break the sentence at each “and” and “or” is that in constructing the Collaboration Diagram we’re trying to figure out the Interface Classes, Entity Classes, Control Classes that will have to be parts of the system, and actions that the system will have to perform, so the rule actually should have been *“a sentence with an “and” or an “or” that describes a system (application) activity is broken into its constituent parts; a sentence describing an actor’s activity isn’t broken down.”*

Introduction to Use-Case Driven Design, CS577a, Fall 2006 19 10/5/2006




University of Southern California  
Center for Software Engineering

## Constructing the Collaboration Diagram (cont.)

- ❑ Since Login’s basic course consists of five “sentences” – see previous slides, Login’s Collaboration Diagram will be constructed in five steps
- ❑ Login’s Collaboration Diagram starts out as an empty/blank diagram, and additions are made to it for each “sentence.”
- ❑ See the next five slides for the step-by-step construction of Login’s Collaboration Diagram.


Introduction to Use-Case Driven Design, CS577a, Fall 2006 20 10/5/2006



University of Southern California  
Center for Software Engineering

## STEP 1


The **Customer** clicks the **Login** button on the **Home Page**.



Note that the Login button is an interface item; if it were any more complex than *just being a button to click on*, the author of Login's Collaboration Diagram might have modeled it as a separate Interface Class.

Note, further, that just as was the case in deciding what the Use-Cases are for the Internet Bookstore, deciding on what's an "object" worthy of going into a Collaboration Diagram is a judgment call. IN SOFTWARE DEVELOPMENT THERE IS OFTEN NO SINGLE *RIGHT ANSWER OR RIGHT ANALYSIS OR DESIGN DECISION*.

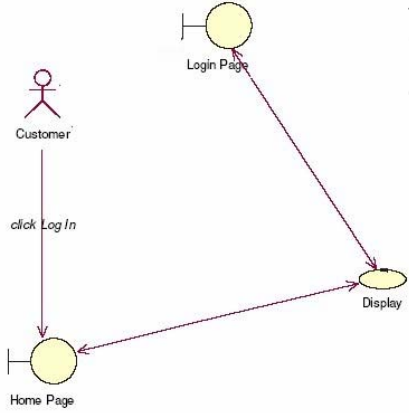
Introduction to Use-Case Driven Design, CS577a, Fall 2006
21
10/5/2006



University of Southern California  
Center for Software Engineering

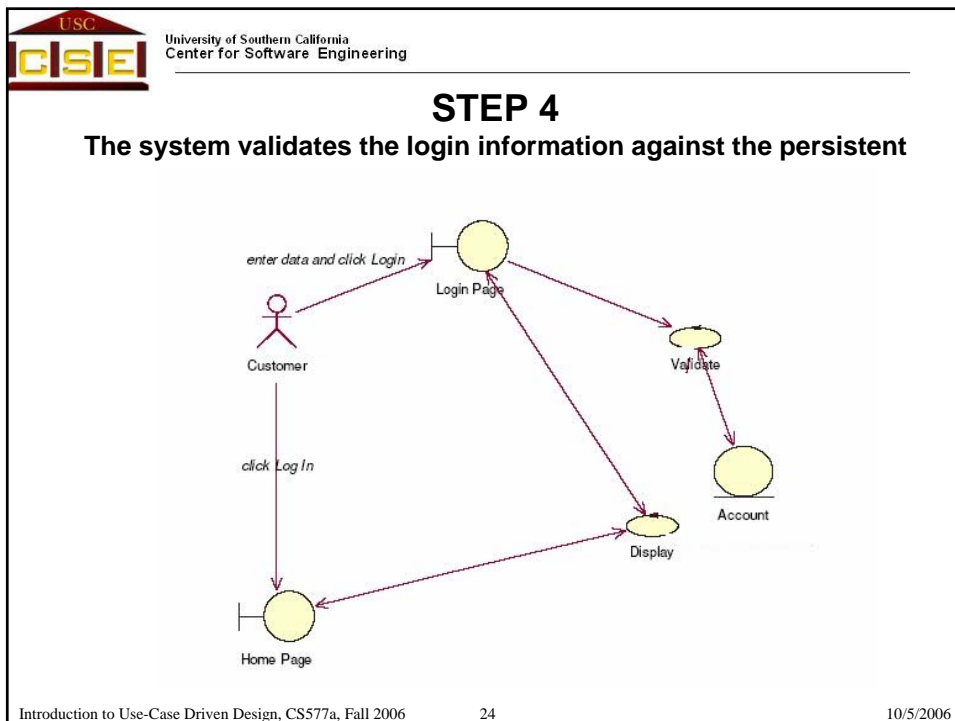
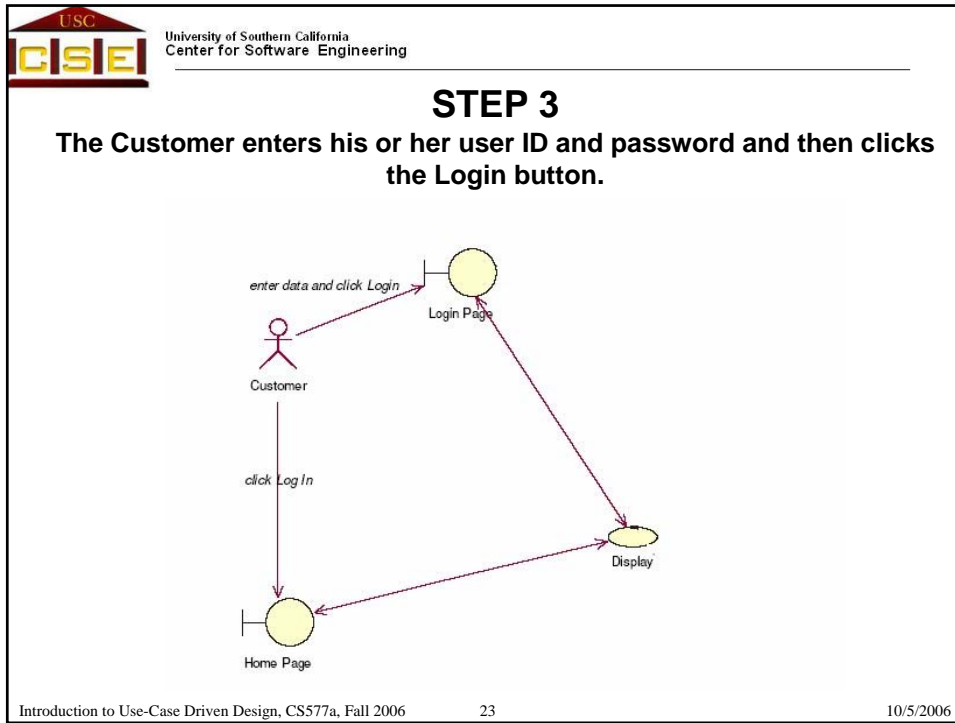
## STEP 2

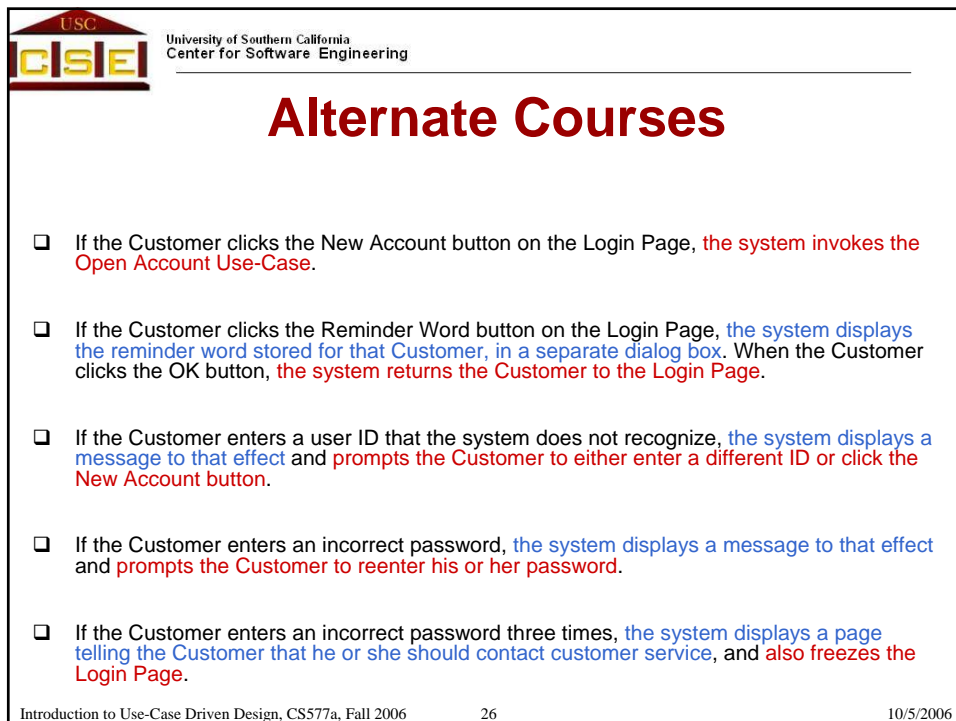
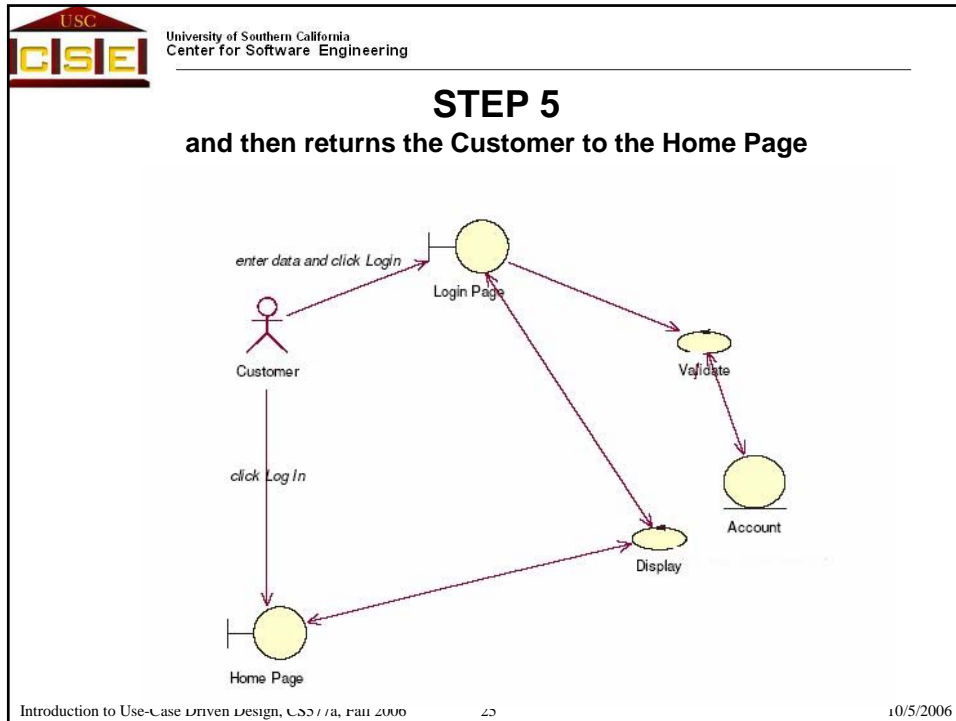
The system displays the **Login Page**.

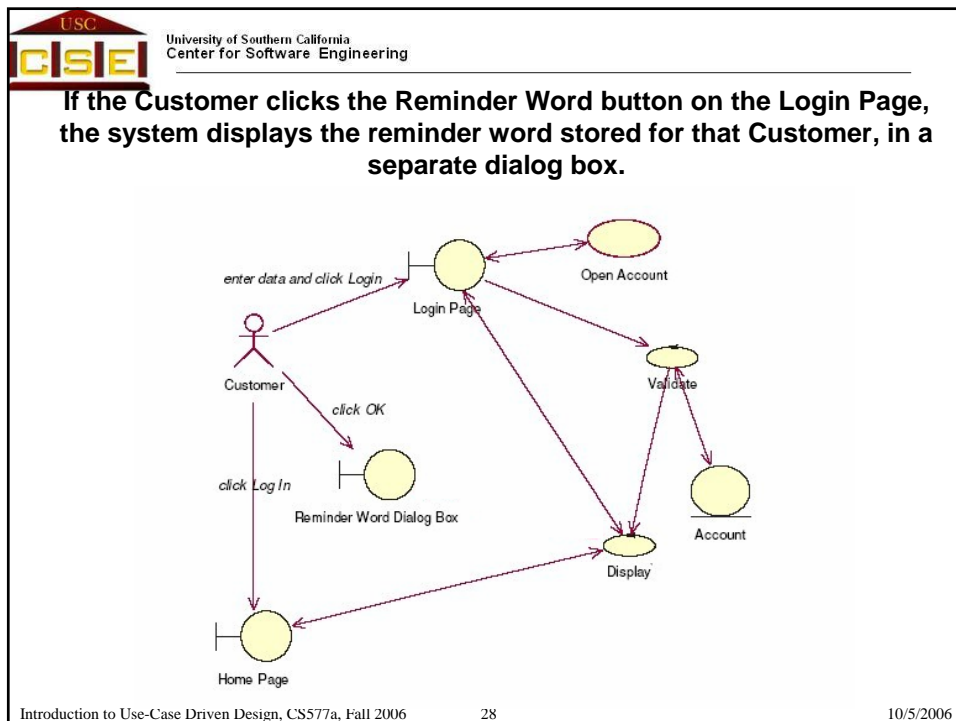
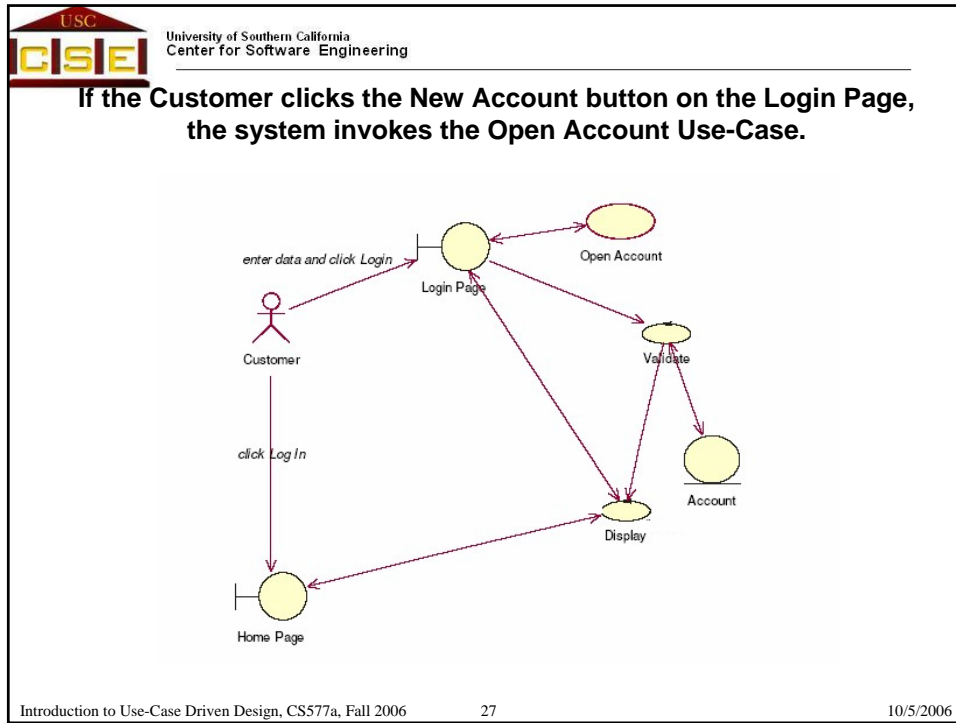


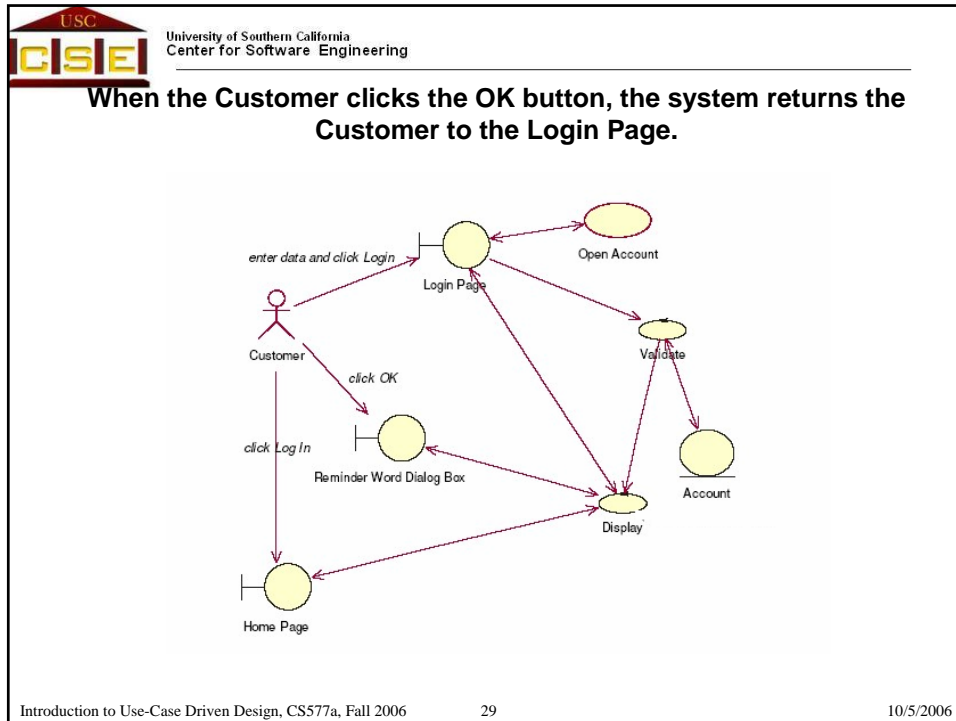
It wouldn't be much of a stretch to guess that oval-shaped Control Classes in Collaboration Diagrams later turn into "methods."

Introduction to Use-Case Driven Design, CS577a, Fall 2006
22
10/5/2006










USC  
CSEI  
University of Southern California  
Center for Software Engineering

## Exercise

❑ Fill in the parts of the Collaboration Diagram for the following three parts of the alternate course

- If the Customer enters a user ID that the system does not recognize, the system displays a message to that effect and prompts the Customer to either enter a different ID or click the New Account button.
- If the Customer enters an incorrect password, the system displays a message to that effect and prompts the Customer to reenter his or her password.
- If the Customer enters an incorrect password three times, the system displays a page telling the Customer that he or she should contact customer service, and also freezes the Login Page.

Introduction to Use-Case Driven Design, CS577a, Fall 2006 30 10/5/2006




University of Southern California  
Center for Software Engineering

---

## More on Use-Case Driven Design

- ❑ In most cases, Use-Case Driven Design results in the realization that Use-Case texts are missing details, so use-case construction and Use-Case Driven Design are iterative processes, during which both Use-Case texts and Collaboration Diagrams improve iteratively until they are deemed to be adequate to go on to the next step – Sequence Diagram construction.
- ❑ In this example, however, the Use-Case texts are already extremely well written
  - In particular, they cover all/most alternate courses
    - The source of the largest number of problems if they aren't done well
  - So we have assumed that they're OK and have constructed Collaboration Diagrams directly from them without trying to improve them.

Introduction to Use-Case Driven Design, CS577a, Fall 2006 31 10/5/2006




University of Southern California  
Center for Software Engineering

---

## More Collaboration Diagrams

- ❑ The next few slides show the Collaboration Diagrams of a number of other Internet Bookstore uses cases. (Use-Case texts are repeated for convenience.)

Introduction to Use-Case Driven Design, CS577a, Fall 2006 32 10/5/2006




University of Southern California  
Center for Software Engineering

## Open Account Use-Case

### Basic Course

The system displays the New Account Page. The Customer types his or her name, an e-mail address, and a password (twice), and then presses the Create Account button. The system ensures that the Customer has provided valid data and then adds an Account to the Master Account Table using that data. Then the system returns the Customer to the Home Page

Introduction to Use-Case Driven Design, CS577a, Fall 2006 33 10/5/2006



University of Southern California  
Center for Software Engineering

## Open Account Use-Case

### Alternate Courses

If the Customer did not provide a name, the system displays an error message to that effect and prompts the Customer to type a name.

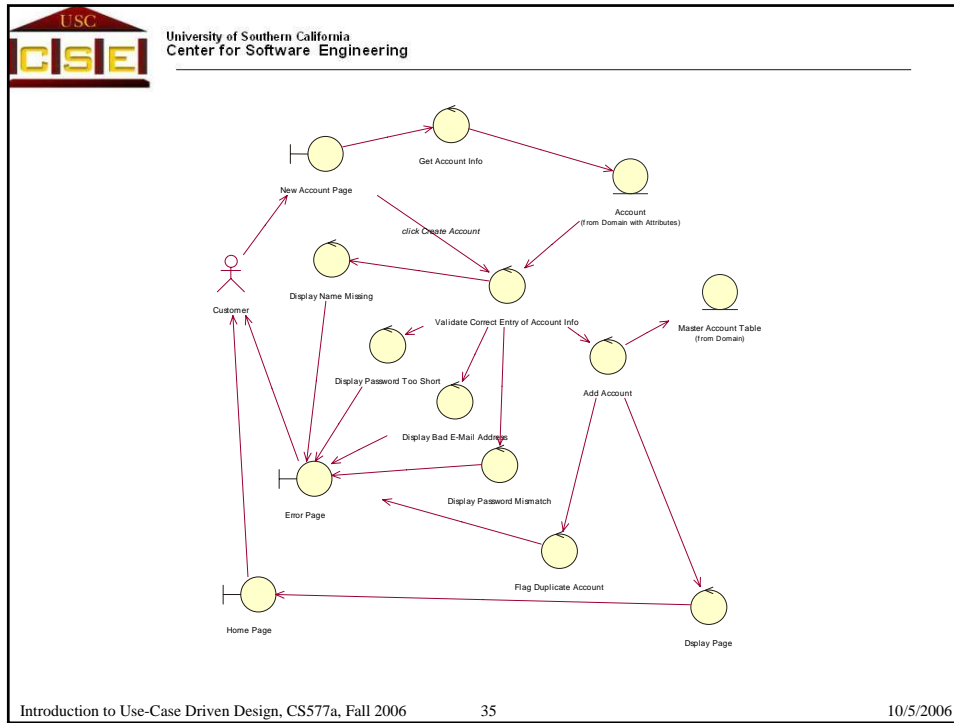
If the Customer provided an email address that's not in the correct form, the system displays an error message to that effect and prompts the Customer to type a different address.

If the Customer provided a password that is too short, the system displays an error message to that effect and prompts the Customer to type a longer password.

If the Customer did not type the same password twice, the system displays an error message to that effect and prompts the Customer to type the password correctly the second time.

If the account is already in the master account table, notify the user.

Introduction to Use-Case Driven Design, CS577a, Fall 2006 34 10/5/2006



University of Southern California  
Center for Software Engineering


## Search by Author Use-Case

### Basic Course

The Customer types the name of an Author on the Search Page and then presses the Search button. The system ensures that the Customer typed a valid search phrase, and then searches the Catalog and retrieves all of the Books with which that Author is associated. Then the system retrieves the important details about each Book, and creates a Search Results object with that information.

Then the system displays the list of Books on the Search Results Page, with the Books listed in reverse chronological order by publication date. Each entry has a thumbnail of the Book's cover, the Book's title and authors, the average Rating, and an Add to Shopping Cart button. The Customer presses the Add to Shopping Cart button for a particular Book. The system passes control to the Add Item to Shopping Cart Use-Case.

Introduction to Use-Case Driven Design, CS577a, Fall 2006 36 10/5/2006



University of Southern California  
Center for Software Engineering

## Search by Author Use-Case

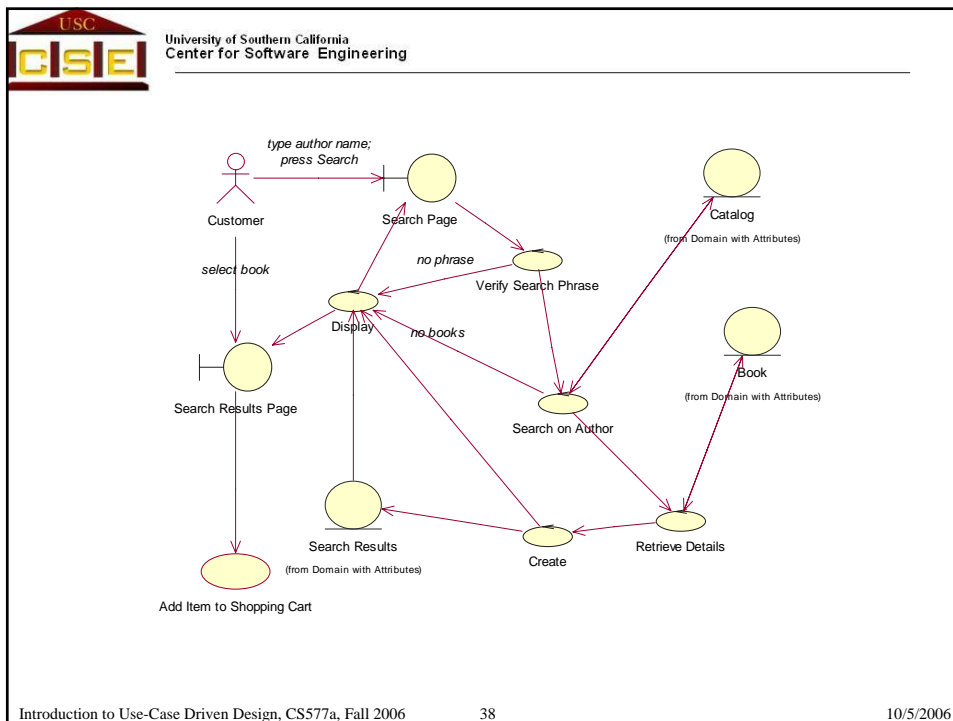
Alternate Courses


If the Customer did not type a search phrase before pressing the Search button, the system displays an error message to that effect and prompts the Customer to type a search phrase.

If the system was unable to find any Books associated with the Author that the Customer specified, the system displays a message to that effect and prompts the Customer to perform a different search.

If the Customer leaves the page in a way other than by pressing an Add to Shopping Cart button, the system returns control to the Use-Case from which this Use-Case received control.

Introduction to Use-Case Driven Design, CS577a, Fall 2006
37
10/5/2006






University of Southern California  
Center for Software Engineering

## Pre-Class Exercise: Part 1

- ❑ The next slide shows the basic course and the alternate courses for the “Edit Contents of Shopping Cart” Use-Case.
- ❑ Construct a Collaboration Diagram for this Use-Case. Be sure to show how you decomposed the text into “sentences” in doing the construction.
- ❑ Remember that robustness analysis, like most types of “modeling” in software engineering, is not like mathematics; that is, there is no single “right answer.” Different developers can develop different, equally good, models of the same thing.

Introduction to Use-Case Driven Design, CS577a, Fall 2006 39 10/5/2006



University of Southern California  
Center for Software Engineering

## Edit Contents of Shopping Cart Use-Case

### Basic Course

On the Shopping Cart Page, the Customer modifies the quantity of an Item in the Shopping Cart and then presses the Update button. The system stores the new quantity and then computes and displays the new cost for that Item. The Customer presses the Continue Shopping button. The system returns control to the Use-Case from which it received control.


### Alternate Courses

If the Customer changes the quantity of the Item to 0, the system deletes that Item from the Shopping Cart.

If the Customer presses the Delete button instead of the Update button, the system deletes that Item from the Shopping Cart.

If the Customer presses the Check Out button instead of the Continue Shopping button, the system passes control to the Check Out Use-Case.

Introduction to Use-Case Driven Design, CS577a, Fall 2006 10/5/2006



University of Southern California  
Center for Software Engineering

## Track Recent Orders Use-Case


### Basic Course

The system retrieves the Orders that the Customer has placed within the last 30 days and displays these Orders on the Order Tracking Page. Each entry has the Order ID (in the form of a link), the Order date, the Order status, the Order recipient, and the Shipping Method by which the Order was shipped.

The Customer clicks on a link. The system retrieves the relevant contents of the Order, and then displays this information, in view-only mode, on the Order Details Page. The Customer presses OK to return to the Order Tracking Page.

Once the Customer has finished viewing Orders, he or she clicks the Account Maintenance link on the Order Tracking Page. The system returns control to the invoking Use-Case.

Introduction to Use-Case Driven Design, CS577a, Fall 2006 41 10/5/2006



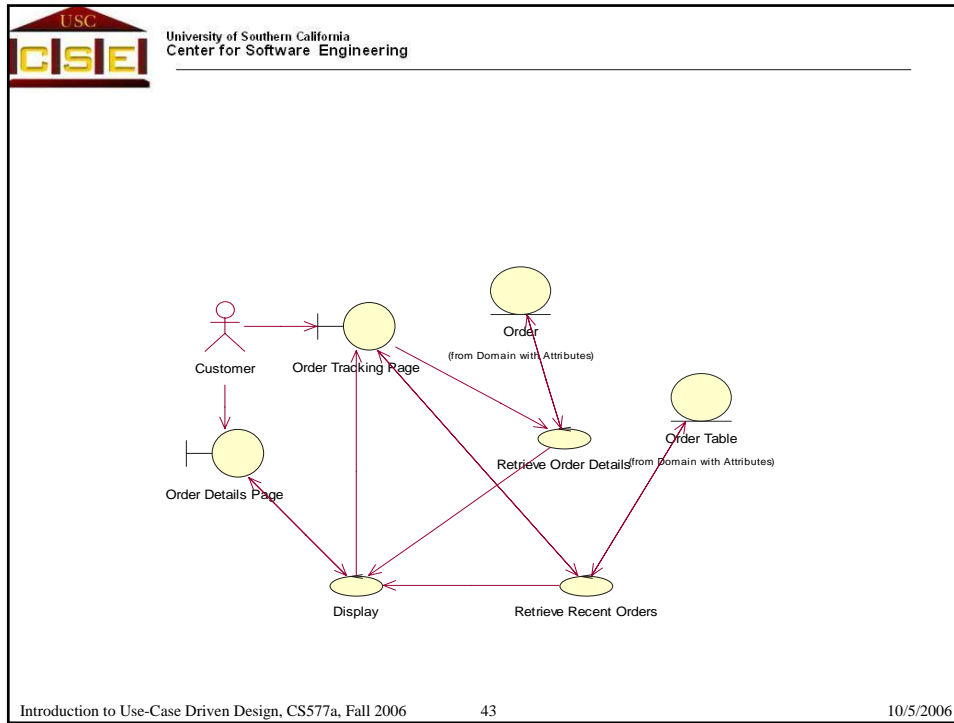
University of Southern California  
Center for Software Engineering

## Track Recent Orders Use-Case

### Alternate Courses

If the Customer has not placed any Orders within the last 30 days, the system displays a message to that effect on the Order Tracking Page.

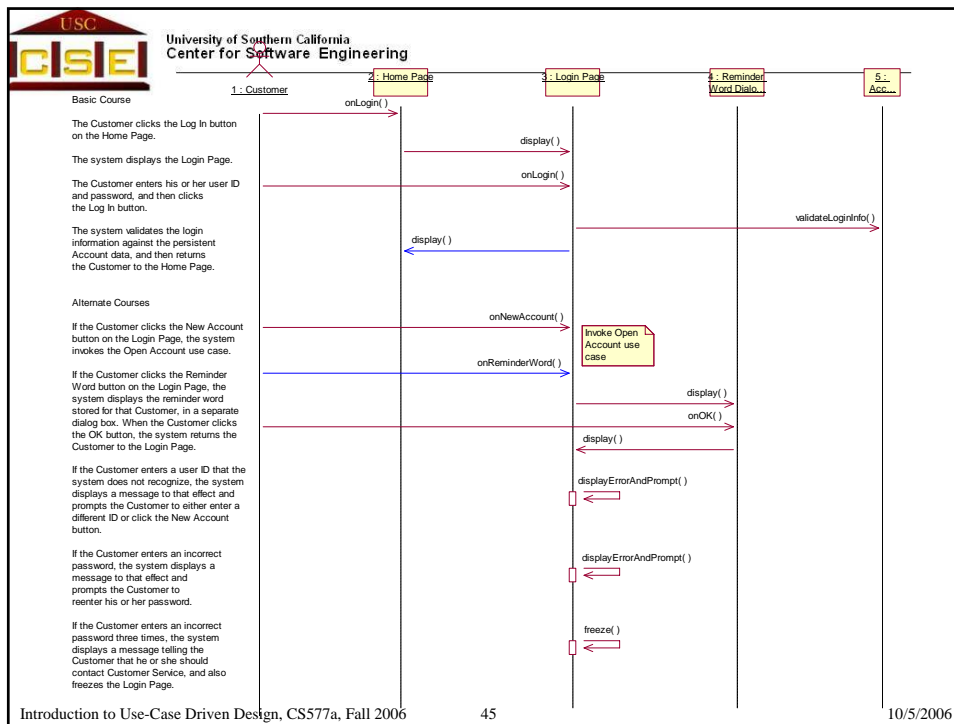
Introduction to Use-Case Driven Design, CS577a, Fall 2006 42 10/5/2006



## Sequence Diagram Construction

- ❑ The next step in “simple version” OOA&D is the construction of Sequence Diagrams from the Collaboration Diagrams
  - One Use-Case (Collaboration Diagram) at a time
- ❑ The next slide shows the sequence diagram for the Login Use-Case
  - Constructed from the Login Collaboration Diagram


Introduction to Use-Case Driven Design, CS577a, Fall 2006 44 10/5/2006



## Sequence Diagram Construction: Items Across the Top

- ❑ The items across the top of a sequence diagram are:
  - the actors involved in the Collaboration Diagram, and, therefore, in the Use-Case; in this case, just the customer
  - the Interface Classes created for the Collaboration Diagram; in this case the Home Page, the Login Page, and the Reminder Word Dialog Box
  - the Entity Classes created for the Collaboration Diagram; in this case just the Account object
- ❑ The items across the top of the sequence diagrams are going to turn into objects in the application's (system's) design
- ❑ If a Control Class represents a sufficiently complex control function, it may make it to the top of the sequence diagram and become an object in the design
- ❑ If, on the other hand, a Control Class is simple enough so that it can be implemented as a method, then it doesn't make it to the top of the sequence diagram; we'll see in a moment where it does go in the diagram; in this case, the two Control Classes, Display and Validate, will become methods.

Introduction to Use-Case Driven Design, CS577a, Fall 2006 46 10/5/2006




University of Southern California  
Center for Software Engineering

---

## Sequence Diagram Construction: Items Down the Left Side

- ❑ The individual sentences of the Use-Case's basic and alternate courses go down the left side of the sequence diagram.

Introduction to Use-Case Driven Design, CS577a, Fall 2006 47 10/5/2006




University of Southern California  
Center for Software Engineering

---

## Sequence Diagram Construction: Messages & Methods

- ❑ Arrows in the sequence diagrams represent messages – method calls
- ❑ The source of an arrow represents the object sending the message, i.e., the object invoking the method
- ❑ The target of an arrow is the object receiving the message, i.e., the object on which the method is invoked
- ❑ The only (design) creativity involved in constructing a sequence diagram is in deciding:
  - what the methods should be
  - which object each method should belong to
  - which object(s) calls which method, and at exactly what point during the course of the Use-Case
- ❑ The decisions are made one (Use-Case) sentence at a time...and the process is iterated until the designer is satisfied with the results

Introduction to Use-Case Driven Design, CS577a, Fall 2006 48 10/5/2006




University of Southern California  
Center for Software Engineering

## Sequence Diagram Construction: Messages & Methods (cont.)

- ❑ Read the assignments of methods, and the decisions as to who invokes each one and when, during the course of the Login Use-Case.
- ❑ On the next next two slides you'll find sequence diagrams for the New Account, Search by Author and Track Recent Orders Use-Cases
- ❑ On the slide following these two Use-Cases is part 2 of the pre-class exercise

Introduction to Use-Case Driven Design, CS577a, Fall 2006
49
10/5/2006



University of Southern California  
Center for Software Engineering

1: Customer
2: New Account Page
3: Error Page
4: Acc. Mail
5: Home Page
6: Get Acc.

**Basic Course**

The system displays the New Account Page. The Customer types his or her name, an email address, and a password (twice), and then presses the Create Account button.

The system ensures that the Customer has provided valid data, and then adds an Account to the Master Account Table using that data. Then the system returns the Customer to the Home Page.

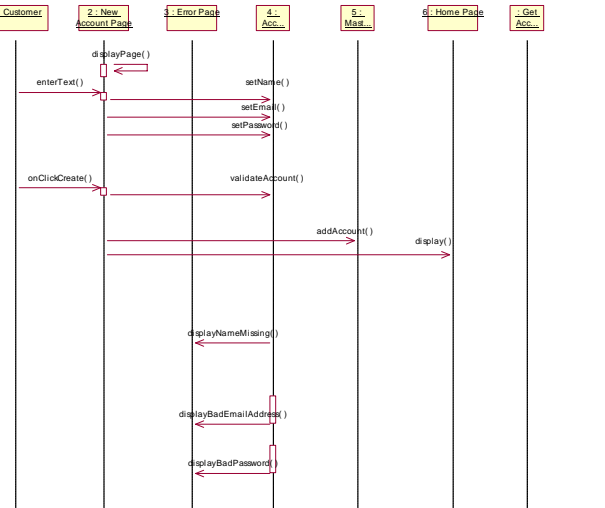
**Alternate Courses**

If the Customer did not provide a name, the system displays an error message to that effect and prompts the Customer to type a name.

If the Customer provided an email address that's not in the correct form, the system displays an error message to that effect and prompts the Customer to type a different address.

If the Customer provided a password that is too short, the system displays an error message to that effect and prompts the Customer to type a longer password.

If the Customer did not type the same password twice, the system displays an error message to that effect and prompts the Customer to type the password correctly the next time.



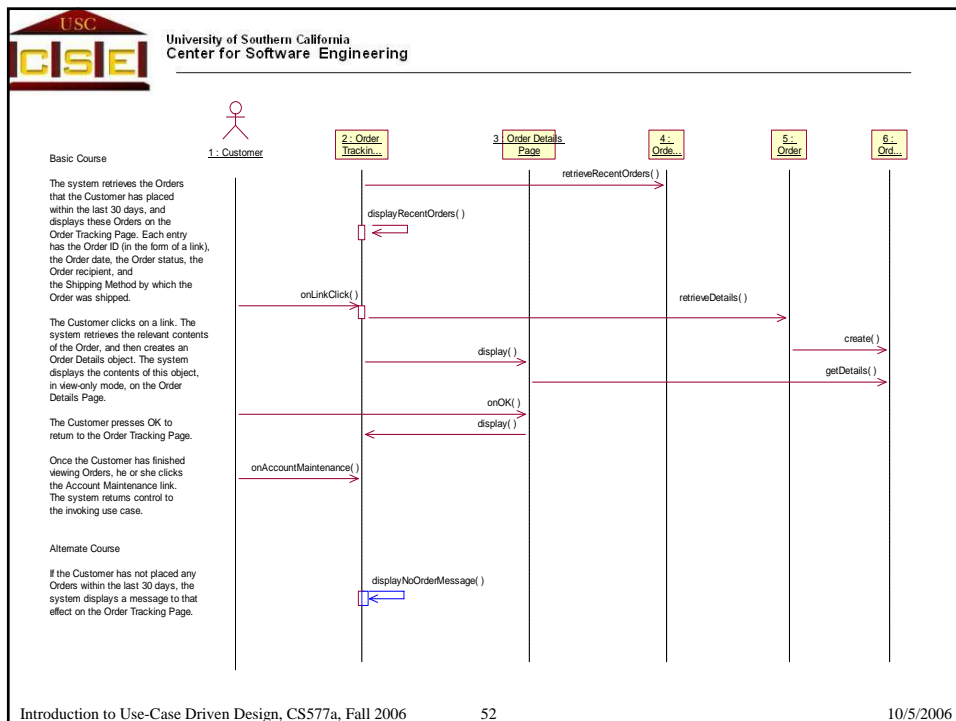
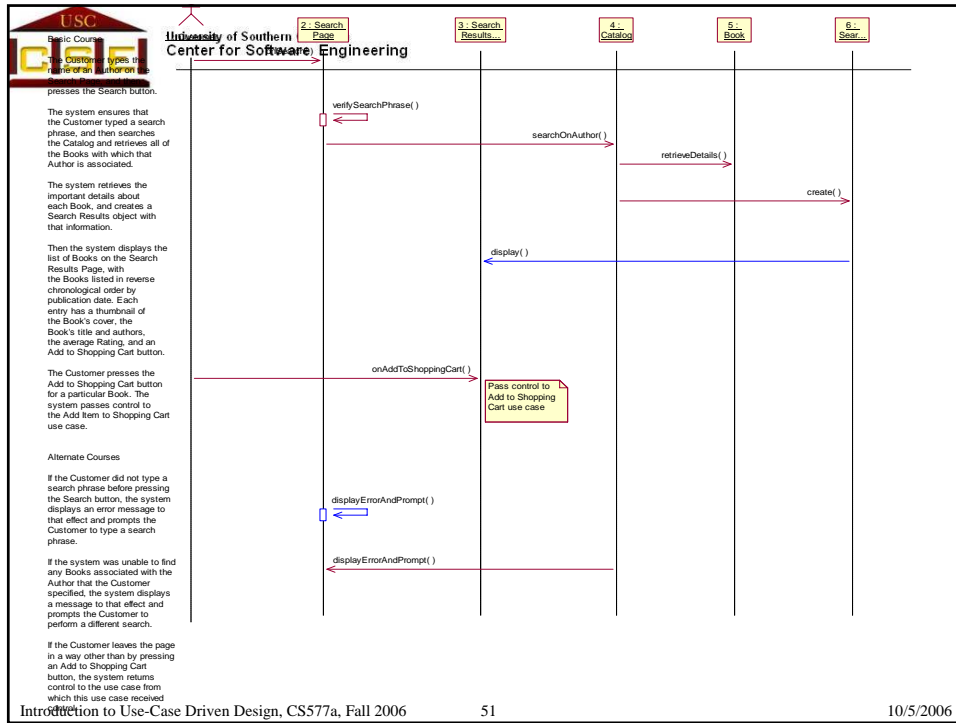
```


sequenceDiagram
    participant C as 1: Customer
    participant NP as 2: New Account Page
    participant EP as 3: Error Page
    participant AM as 4: Acc. Mail
    participant HP as 5: Home Page
    participant GA as 6: Get Acc.

    NP->>NP: displayPage()
    C->>NP: enterText()
    NP->>AM: setName()
    NP->>AM: setEmail()
    NP->>AM: setPassword()
    NP->>AM: setPassword()
    NP->>EP: validateAccount()
    NP->>AM: addAccount()
    NP->>HP: display()

    EP-->>EP: displayNameMissing()
    EP-->>EP: displayBadEmailAddress()
    EP-->>EP: displayBadPassword()
                    
```

Introduction to Use-Case Driven Design, CS577a, Fall 2006
50
10/5/2006






University of Southern California  
Center for Software Engineering

---

## Pre-Class Exercise: Part 2

- The next slide contains the Edit Contents of Shopping Cart Use-Case text
- Construct a sequence diagram for this Use-Case

Introduction to Use-Case Driven Design, CS577a, Fall 2006 53 10/5/2006




University of Southern California  
Center for Software Engineering

---

## Edit Contents of Shopping Cart Use-Case

- Basic Course
  - On the Shopping Cart Page, the Customer modifies the quantity of an Item in the Shopping Cart and then presses the Update button. The system stores the new quantity and then computes and displays the new cost for that Item. The Customer presses the Continue Shopping button. The system returns control to the Use-Case from which it received control.
- Alternate Courses
  - If the Customer changes the quantity of the Item to 0, the system deletes that Item from the Shopping Cart.
  - If the Customer presses the Delete button instead of the Update button, the system deletes that Item from the Shopping Cart.
  - If the Customer presses the Check Out button instead of the Continue Shopping button, the system passes control to the Check Out Use-Case.

Introduction to Use-Case Driven Design, CS577a, Fall 2006 54 10/5/2006



University of Southern California  
Center for Software Engineering

---

## Class Diagram Construction

- ❑ The application's (system's) class diagram is constructed directly from the sequence diagrams
- ❑ The class diagrams for the Internet Bookstore are posted as a separate file.

Introduction to Use-Case Driven Design, CS577a, Fall 2006

55

10/5/2006