

Design Use Case (concluded)

Alternate Courses:	<p>Alt-Step 3a: If the member clicks on the item name, the system displays a pop-up window, <i>W15—Product Detail Display</i>, which contains all the product details, including a graphic of its cover. The member clicks the [Close] button to close the pop-up window.</p> <p>Alt-Step 3b: If member wants to perform keyword search, invoke abstract use case <i>MSS-AUC006.00 Search Product Catalog by Keyword</i>.</p> <p>Alt-Step 5: If member wants to change demographic information, invoke abstract use case <i>MSS-AUC007.00 Change Member Profile</i>.</p> <p>Alt-Step 7: If the order requires changes the member can delete any item no longer wanted by deselecting the check box by item and/or changing the order quantity. Once the member has completed the order changes, he or she clicks the [Update Order] button. The system reprocesses the order (go to step 6). If the member clicks the [Do More Shopping] button, go to Step 3. If the member clicks the [Update Member Profile] button, invoke abstract use case <i>MSS-AUC007.00 Change Member Profile</i> and then go to step 6.</p> <p>Alt-Step 8: If the member's account is not in good standing, display to the member using window <i>W09—Member Account Status Display</i>, the account status, the reason the order is being held, and what actions are necessary to resolve the problem. In addition an e-mail is sent to the member with the same information. Invoke abstract use case <i>MSS—AUC004.00 Send Electronic Member Correspondence</i>. The system prompts the member to hold the order for later processing or cancel the order. If the member wishes to hold the order by clicking the [Save Order] button, the system records the order information, places it in hold status, and then displays the SoundStage main page, window <i>W00—Member Home Page</i>. If the member chooses to cancel the order by clicking the [Cancel Order] button, the system erases the inputted information, and then displays the SoundStage main page, window <i>W00—Member Home Page</i>. Terminate the use case.</p> <p>Alt-Step 10: If the member selects the option to pay by credit card, invoke abstract use case <i>MSS-AUC012.00 Pay by Credit Card</i>. If the member cannot pay by credit card, the system prompts the member to hold the order for later processing or cancel the order. If the member wishes to hold the order by clicking the [Save Order] button, the system records the order information, places it in hold status, and then displays the SoundStage main page, window <i>W00—Member Home Page</i>. If the member chooses to cancel the order by clicking the [Cancel Order] button, the system erases the inputted information, and then displays the SoundStage main page, window <i>W00—Member Home Page</i>. Terminate the use case</p>
---------------------------	---

Modeling Object Interactions and Behaviors

Step 1: Identify and Classify Use-Case Design Objects

Step 2: Identify Object Attributes

Step 3: Model High-Level Object Interactions

Step 4: Identify Object States, Behaviors, and Responsibilities

Step 5: Model Detailed Object Interactions

Updating Object Model to Reflect Implementation Environment

Design class diagram – a diagram that depicts classes that correspond to software components that are used to build the software application.

Includes:

- Classes
- Associations and gen/spec and aggregation relationships
- Attributes and attribute-type information
- Methods with parameters
- Navigability
- Dependencies

Transforming Analysis Class Diagram to Design Class Diagram

- Add design objects to diagram
- Add attributes and attribute-type information to design objects
- Add attribute visibility
- Add methods to design objects
- Add method visibility
- Add association navigability
- Add dependency relationships

Additional UML Design and Implementation Diagrams

- **Activity diagrams** model actions that will be performed when an operation is executing as well as the results of those actions.
- **Component diagrams** graphically depict the physical architecture of the software of the system. They can be used to show how programming code is divided into modules and to depict the dependencies between those components.
- **Deployment diagrams** describe the physical architecture of the hardware and software in the system.