

Diagram: Business Domain Model

pkg Business Domain Model

Business Domain Model

Below are the key packages in the Business Domain Model of TheMurray.com.au bookshop.
Double-click on the packages to view the key diagrams.

Domain Model

- + Account
- + Line Item
- + Order
- + OrderStatus
- + ShoppingBasket
- + Stock Item
- + Transaction

The Domain Model captures a description of what the software knows about the domain and the objects it contains.

Process Model

- + Manage Customer Orders
- + Sell Books On-Line
- + User
- + Customer Order
- + User Enquiry
- + Book Catalogue
- + Delivered Order
- + Order
- + Ship Order
- + Shipping Company
- + Shopping Basket
- + Shopping Basket Item
- + Take Customer Orders
- + Warehouse Inventory
- + Web Pages

The Process Model reflects the main business activities in which the system will be involved.

Opportunity Definition

- + Stakeholders Interests
- + Positioning
- + Stakeholders

This section defines the Stakeholders, their interests and the key points in their positioning towards the implementation of a new system.

Diagram: Domain Model

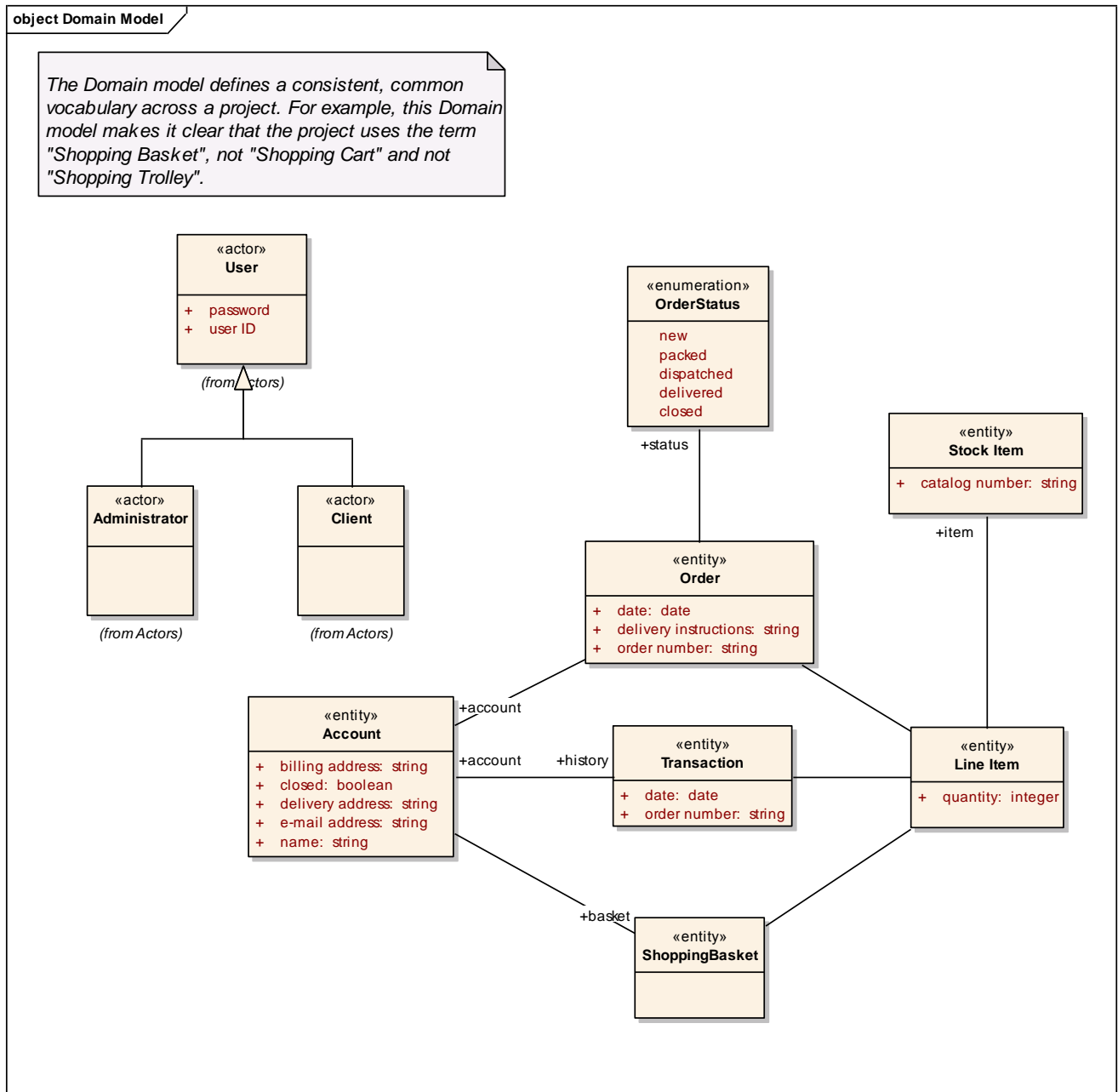


Diagram: Opportunity Definition

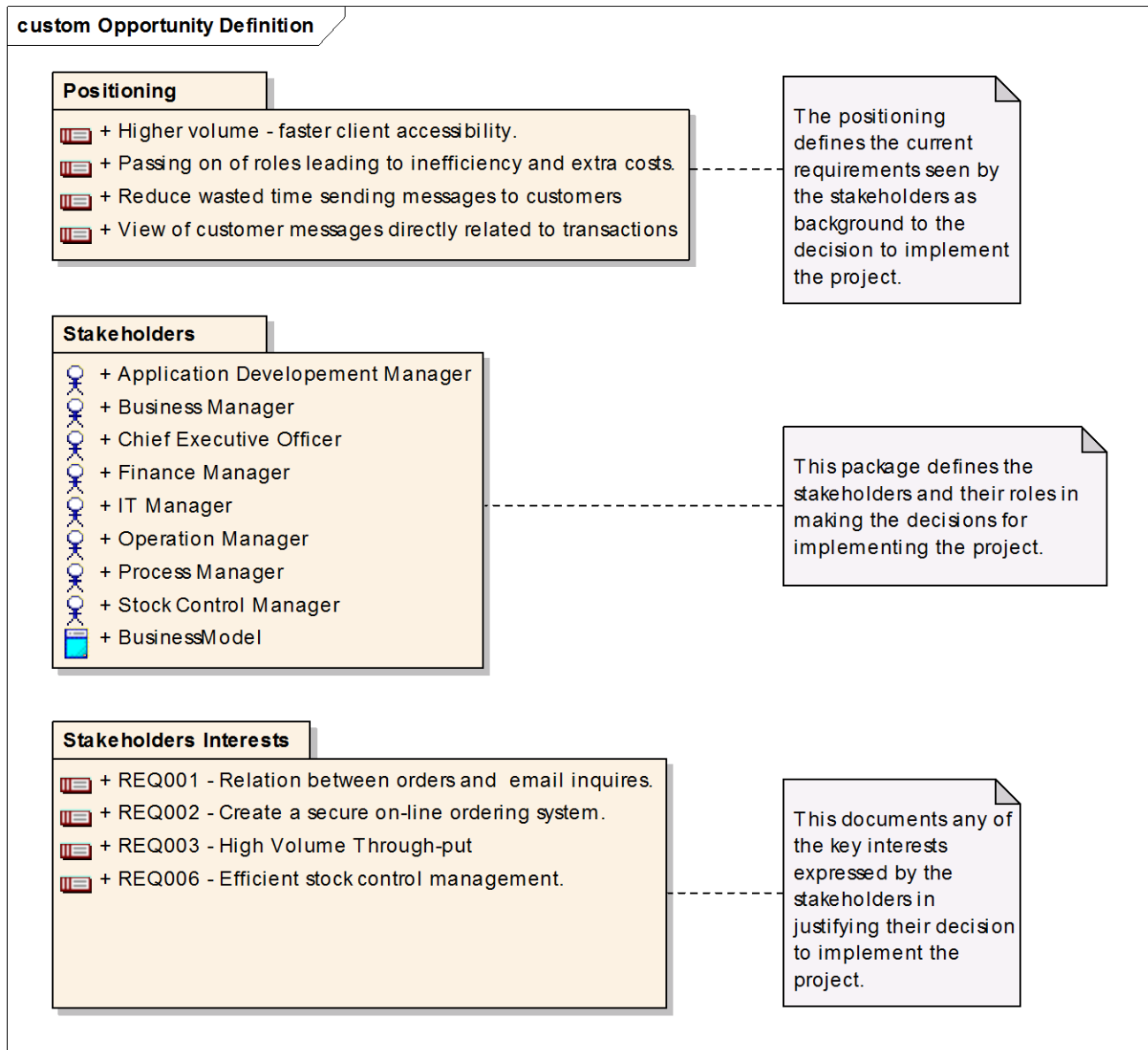


Diagram: Stakeholders

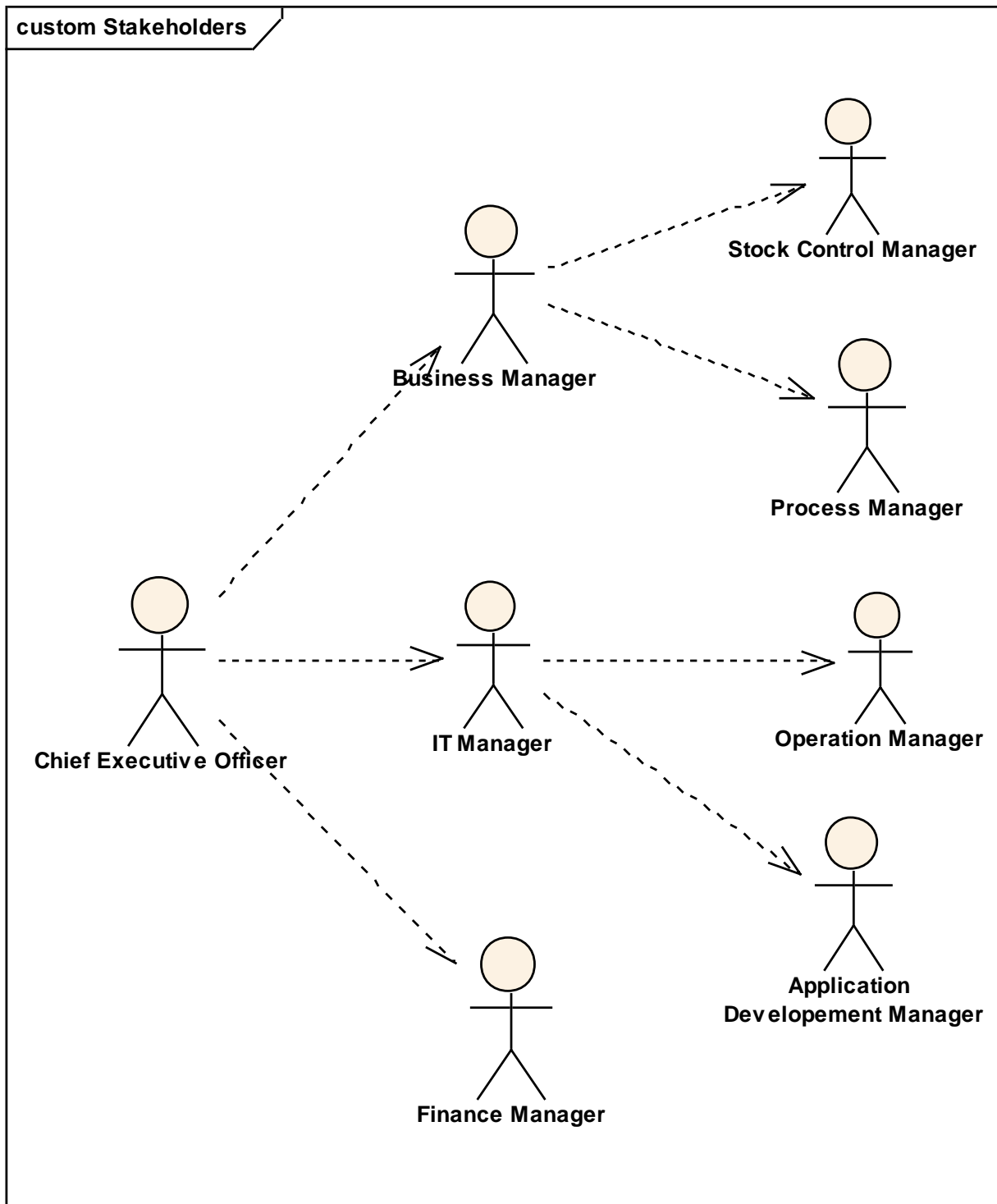


Diagram: Stakeholders Images

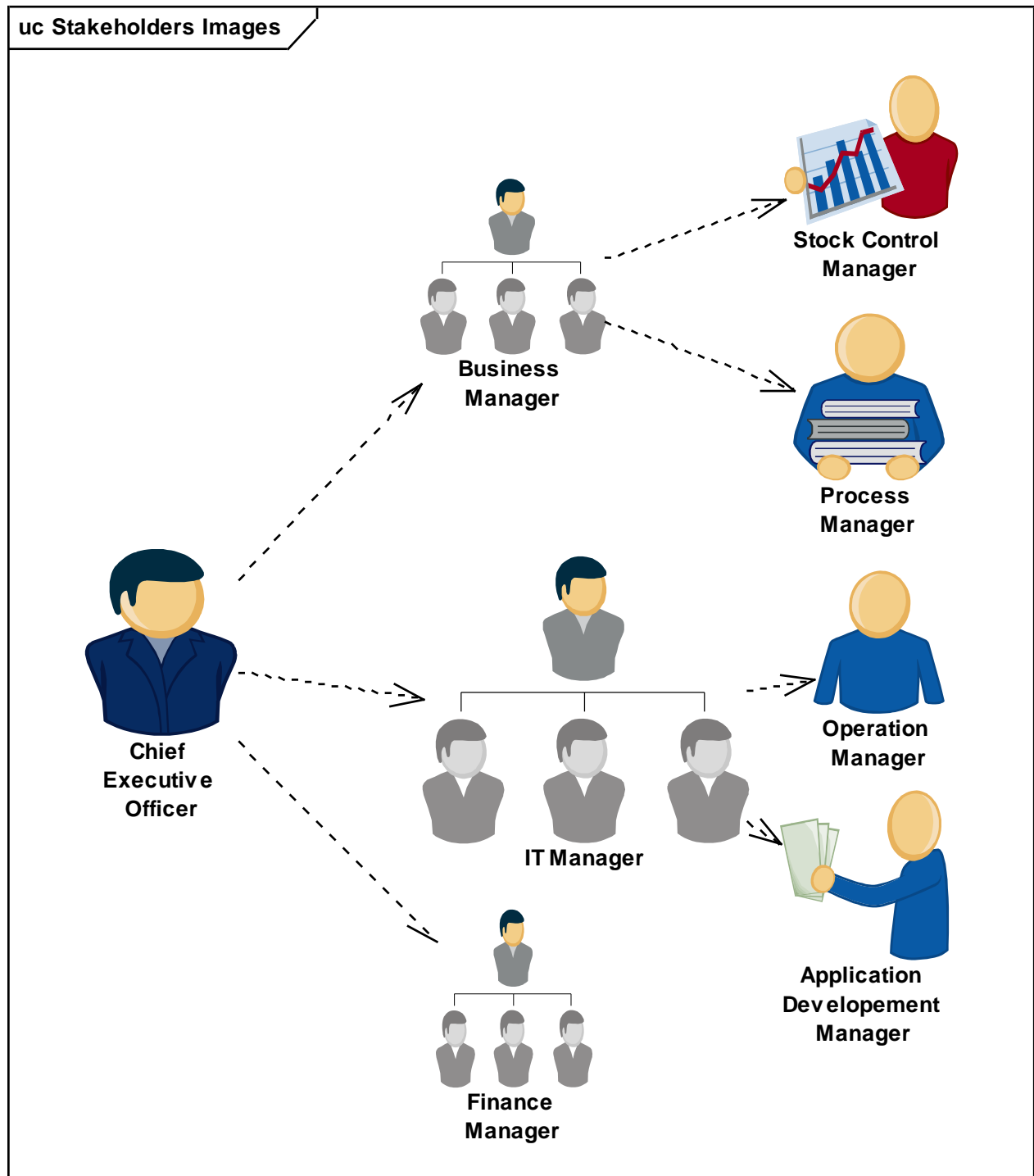


Diagram: Stakeholders Interests

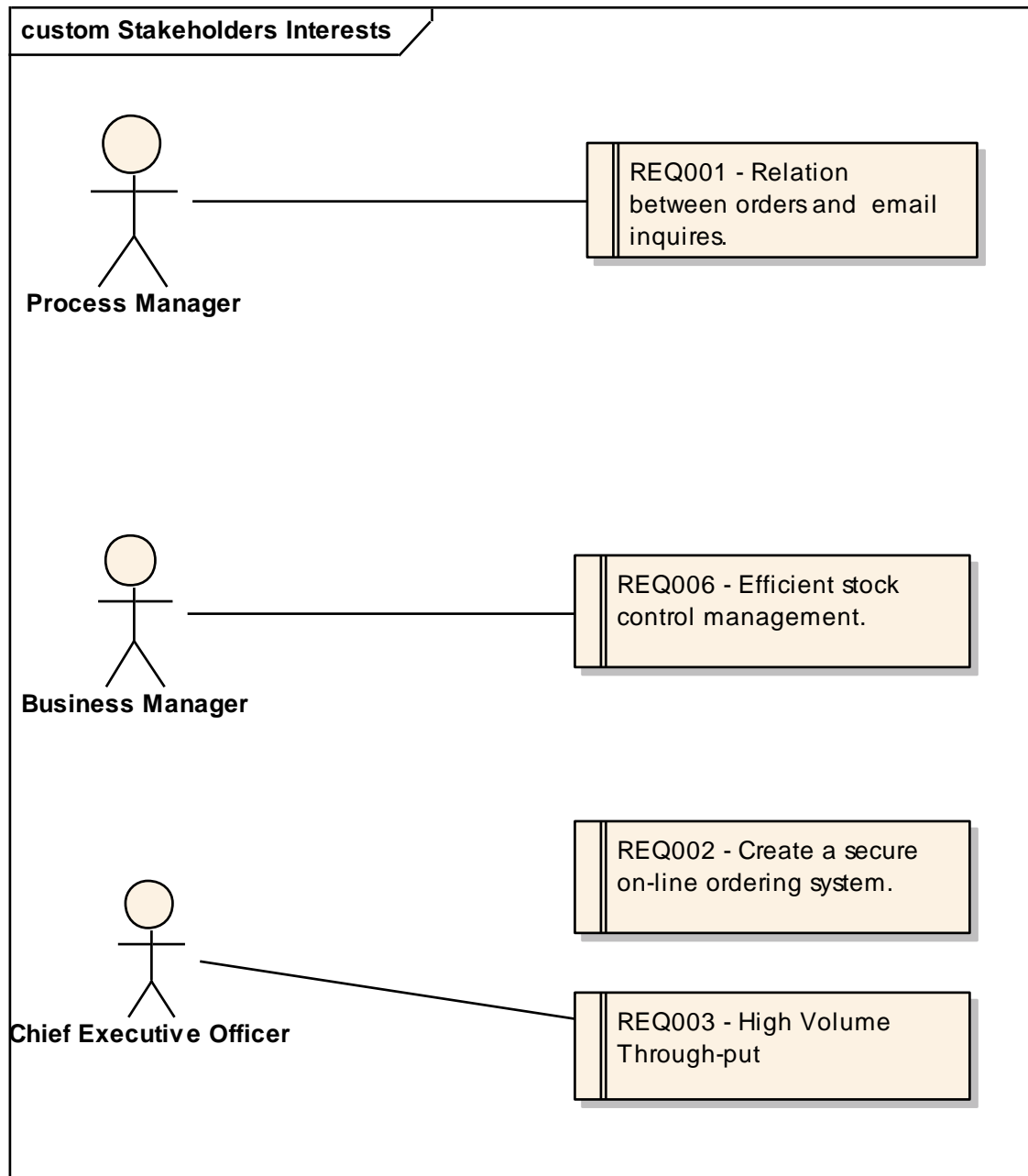


Diagram: Positioning

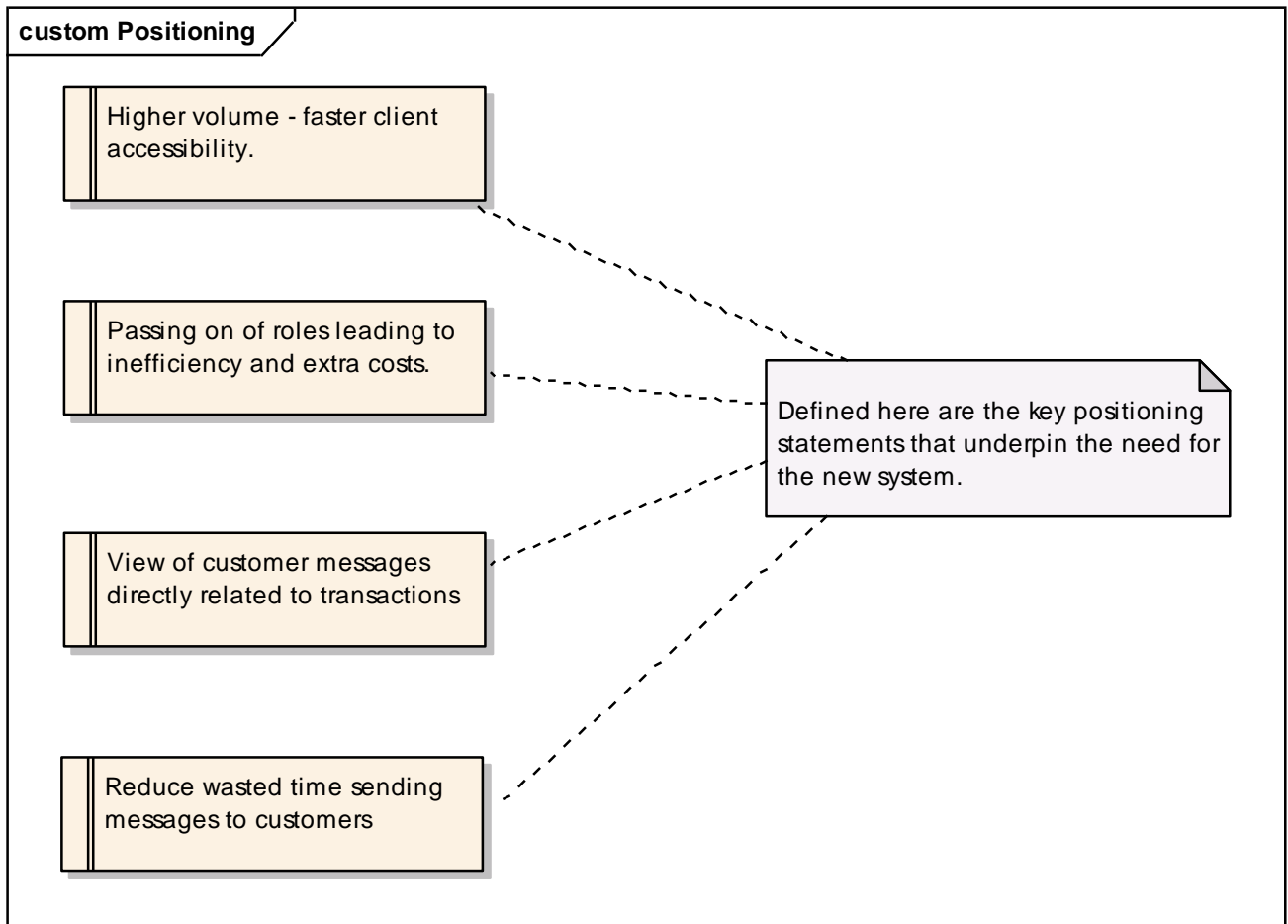


Diagram: Process Model

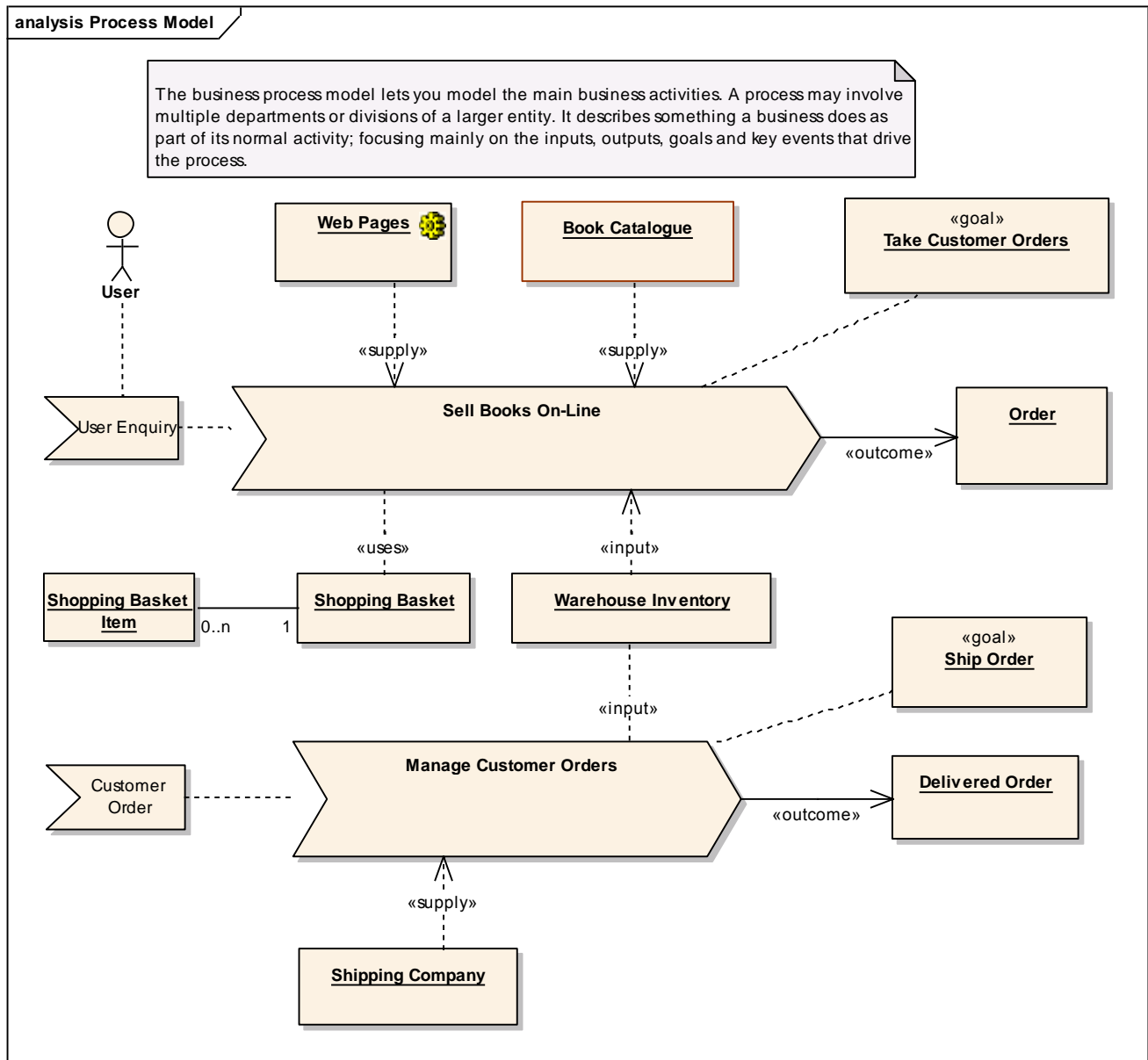


Diagram: Requirements Model

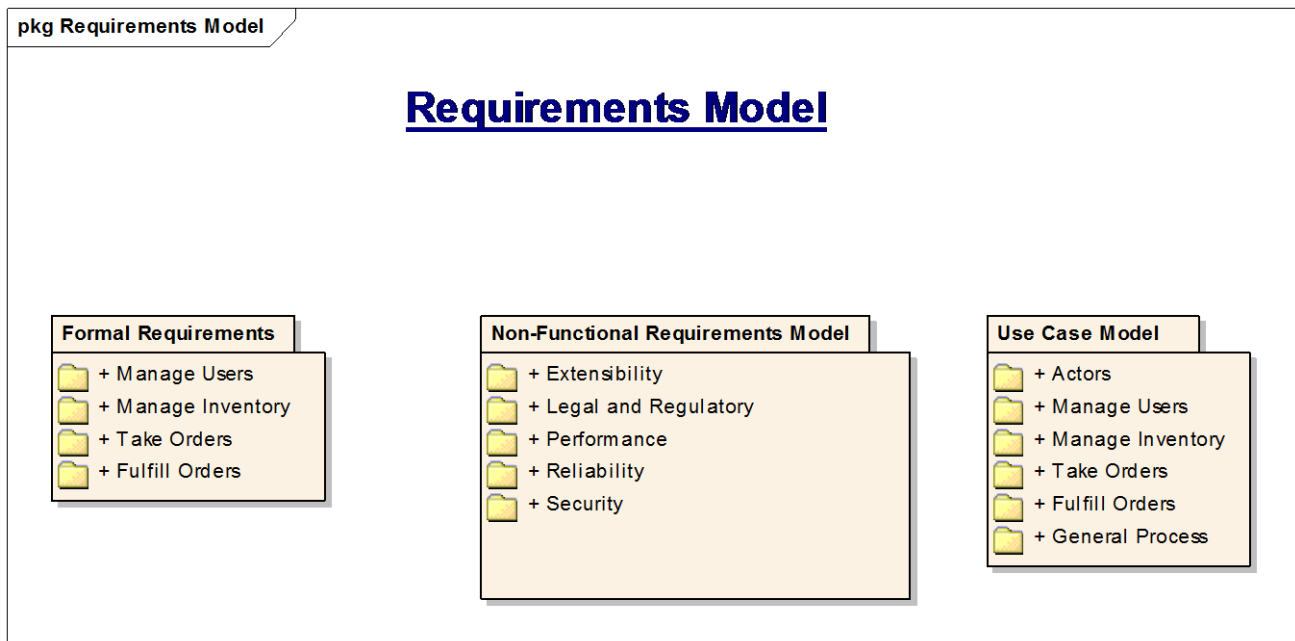


Diagram: Formal Requirements

pkg Formal Requirements

Formal Requirements

Enterprise Architect allows you to document requirements graphically using the Requirement element. The Requirement element is available from the 'Requirements' Toolbox folder.

Using a Requirement element in the UML model, allows relationships to be drawn between requirements. It also allows for direct traceability to other aspects of the model such as Use Cases, Test Cases and other Analysis or Design elements.

The requirement element can be used to model or document any requirements, ranging from formal business requirements through to performance or security requirements.

Double-click on the package elements below to view the diagrams for these packages.



[See Help:Requirements](#)

Manage Users

- + REQ011 - Manage User Accounts
- + REQ016 -Add Users
- + REQ017 -Remove User
- + REQ018 - Report on User Account
- + REQ024 - Secure Access
- + REQ025 - Store User Details
- + REQ026 - Validate User

Manage Inventory

- + REQ019 - Manage Inventory
- + REQ020 -Receive Books
- + REQ021 - List Stock Levels
- + REQ022 -Order Books
- + REQ023 -Store and Manage Books
- + REQ027 - Add Books
- + REQ032 - Update Inventory

Take Orders

- + REQ012 - Provide Online Sales
- + REQ014 -ShoppingBasket
- + REQ015 -Process Credit Card Payment

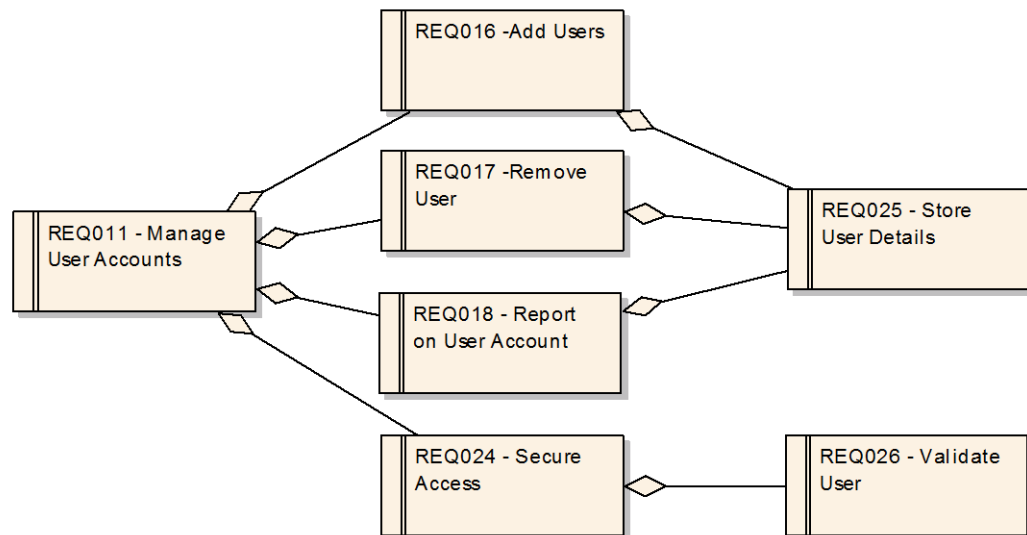
Fulfill Orders

- + REQ013 -Manage Deliveries
- + REQ028 - Process Order
- + REQ029 - Ship Order
- + REQ030 - Package Order
- + REQ031 - List Current Orders
- + REQ033 - Retrieve Books

Diagram: Manage Users

custom Manage Users

Requirements are defined using the Custom Element of type Requirement. To view the detailed description of a Requirement, double-click on the element to view the properties.



Requirements can have relationships with other elements such as other Requirements, Use Cases etc. To view the Traceability of a requirement use the Hierarchy window. The Hierarchy window can be accessed from the main menu: View | Hierarchy or (Ctrl-Shift-4).

For more information on Requirements and the Hierarchy View see the following help topics:

 [Requirements Management](#)

 [Hierarchy View](#)

Diagram: Manage Inventory

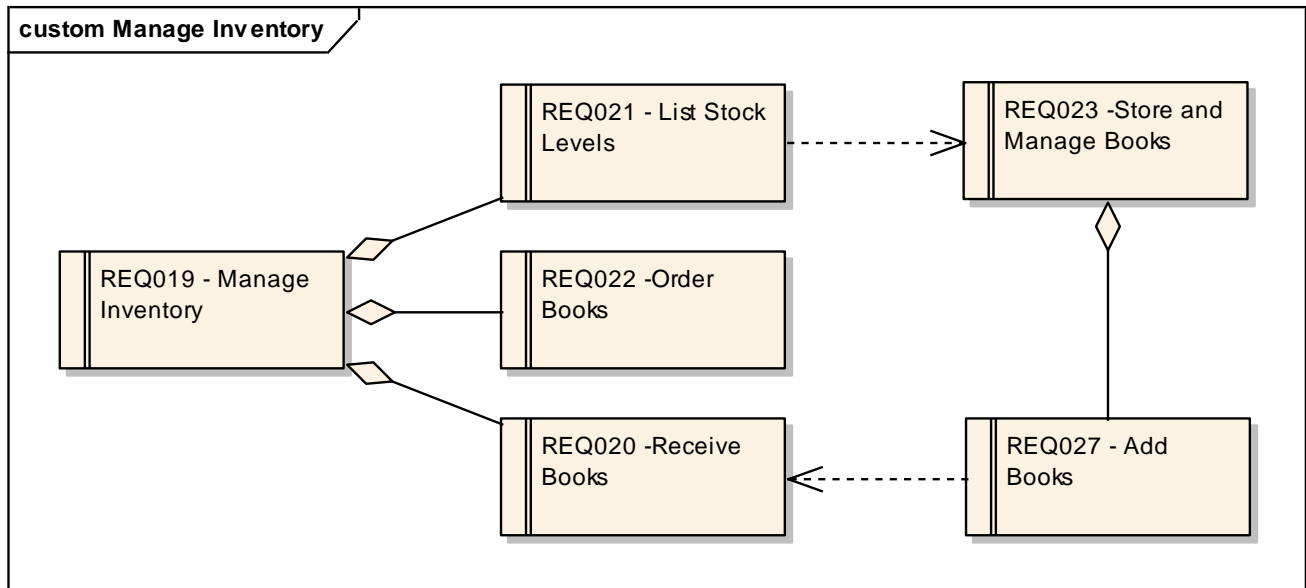


Diagram: Take Orders

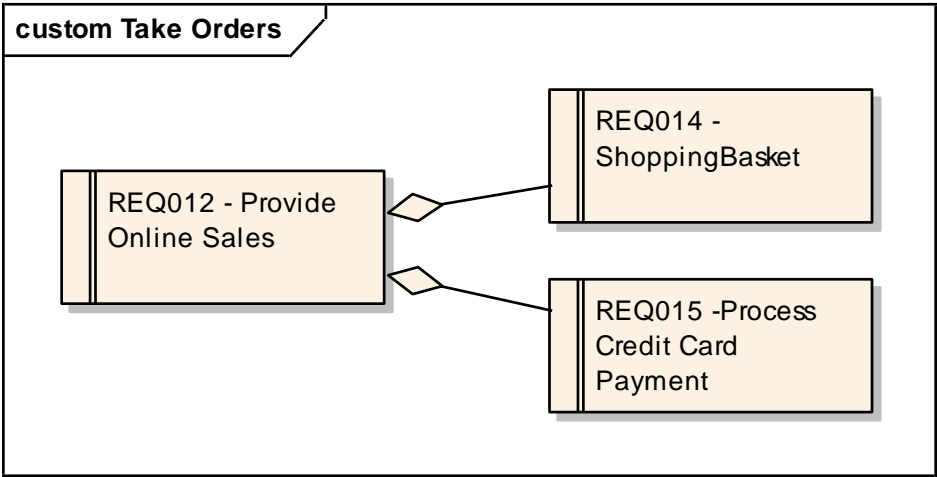


Diagram: Fulfill orders

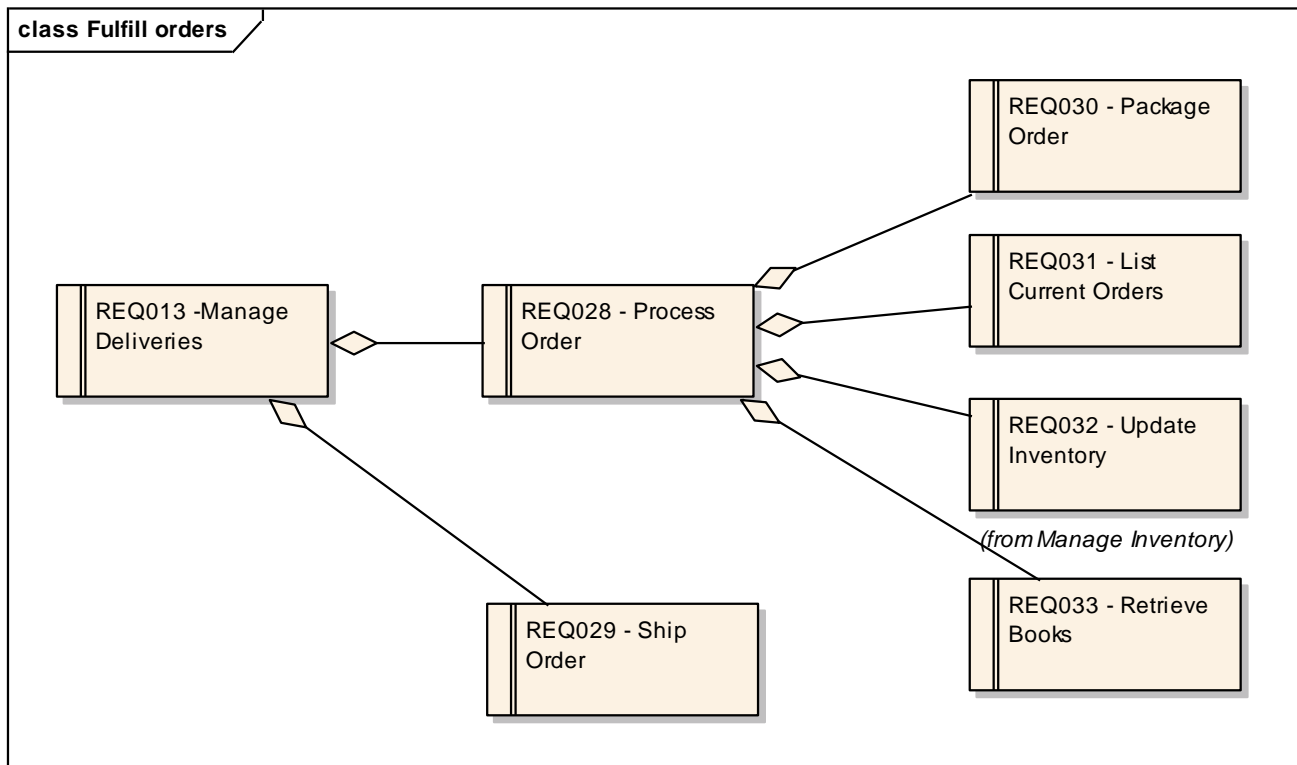


Diagram: Use Case Model

uc Use Case Model

Use Case

This section covers the Use Case model of the system as defined in the requirements section.

To view the Use Case diagrams double click on any package below.



To View the Traceability of the Use Cases - double-click here.

Manage Users

- + Close Account
- + Create Account
- + Delete User
- + Login
- + View Account details
- + View History
- + View Open Orders

Fulfill Orders

- + List Current Orders
- + Package Order
- + Process Order
- + Ship Order

Take Orders

- + Add To Shopping Basket
- + Go To Checkout
- + Remove From Shopping Basket
- + Search for Books
- + View Shopping Basket

Manage Inventory

- + Add New Titles
- + Create Orders
- + Edit Titles
- + List Stock Levels
- + Manage Publishers
- + Manage Titles
- + Receive Orders

Actors

- + Administrator
- + Client
- + Shipping Company
- + Storeroom Worker
- + User

Diagram: Actors

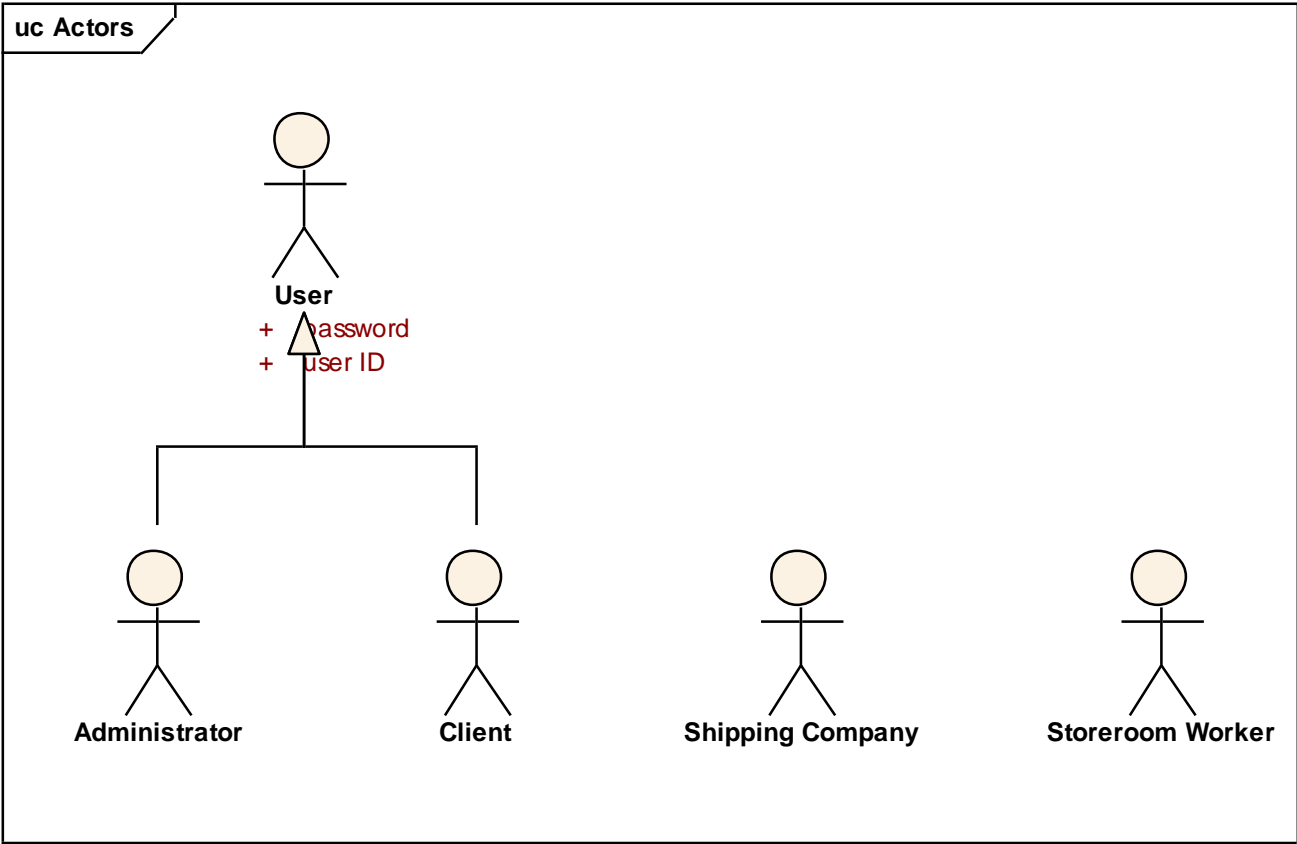


Diagram: Actors - Images

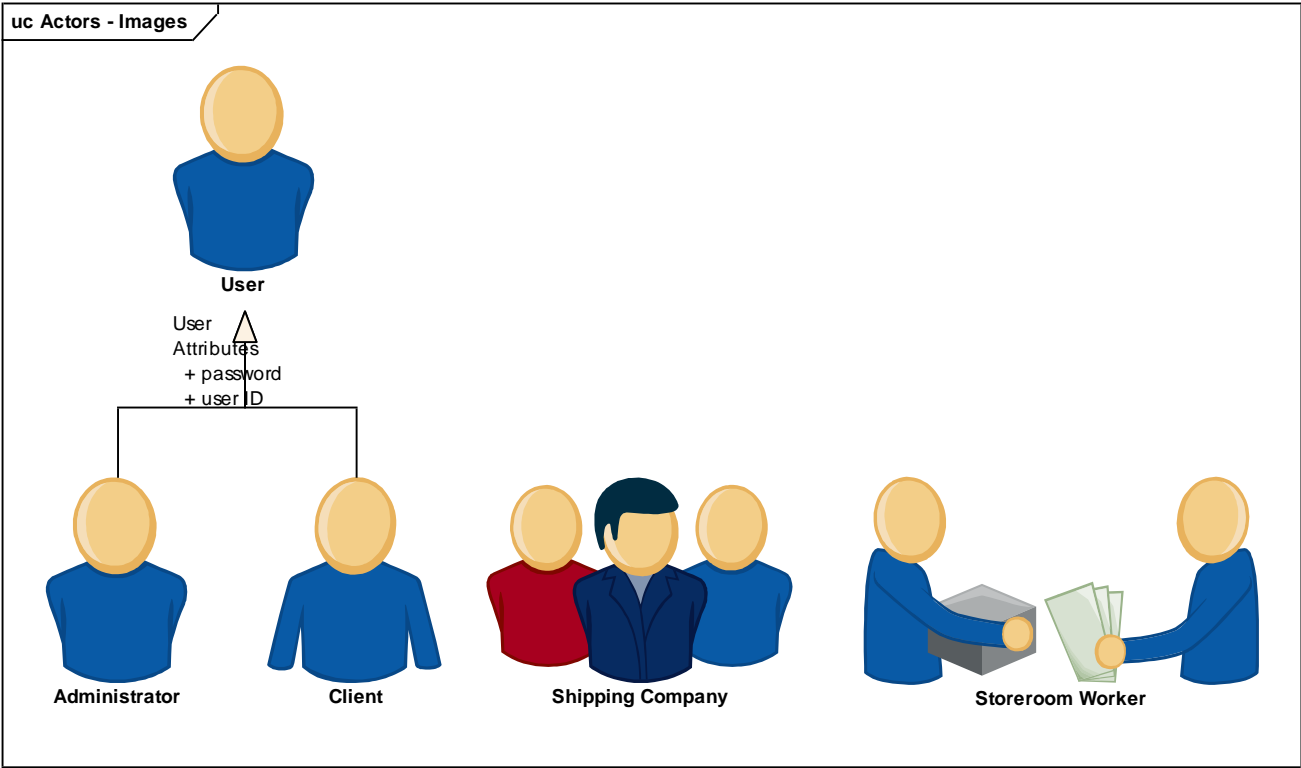


Diagram: Close Account

sd Close Account

Basic Path - Client

The use case starts when a Client selects the Close Account command from the Account Details Screen. The system will load the Client's account details and check whether the Client has any outstanding orders. If not, the Client is shown a Confirmation Dialog. If the Client replies yes, the account is marked as closed. Note that none of the details are removed from the system, for auditing purposes.

Basic Path - Administrator

This scenario is the same as "Basic Path - Client" except that the user is Administrator and instead of closing the account for the current user, it could be for any user of the system.

No to Close

If the user replies "No" to the "Are You Sure?" dialog, a message is displayed confirming that nothing has been changed and then the use case terminates.

Outstanding Transactions

If the account has any outstanding transactions, a message is displayed to the effect that the account cannot be closed, giving the reasons.



Sequence Diagram :
Close Account

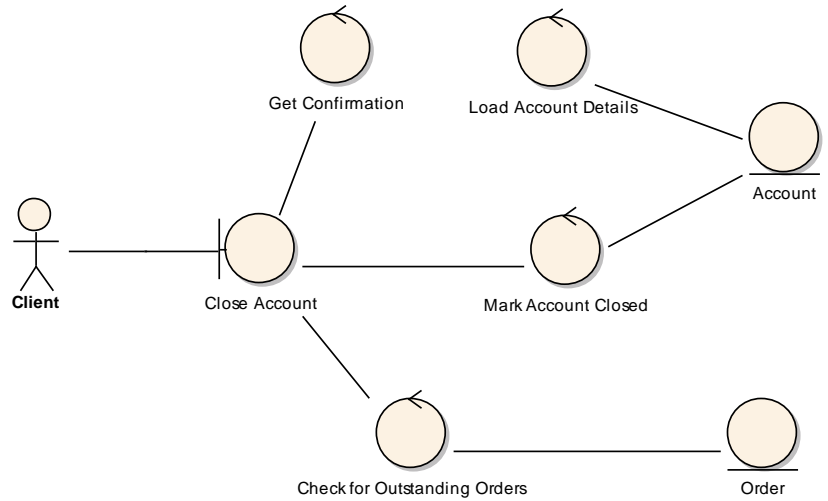


Diagram: Close Account

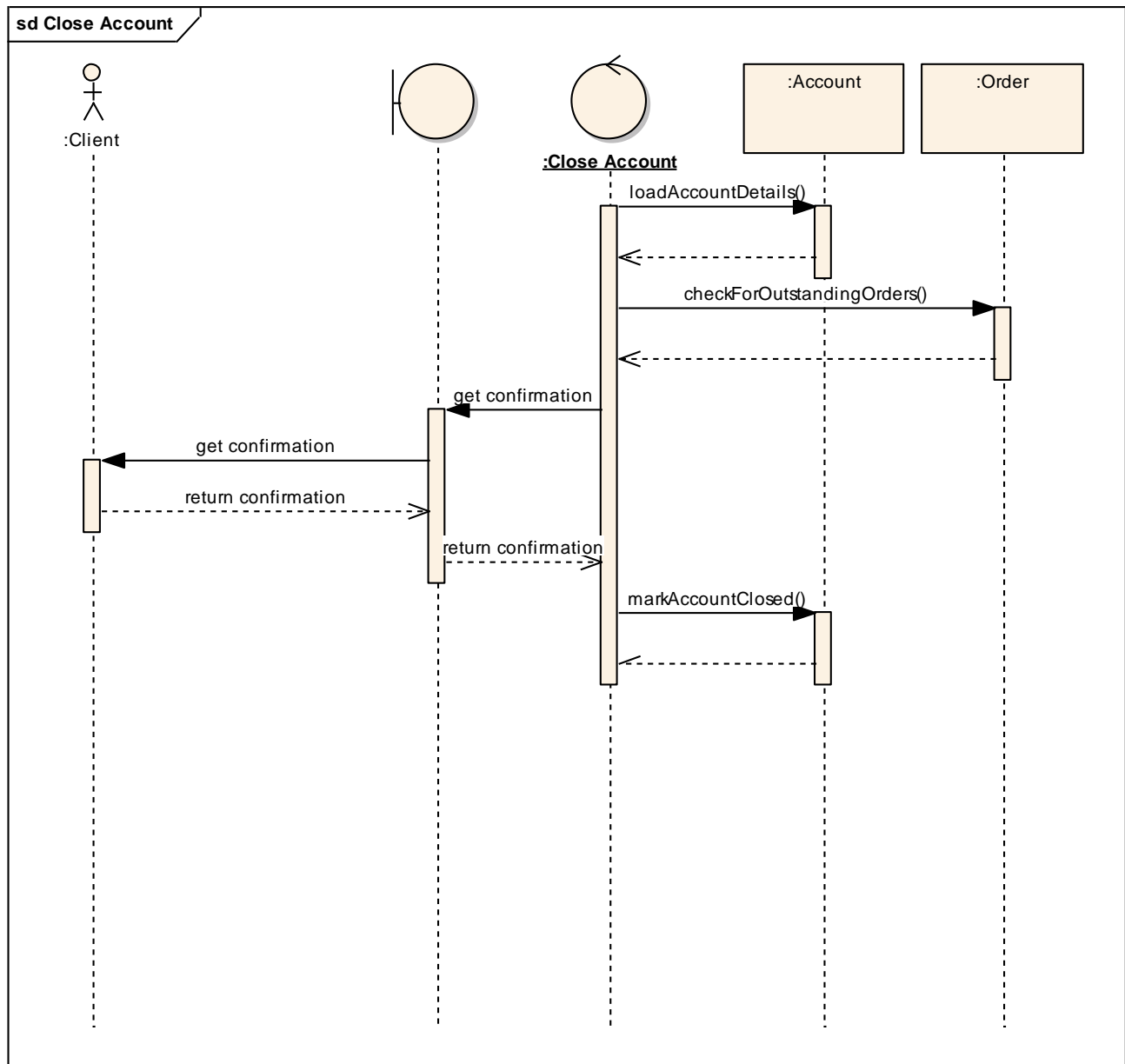


Diagram: Create Account

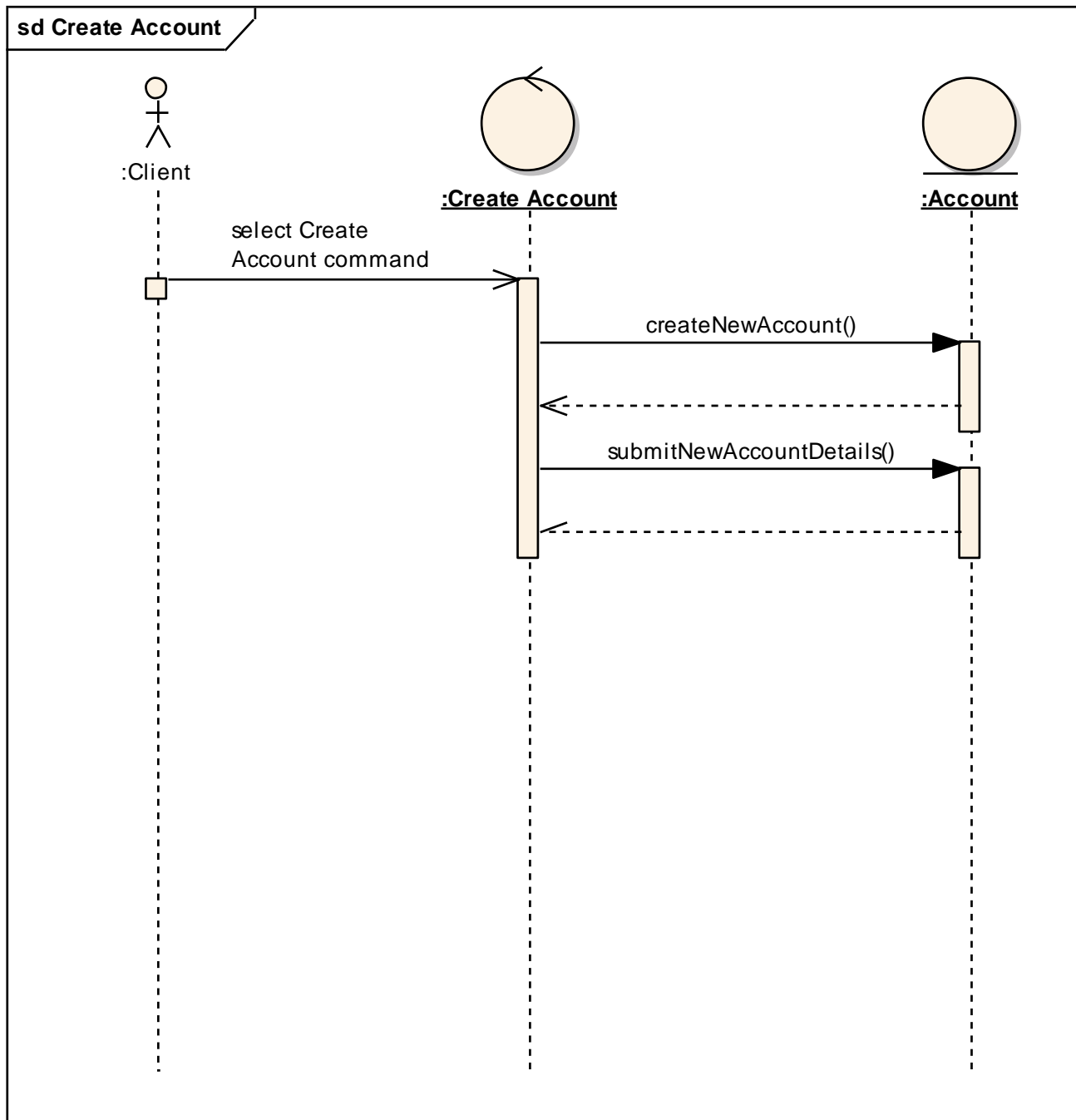


Diagram: Create Account

analysis Create Account

Basic Path

1. Use case begins when the Client selects the "Create Account" command.
2. The Client enters name, address, e-mail address, password and password confirmation.
3. Client presses Submit button.
4. Submitted details are validated.
5. Processing message is displayed to the Client.
6. New account is created in the database.
7. Success message is displayed to Client.

No Submit

The Client may press the Exit or Back button at step 3. The use case terminates with no action performed.

Validation fails

At step 4, if validation fails then an error message is displayed and the Client is returned to step 2 with the invalid field or fields highlighted.

Cannot Create New Account

In step 6, if the database create fails, the Client is informed and the use case terminates.



Sequence Diagram :
Create Account

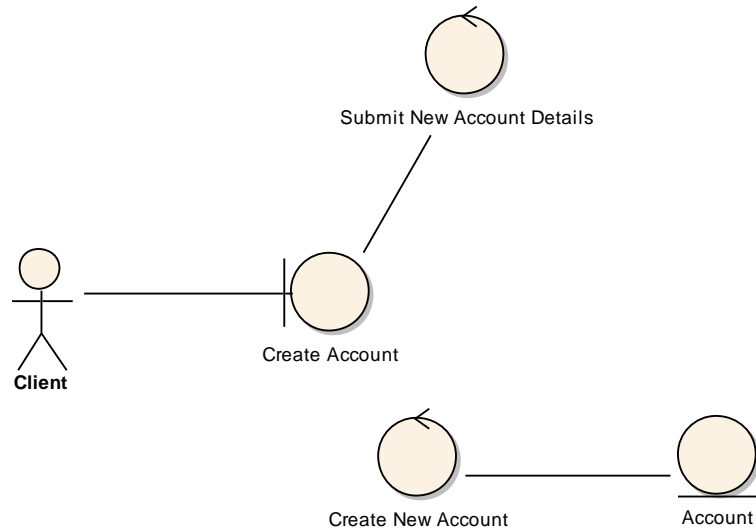


Diagram: Delete User

analysis Delete User

Basic Path

This use case begins when an Administrator selects the Delete User command. The Administrator is prompted to enter a user ID. The ID is used to retrieve the user's name and address which are displayed on the screen. The administrator is then asked to confirm that they want to delete this particular user and asked to re-enter their password as an added security measure. The Close Account use case is executed to perform the closure and that use case will display the success or failure notification. The administrator is able to enter another user ID and repeat the process. The use case terminates when the administrator selects the Back or Exit button.

Bad ID

If the user ID doesn't match any existing account, an error message is displayed and the administrator is prompted to re-enter it.

Bad password

If the administrator enters an incorrect password, an error message is displayed and the administrator is allowed a maximum of two more attempts before the use case terminates.

No confirm

If the administrator answers No to the Are You Sure request, it is assumed that the wrong user ID was entered so the user ID is cleared and the administrator is permitted to try again.



Sequence Diagram :
Delete User

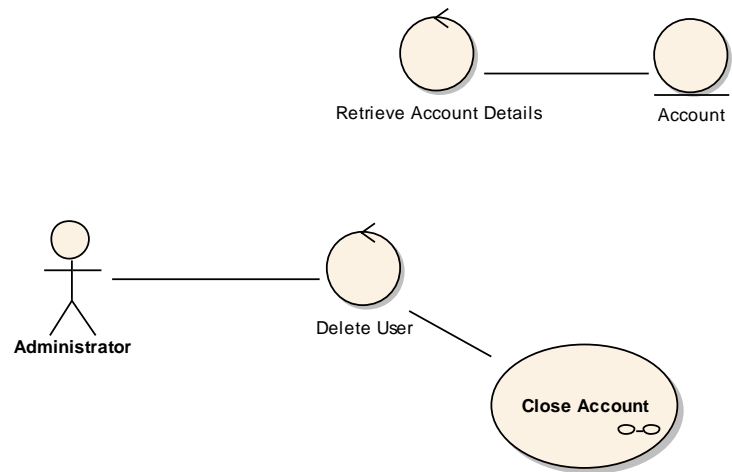


Diagram: Delete User

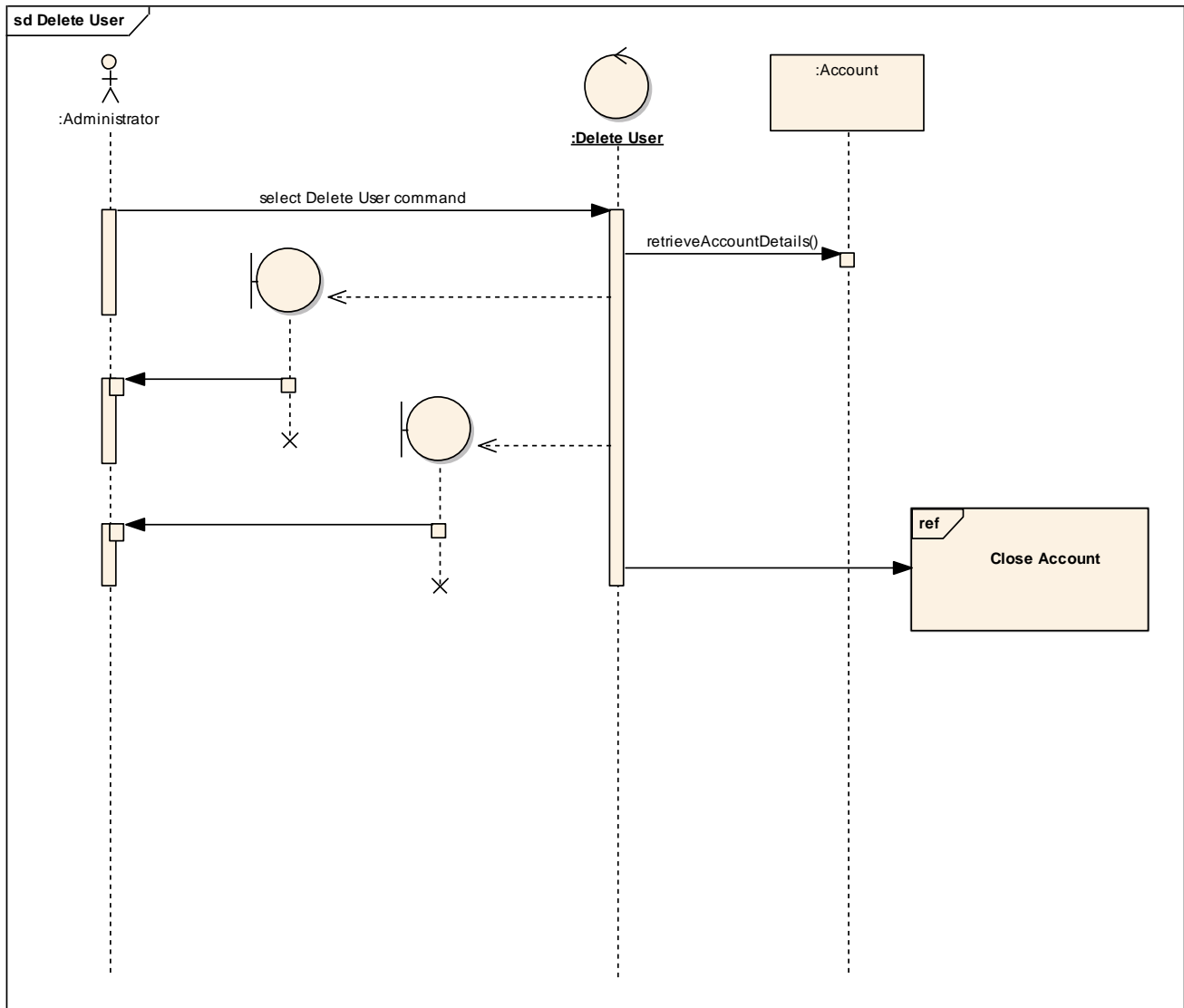


Diagram: Login

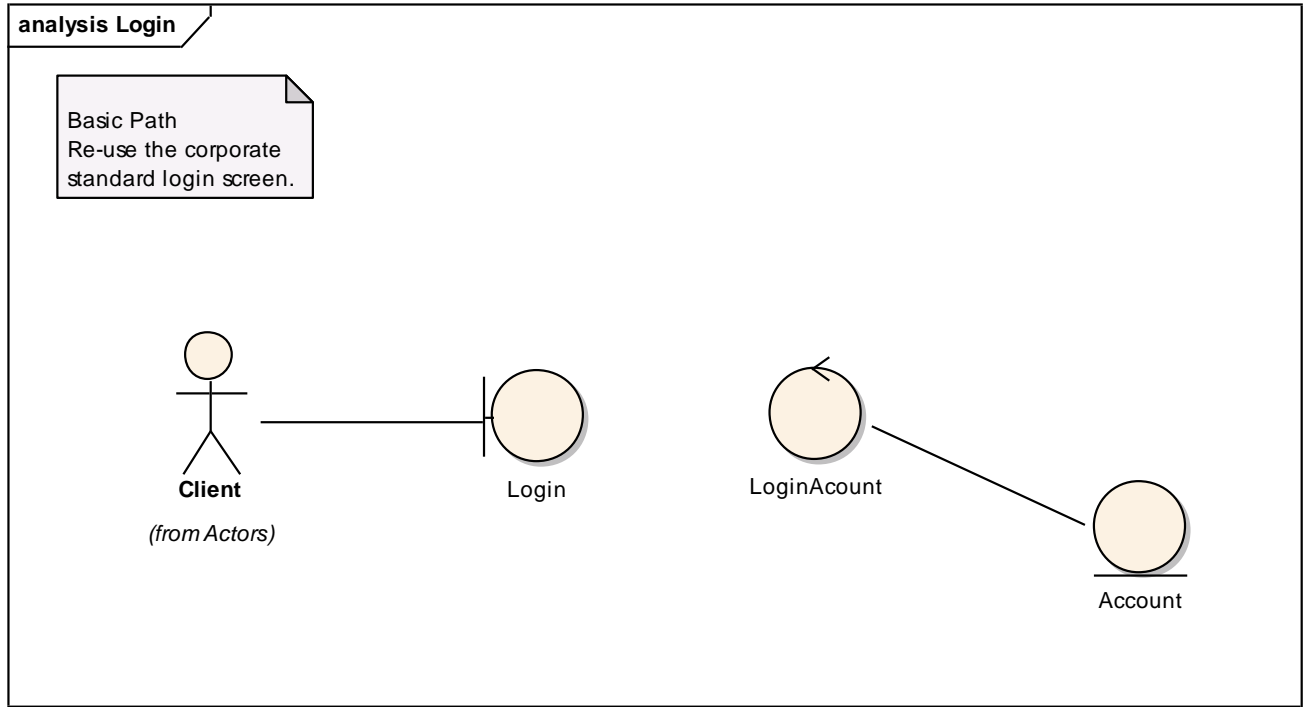


Diagram: Login

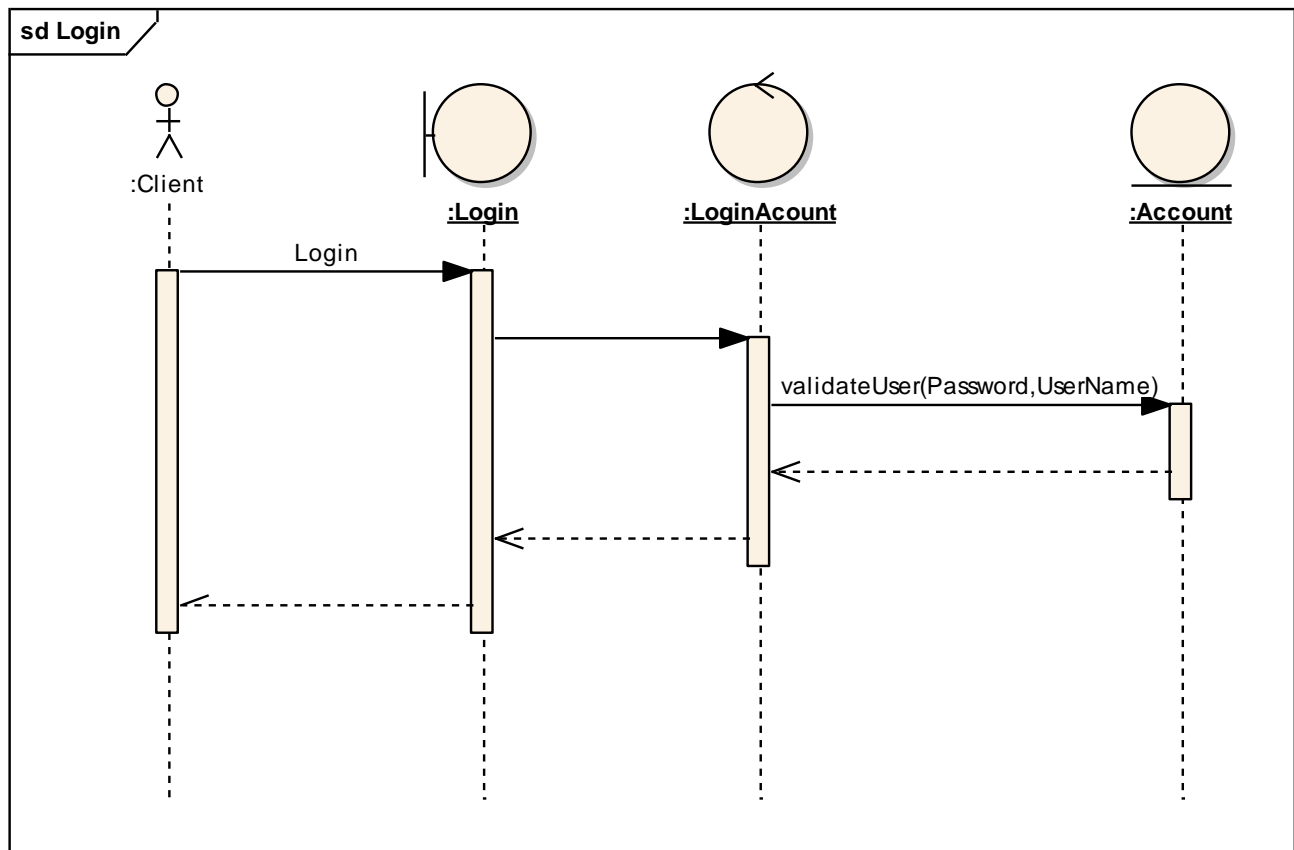


Diagram: Manage Users

uc Manage Users

The following diagram illustrates the Use Cases relating to Managing Users. Some of the use cases contain nested Sequence and Communication diagrams. You can double-click these use cases to view the child diagrams

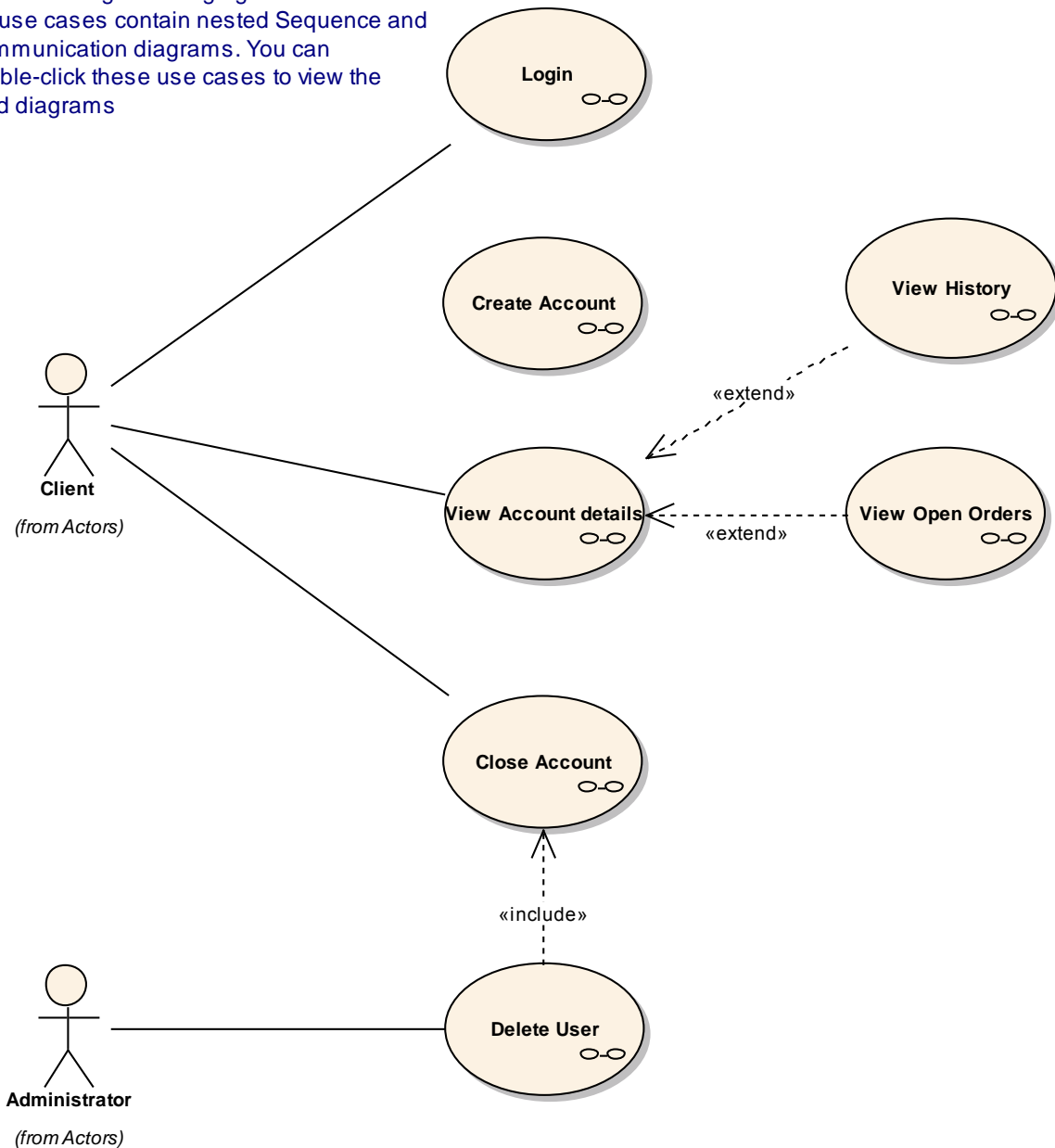


Diagram: View Account details

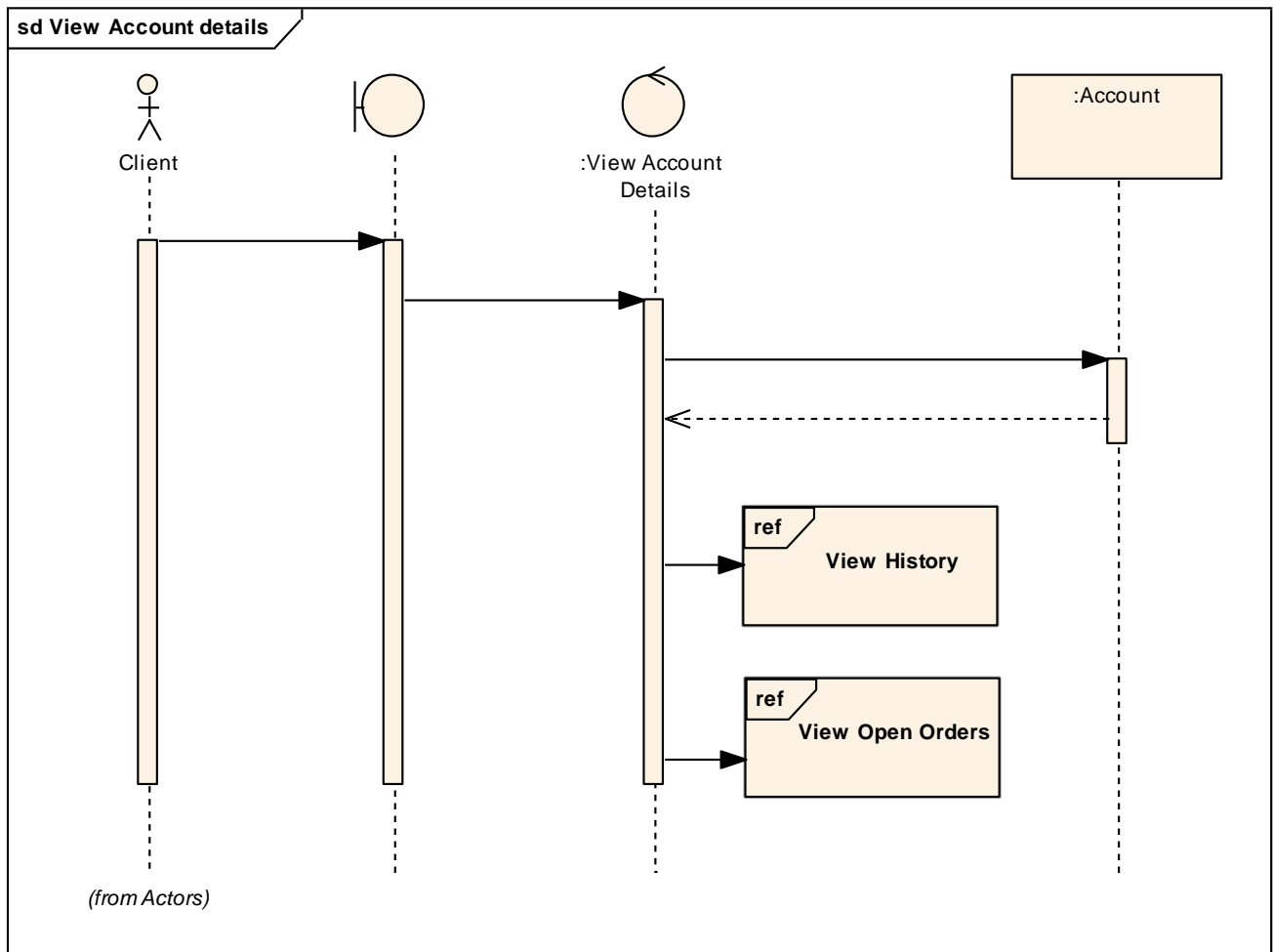


Diagram: View Account details

analysis View Account details

Basic Path

This use case begins when a Client requests a display of their account details. General account information is displayed (i.e. ID, username, billing address, delivery address etc). Command buttons are displayed to allow the user to view Open Orders or History. The use case terminates when the user selects the Exit or Back button.

View Open Orders

If the View Open Orders command is selected, the View Open Orders use case is executed.

View History

If the user selects the View History command, the View History use case is executed.



Sequence Diagram :
View Account details

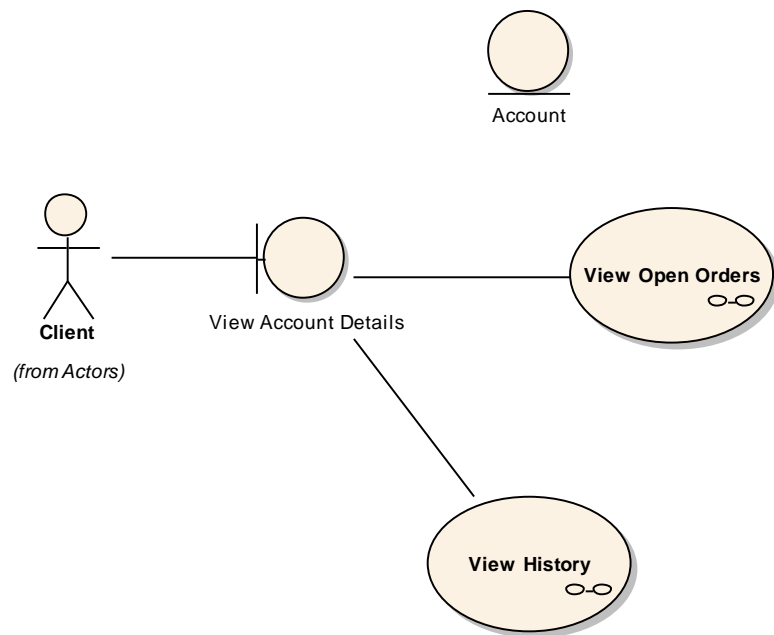


Diagram: View History

analysis View History

Basic Path

This use case begins when the user requests to view a history of transactions against their account. The account ID is used as a key to lookup the appropriate records in the database. The results are then displayed sorted in date order.

No History

If the database search finds no previous transactions, then you don't display a sorted list of nothing; instead you put a message saying nothing was found. Note there is no need to put up an error message dialog.



Sequence Diagram
: View History

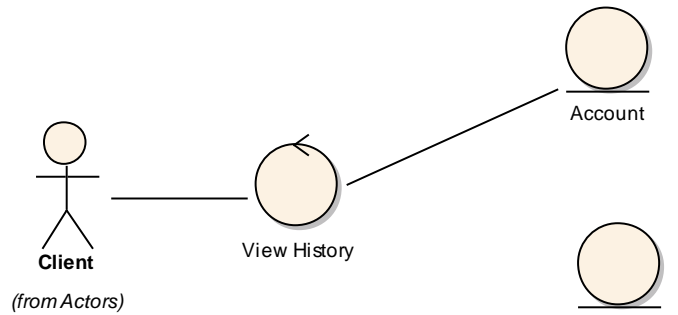


Diagram: View History

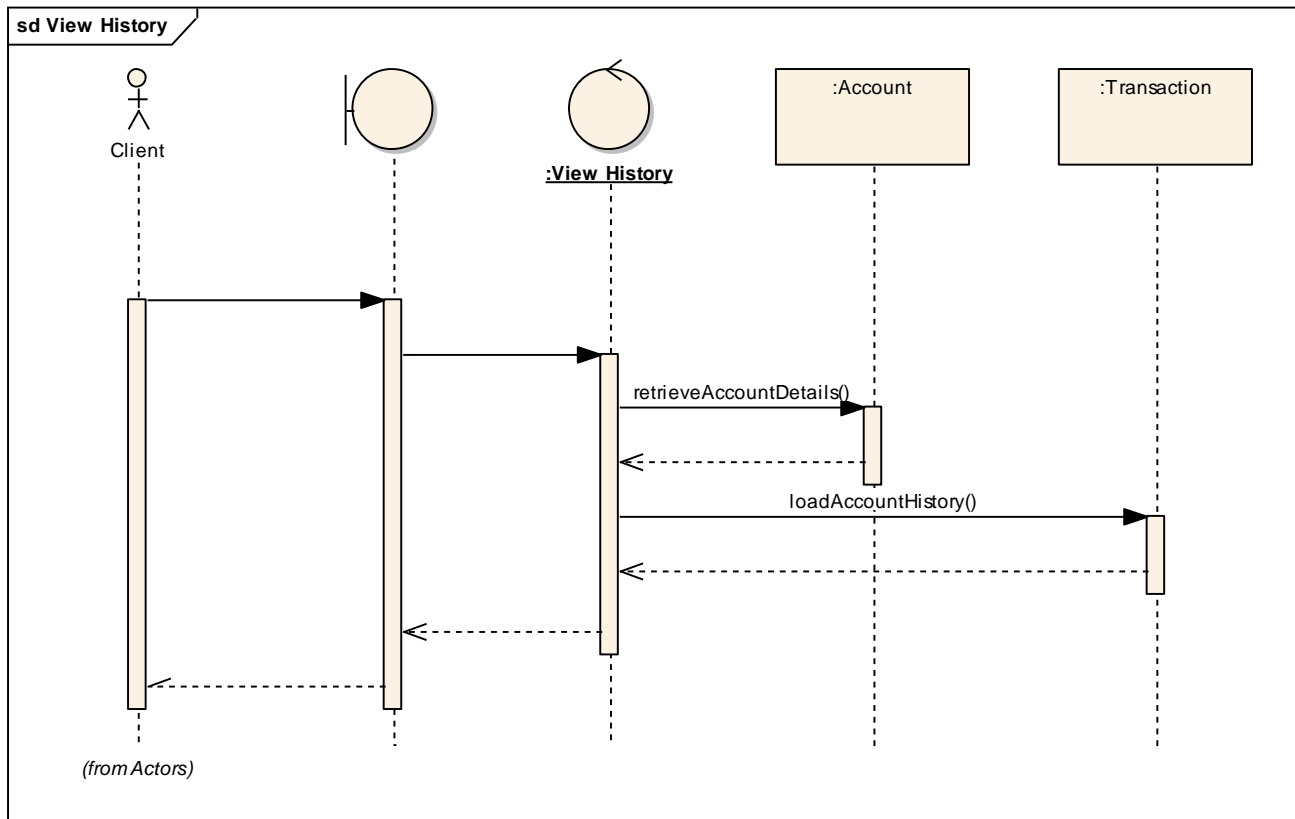


Diagram: View Open Orders

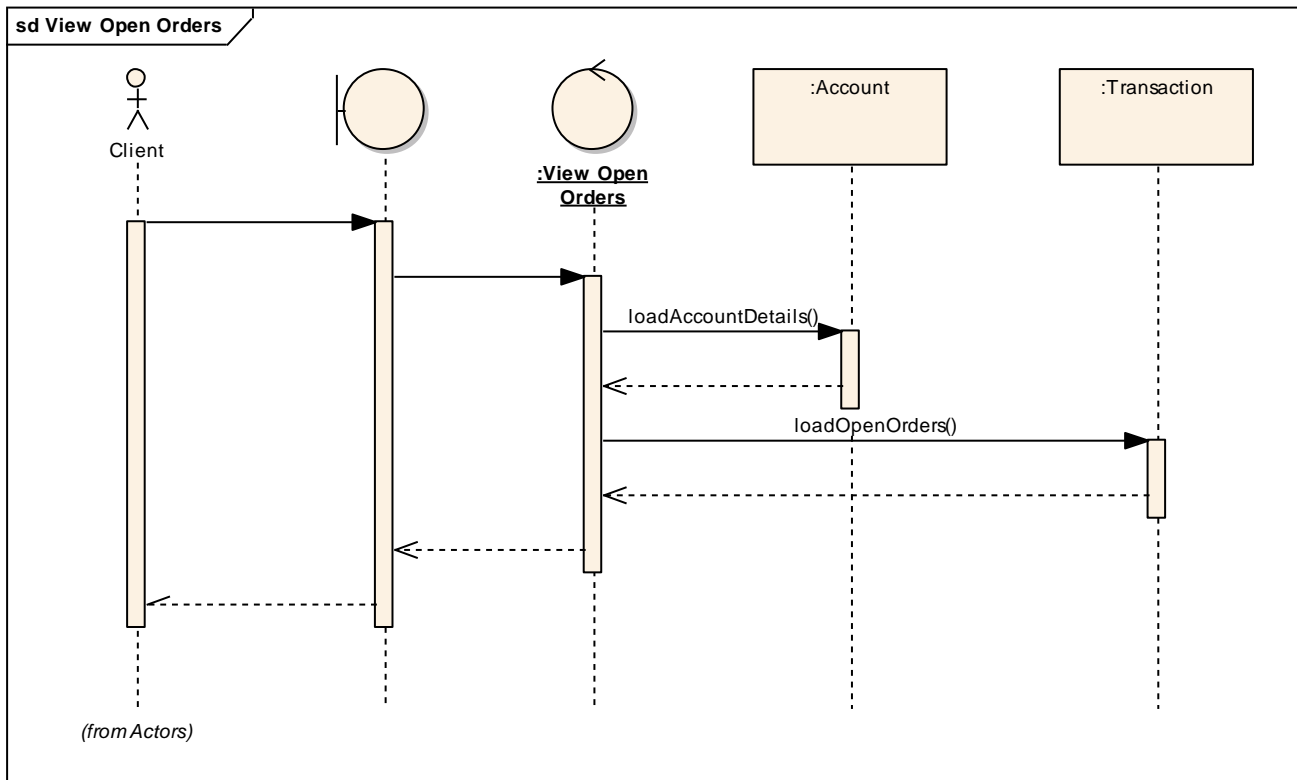


Diagram: View Open Orders

analysis View Open Orders

Basic Path

This use case begins when the user request to view a list of current transactions against their account. The account ID is used as a key to lookup the appropriate records in the database. The results are then displayed sorted in date order.

No Current Transactions

If the database search finds no current transactions, then you don't display a sorted list of nothing: instead you put a message saying nothing was found. Note there is no need to put up an error message dialog.



Sequence Diagram :
View Open Orders

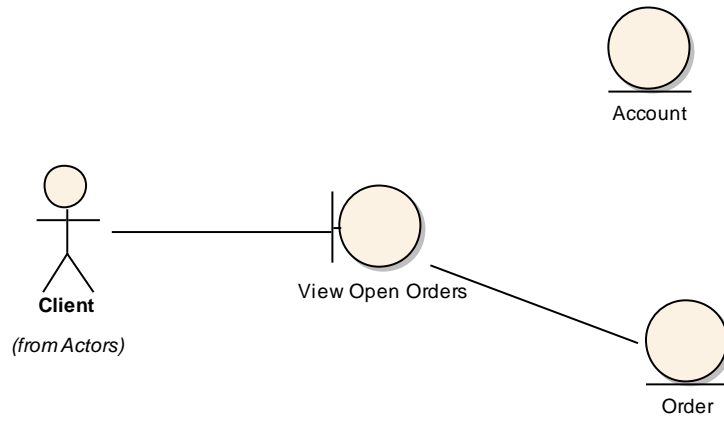


Diagram: Create Orders

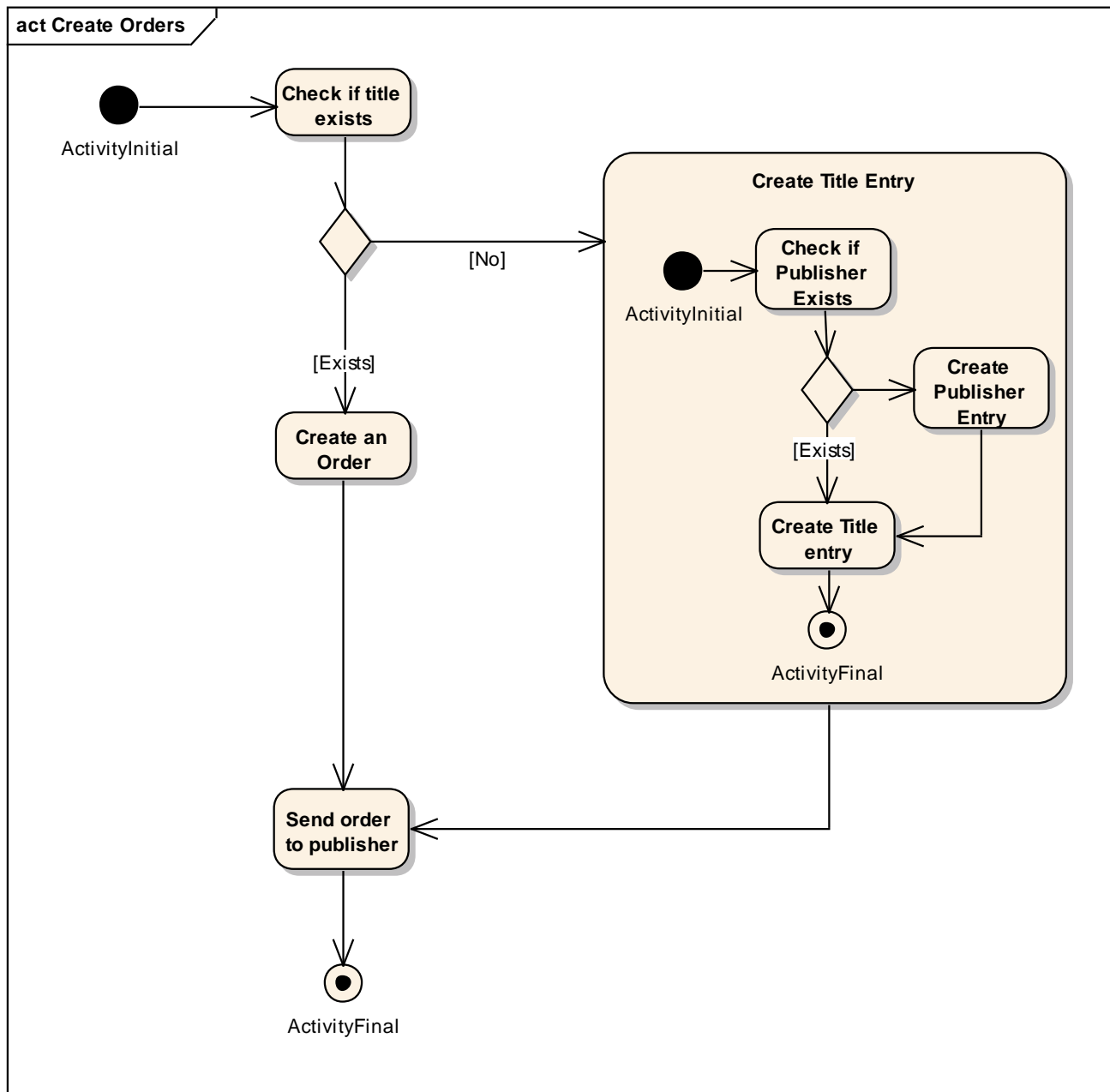


Diagram: Manage Inventory

uc Manage Inventory

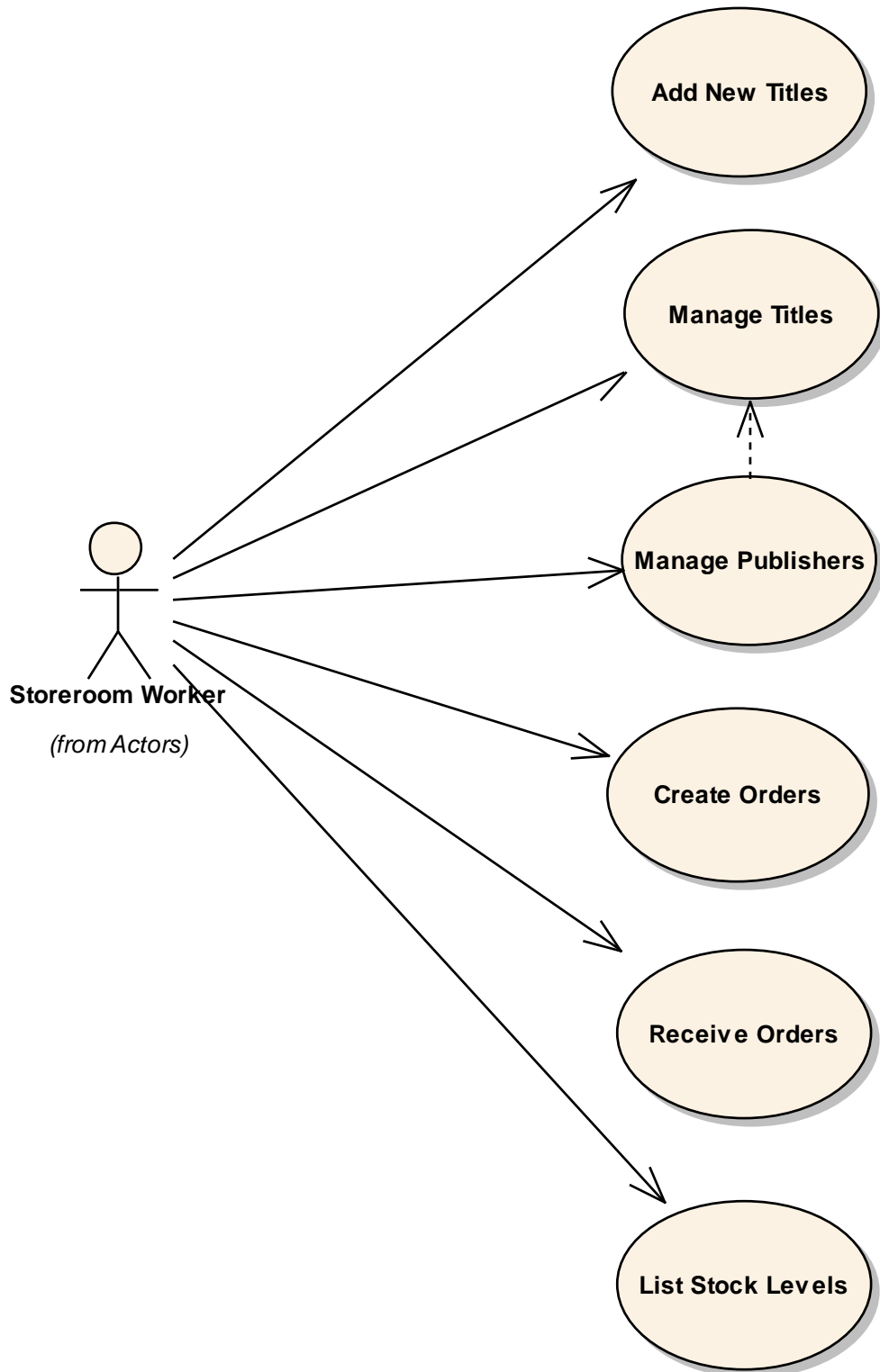


Diagram: Manage Titles State

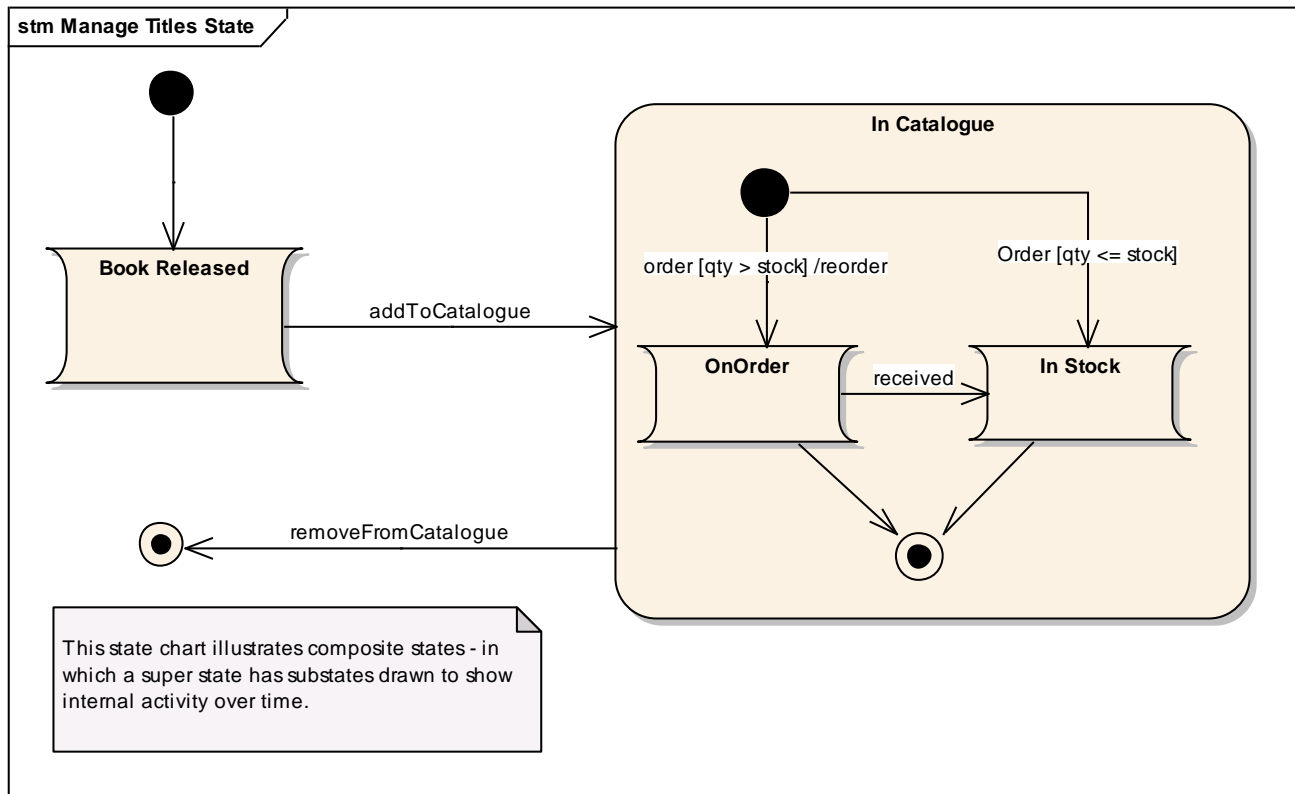


Diagram: Add To Shopping Cart

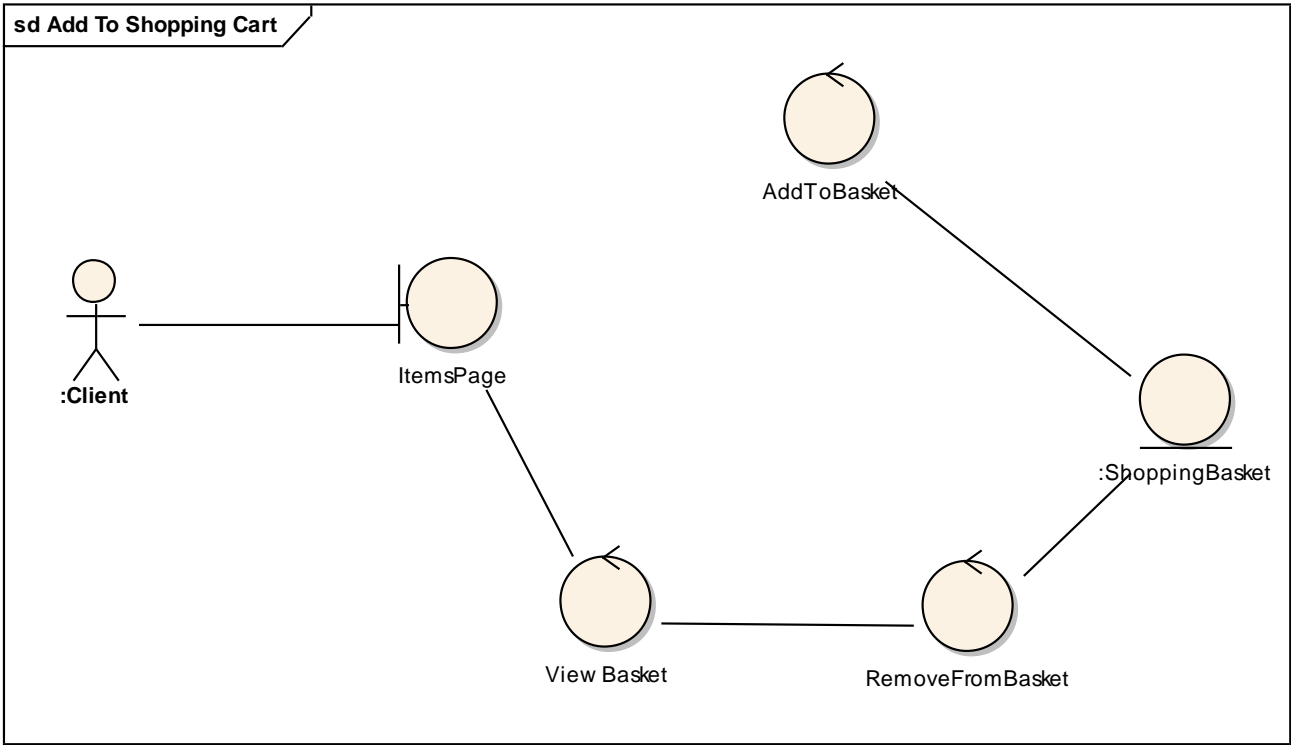


Diagram: Add To Shopping Cart

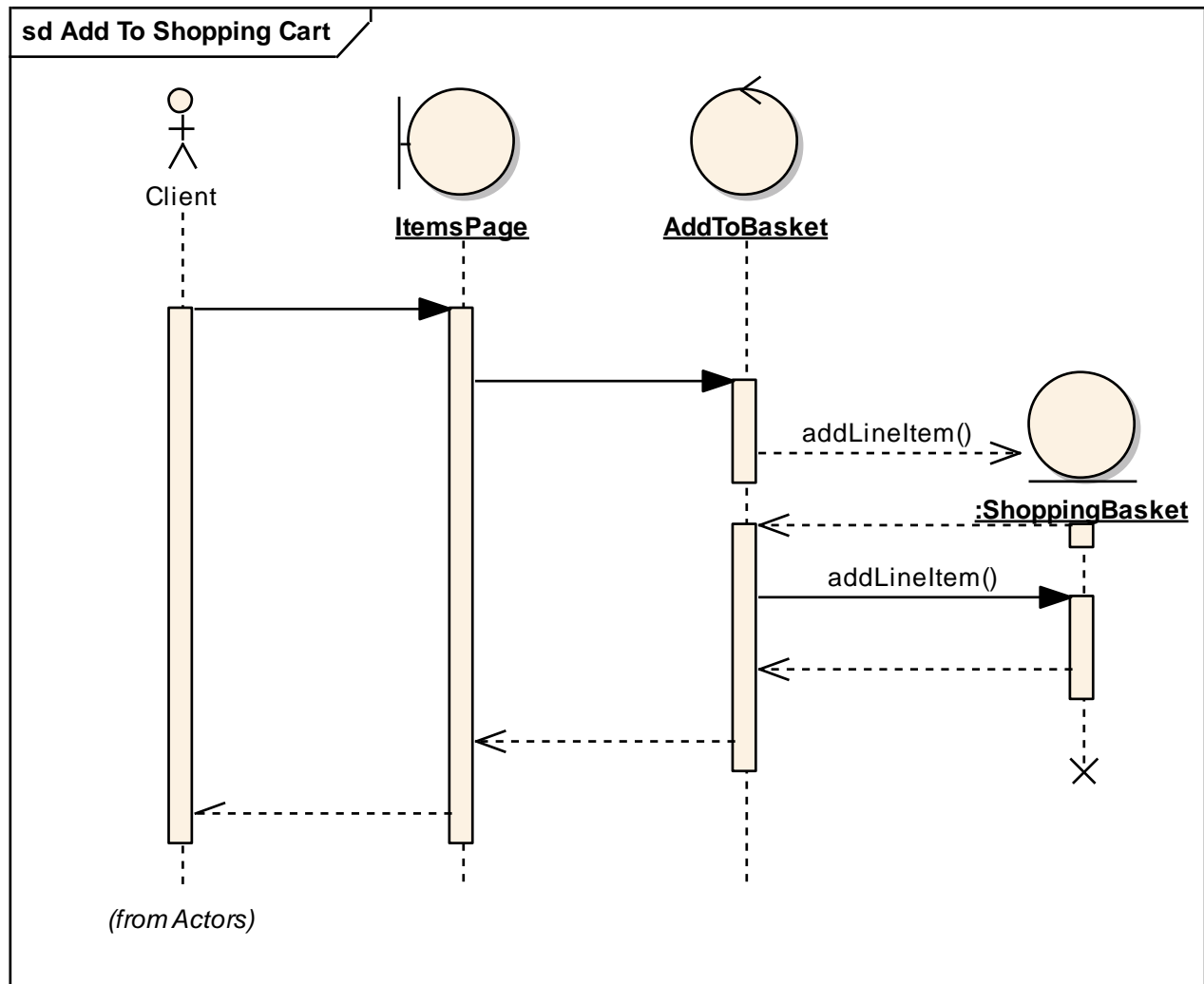


Diagram: Sell Books

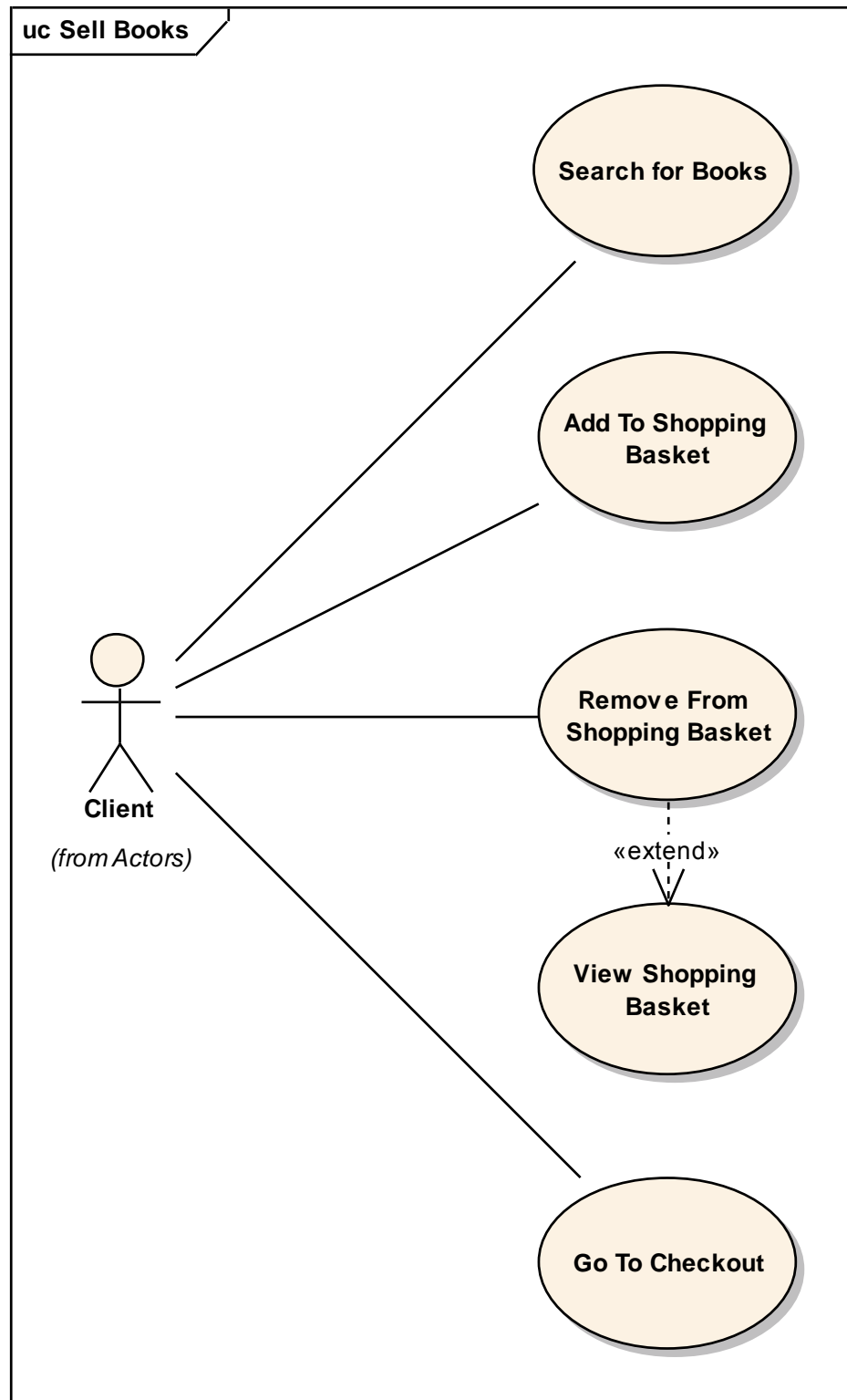


Diagram: Credit Card Problem

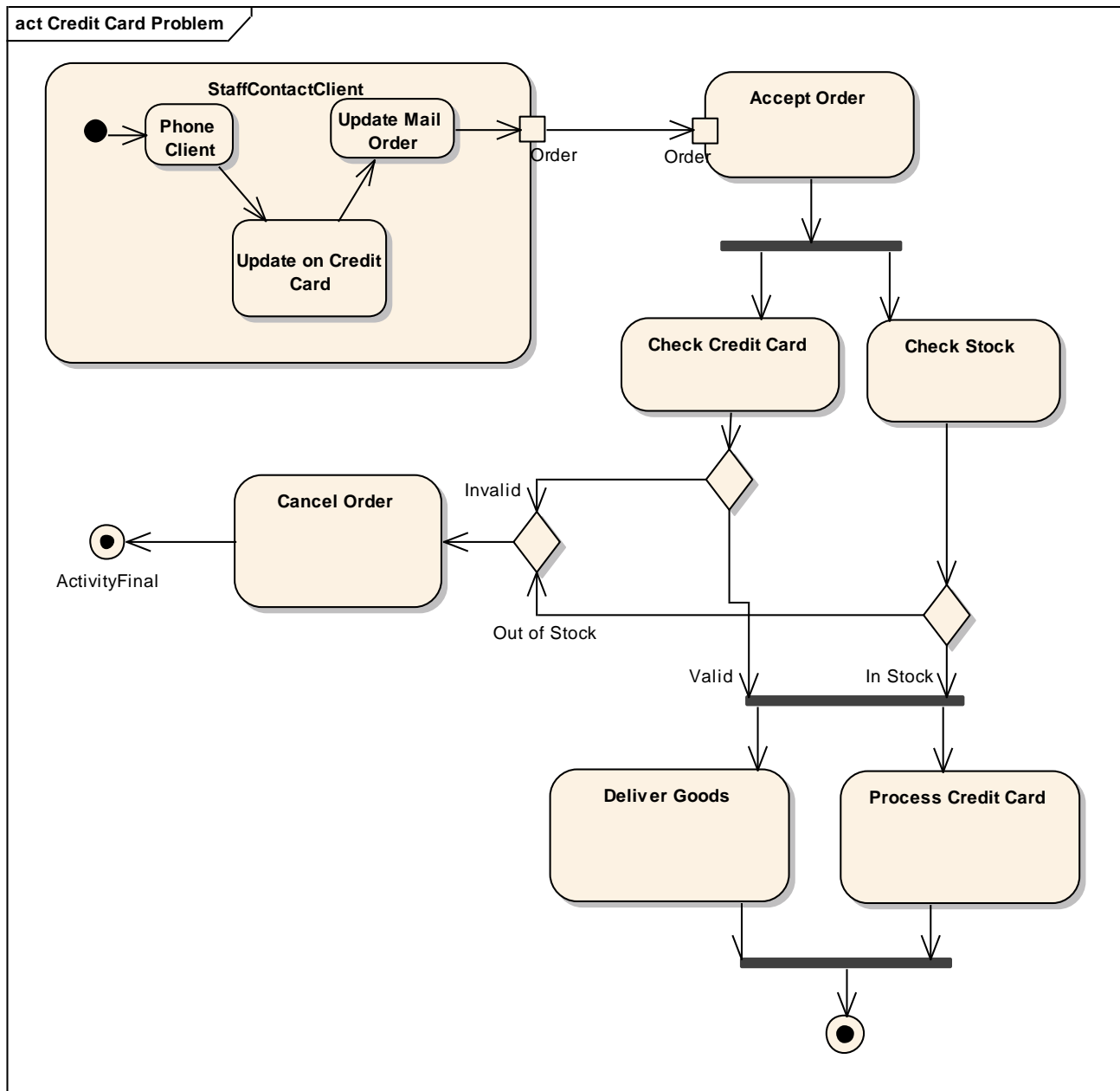


Diagram: Deliver Books

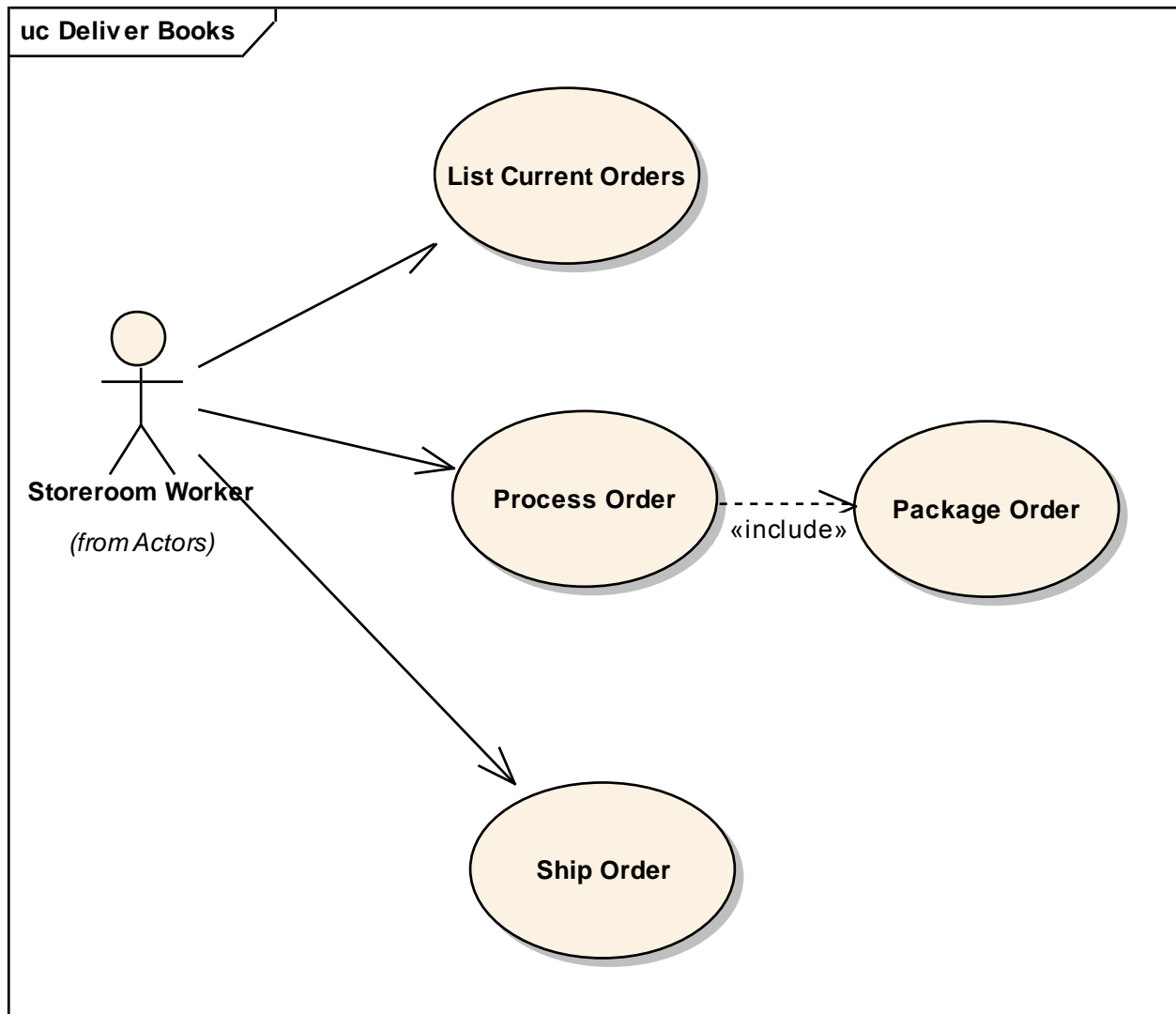


Diagram: Invoice Payment

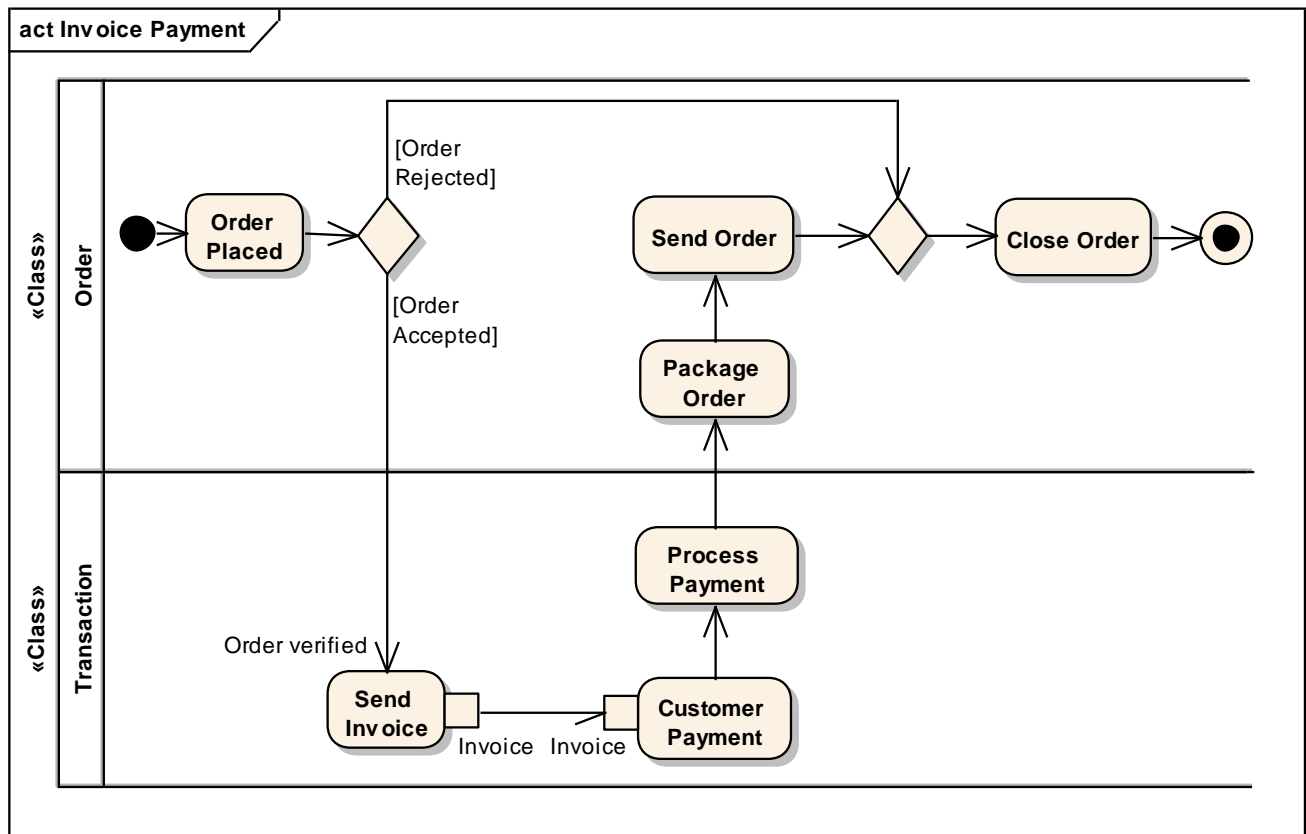


Diagram: Customer Process

act Customer Process

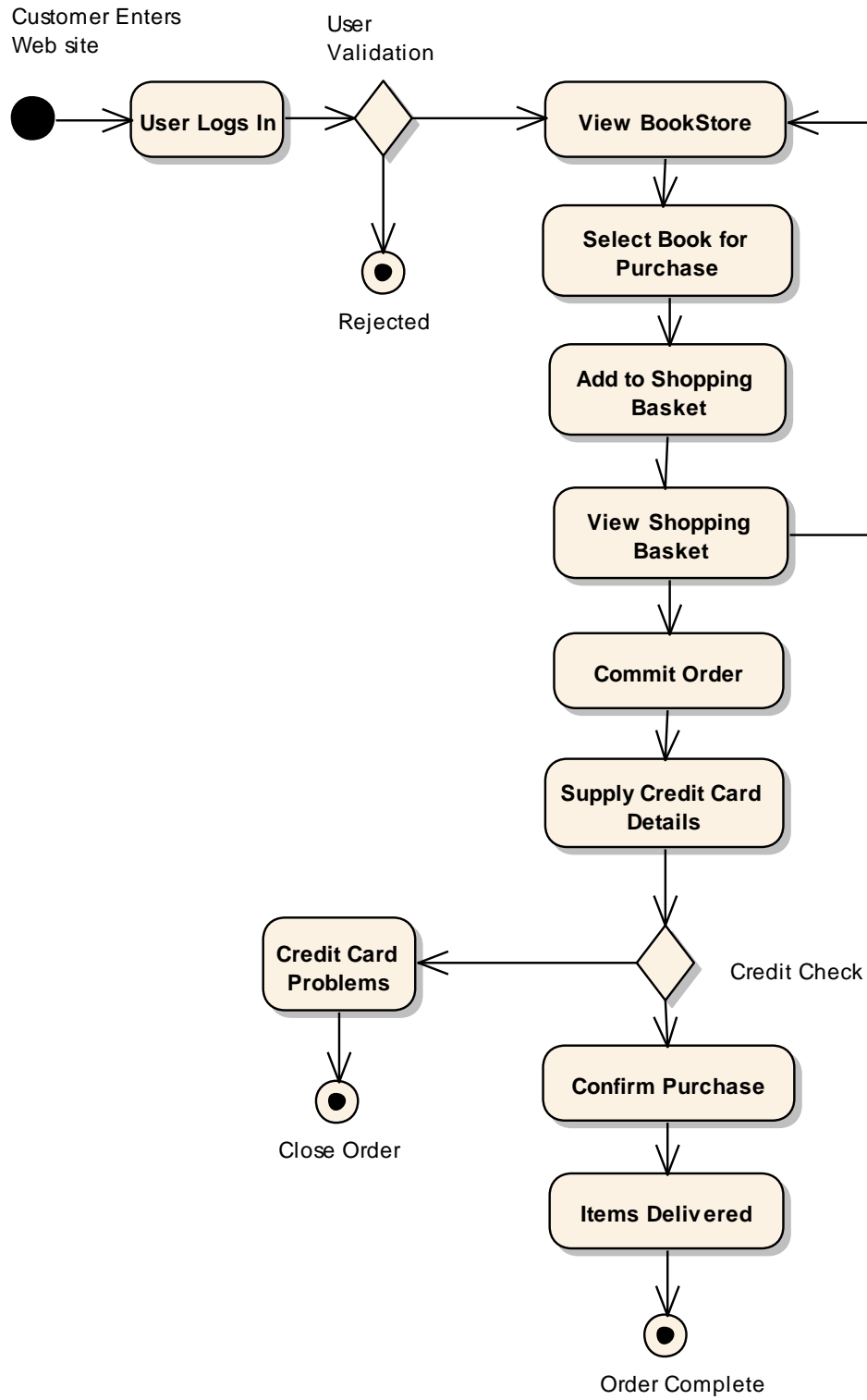


Diagram: InterruptibleActivityRegion

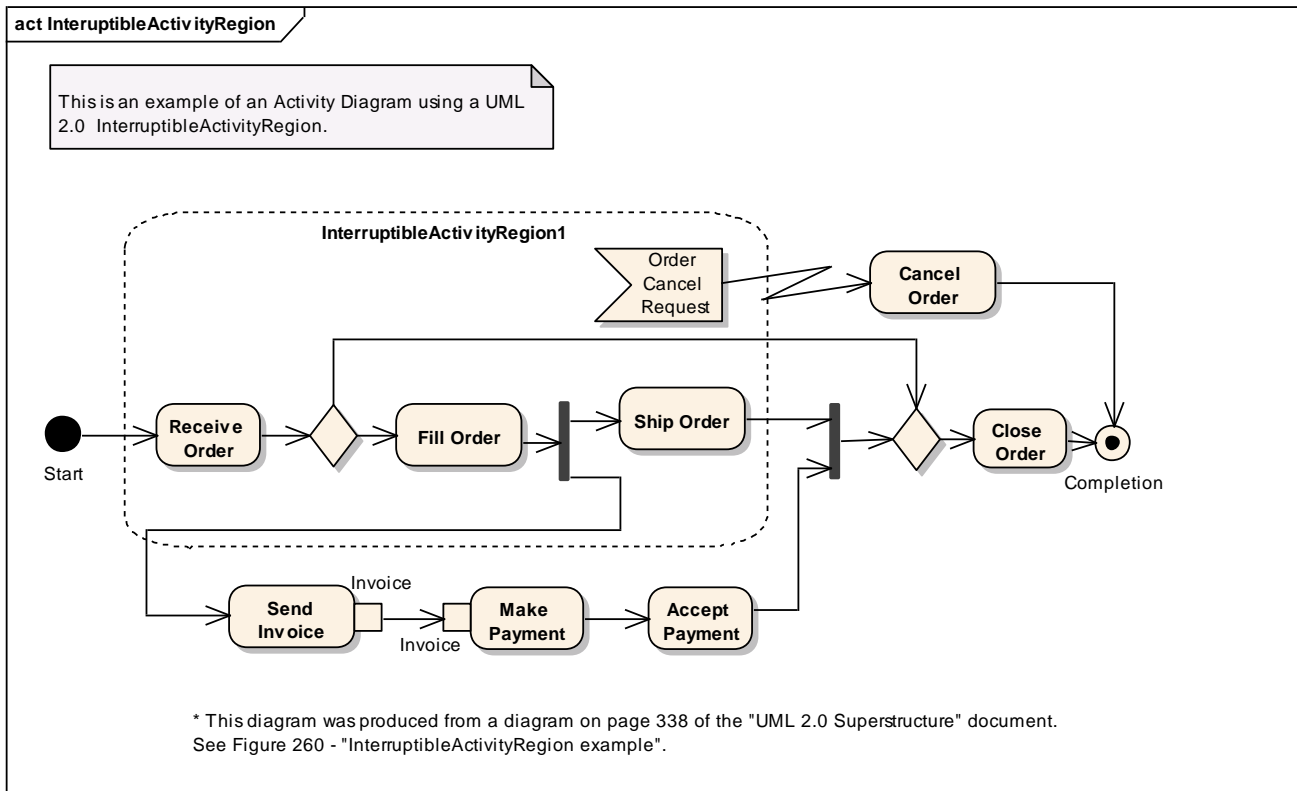


Diagram: Manage Inventory

custom Manage Inventory

These relationships were created using the Relationship Matrix. This diagram is simply a set of elements from the Requirements section and the Use Case section dragged onto this diagram.

The Relationship Matrix is accessible from the main menu - under: View | Relationship Matrix.

To view the relationship Matrix for the two packages that contain these elements - open:



Manage Inventory

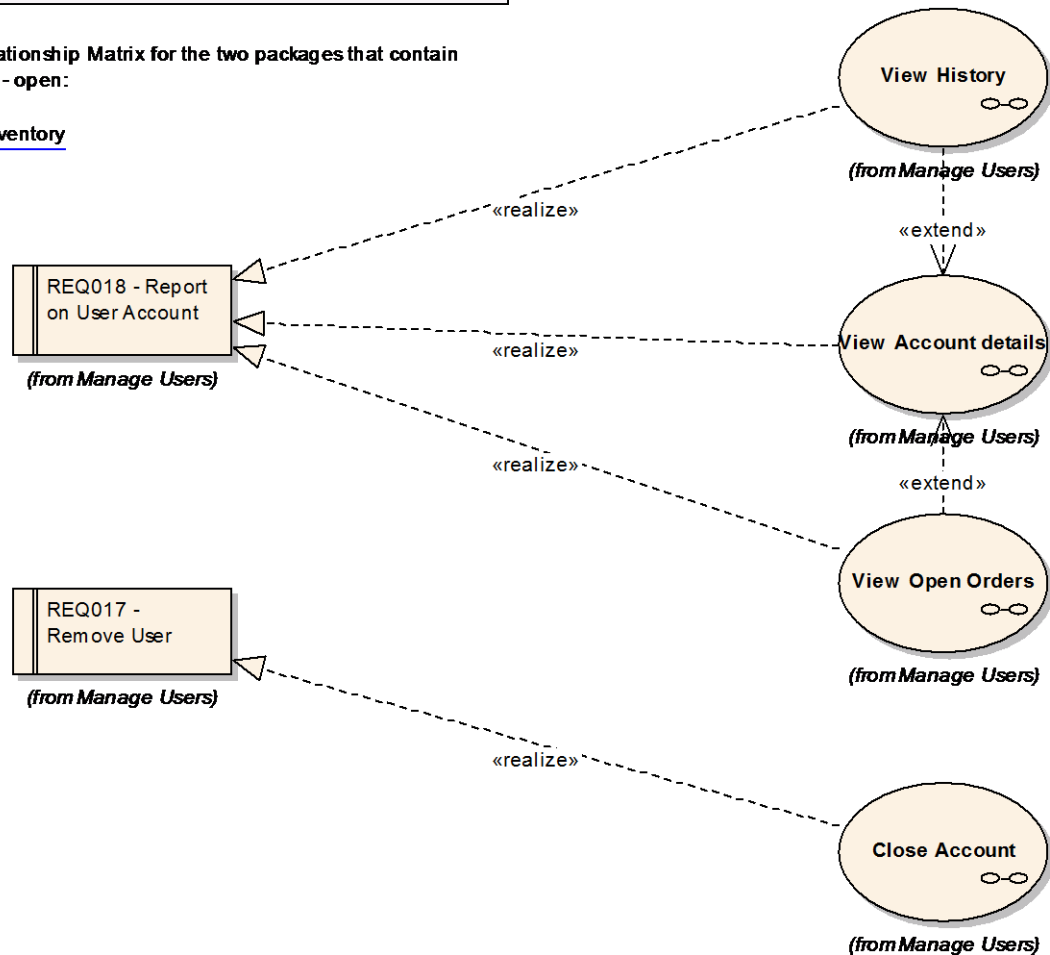


Diagram: ManageUsers

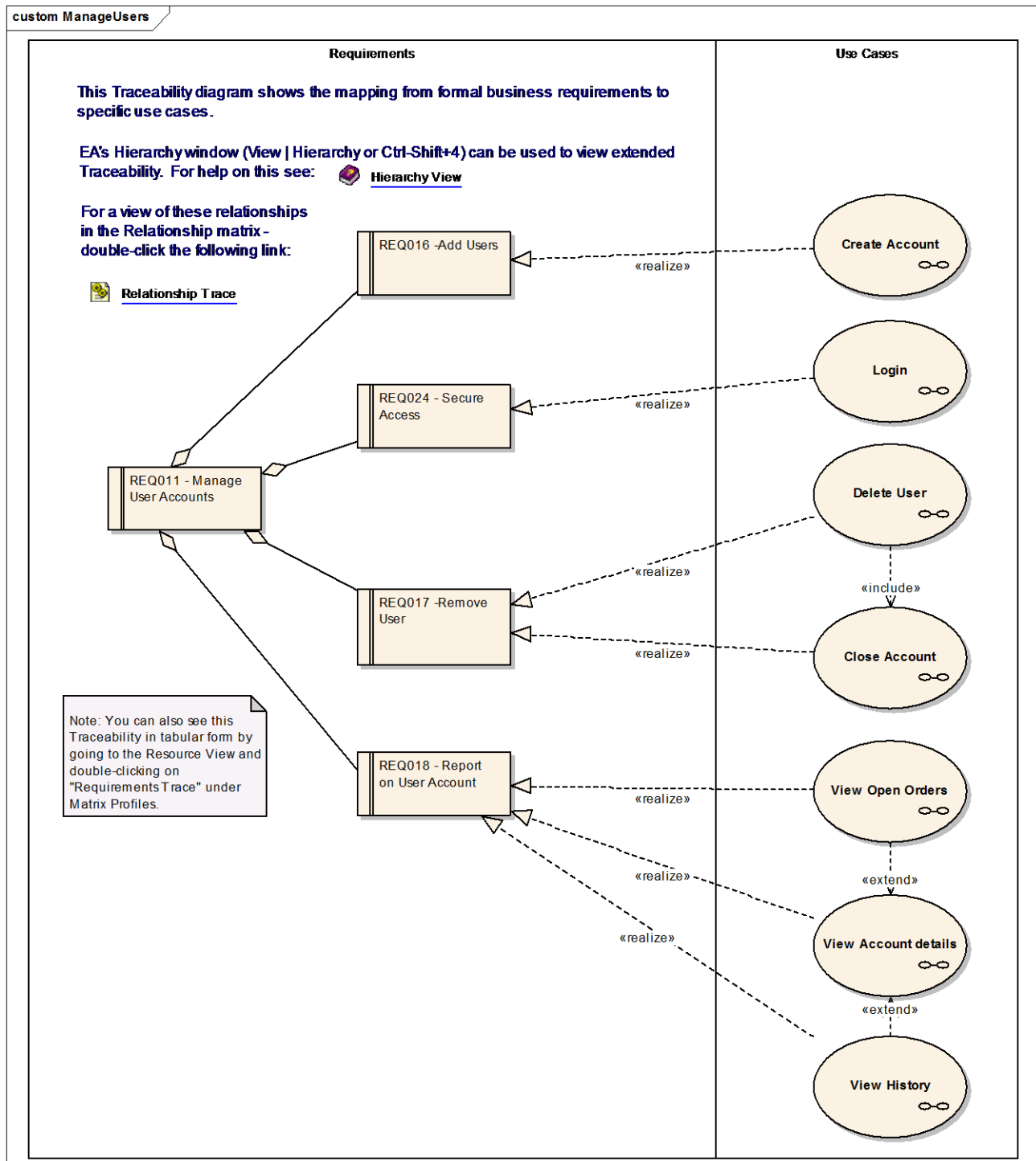


Diagram: Non-Functional Requirements Model

pkg Non-Functional Requirements Model

The Non-Functional requirements are used to state the set of general requirements that are more defined on the business level rather than the functional level. Below are some examples of these.

Extensibility

- + REQ100 - System must be easily extendible
- + REQ101 - Other product types options can be added easily.
- + REQ102 - System must be able to cope with regular retail sales

Legal and Regulatory

- + REQ103 - Orders and dispatch information must be kept for seven years.
- + REQ104 - Non storage of customer credit card details

Reliability

- + REQ112 - 2000 hours mean time between failure.
- + REQ113 - Must be recoverable quickly.
- + REQ114 - 99.999% accuracy.
- + REQ115- 99.999% precision.

Security

- + REQ108 - Processed information must be kept secure.
- + REQ109 - All transactions must be secure.
- + REQ110- Wherever possible existing security definitions should be used.
- + REQ111 - Physical storage locations should be secure.

Diagram: Extensibility Main

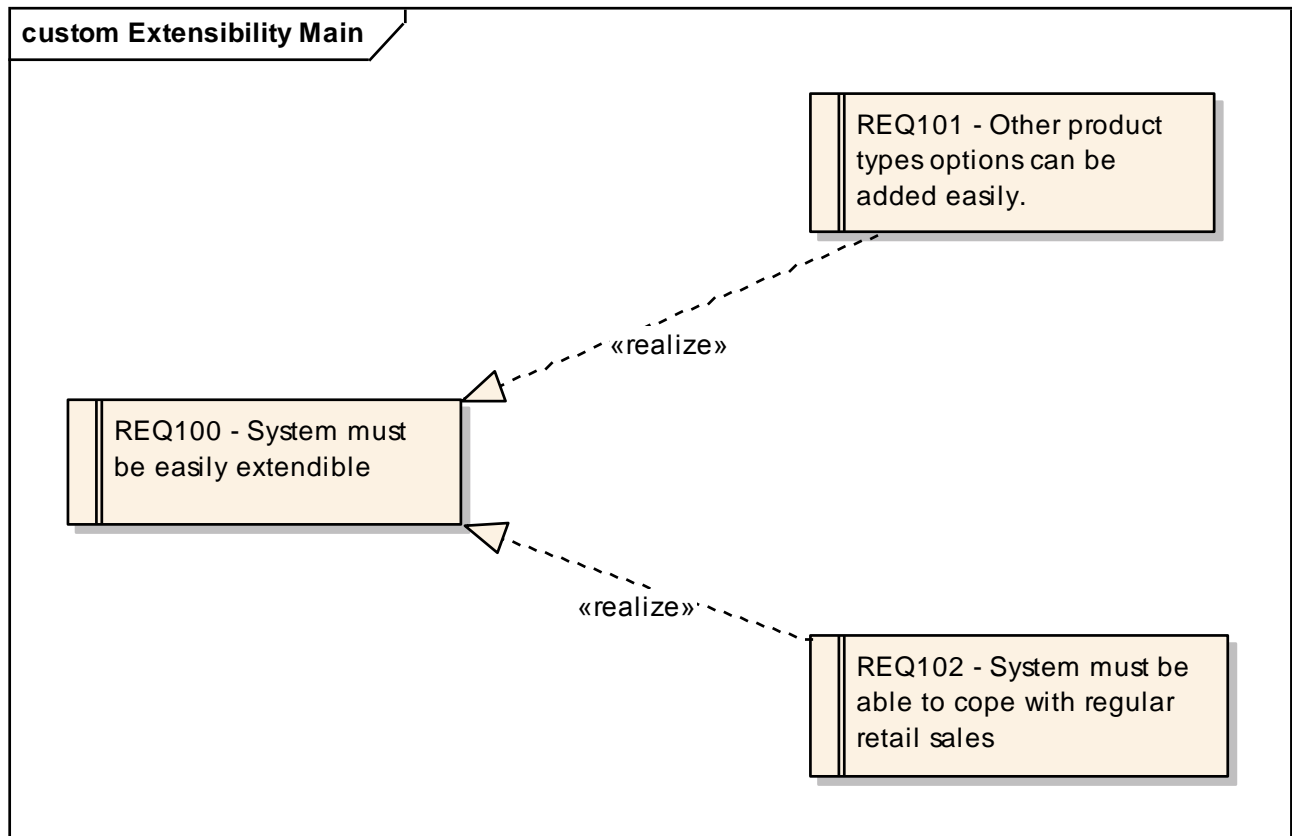


Diagram: Legal and Regulatory

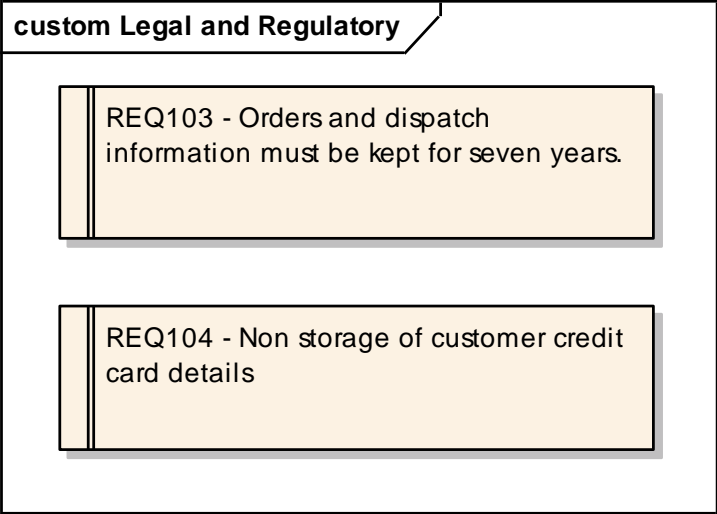


Diagram: Performance Main

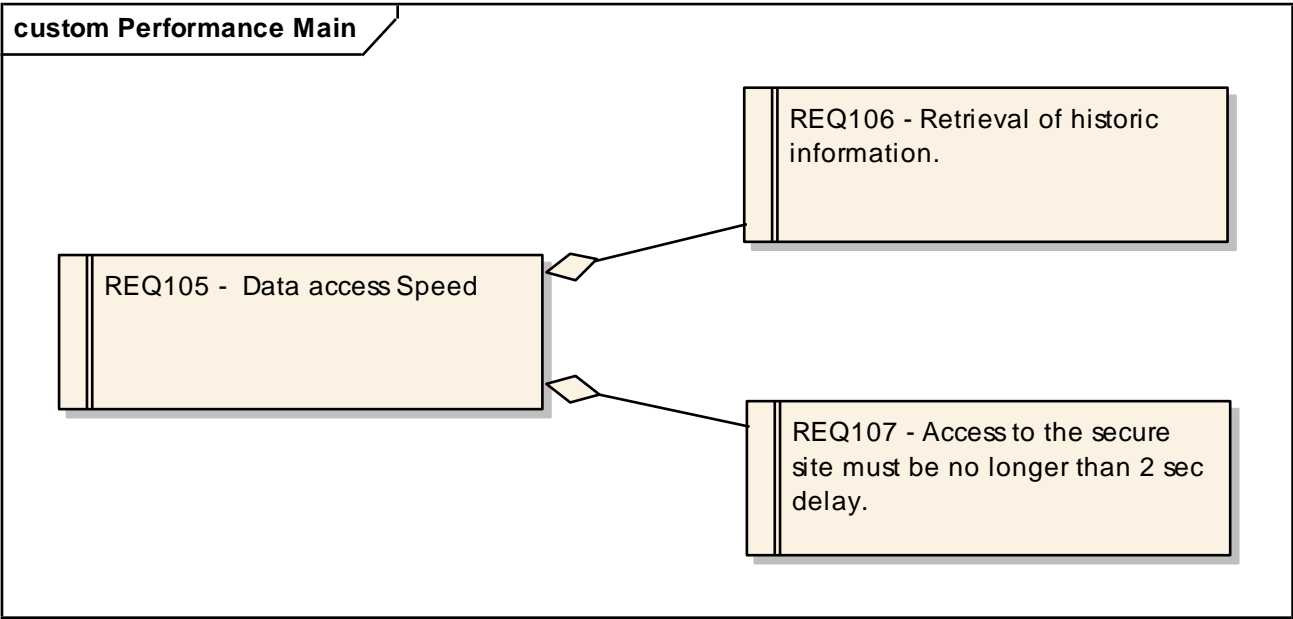


Diagram: Reliability Main

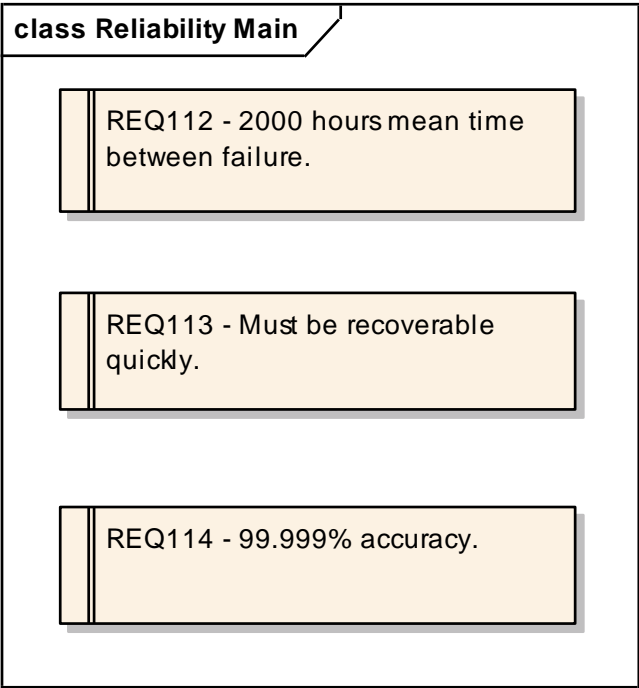


Diagram: Security Main

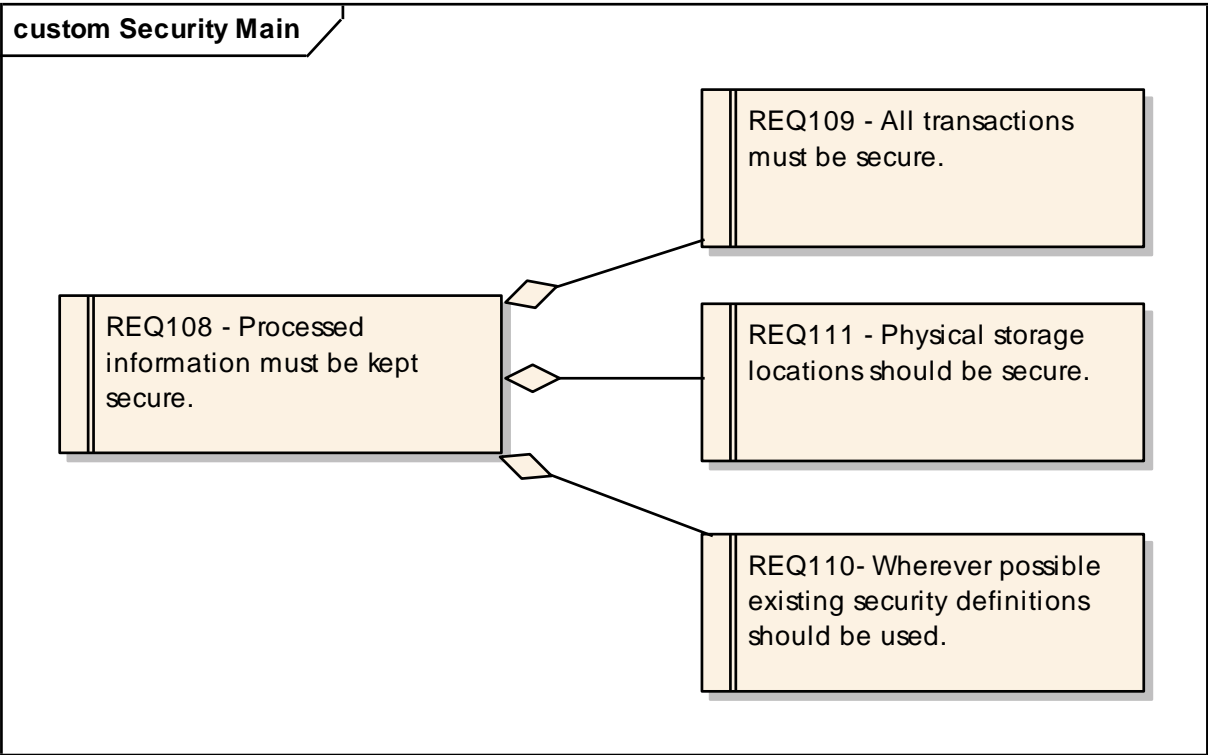


Diagram: System Model

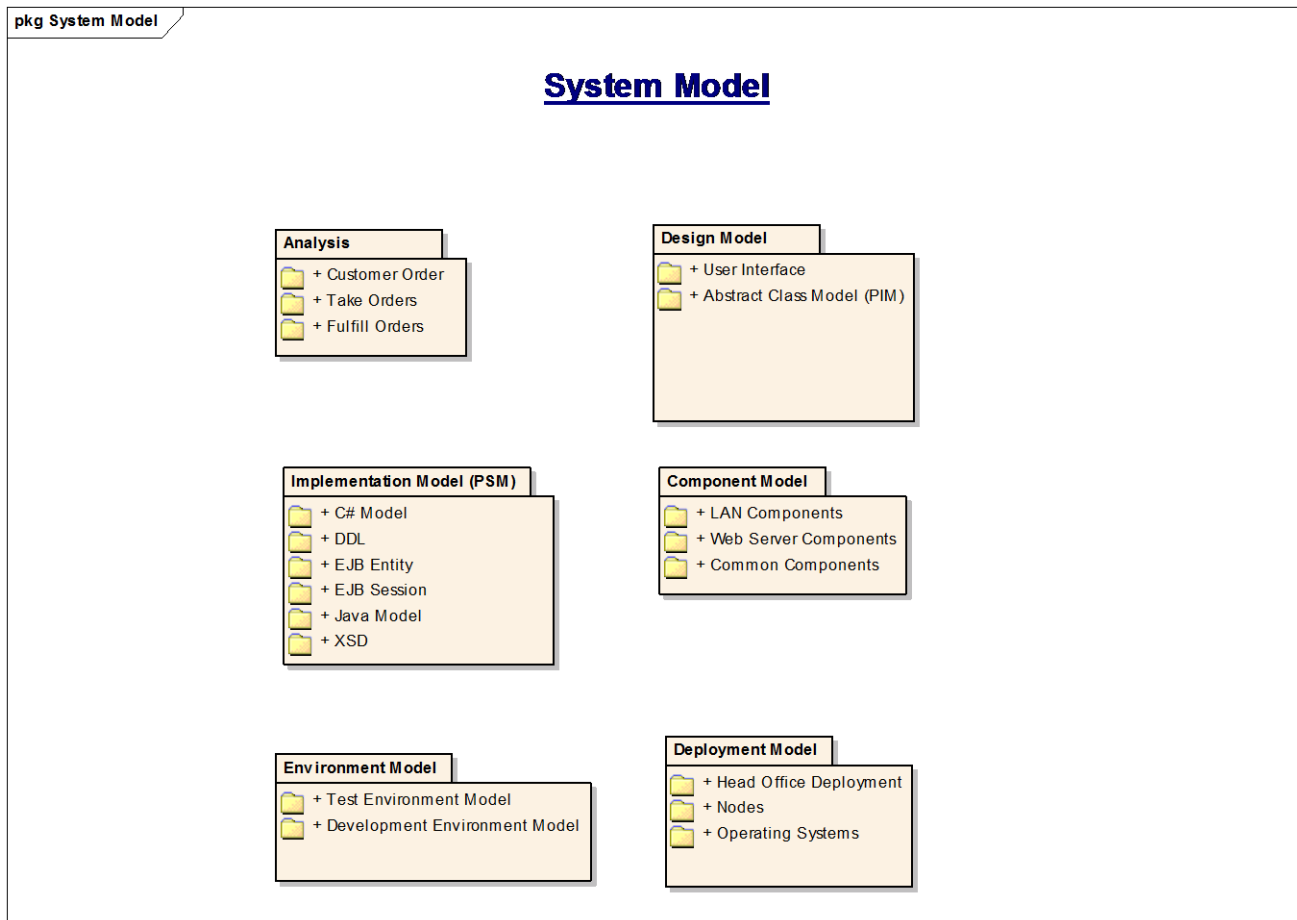


Diagram: Design Model

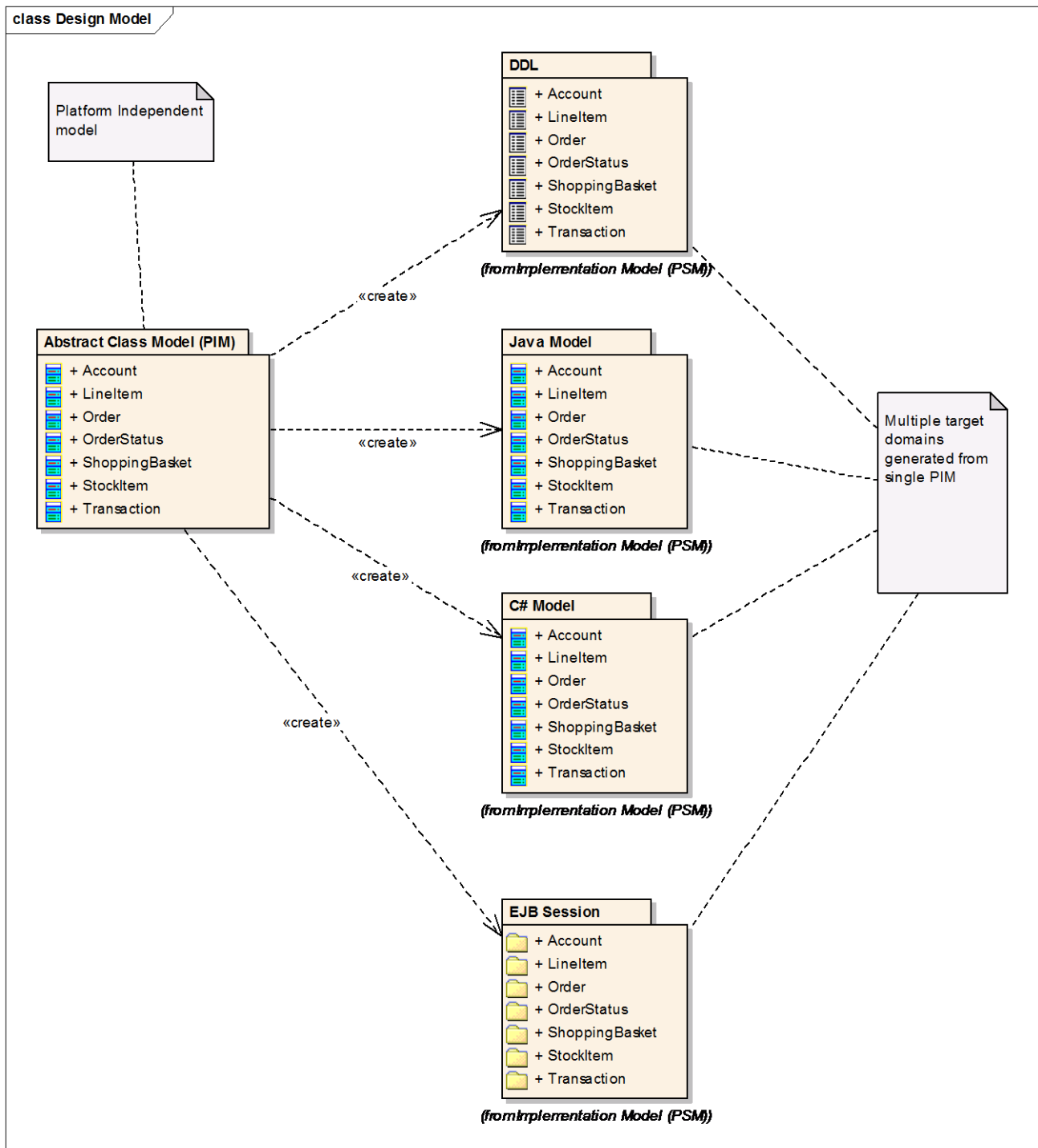


Diagram: Class Model

object Class Model

Abstract Class Model

This is an example of an abstract Class Model (an MDA PIM). The abstract class model can be used to create the implementation Model (an MDA PSM).



MDA Transforms

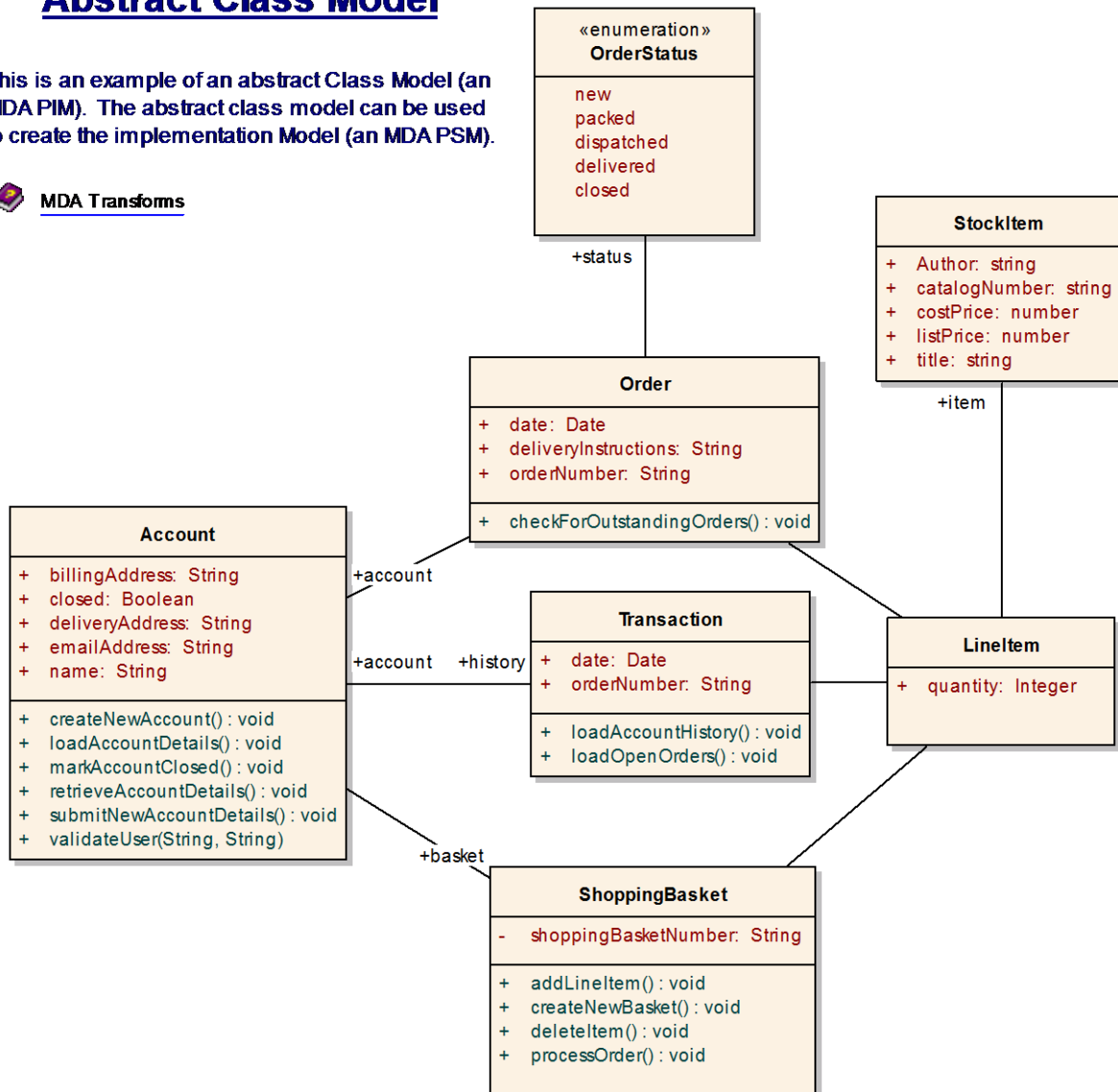


Diagram: User Interface

pkg User Interface

User Interface

EA supports the creation of User Interface screens. Below are links to the diagrams defining the User Interface design for the system under development.



[Browse Catalogue](#)



[Register User : Register User](#)



[PlaceOrder : Place Order](#)



[View Basket](#)



[Login Screen : Login Screen](#)

The Elements used to create these are available from : Toolbox | Custom. The custom elements used are the 'Screen' and 'UI Control' Elements.

Changing the Stereotype on a UI element allows different types of User Interface elements to be selected.

For more information on using these see:



[User Interface Group](#)



[UI Elements](#)

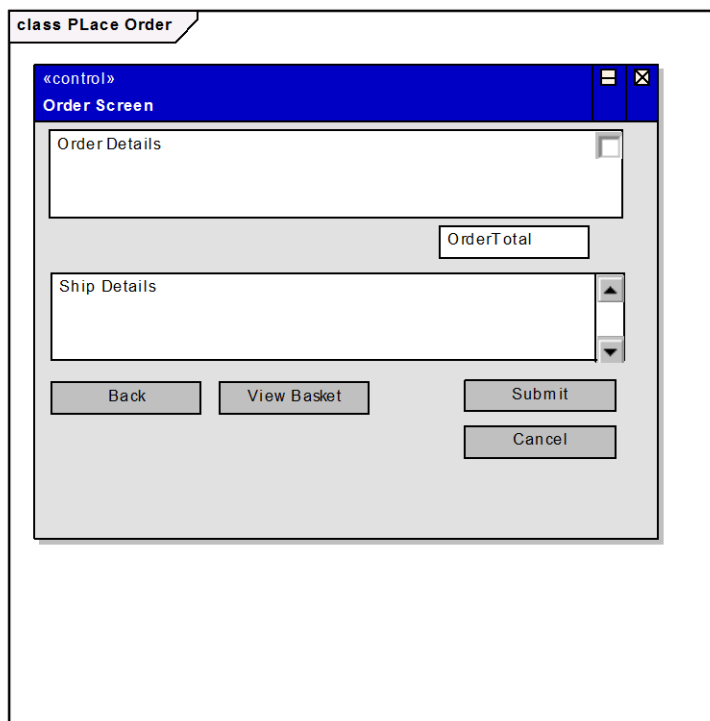


Diagram: User Management

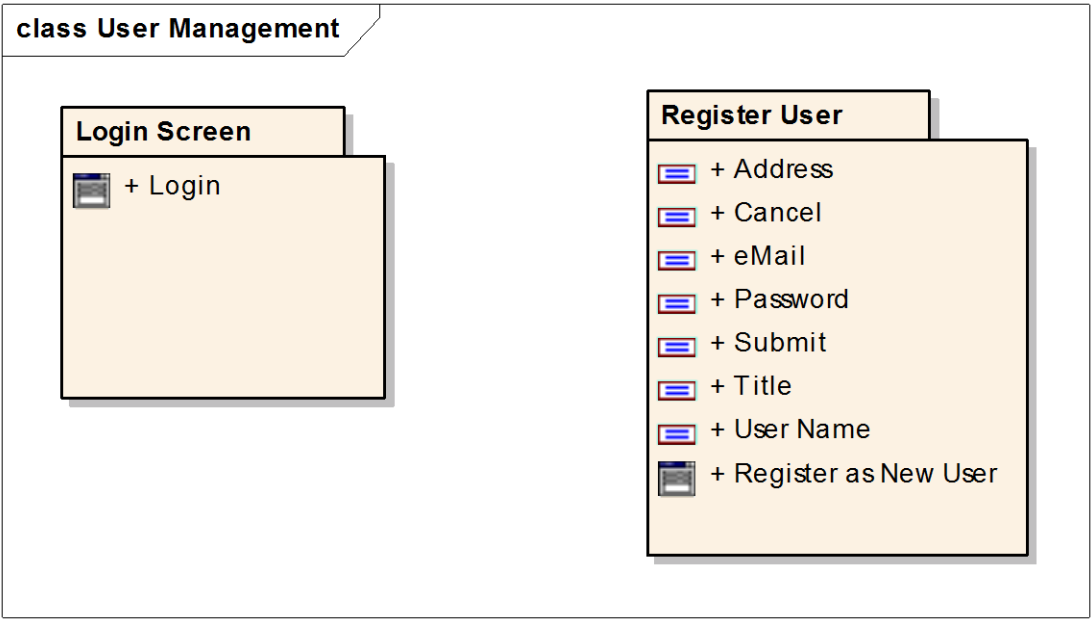


Diagram: Implementation

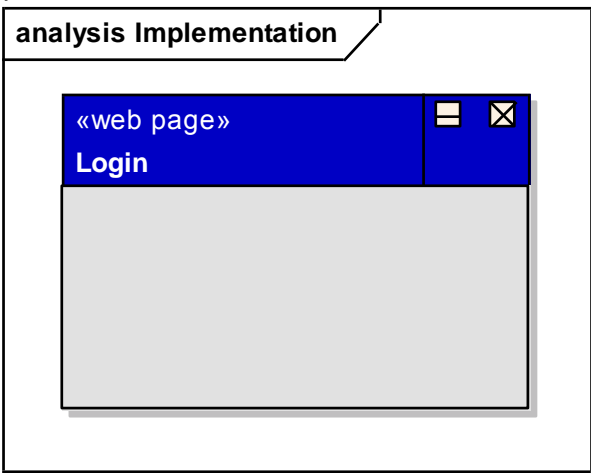


Diagram: Login Screen

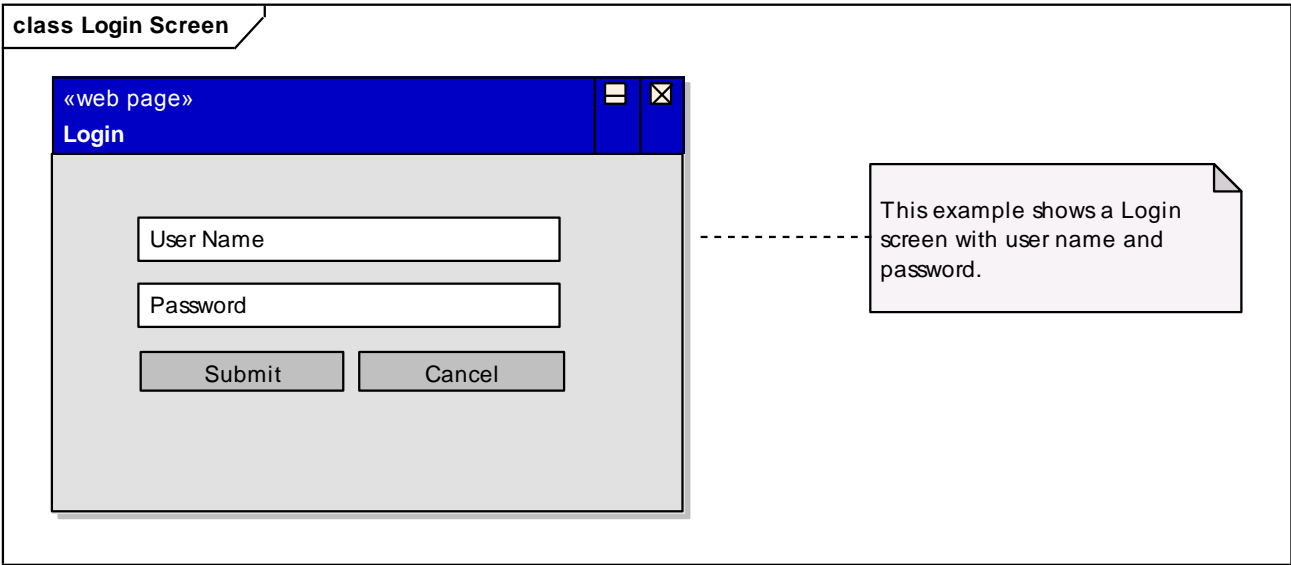


Diagram: Register User

class Register User

«web page»

Register as New User

User Name

Title

Password

City

Address

eMail

Submit

Cancel

Diagram: Inventory

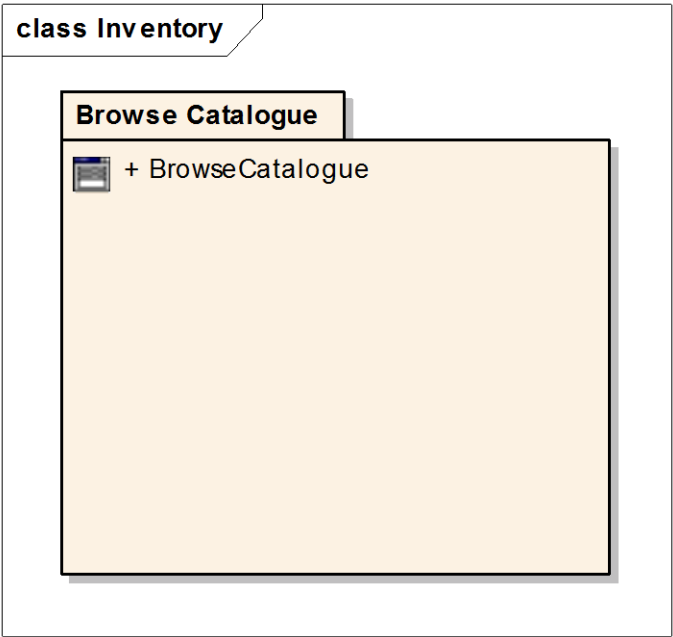


Diagram: Browse

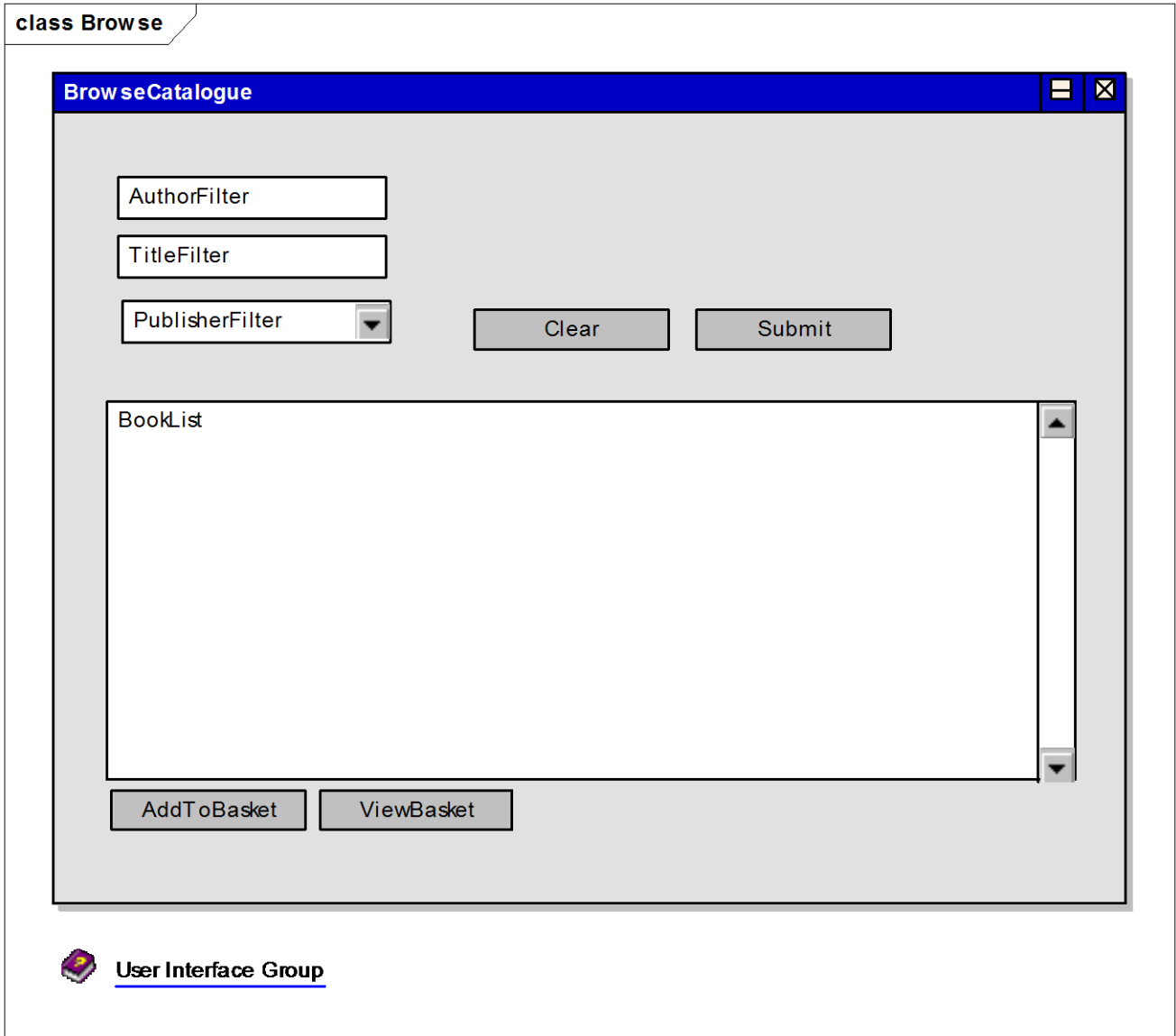


Diagram: Orders

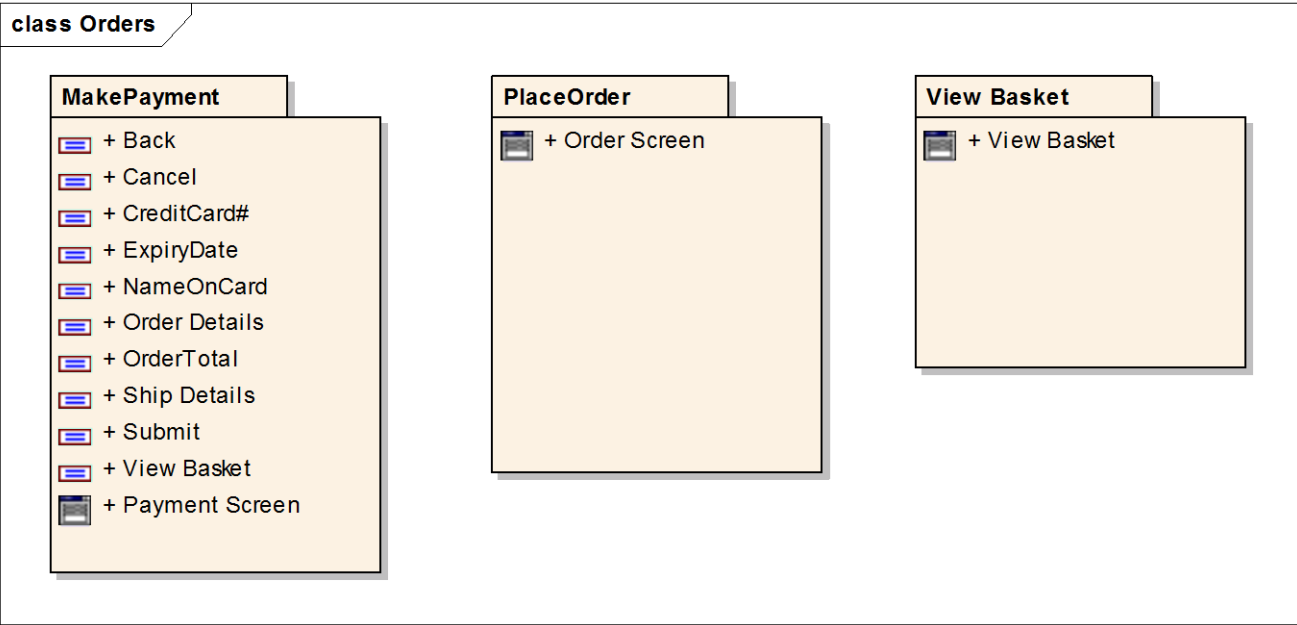


Diagram: Make Payment

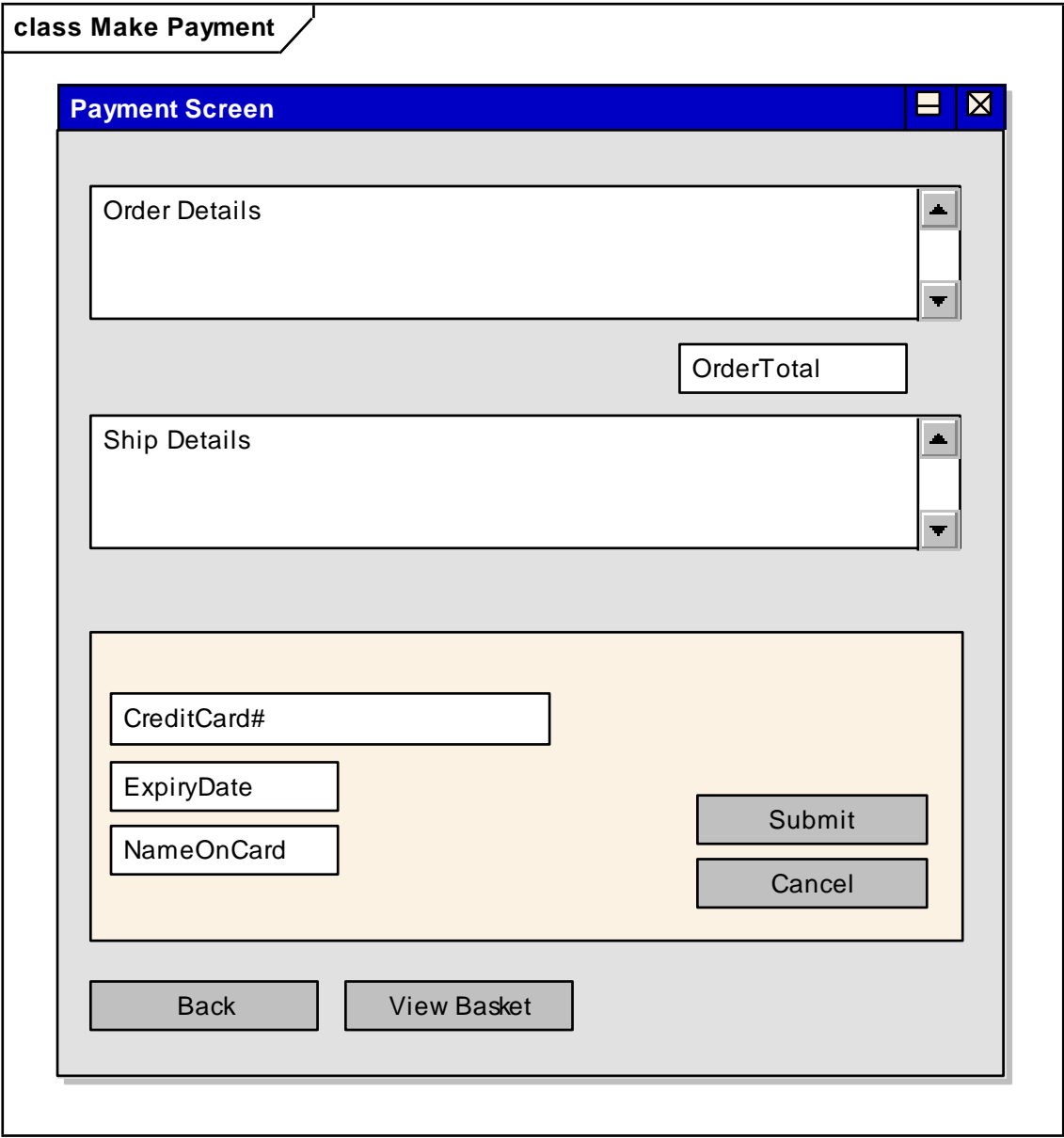


Diagram: PPlace Order

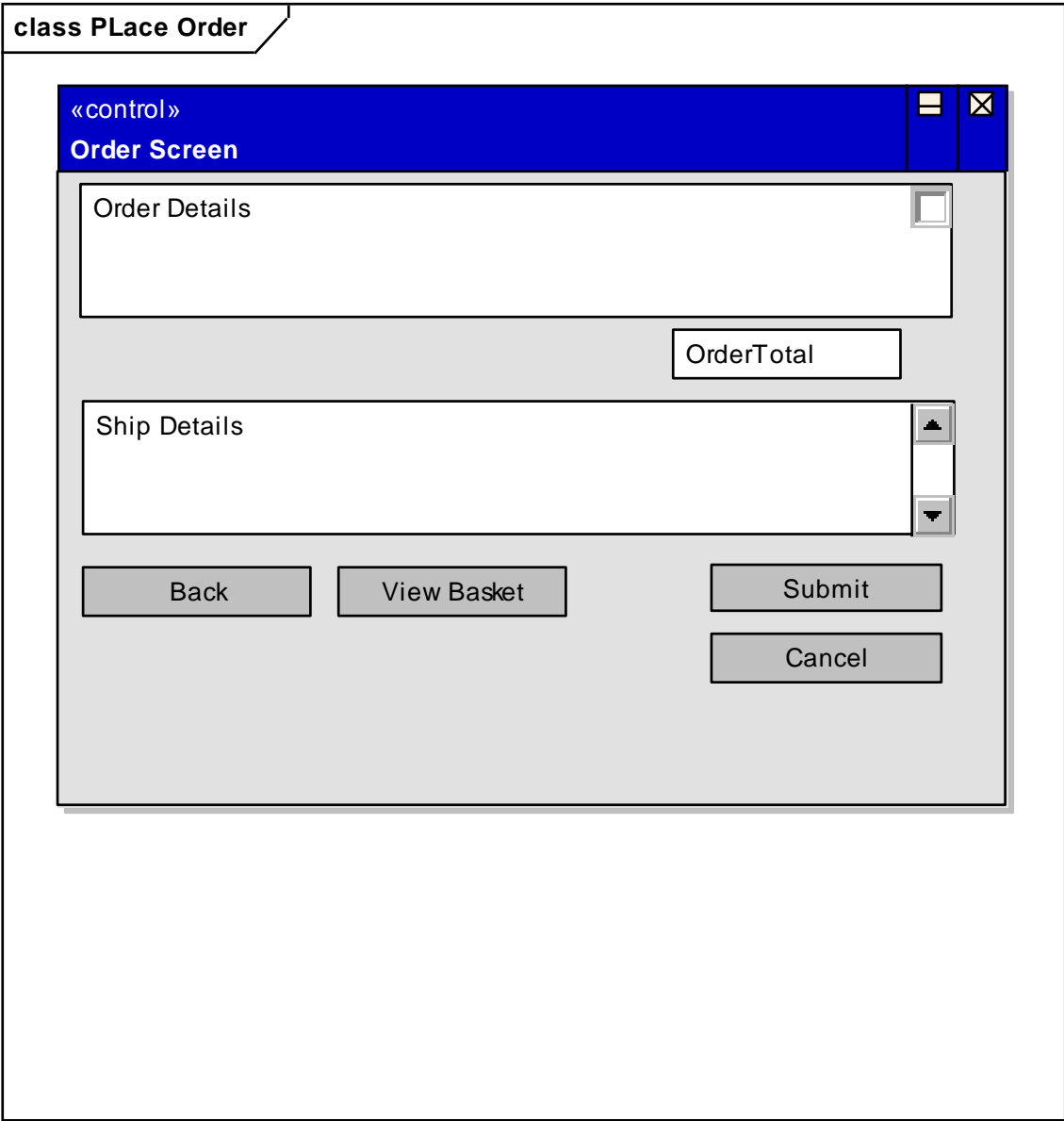


Diagram: View Cart

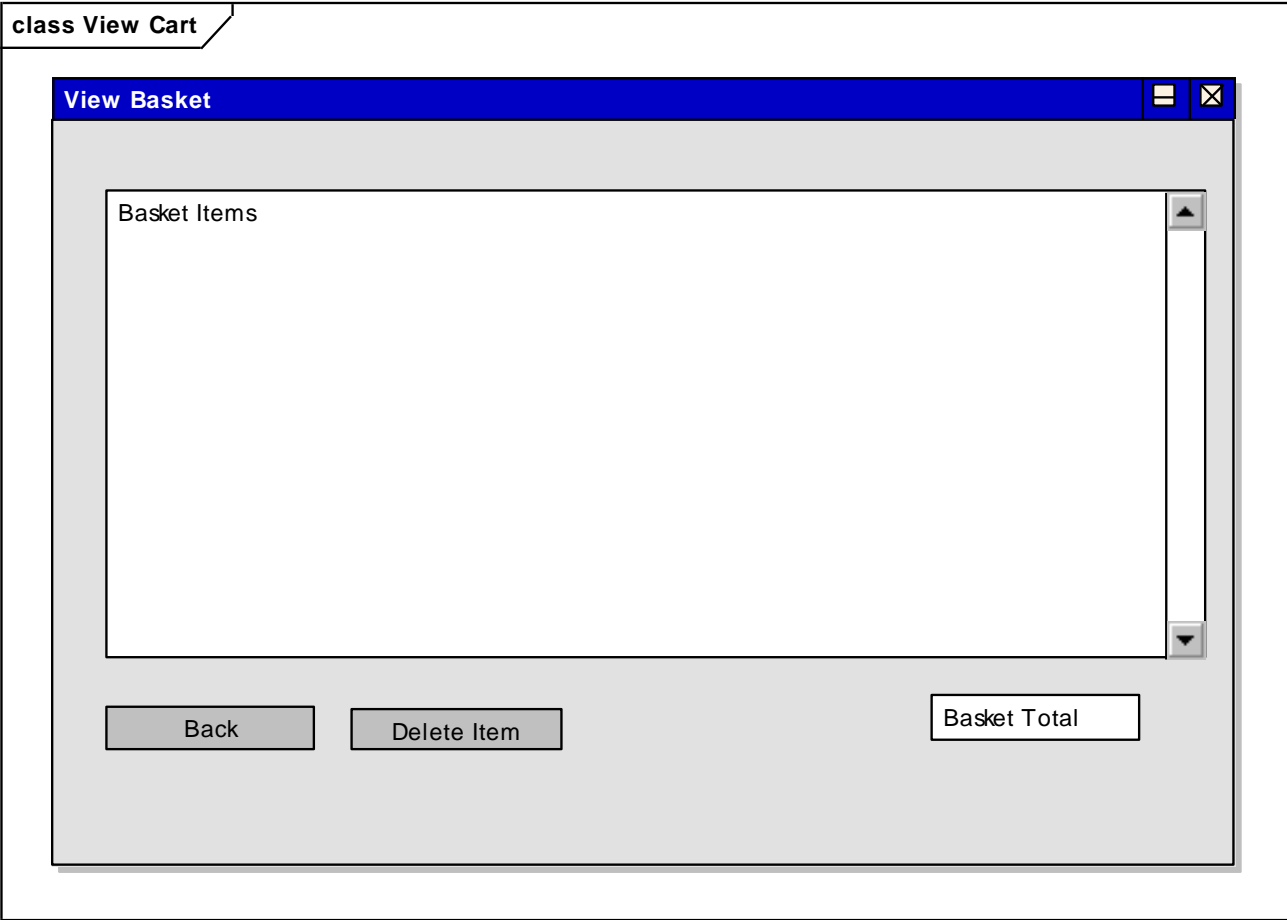


Diagram: Traceability

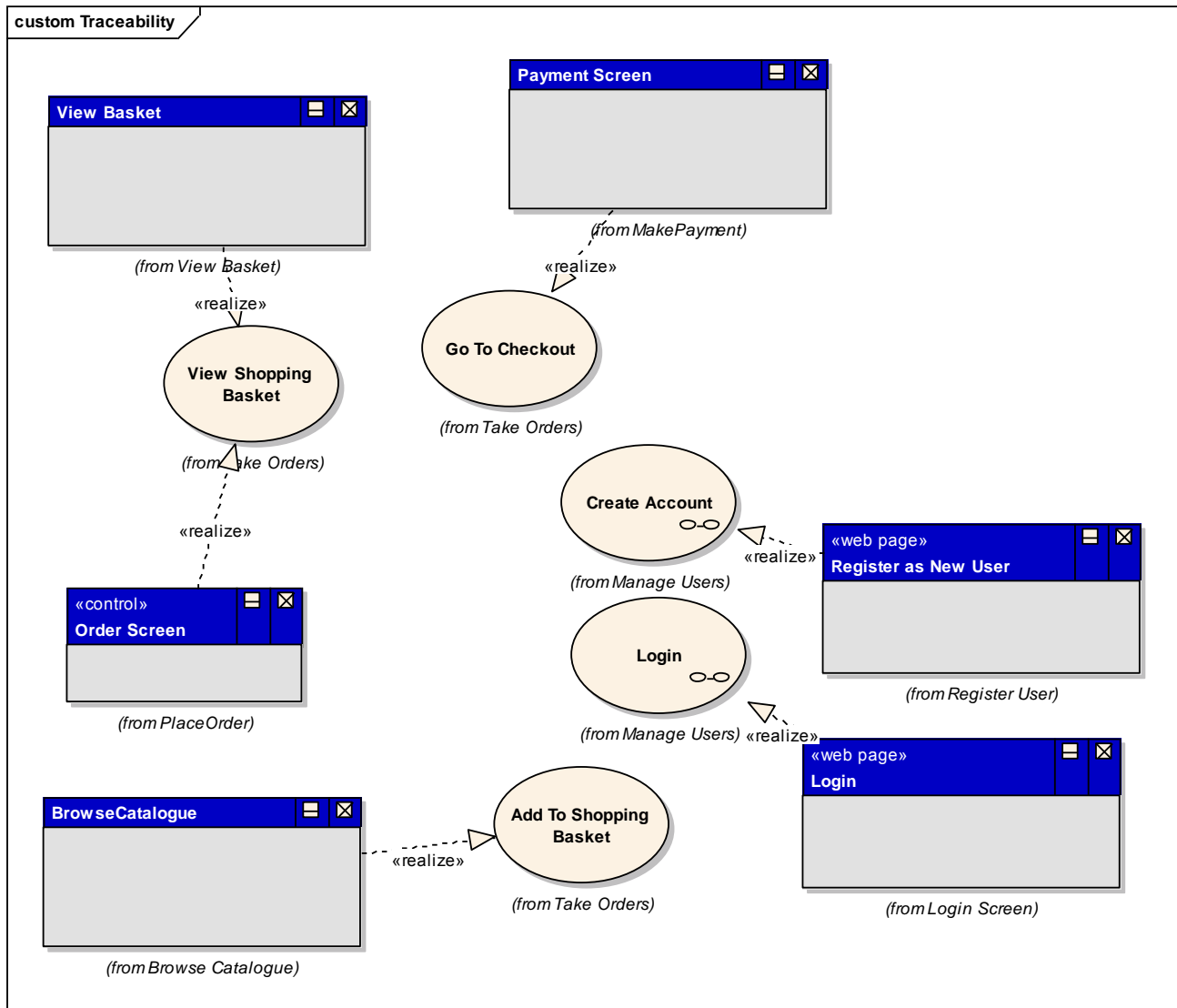


Diagram: Component Model

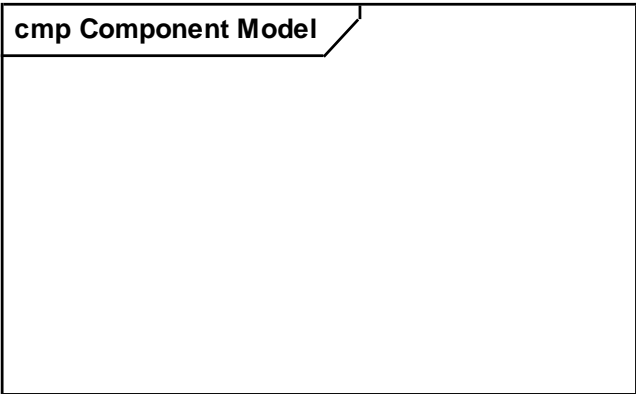


Diagram: SQL Server 2003

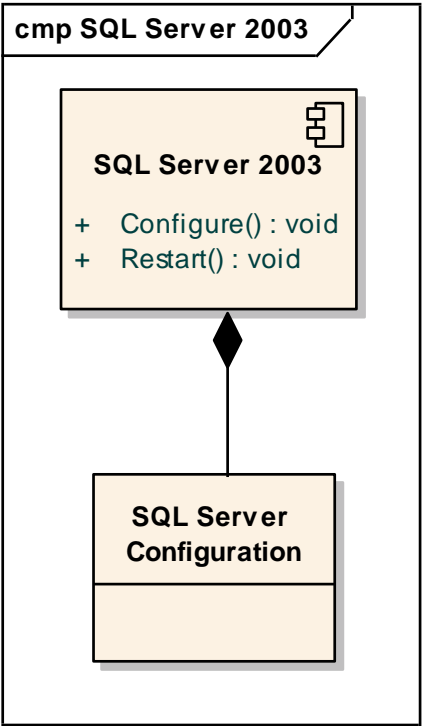


Diagram: Implementation

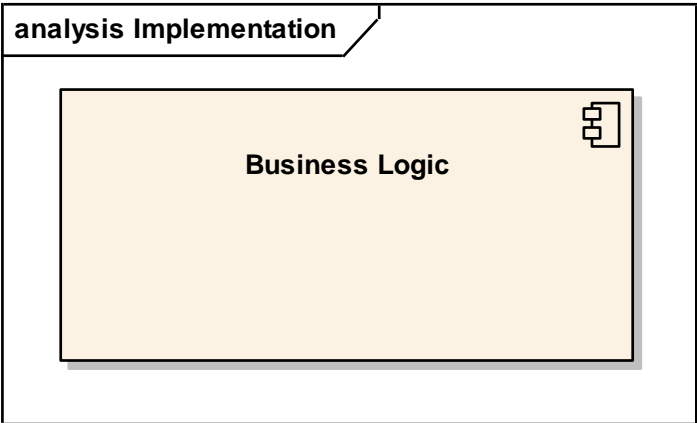


Diagram: Server Components

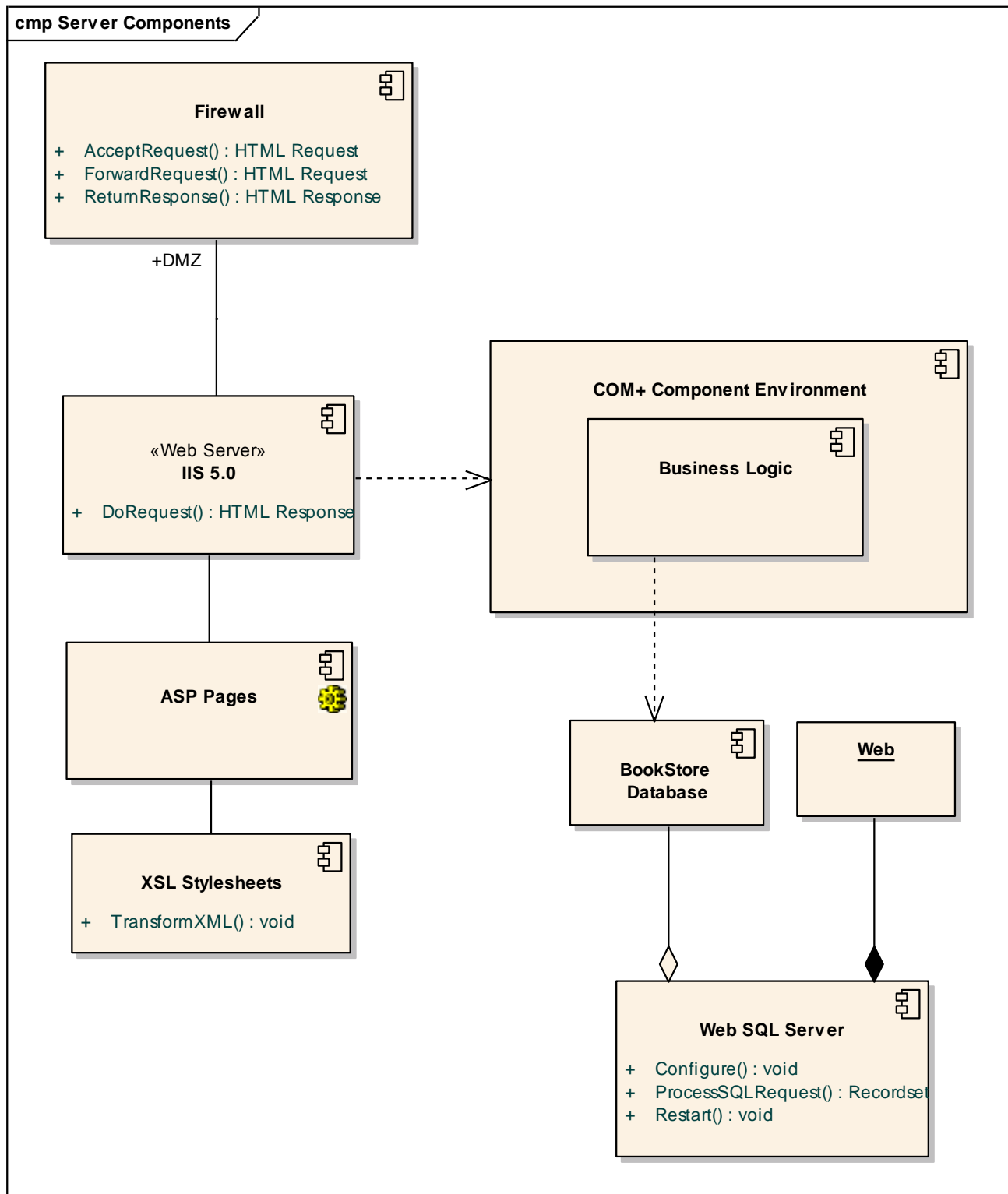


Diagram: LAN Components

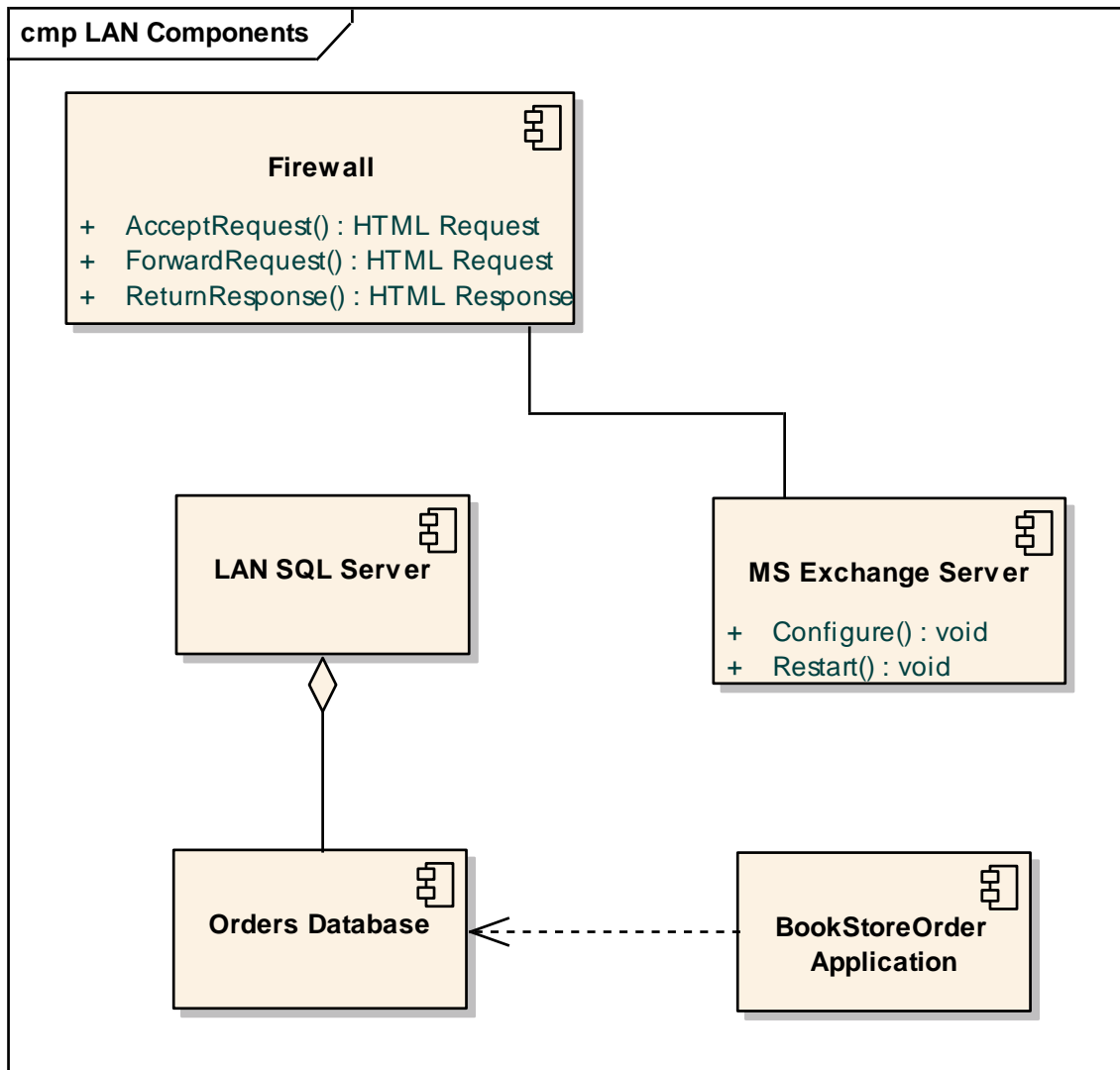


Diagram: Deployment Model

deployment Deployment Model

Deployment Model

To view the deployment of the key servers based in the Head Office see the following links:



[Head Office Deployment - Servers](#)



[Head Office Deployment using Images](#)

For details on the operating system and component configuration of these servers see:



[Server configurations](#)

Below are the key packages used in the deployment model.
Double-click on these to view the diagrams.

Head Office Deployment

- + 192.168.0.3
- + 192.168.0.2
- + 216.239.46.95
- + 216.239.46.96
- + Client Data Server
- + FRR01
- + HOES01
- + HOES02
- + HOFW
- + Mail Server
- + Web Server
- + WebDataServer
- + Clients
- + Servers
- + Networking Equipment

Nodes

- + Computers
- + Networking

Operating Systems

- + Mac OS X
- + RedHat Linux 9.0
- + Sun Linux JVM 1.4.0
- + Windows 2000 Professional
- + Windows 2000 Server
- + Windows 2003 Server
- + Windows XP Professional

Diagram: HO Server Images

deployment HO Server Images

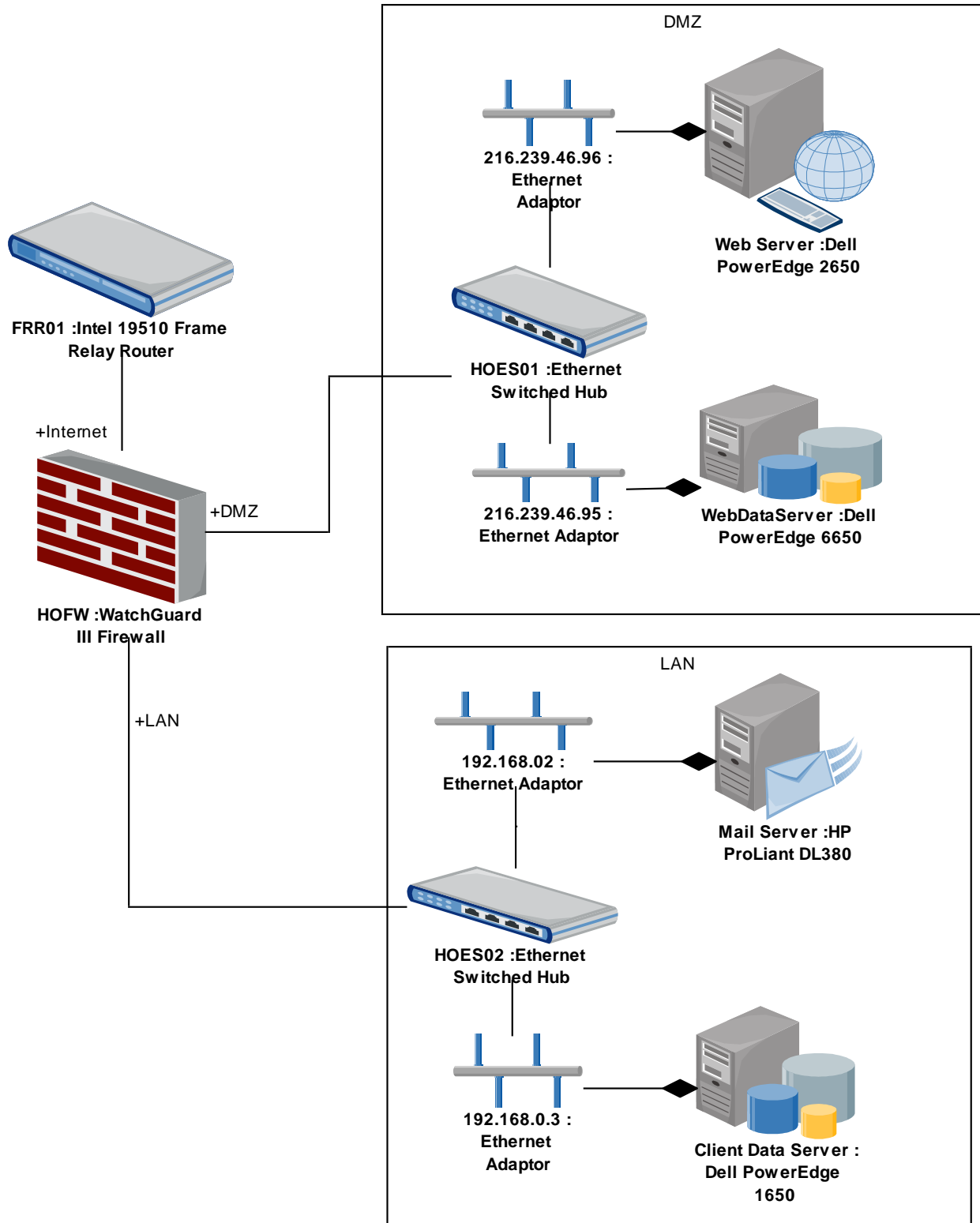


Diagram: HO Servers

deployment HO Servers

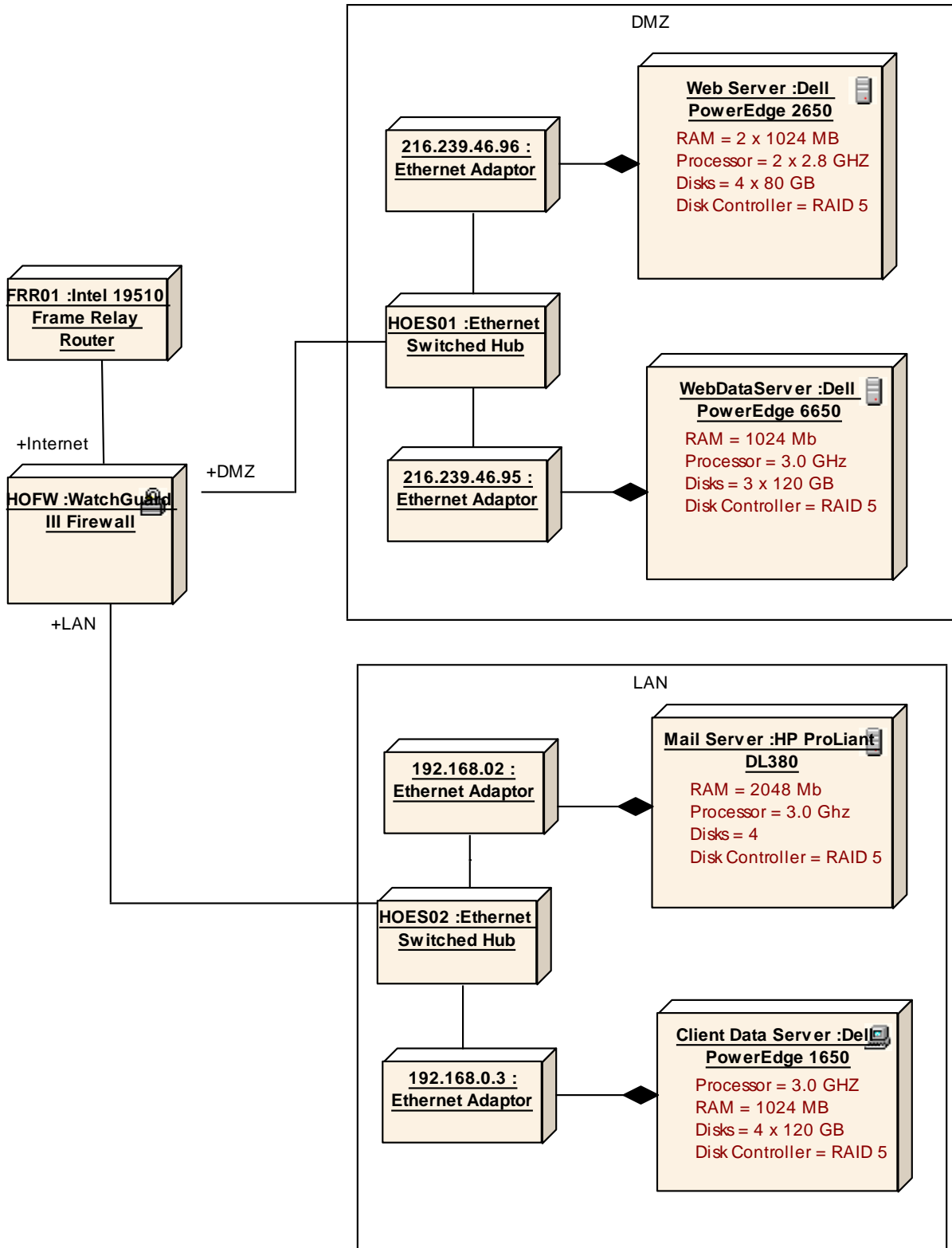


Diagram: Office Client 1

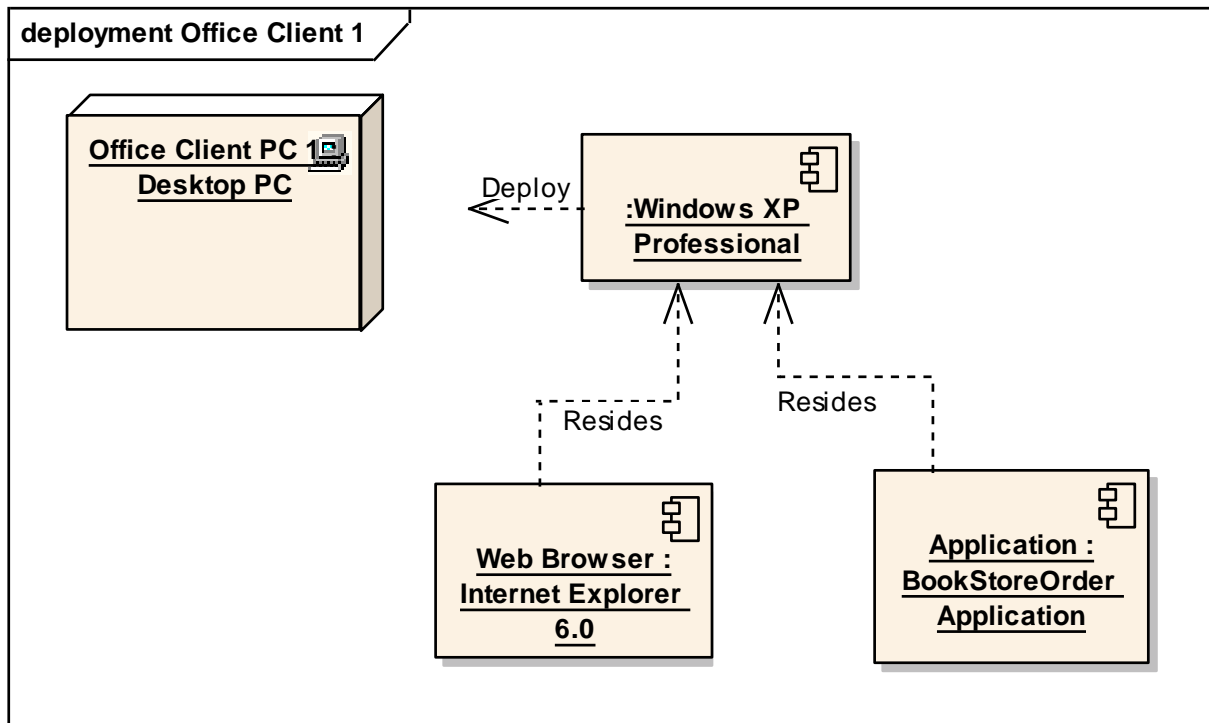


Diagram: Office Client 2

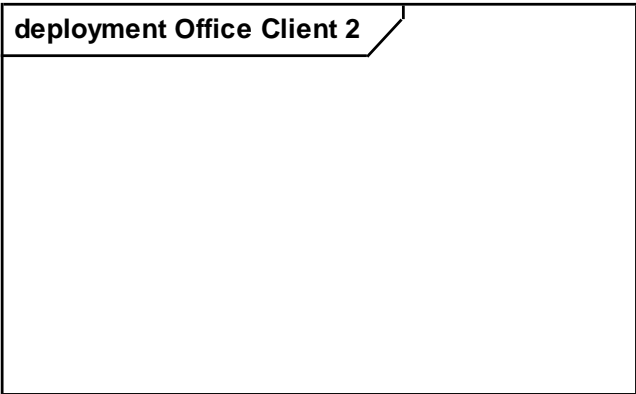


Diagram: Servers

deployment Servers

Server Configurations

Below are the packages containing the server configurations. These packages include diagrams with the details of the components deployed on these servers. To access a server configuration diagram double click on the package.

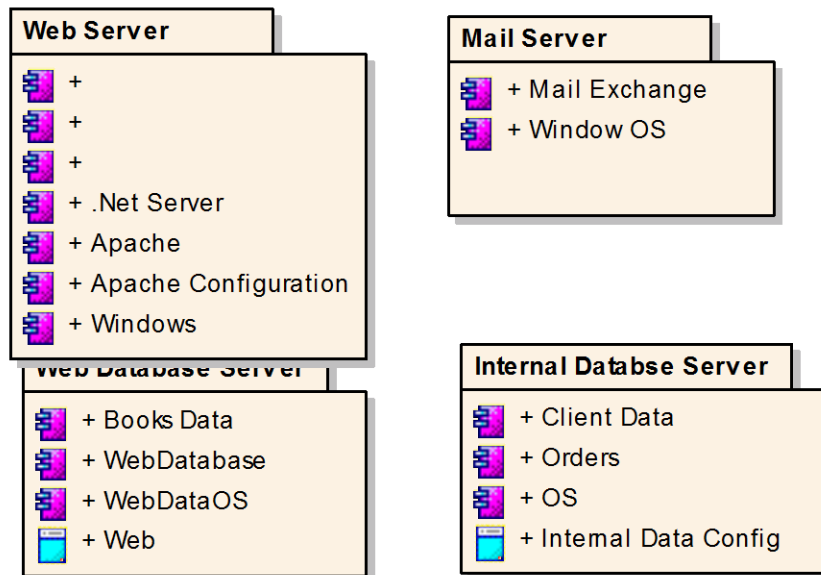


Diagram: Web Server

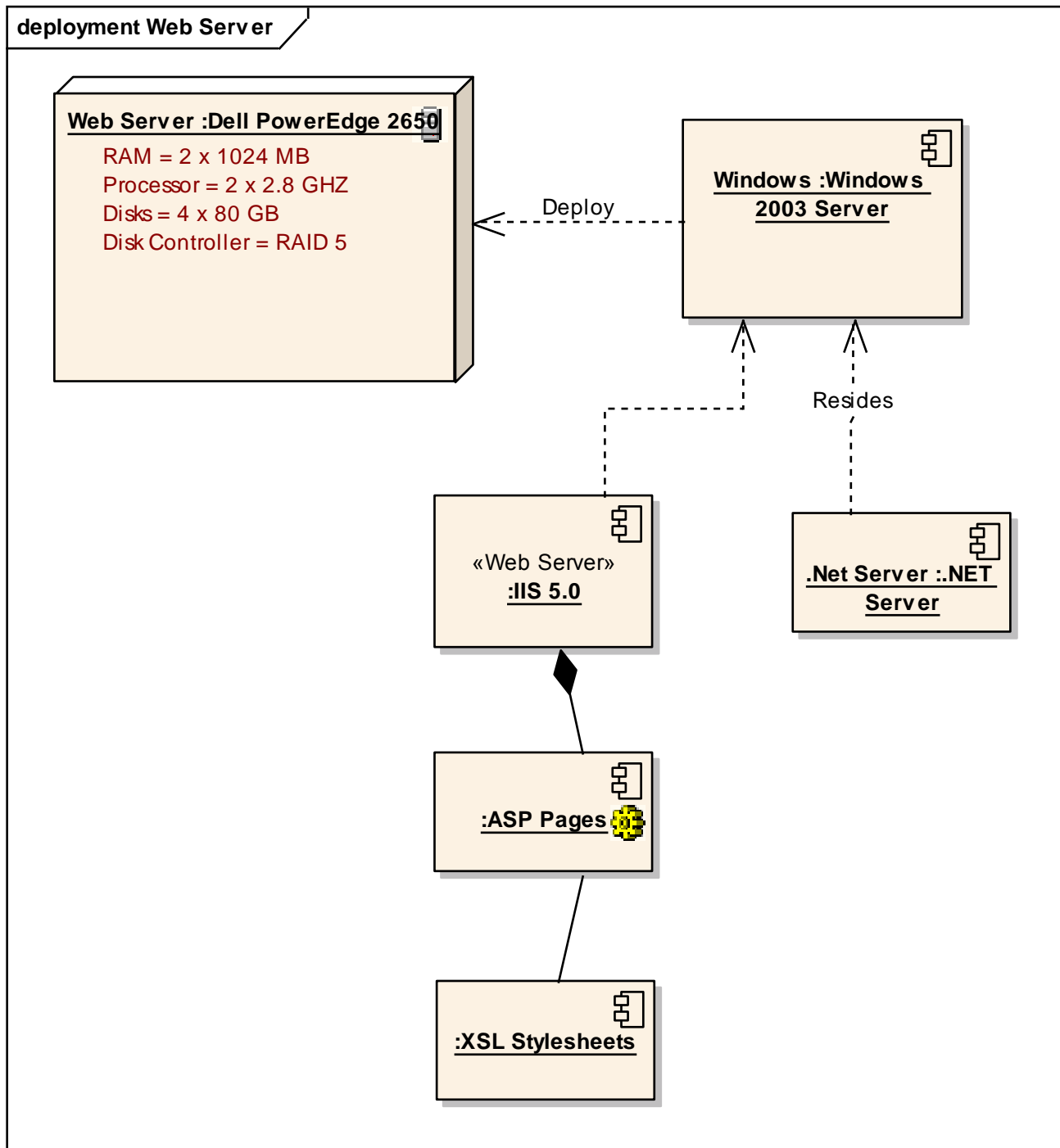


Diagram: Mail Server

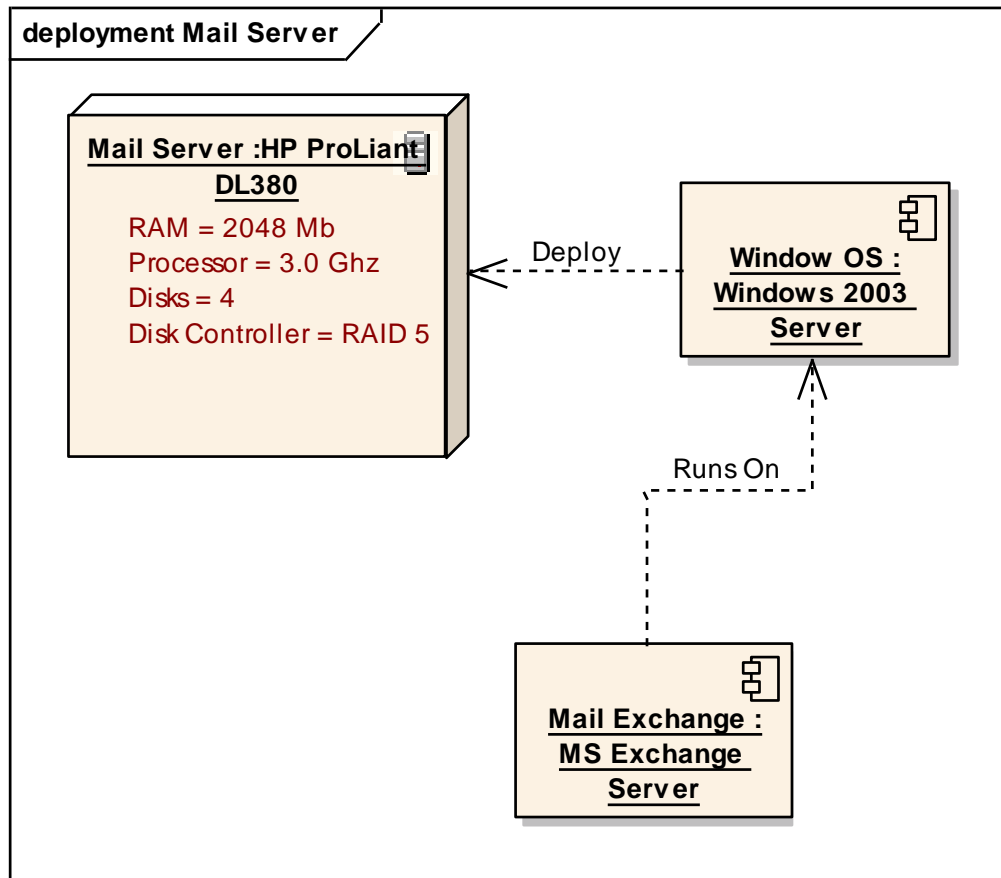


Diagram: Database Server

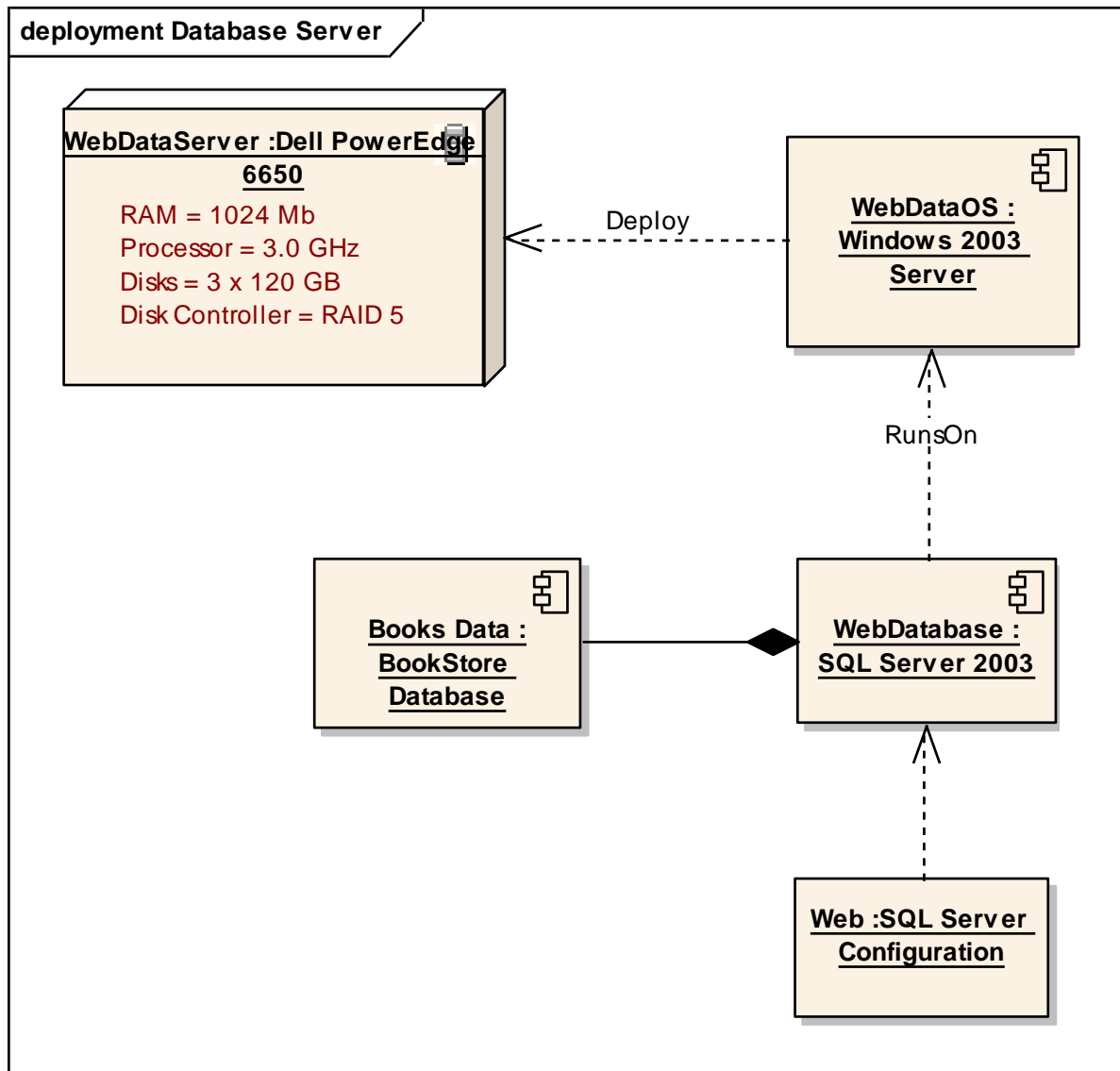


Diagram: Internal Database Server

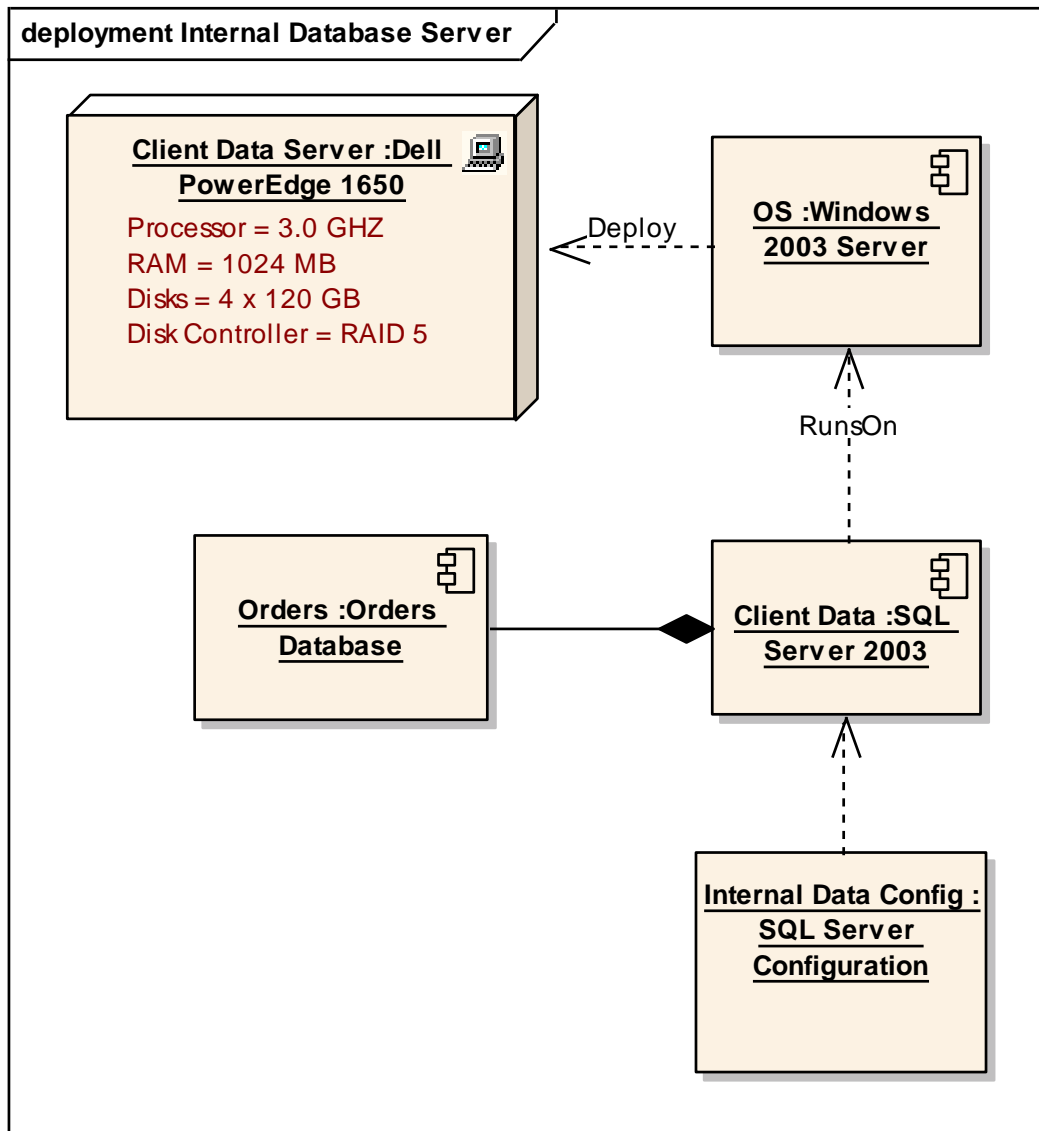


Diagram: Implementation Model (PSM)

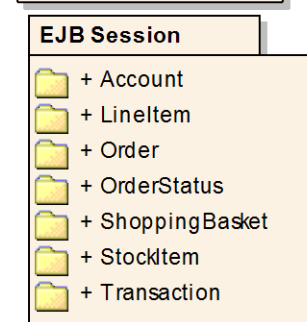
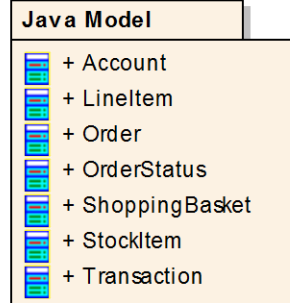
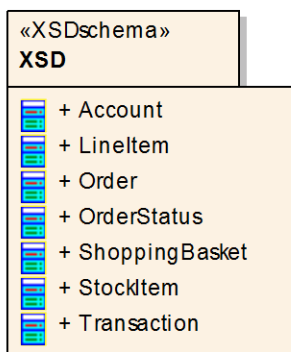
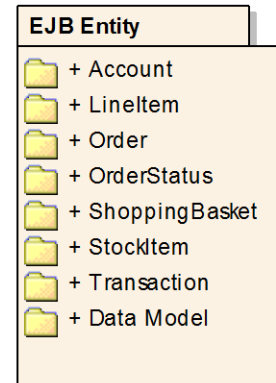
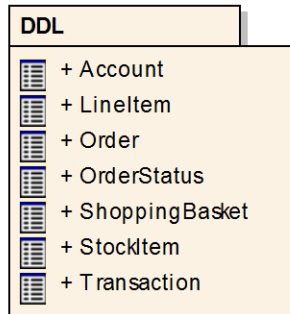
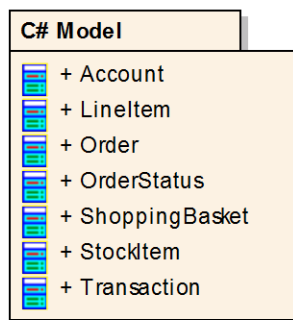
class Implementation Model (PSM)

Implementation Model

The implementation model has been generated from the Abstract Class Model using the MDA transform process. This has generated the following:

- A Data Model for an Oracle DBMS - (see DDL)
- C# classes
- Java classes
- EJB Entity and Session classes
- An XSD schema

To view the details of these, double-click on the packages below.



For more information on using MDA transforms see:  [MDA Transforms](#)

Diagram: C# Model

Class Model

This Class diagram represents the MDA transform generated from the Abstract Class model PIM. For more information on MDA transforms see:



MDA Transforms

This Class model can be forward engineered to the C# code. For more information on processes of code generation, reverse engineering of source code and synchronization between the source code and model - see:



Code Engineering

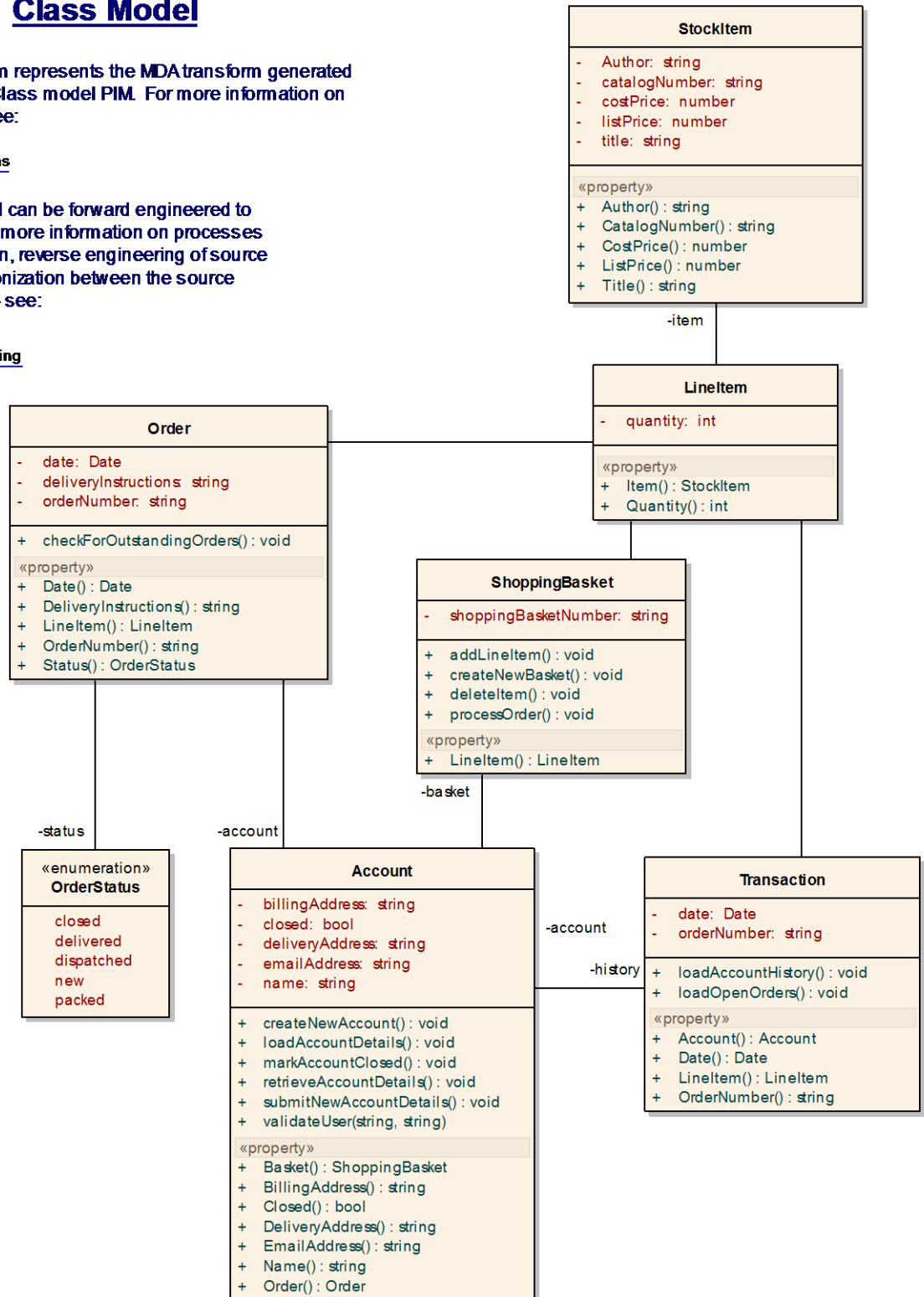


Diagram: C# Model - No Attributes

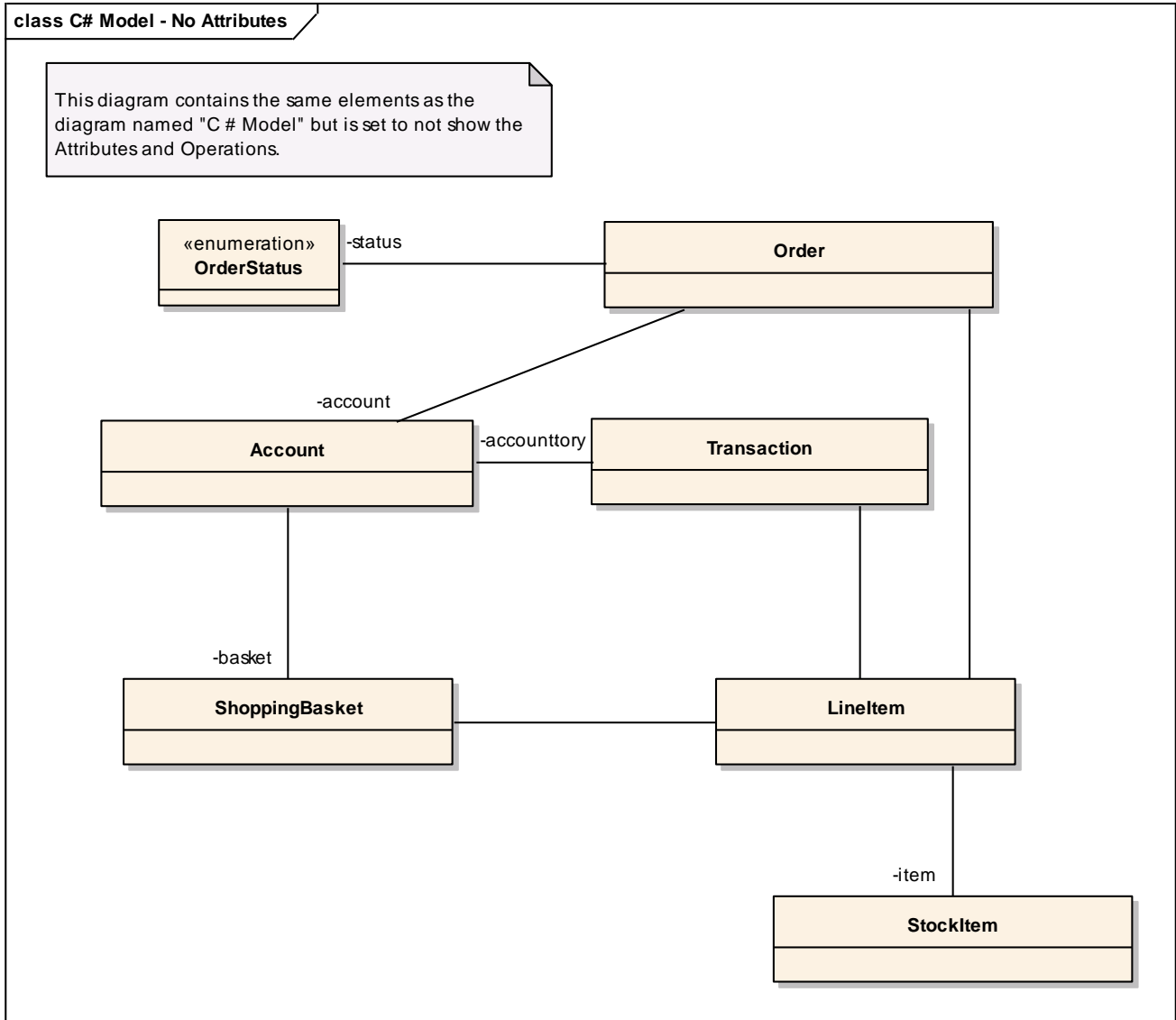


Diagram: DDL

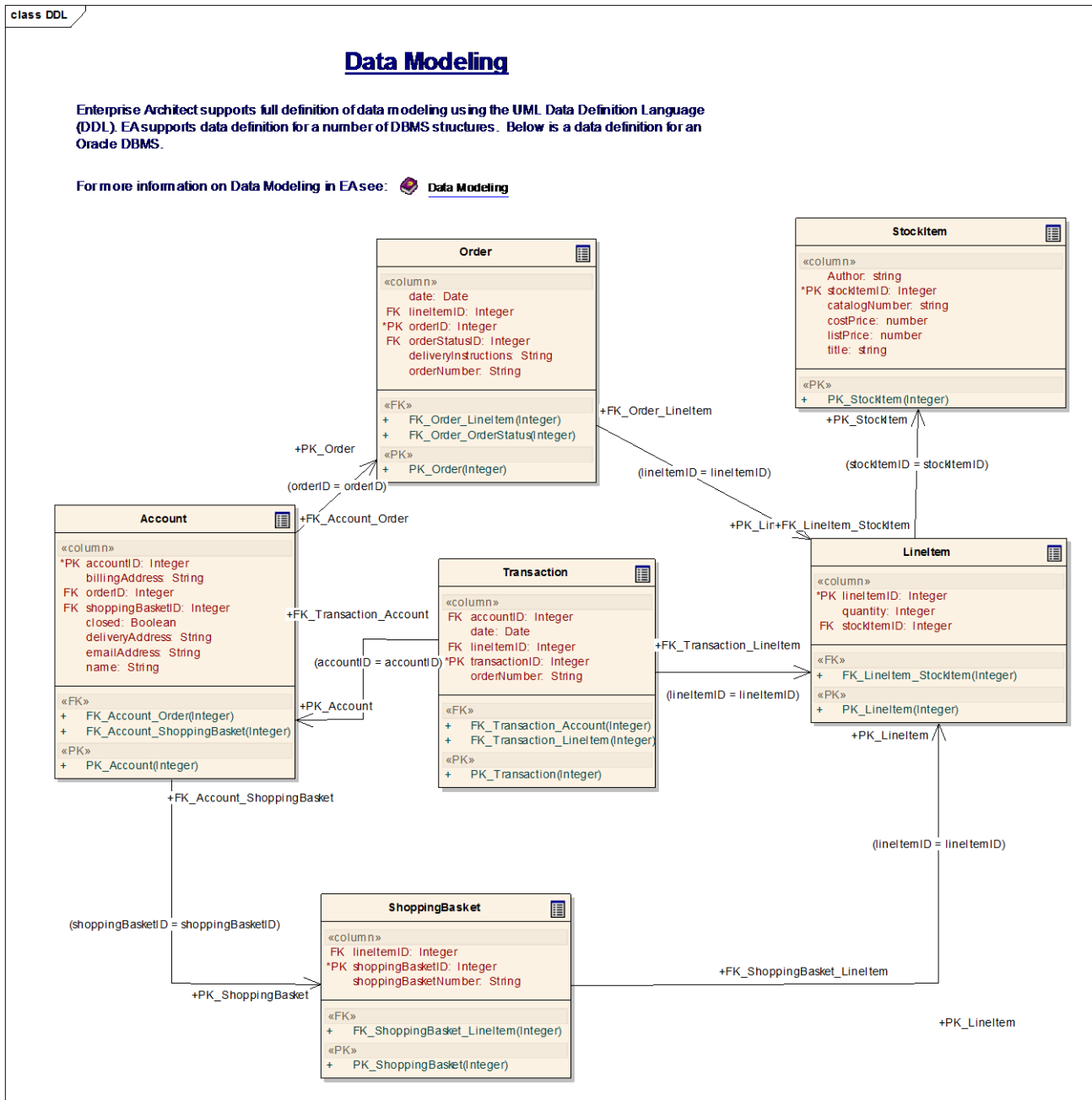


Diagram: EJB Entity

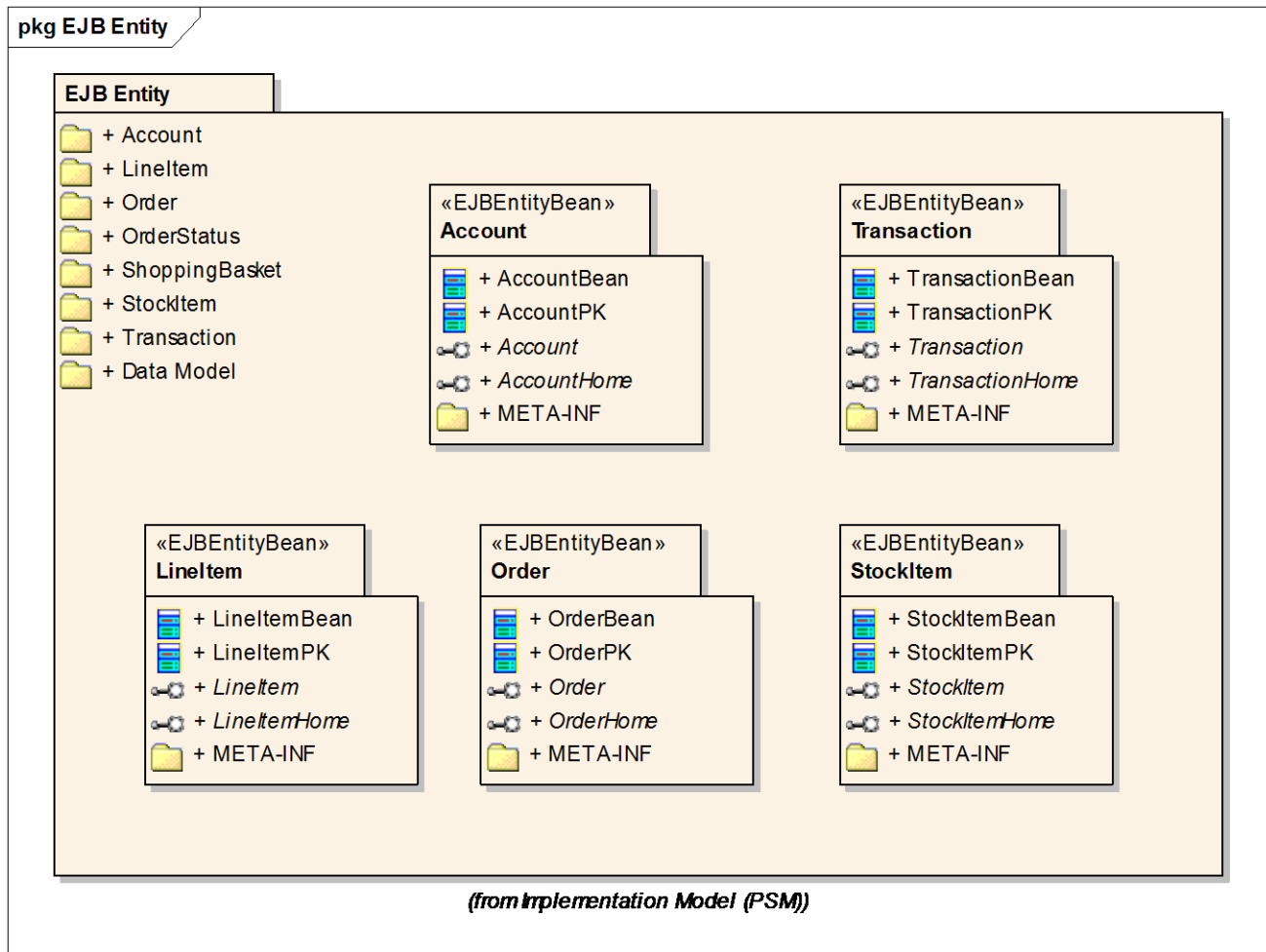


Diagram: Data Model

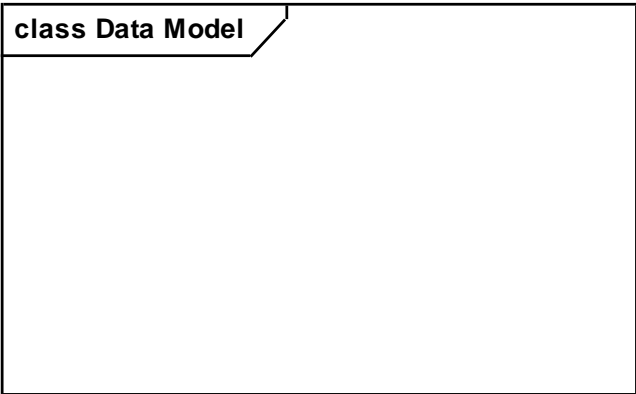


Diagram: Account

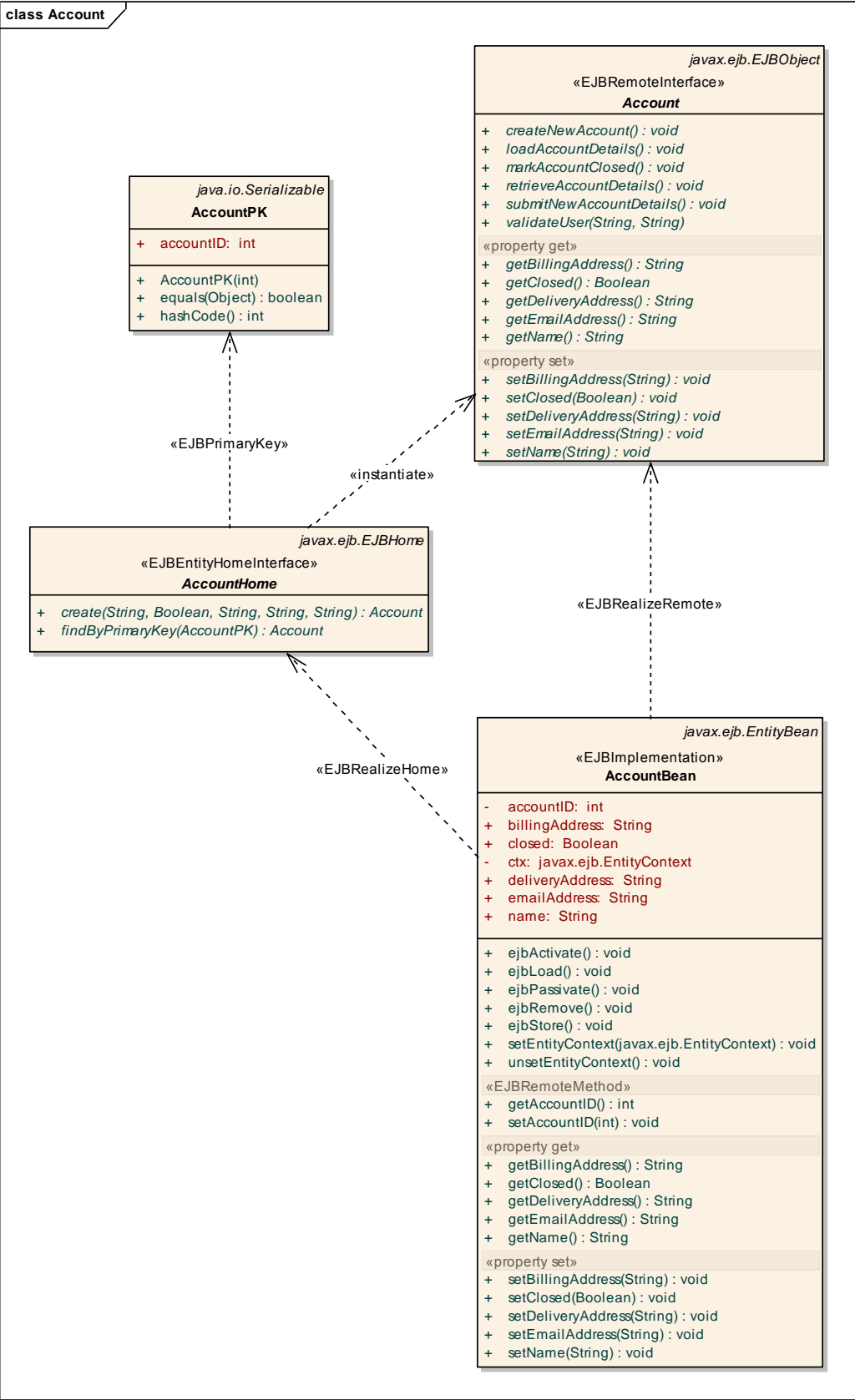


Diagram: META-INF

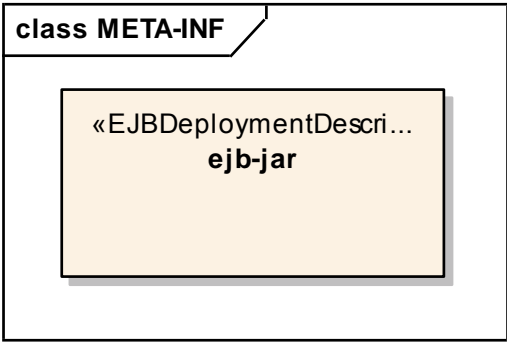


Diagram: LineItem

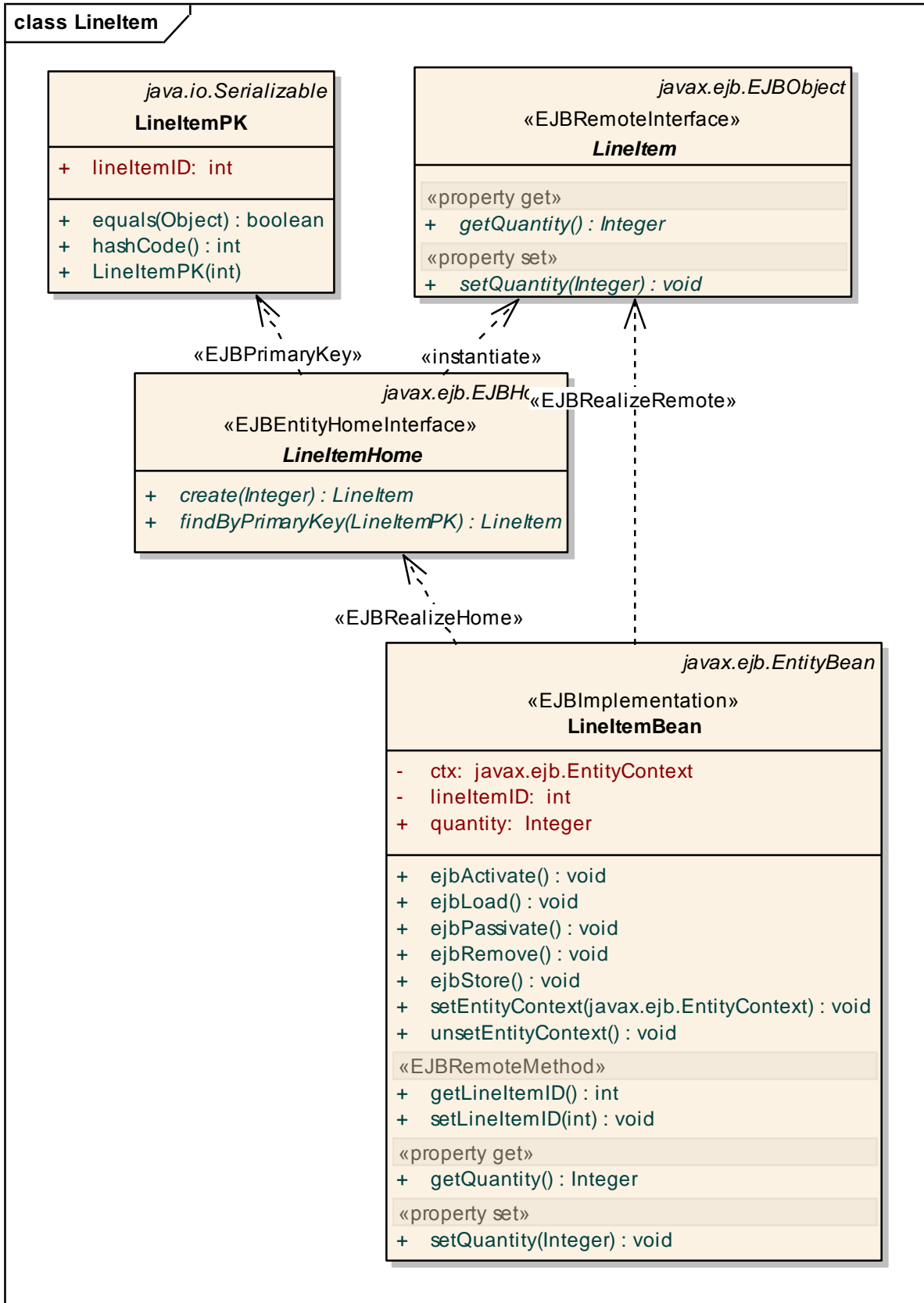


Diagram: META-INF

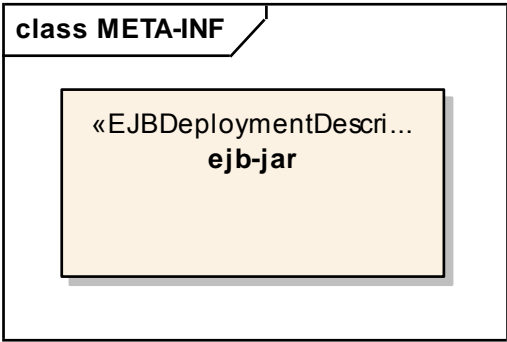


Diagram: Order

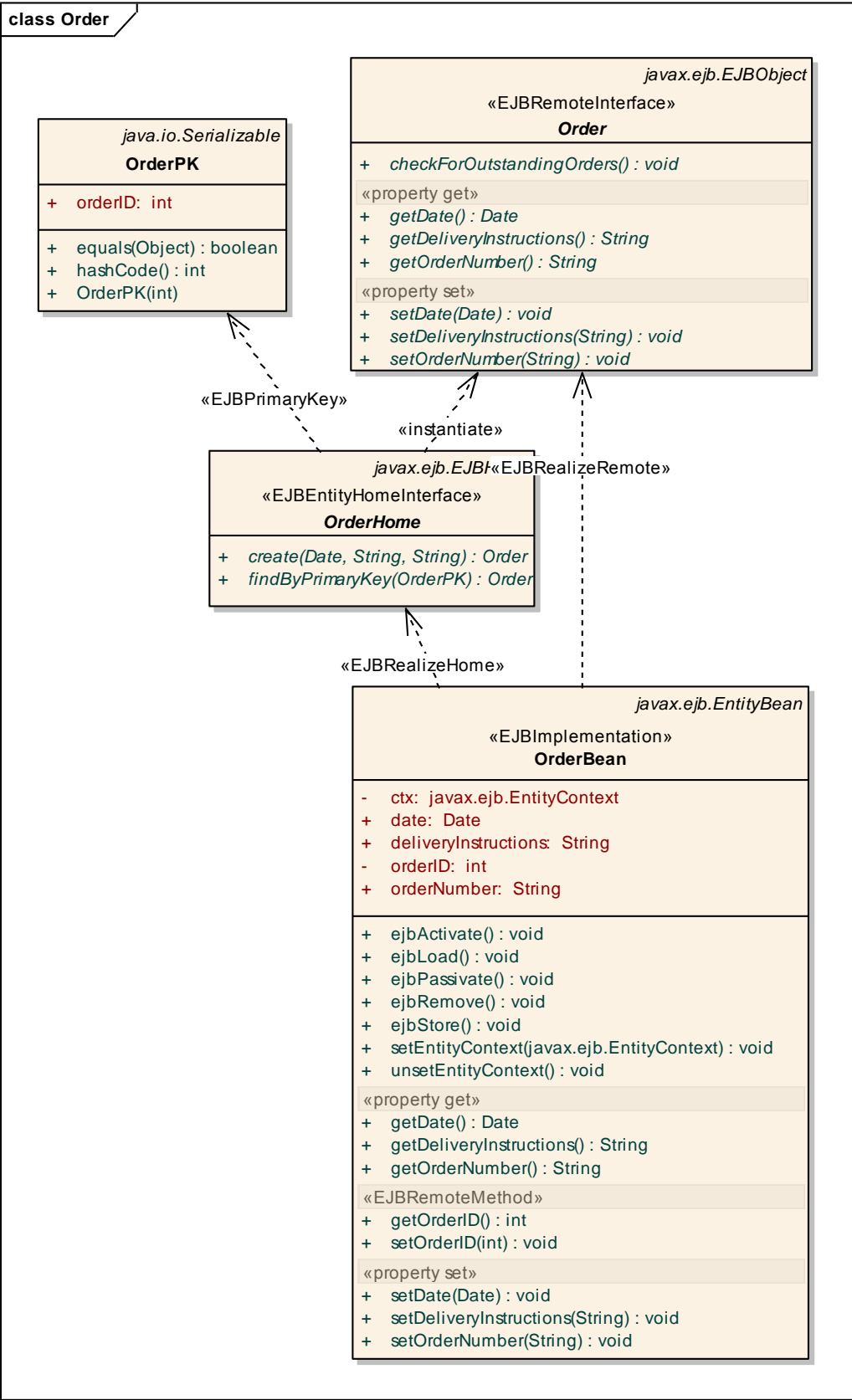


Diagram: META-INF

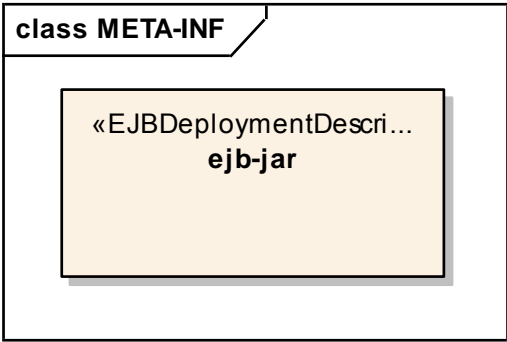


Diagram: StockItem

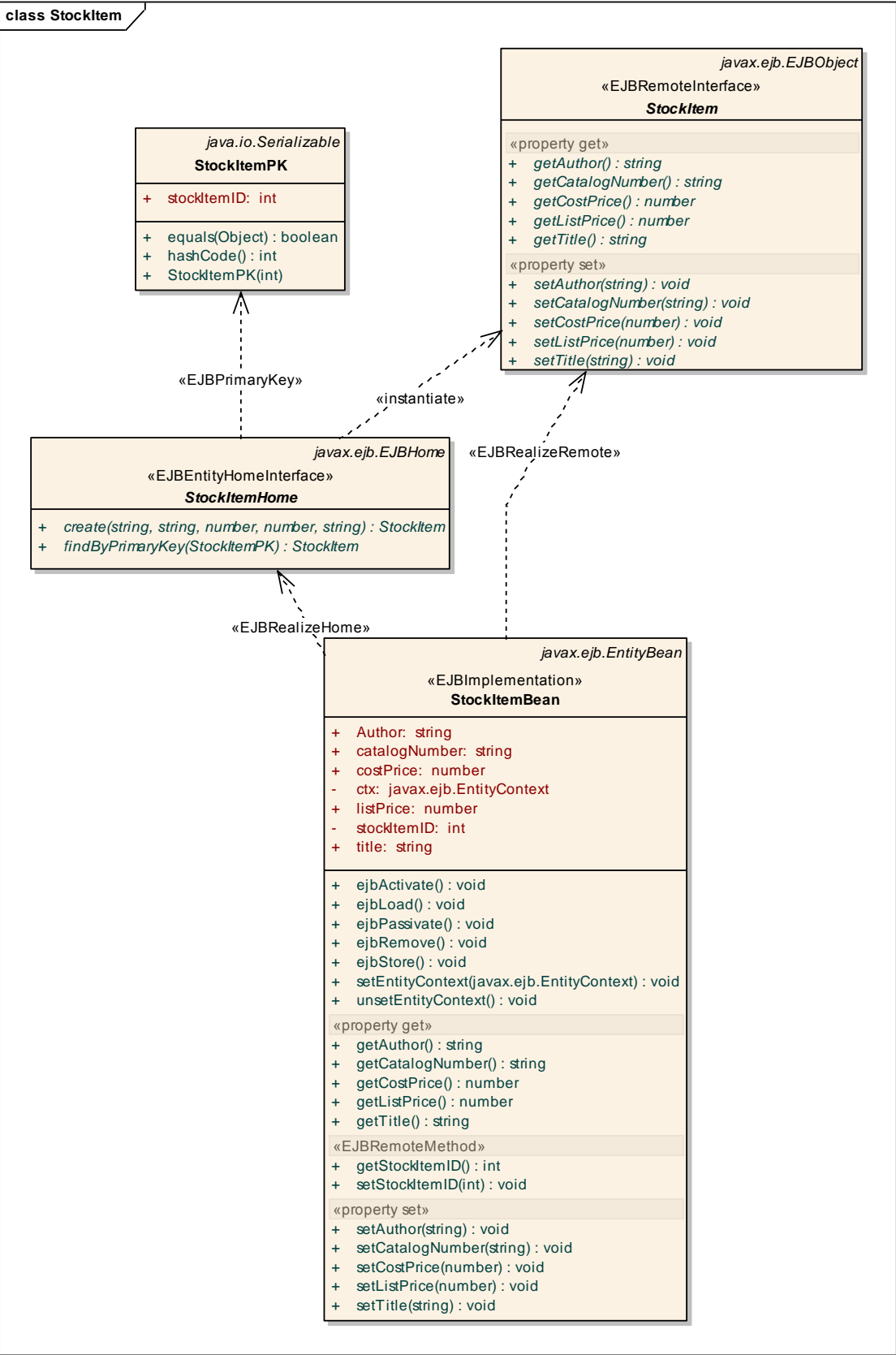


Diagram: META-INF

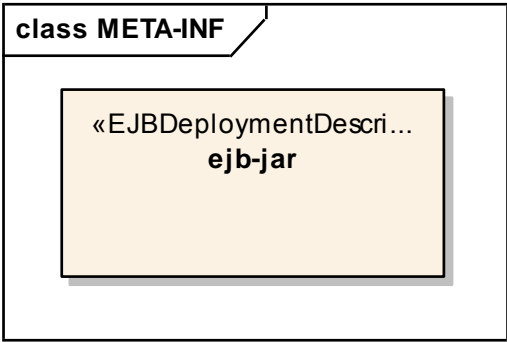


Diagram: Transaction

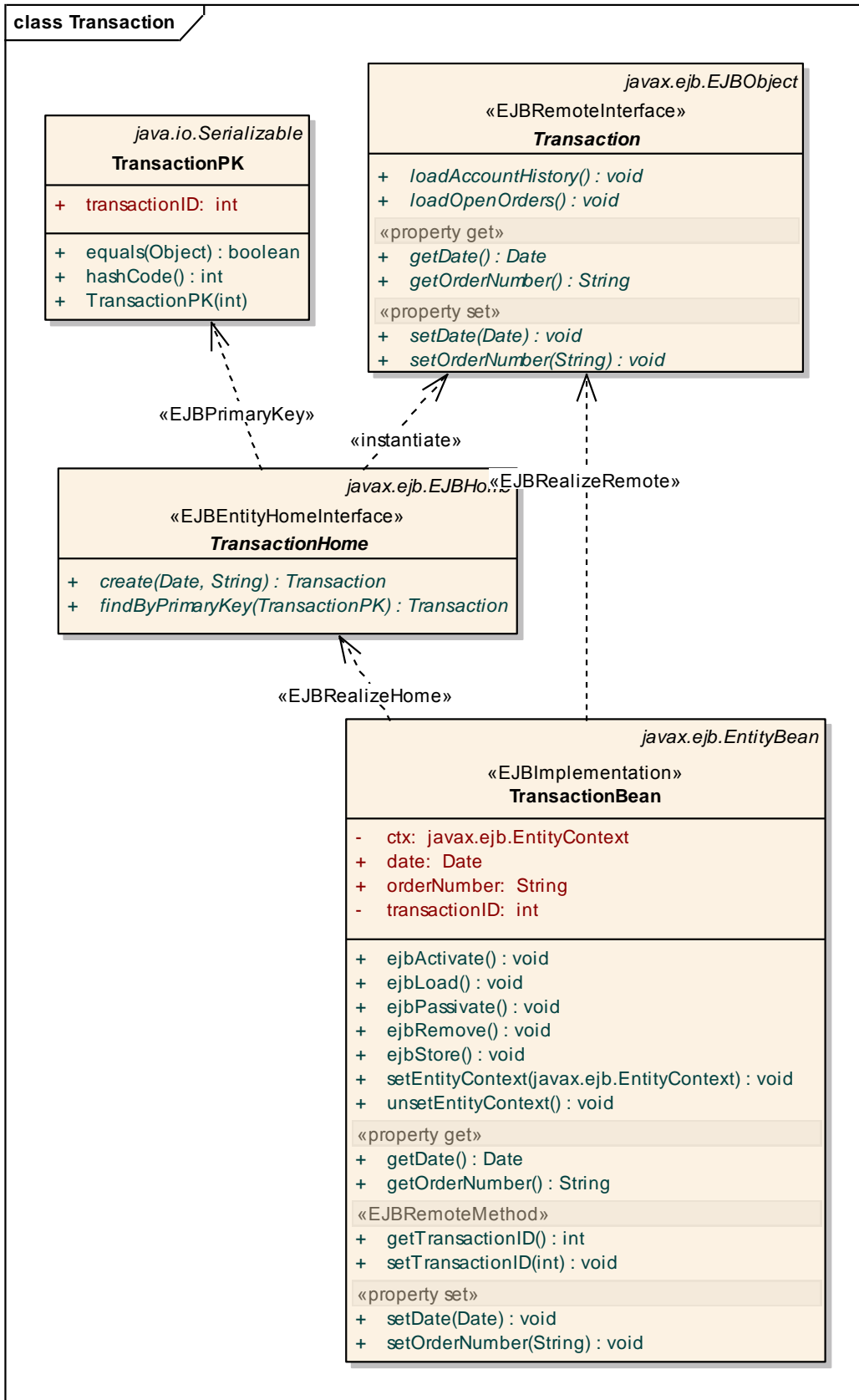


Diagram: META-INF

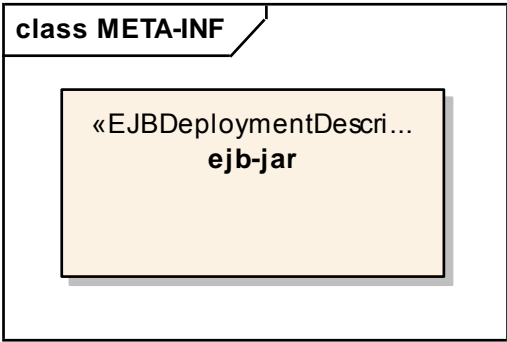


Diagram: OrderStatus

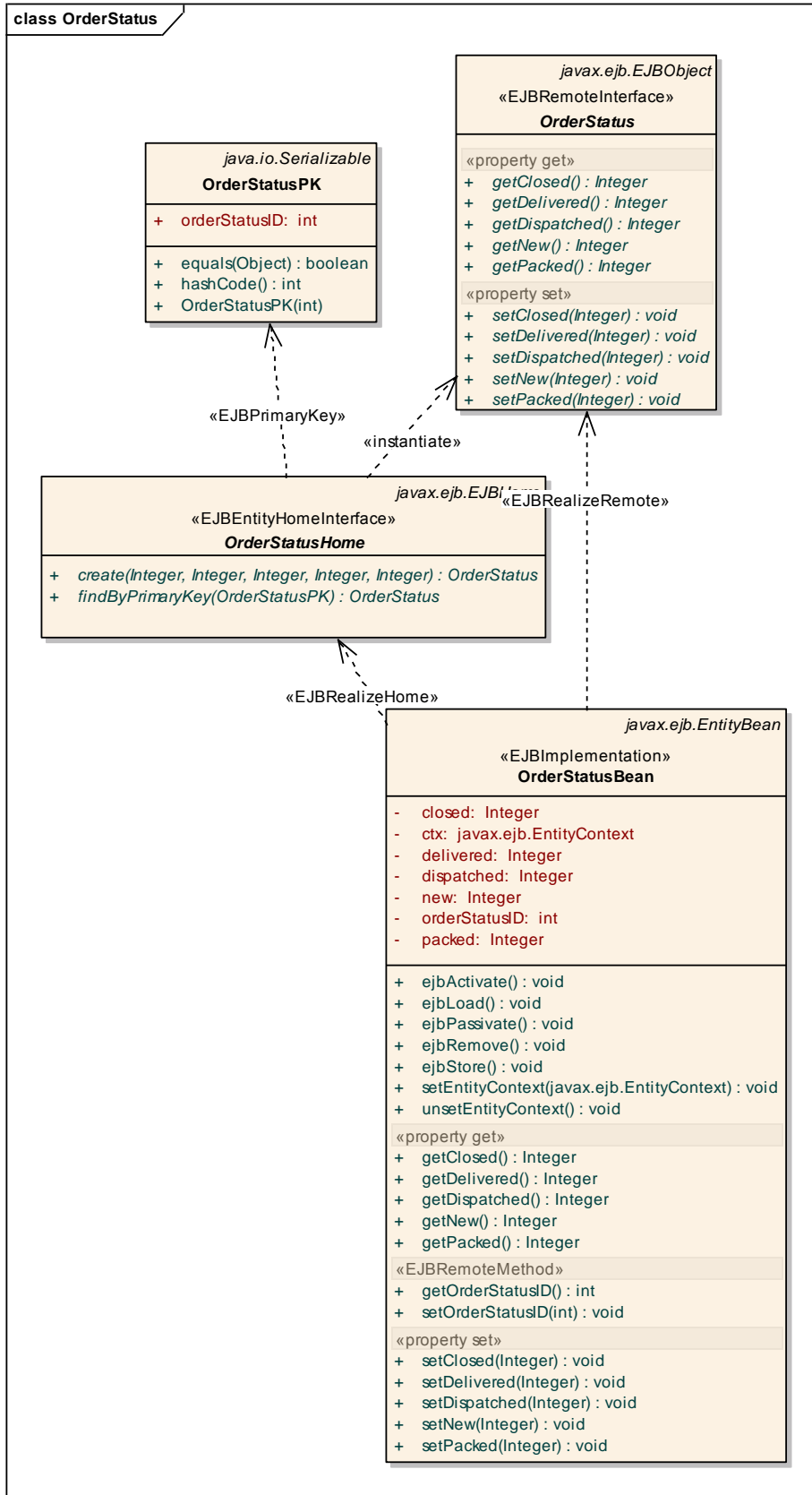


Diagram: META-INF

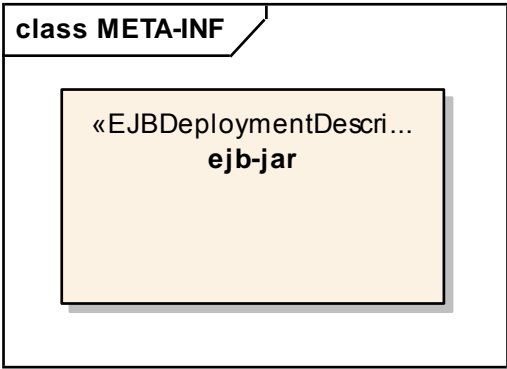


Diagram: ShoppingBasket

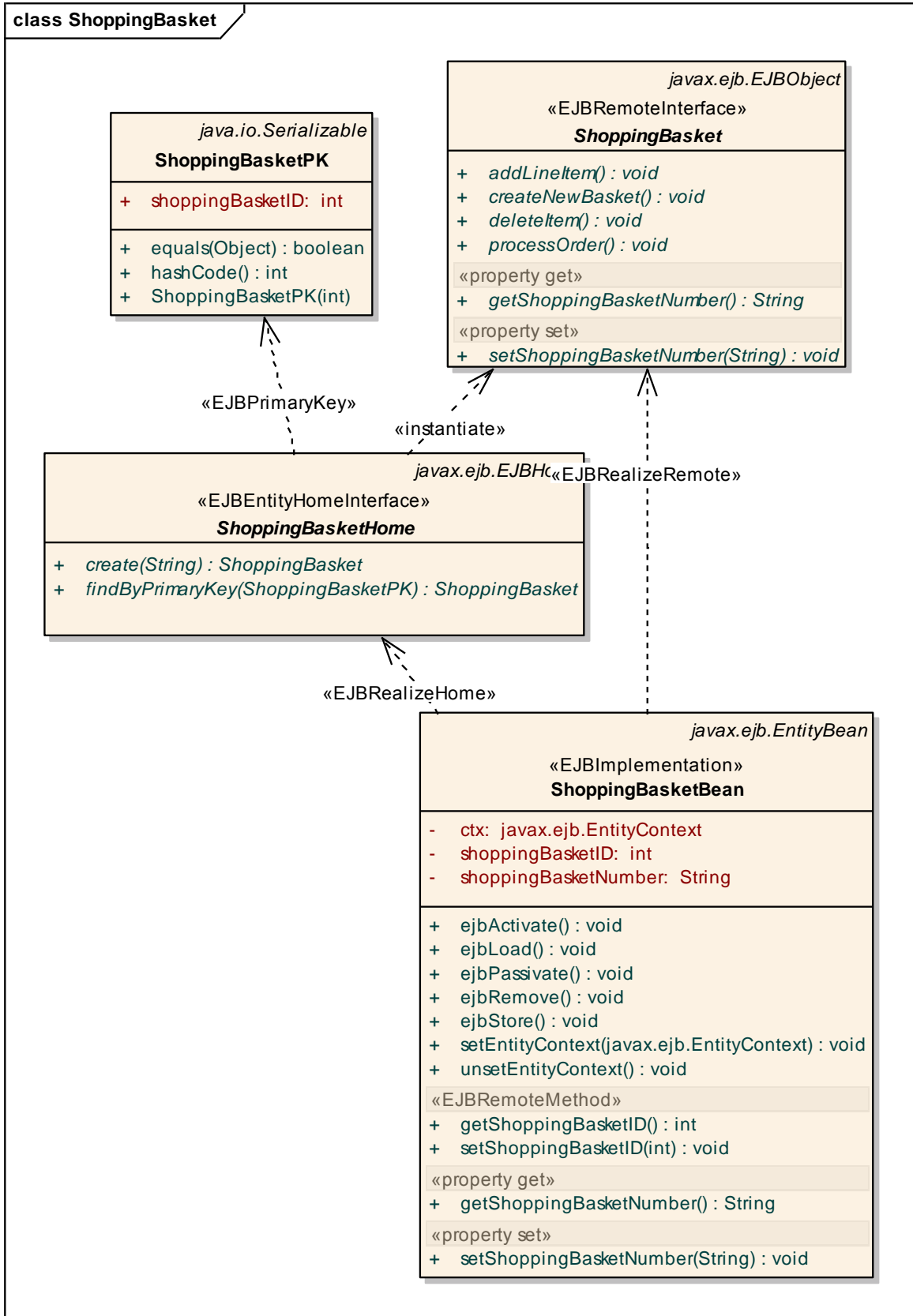


Diagram: META-INF

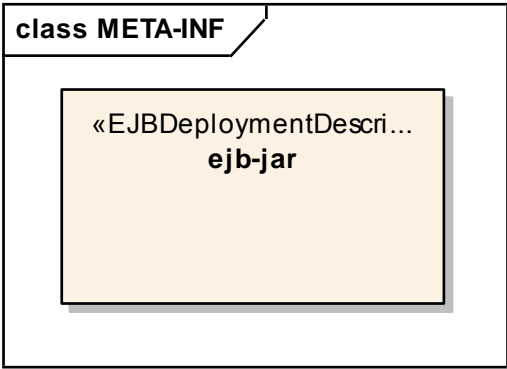


Diagram: EJB Session



Diagram: Account

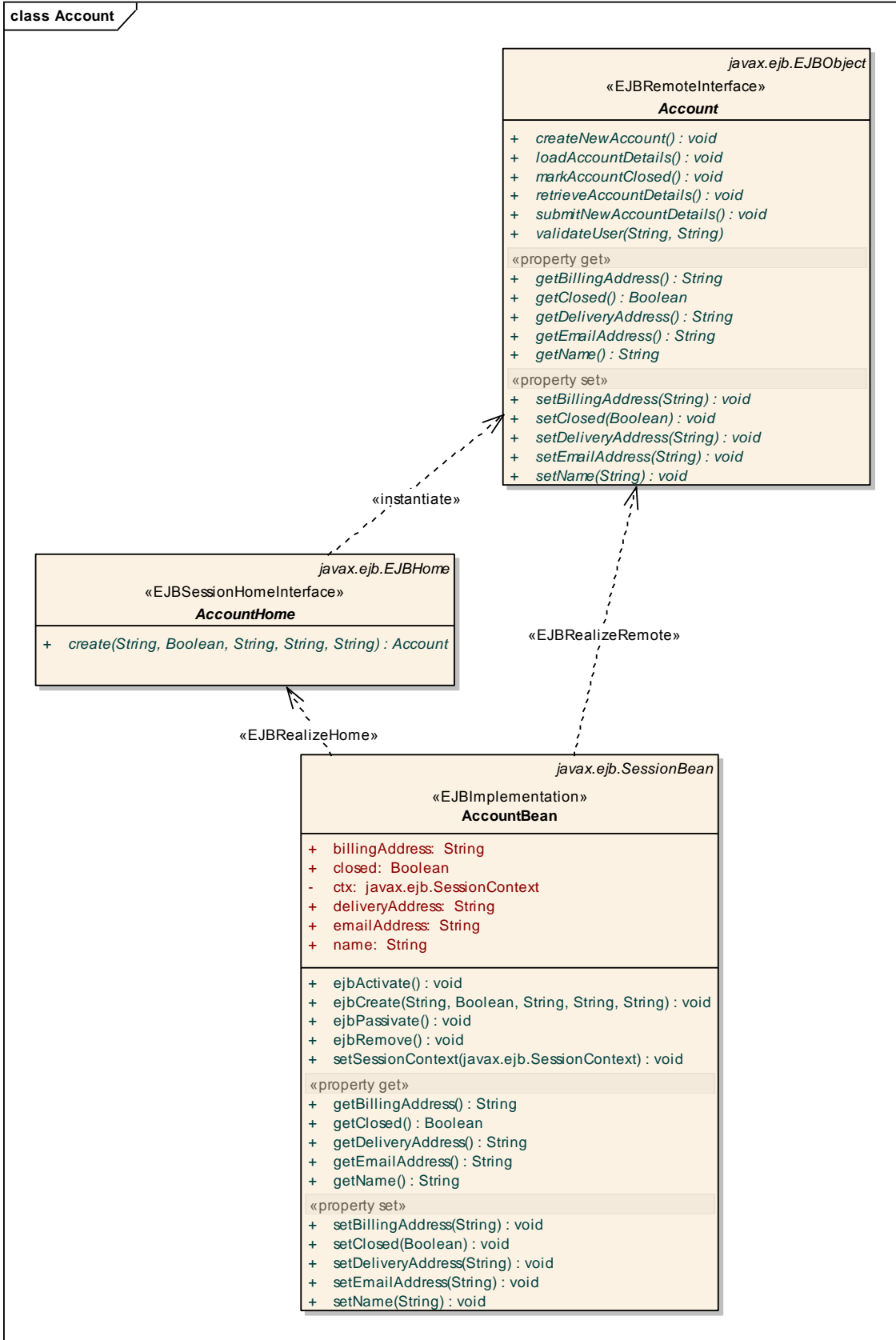


Diagram: META-INF

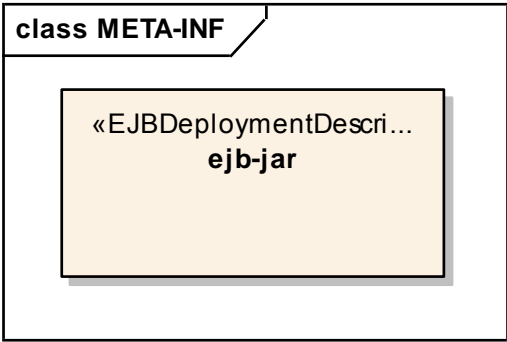


Diagram: LineItem

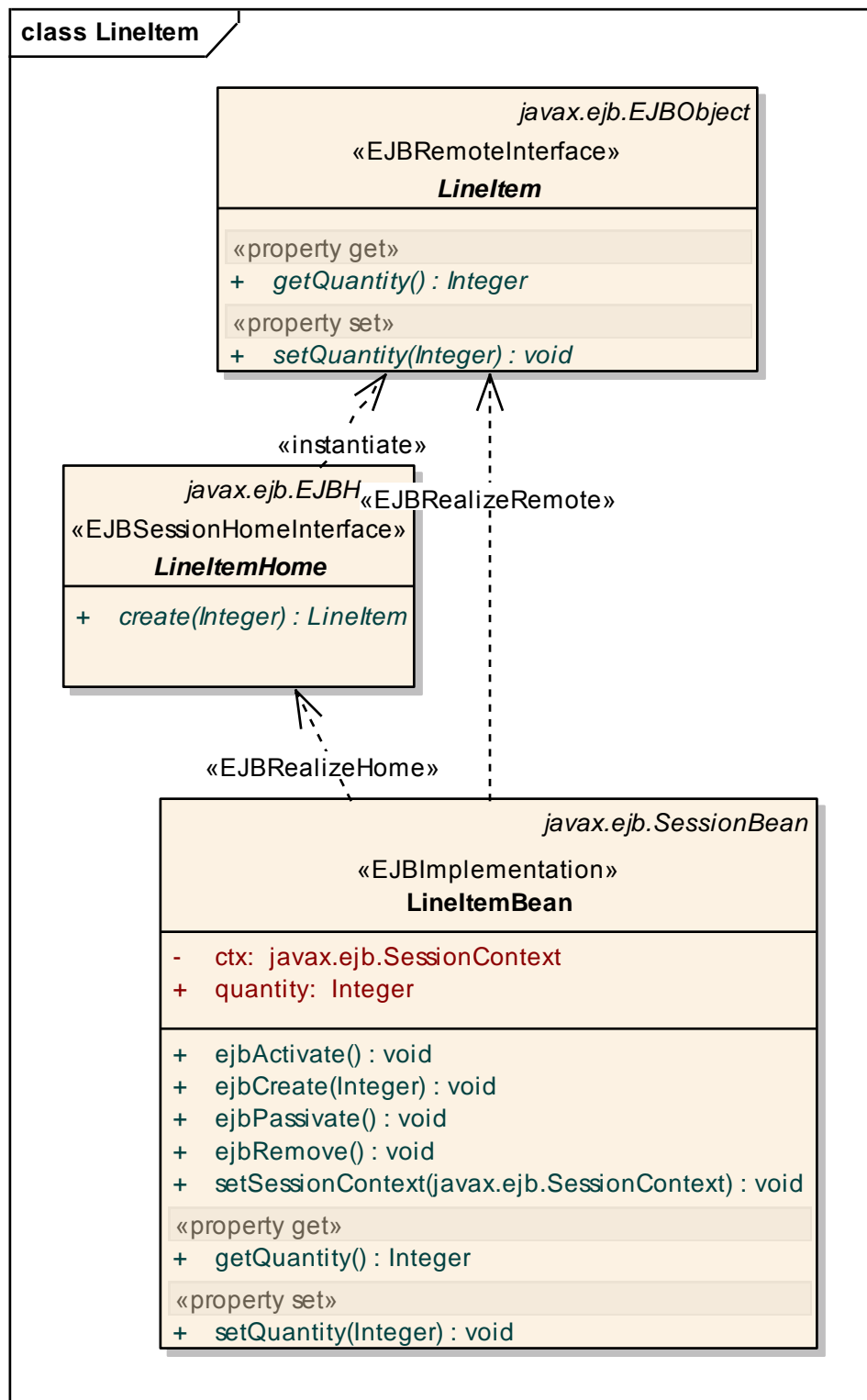


Diagram: META-INF

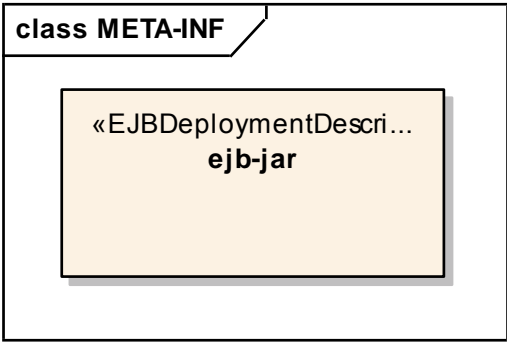


Diagram: Order

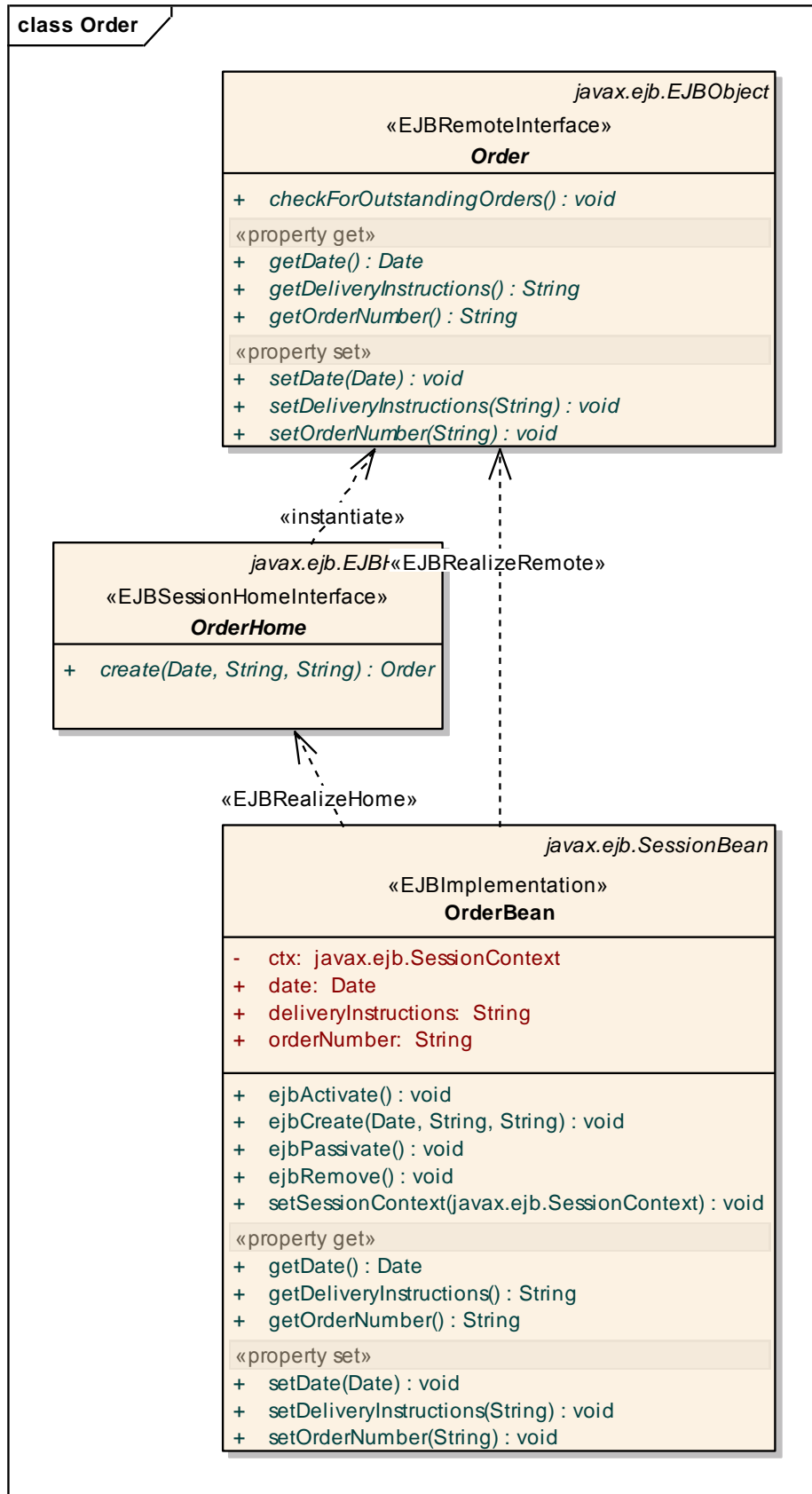


Diagram: META-INF

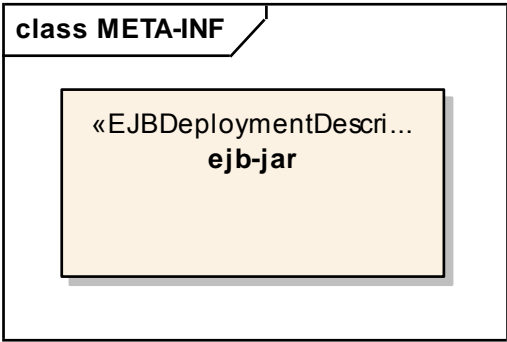


Diagram: ShoppingBasket

class ShoppingBasket

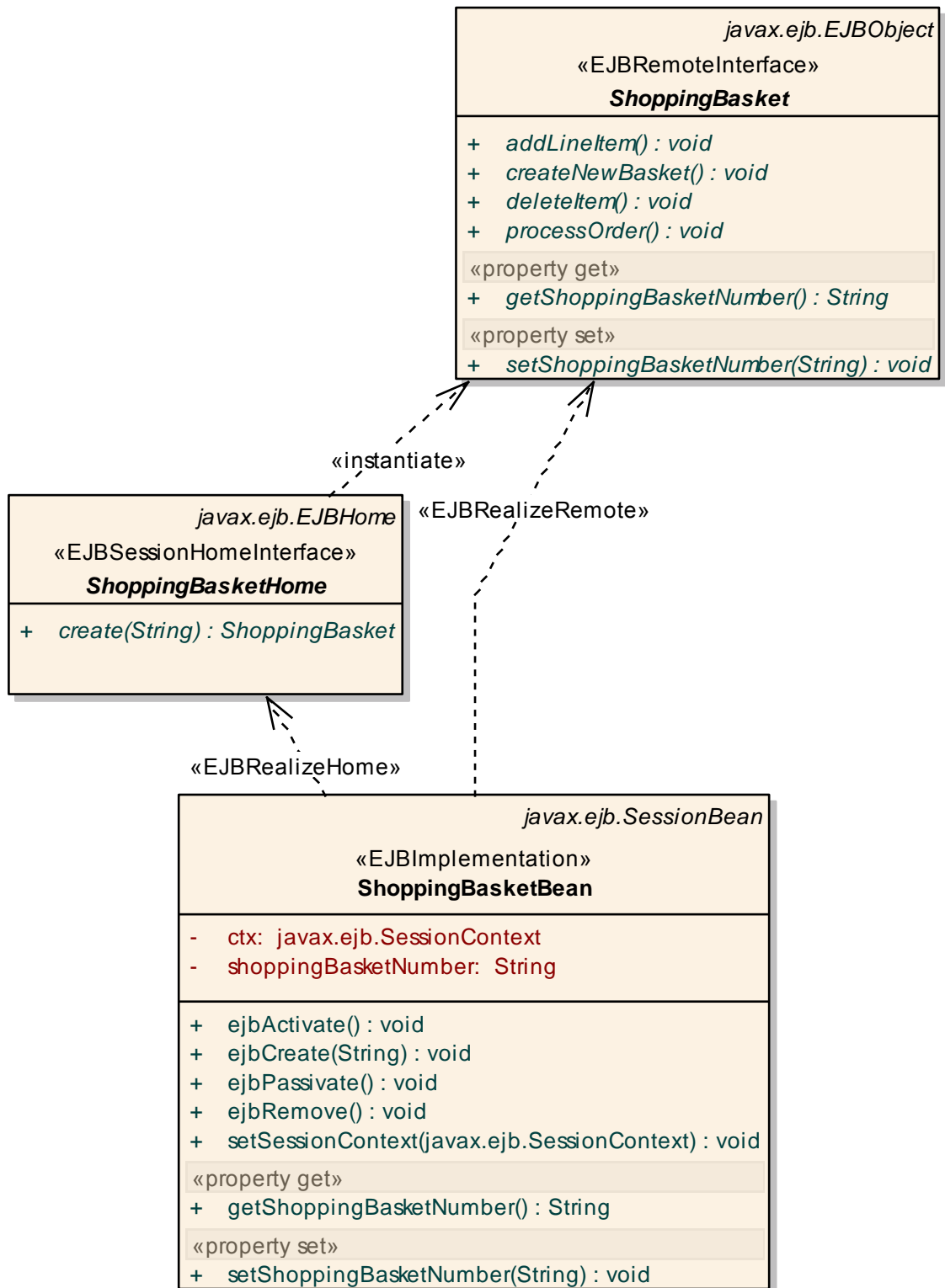


Diagram: META-INF

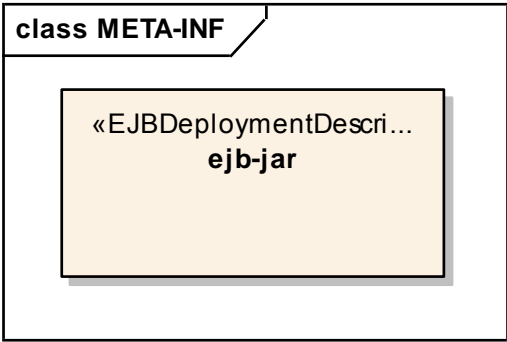


Diagram: StockItem

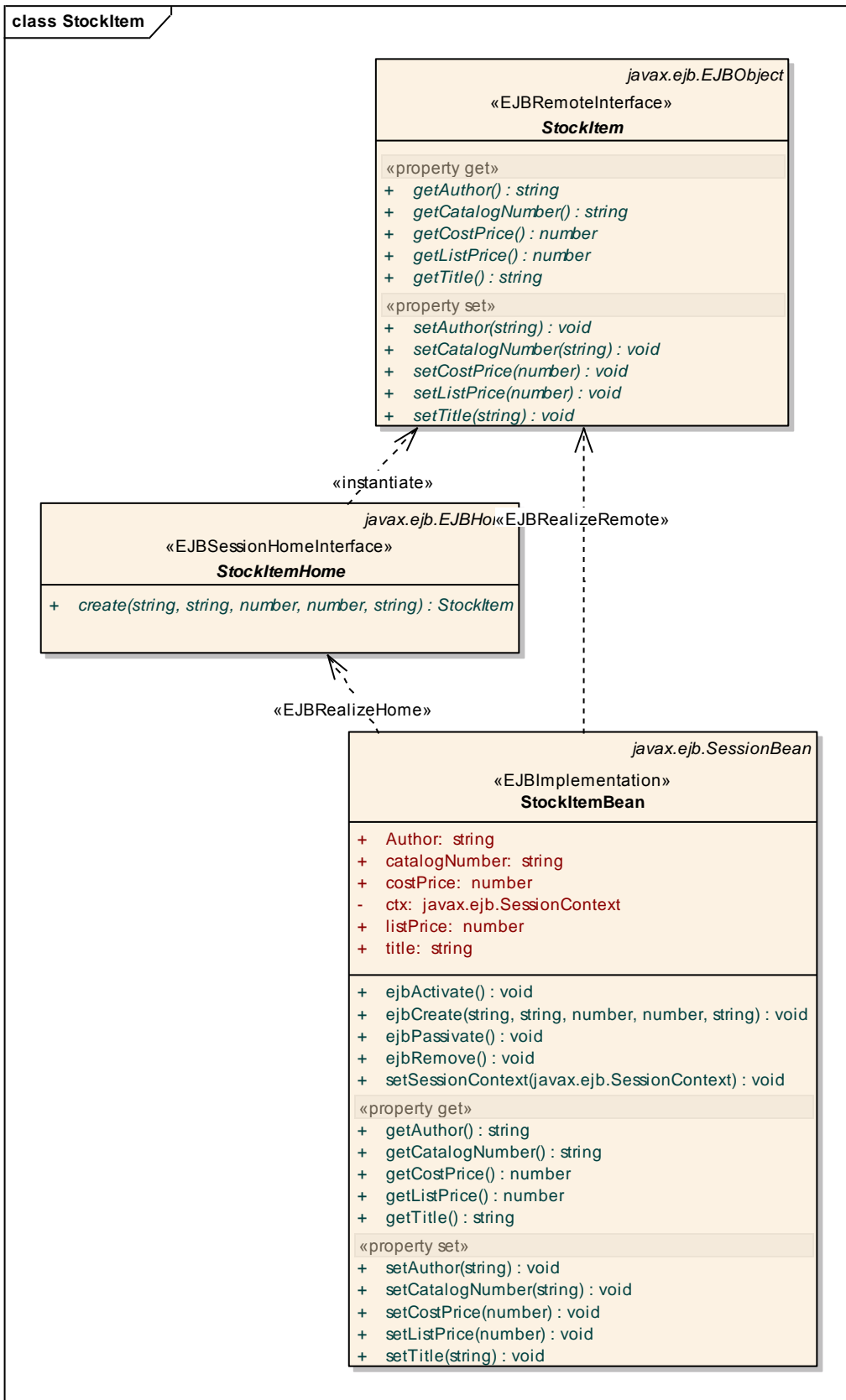


Diagram: META-INF

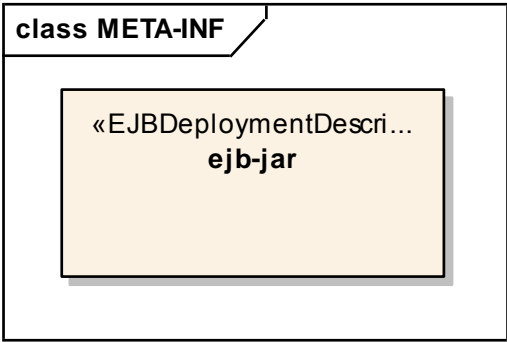


Diagram: Transaction

class Transaction

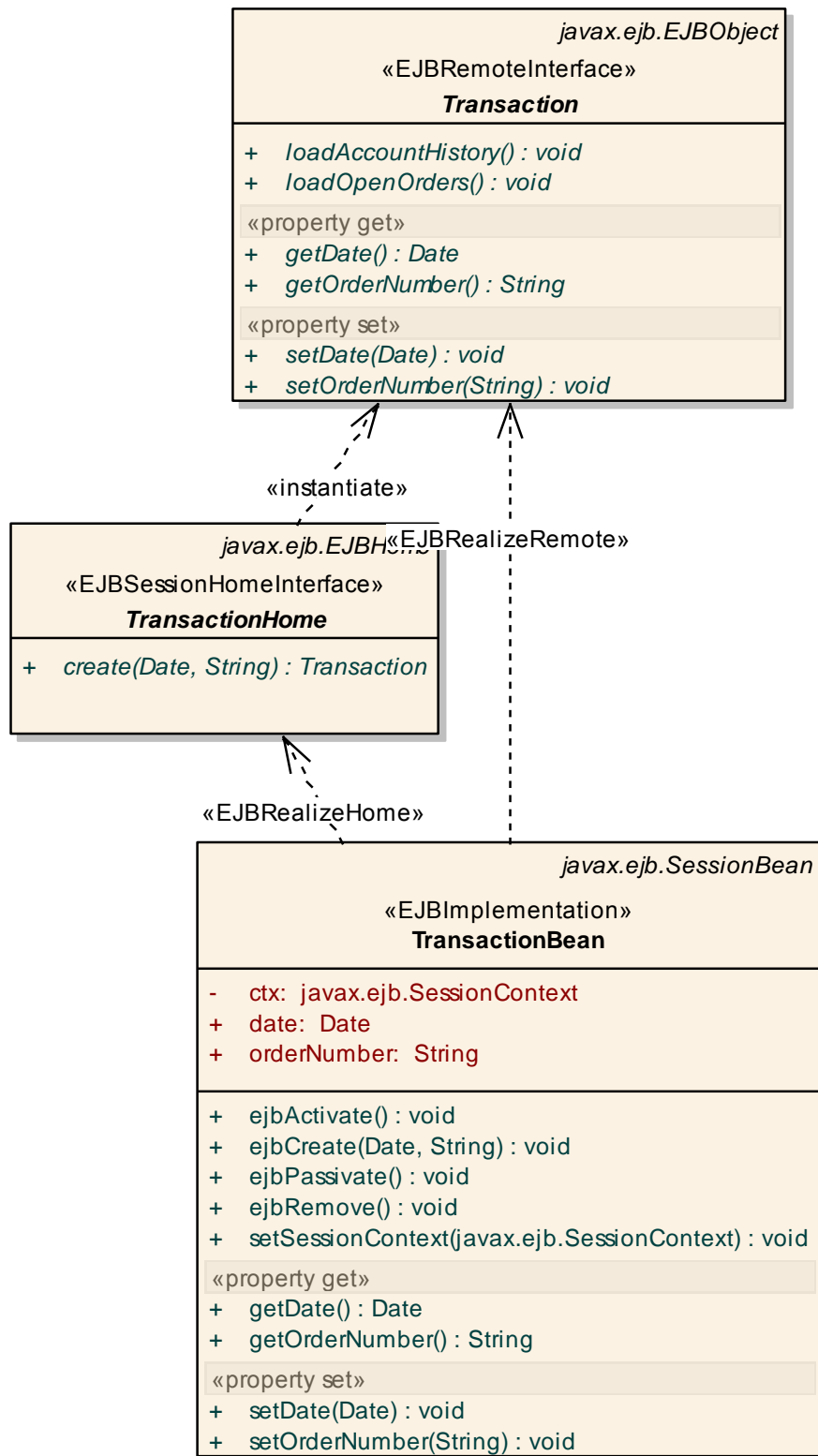


Diagram: META-INF

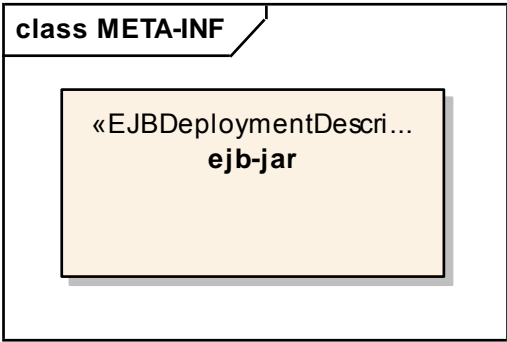


Diagram: OrderStatus

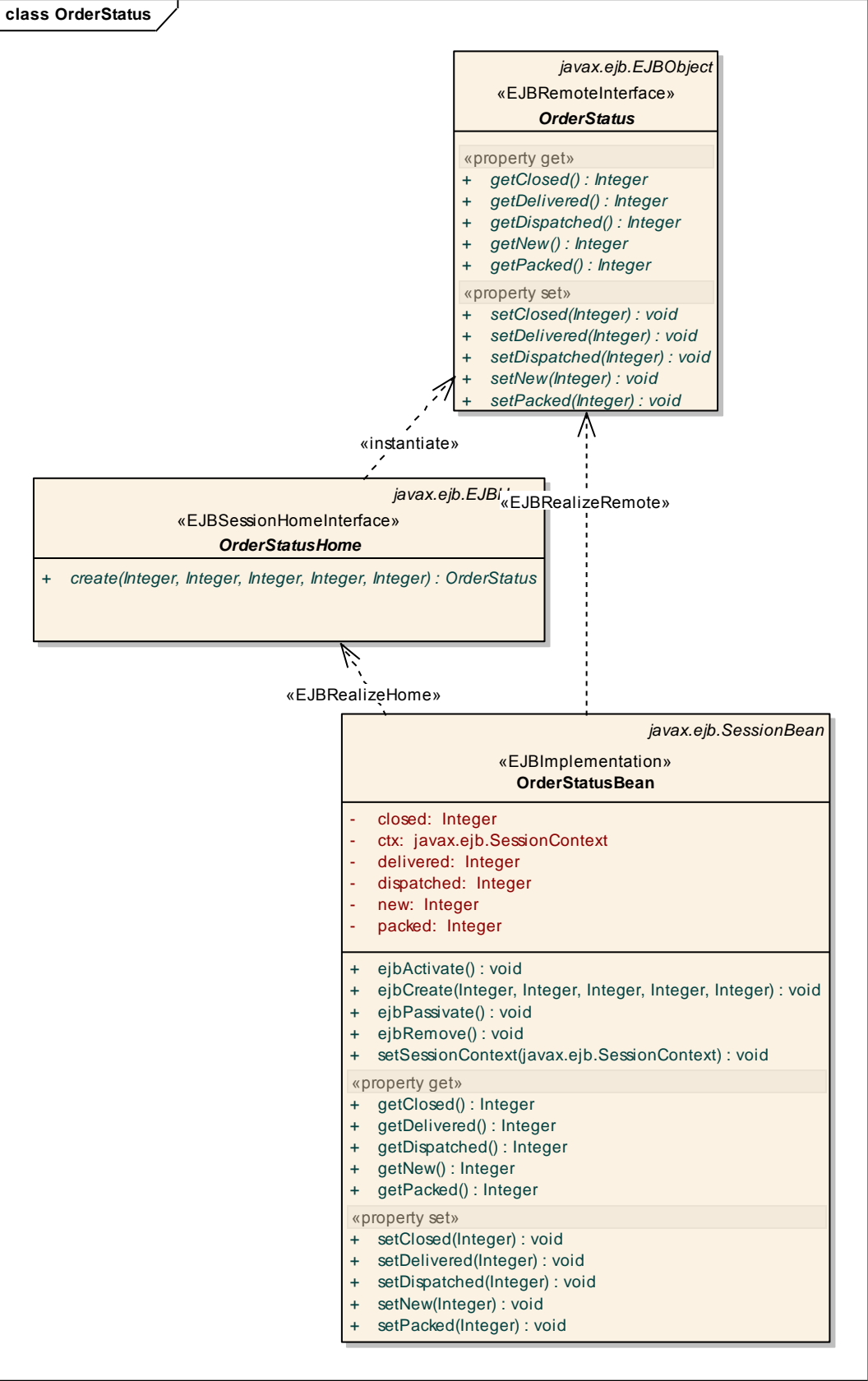


Diagram: META-INF

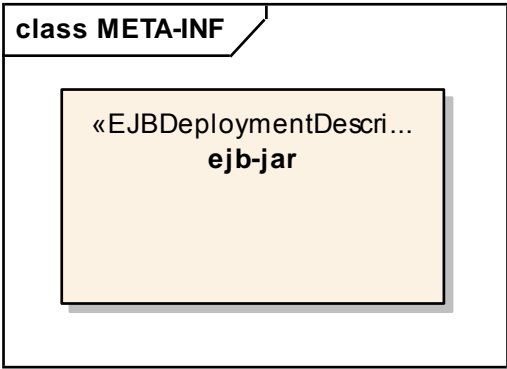


Diagram: XSD

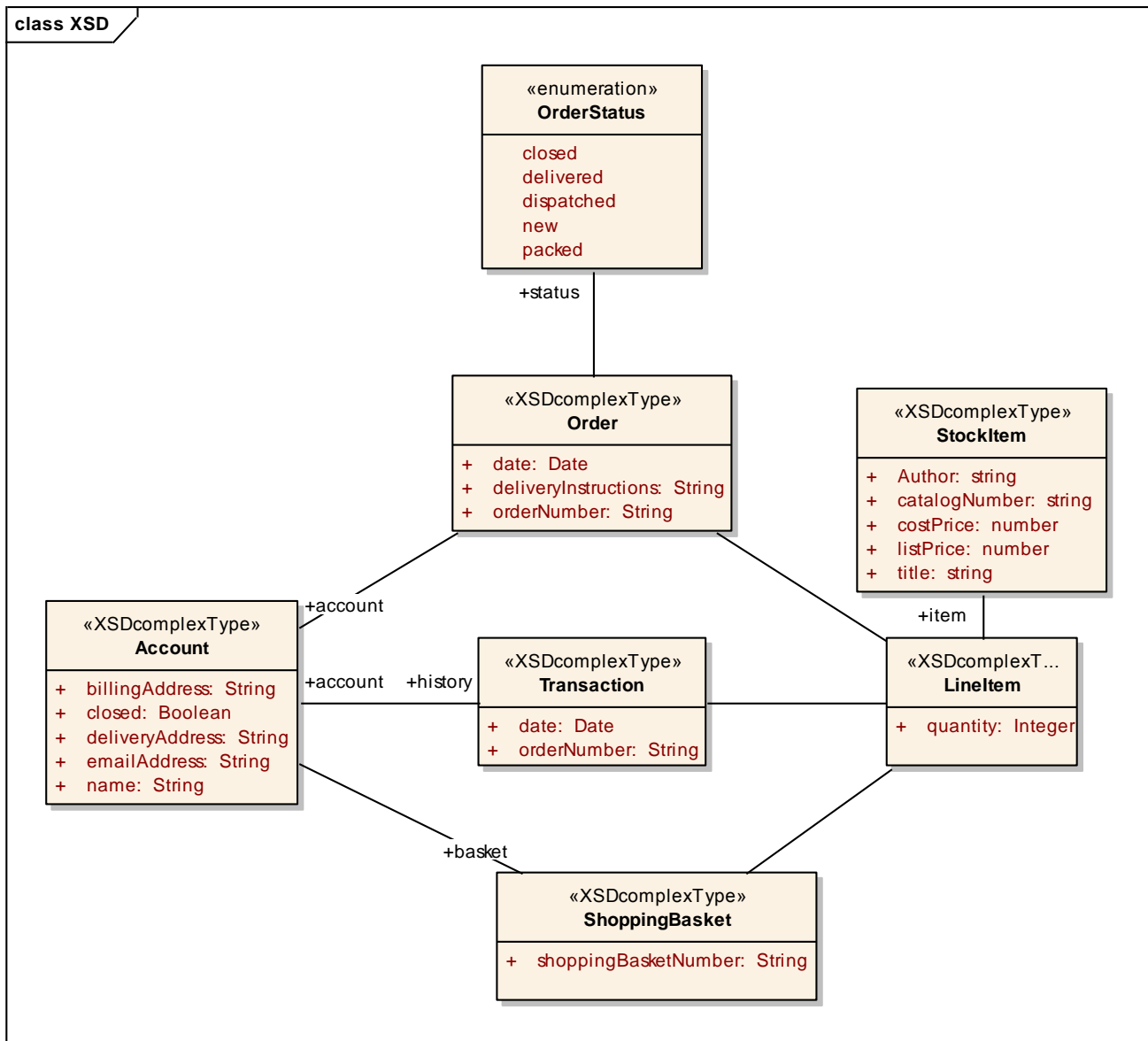


Diagram: Java Model

class Java Model

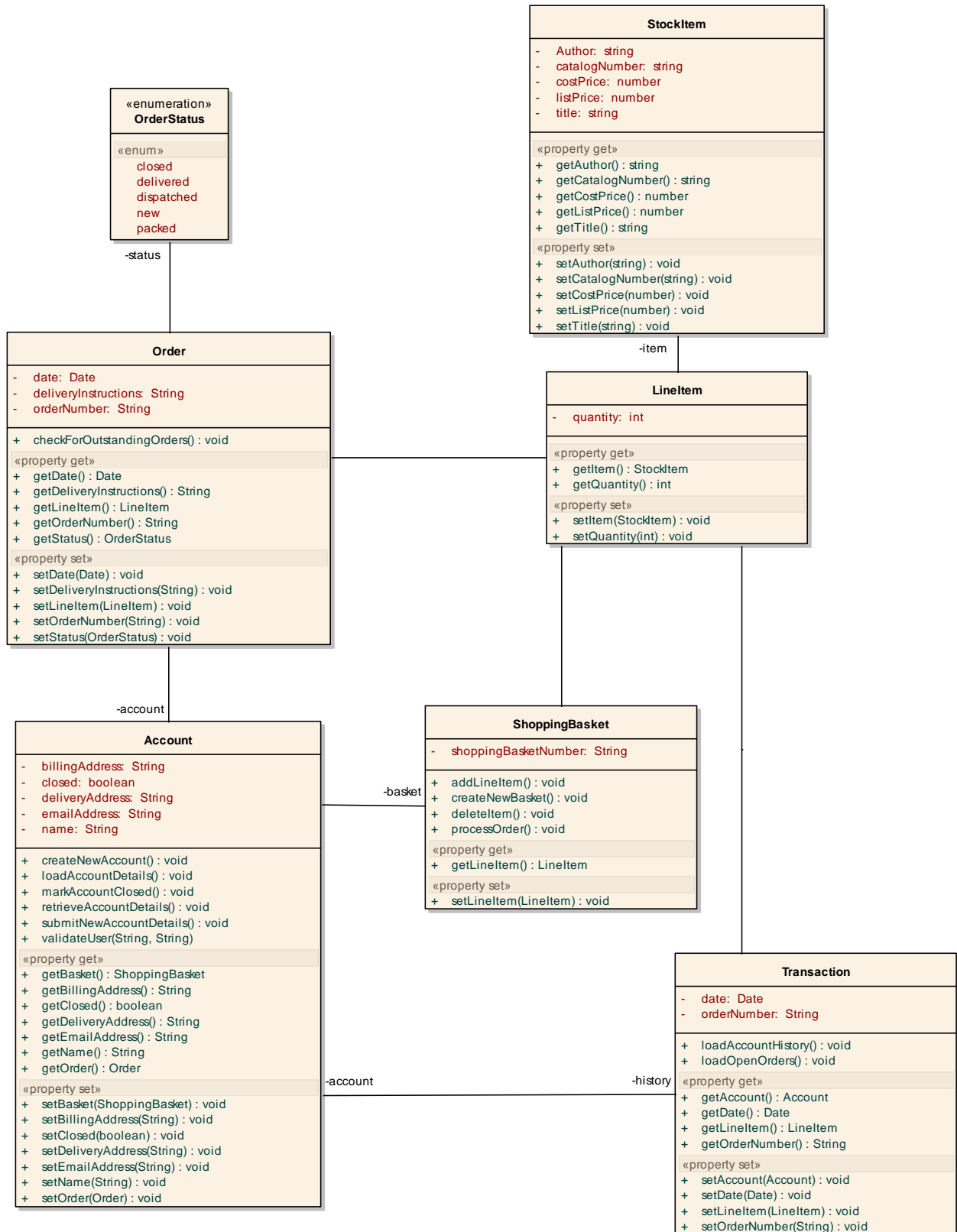


Diagram: Test Environment Model

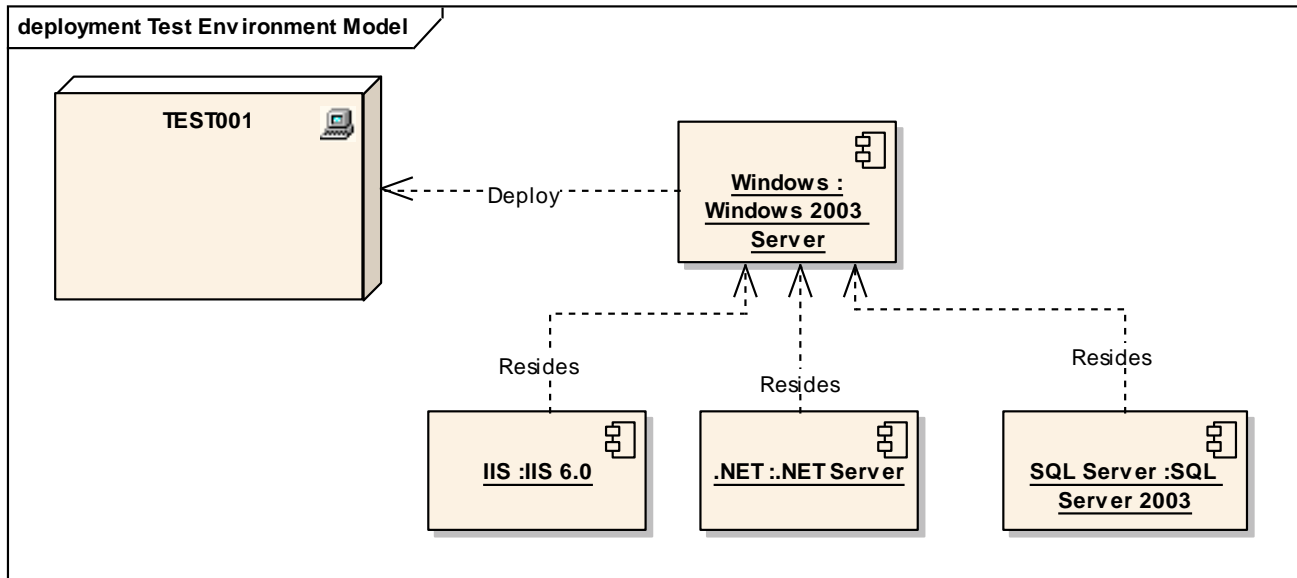


Diagram: Development Environment Model

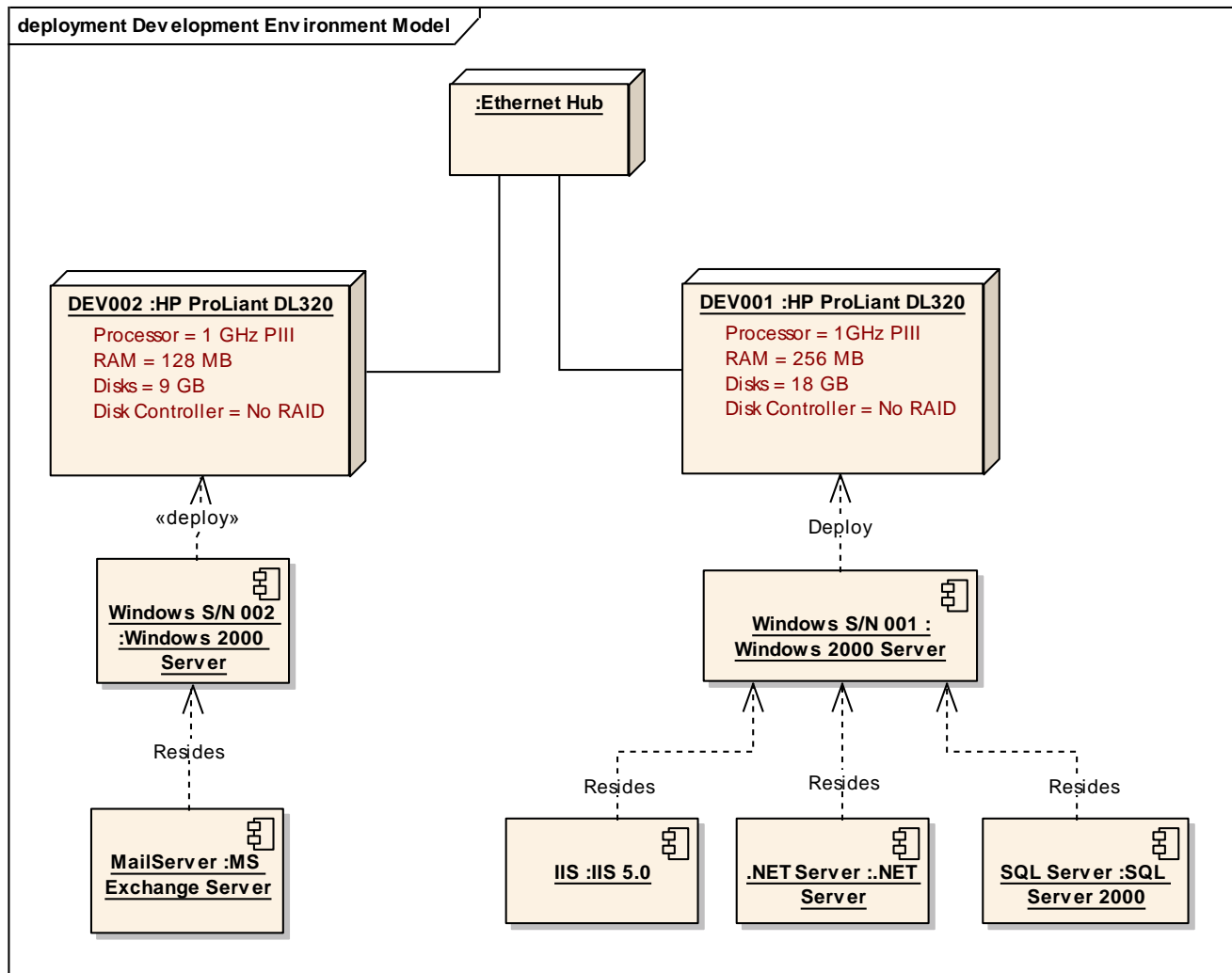


Diagram: View Orders

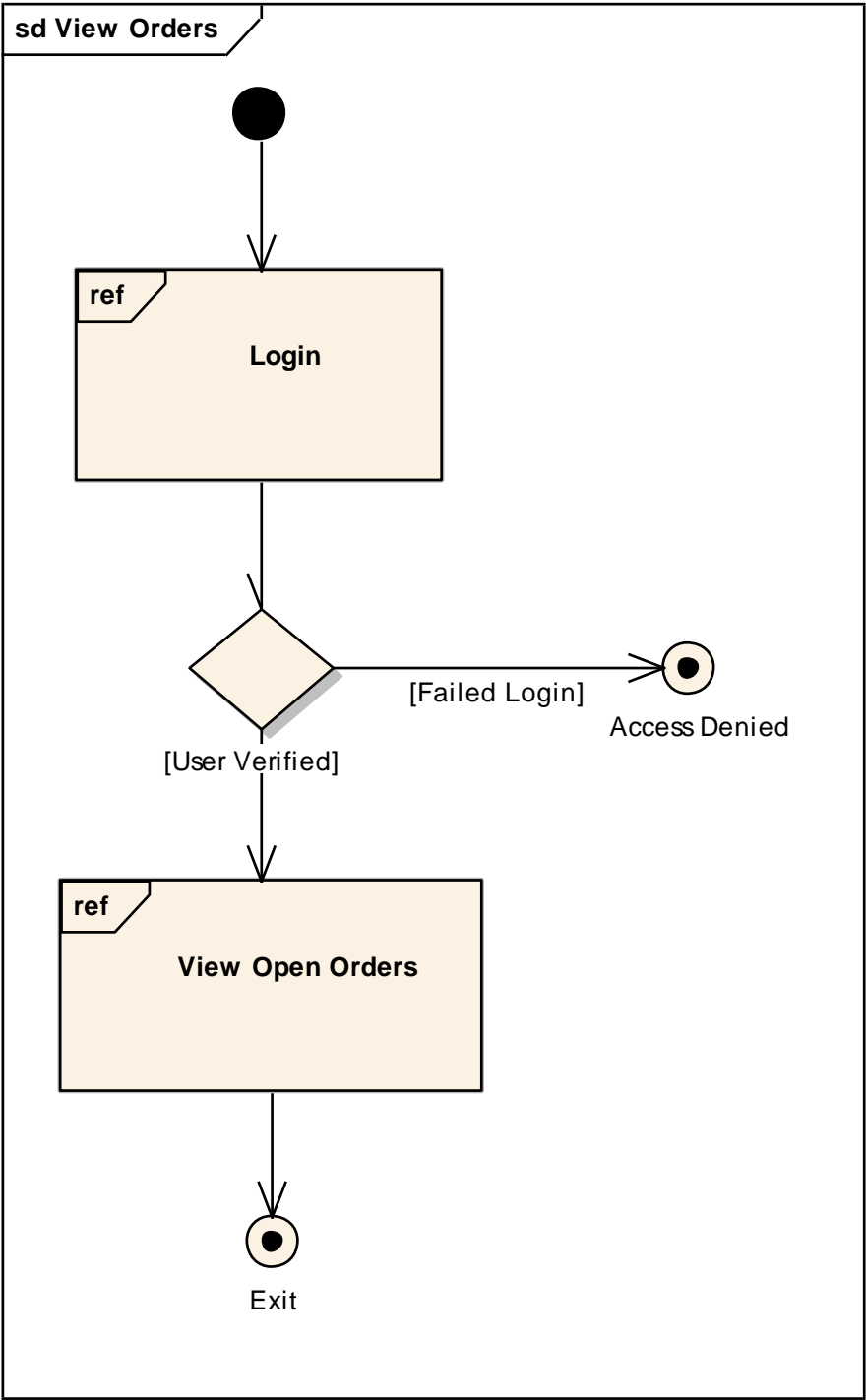


Diagram: Login

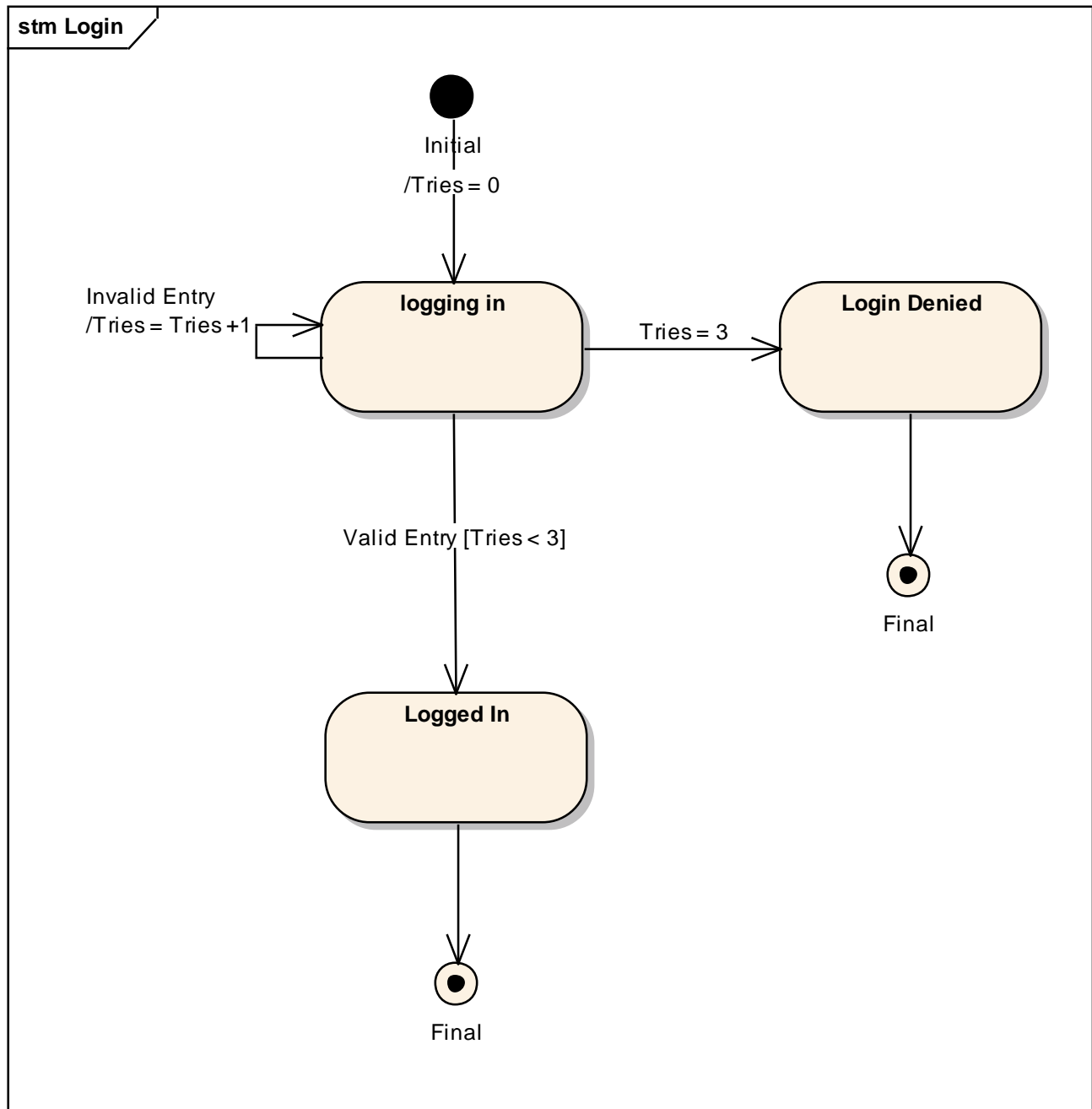


Diagram: Search

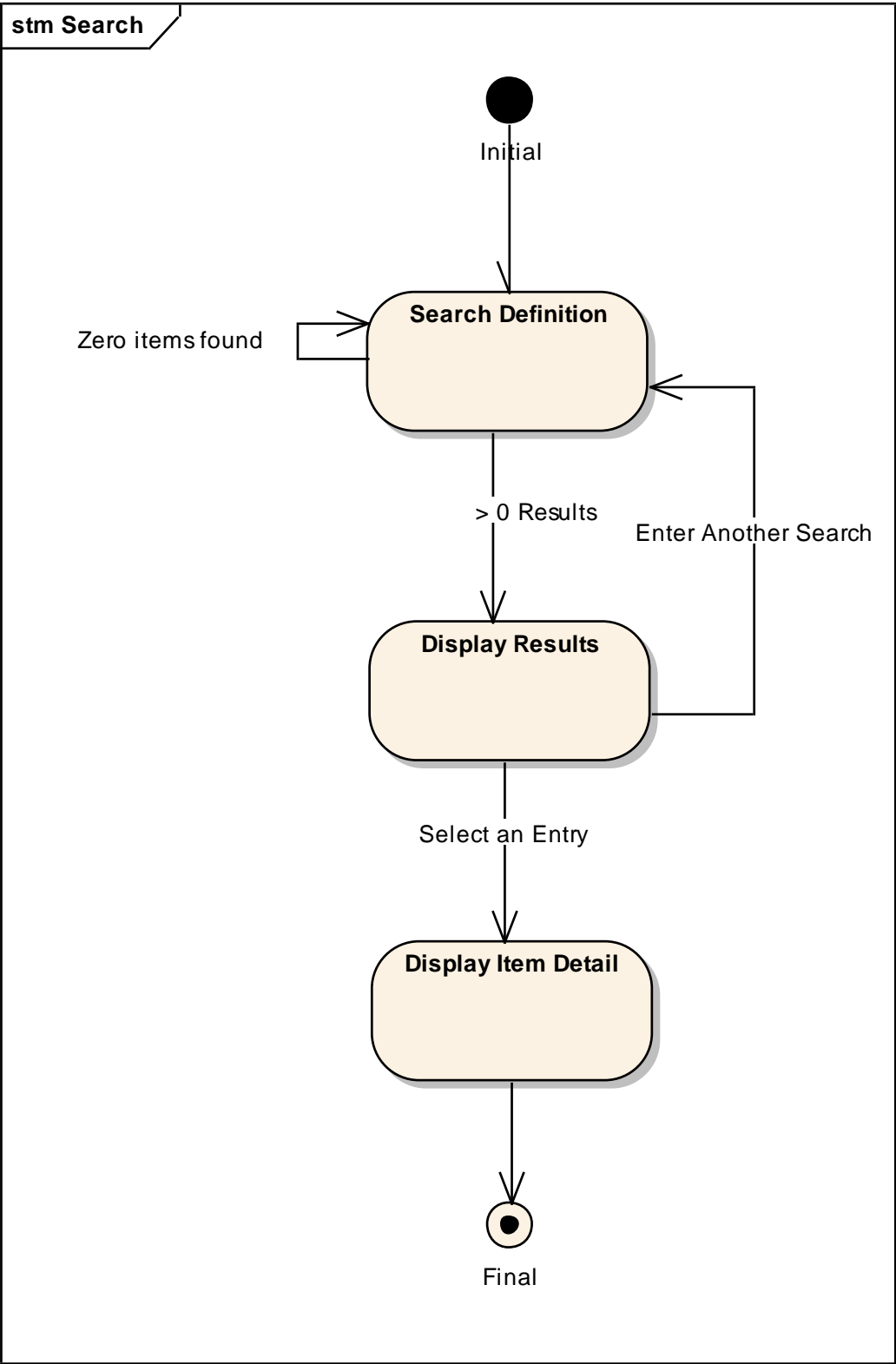


Diagram: Customer Order

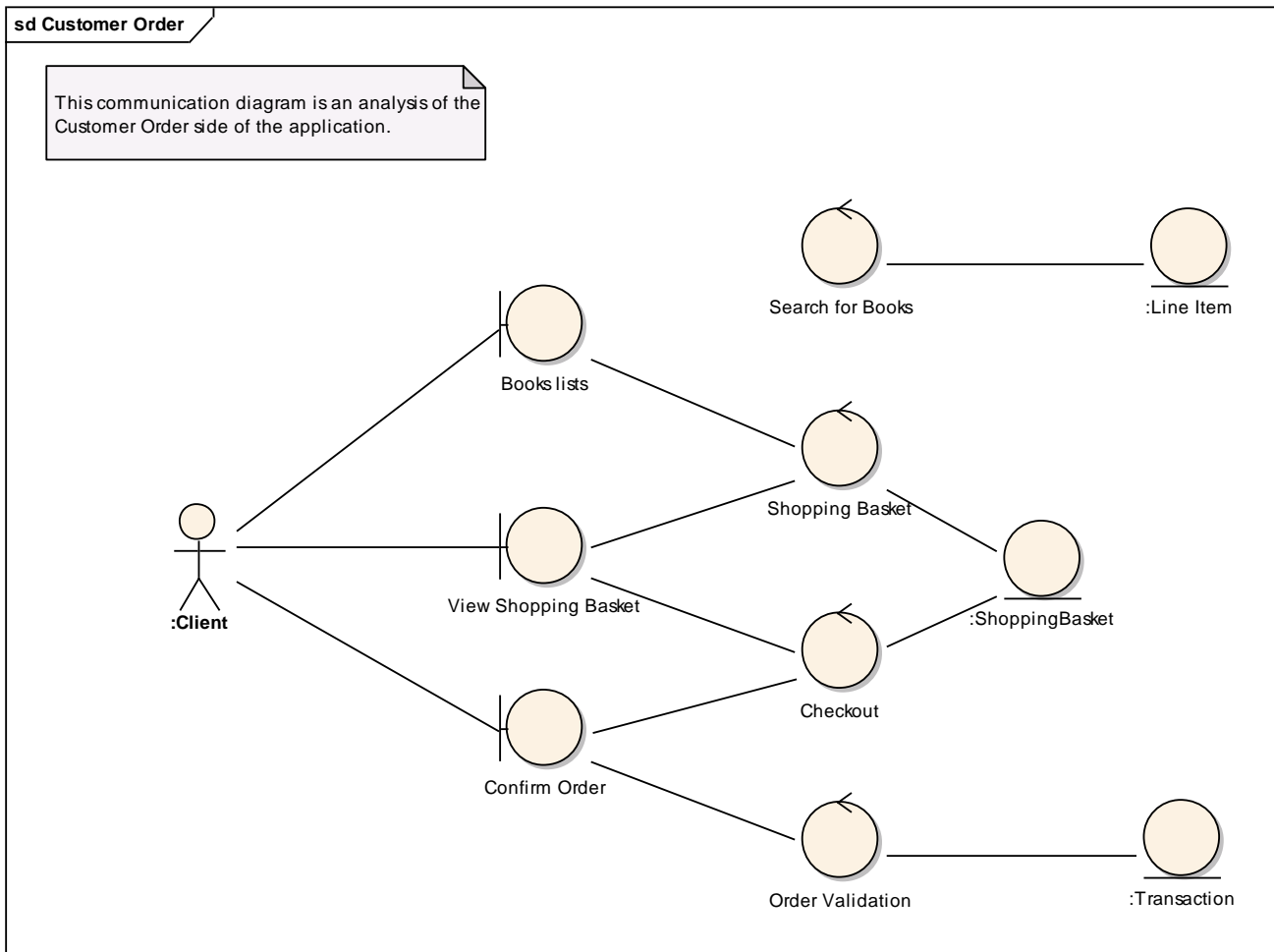


Diagram: QA Model

class QA Model

Quality Assurance

There are a number methods for setting up testing definitions and tracking test results in Enterprise Architect. Two key formats that can be used for this are:

1. Internal test definitions within an element.
- 2.. Attaching a custom element of type "Test Case" to existing elements.

Below are links with examples of using these two methods.



[1. Display internal testing in an Element \(i.e. Class\)](#)



[2. Use a Test Case Element](#)

Enterprise Architect also supports custom elements for Issues and Changes. These are useful for logging maintenance updates. These can be also be viewed with the test definitions displayed.

For examples of this see:



[Changes Example](#)



[Issues Example](#)

For a document of the Test Plan see the internal RTF document covering the test strategy:

Testing::Test Plan



Diagram: Take Orders

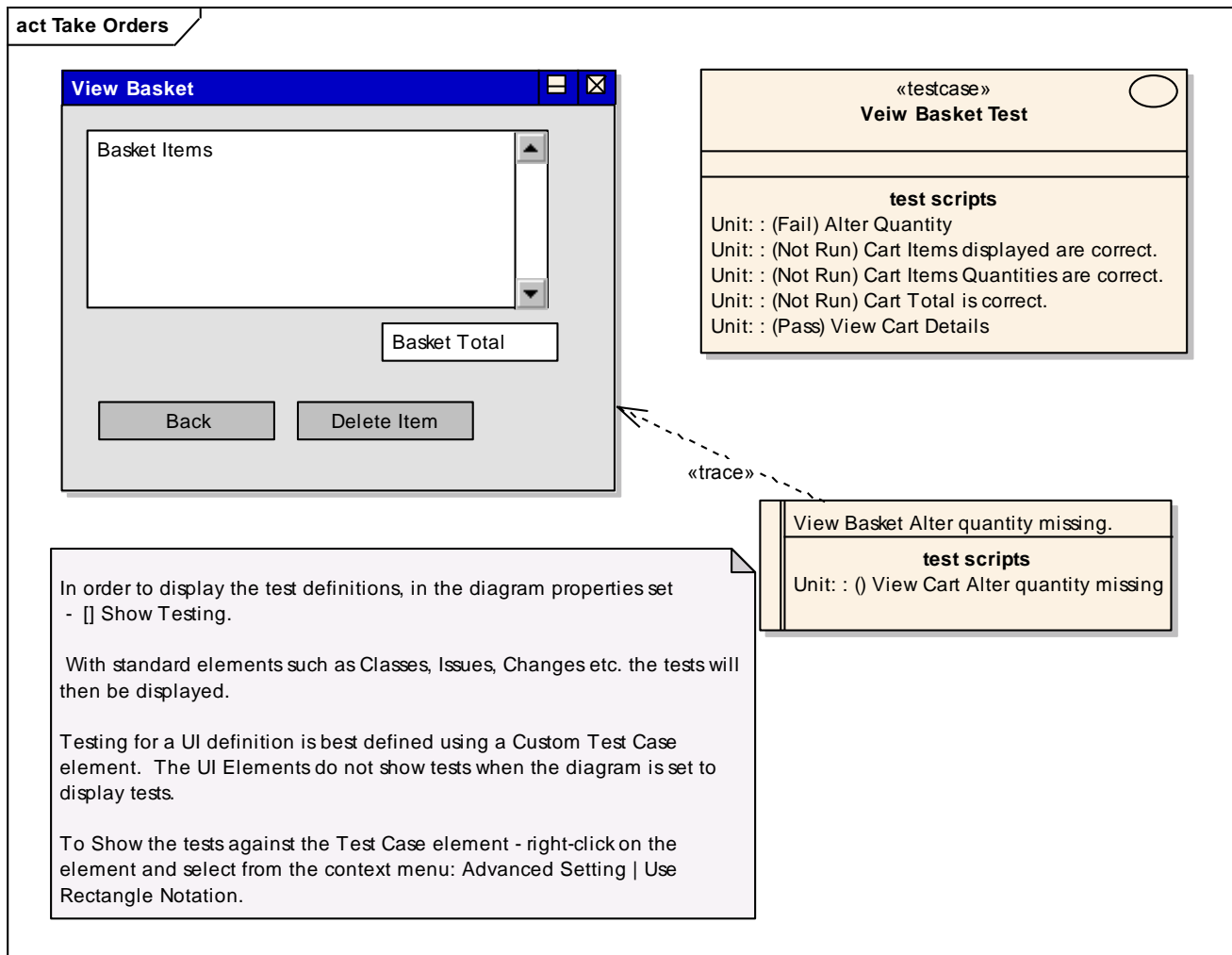


Diagram: Place Order

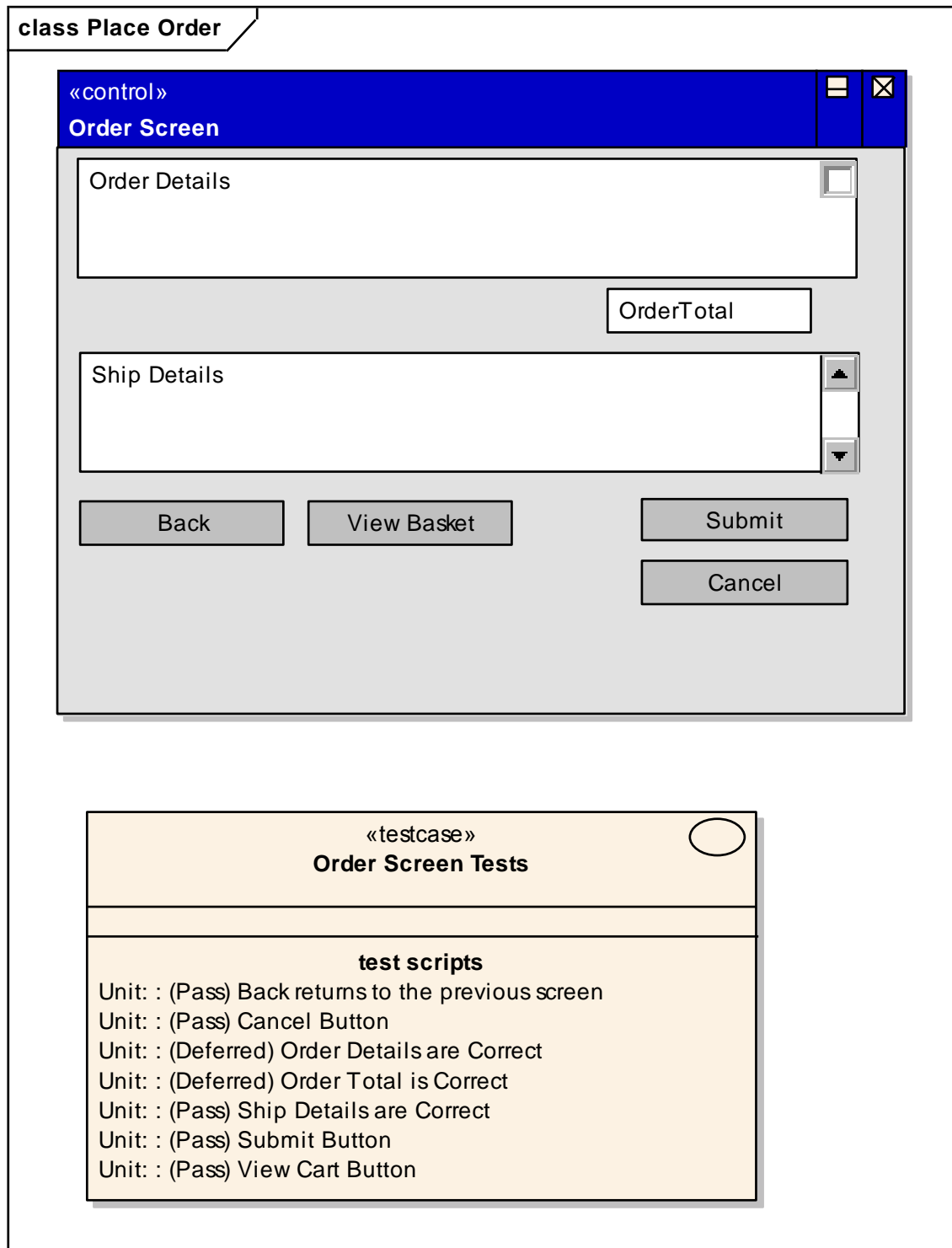


Diagram: Account

custom Account

Below are some classes defined in the Implementation Model. These classes have been copied to this diagram. The diagram is set to display Test scripts.

Abstract Class Model (PIM)::Transaction

test scripts

Unit: : (Not Run) Confirm quantity
Unit: : (Deferred) No History
Unit: : (Not Run) View History
Unit: : (Not Run) Zero Quantity

Abstract Class Model (PIM)::Account

test scripts

Unit: : (Not Run) Basic Path
Unit: : (Not Run) Basic Path - Administrator
Unit: : (Not Run) Basic Path - Client
Unit: : (Not Run) Cannot Create New Account
Unit: : (Not Run) No Submit
Unit: : (Not Run) No to Close
Unit: : (Not Run) Outstanding Transactions
Unit: : (Not Run) Validation fails

Diagram: Orders

class Orders

Internal Test Scripts

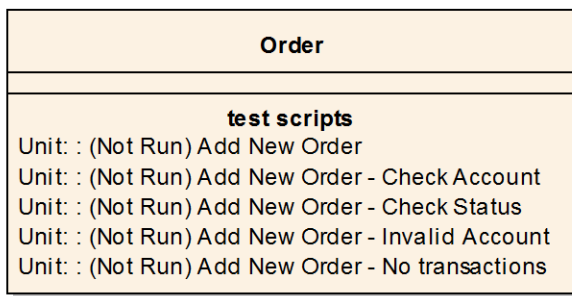
Test scripts can be defined within any element. Below are some classes defined in the Implementation Model that have been copied to this diagram.

The diagram properties have been set to show test cases defined in the elements. To set this:

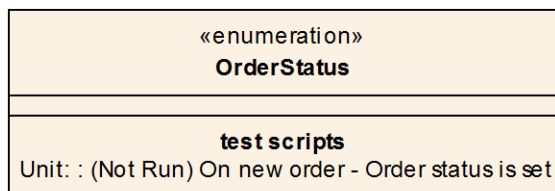
- Open the diagram properties (F5).
- Set ☐ Show Testing.



Show Test Scripts



This is the Order Class as defined in:
System Model | Implementation Model
| C# Model.



This is the OrderStatus Class as defined
in: System Model | Implementation
Model | C# Model.

Diagram: Shopping Cart

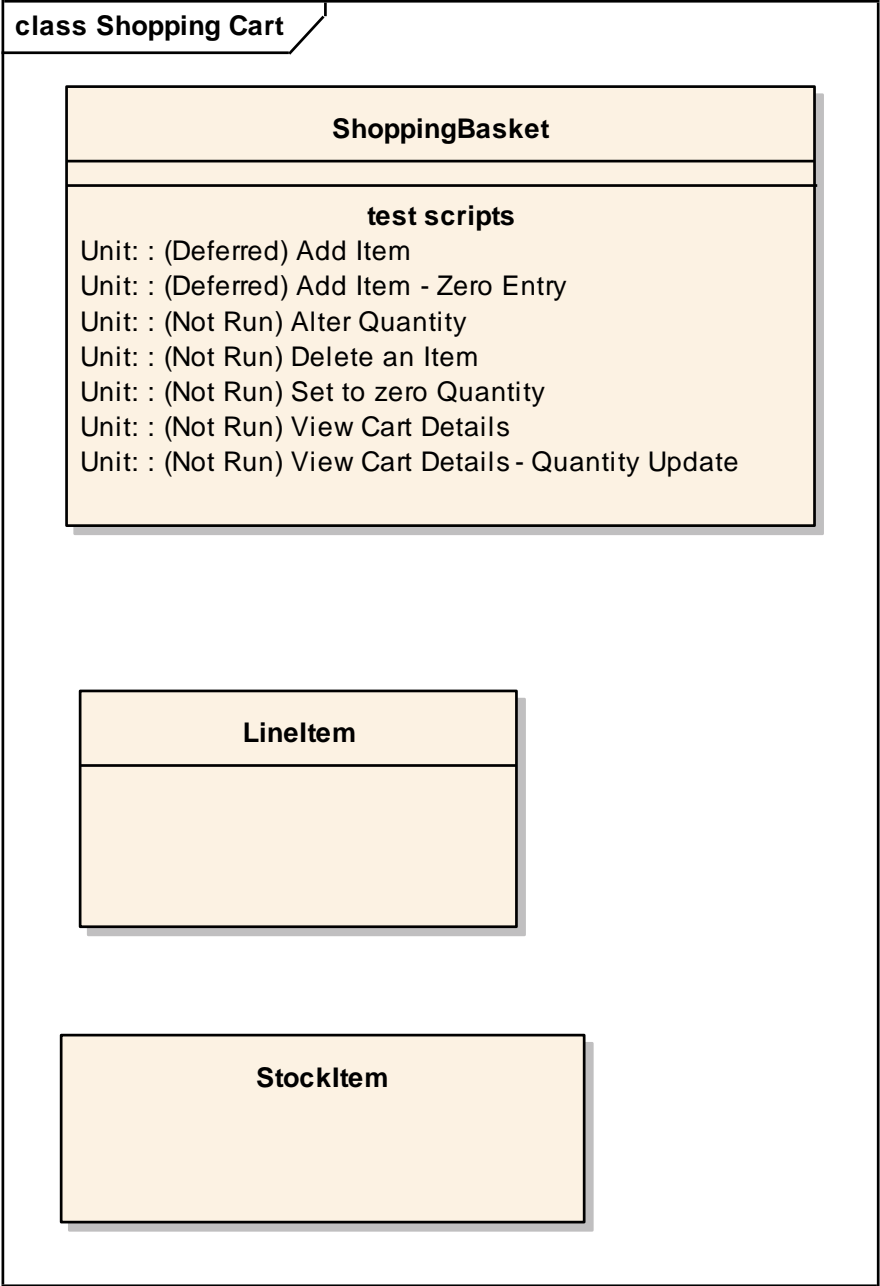


Diagram: User Interface

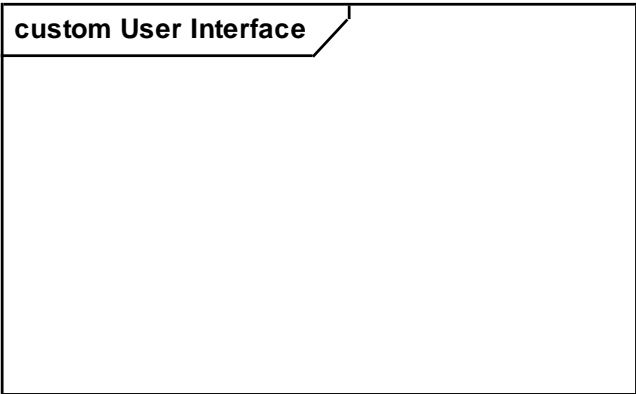


Diagram: Back End Services

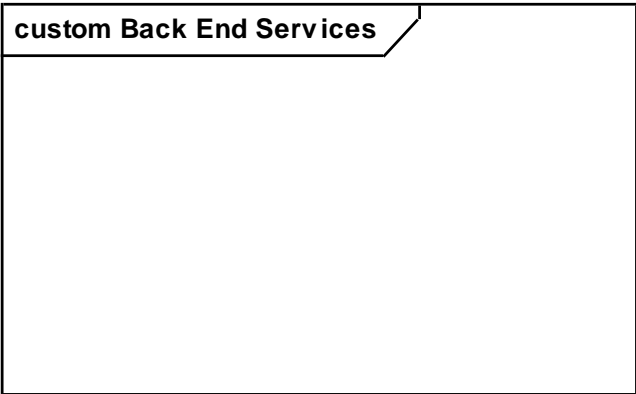


Diagram: Delivery Process

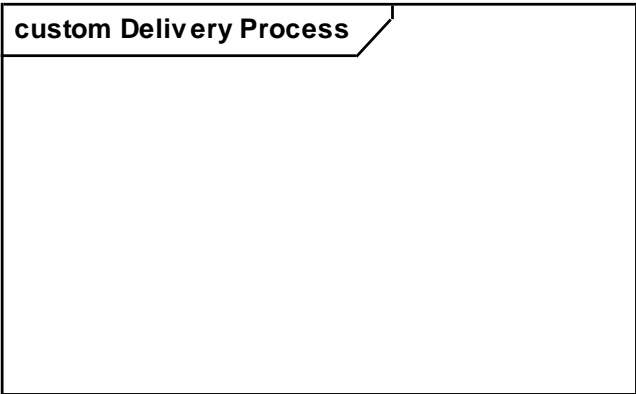


Diagram: Issues

custom Issues

Issues

EA supports custom Elements of type 'Issue'. These can be linked to other elements in the repository or used as a separate lists of issues.

Below are a set of issues for the shopping basket with test definitions listed against these issues ready for checking once they are confirmed as corrected.

The color markings reflect the Status of the element.

View Basket Alter quantity missing.
test scripts
Unit: : () View Cart Alter quantity missing

(from Take Orders)

View basket total cost - not correct
test scripts
Unit: : (Not Run) Check the Total Cost is correct

Remove if quantity is modified and returns Zero
test scripts
Unit: : (Not Run) Delete an Item
Unit: : (Not Run) Set to zero Quantity

Diagram: Changes

custom Changes

Changes

EA supports custom Elements of type 'Change'. These can be linked to other elements in the repository or used as a separate lists of any changes proposed for the model.

Below are a set of Change elements for the shopping basket with test definitions listed against them. They are ready for checking once they are confirmed as corrected.

The color markings reflect the Status of the element.

View Basket - add: alter quantity against the entries.
test scripts
Unit: : (Not Run) Alter Quantity

View Basket - Add button to update Change
test scripts
Unit: : (Deferred) Update Cart Button operative

Create Account: password confirmation fails
test scripts
Unit: : (Not Run) Password Confirmation

Diagram: Issues

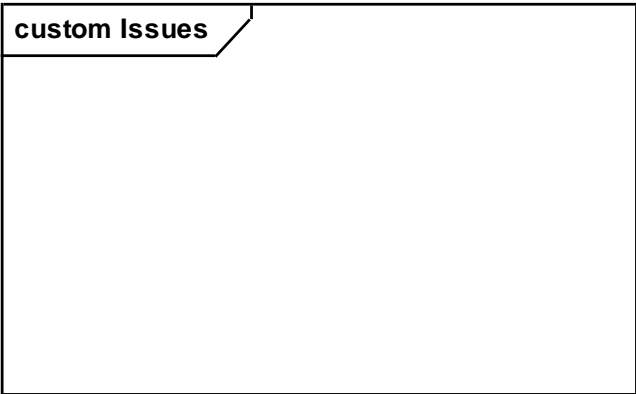


Diagram: Changes

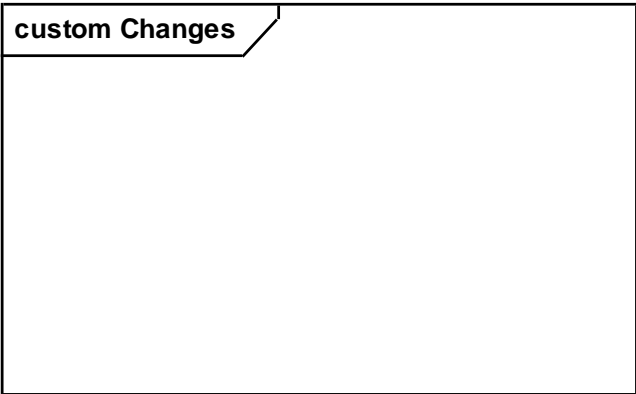


Diagram: Resources

analysis Resources

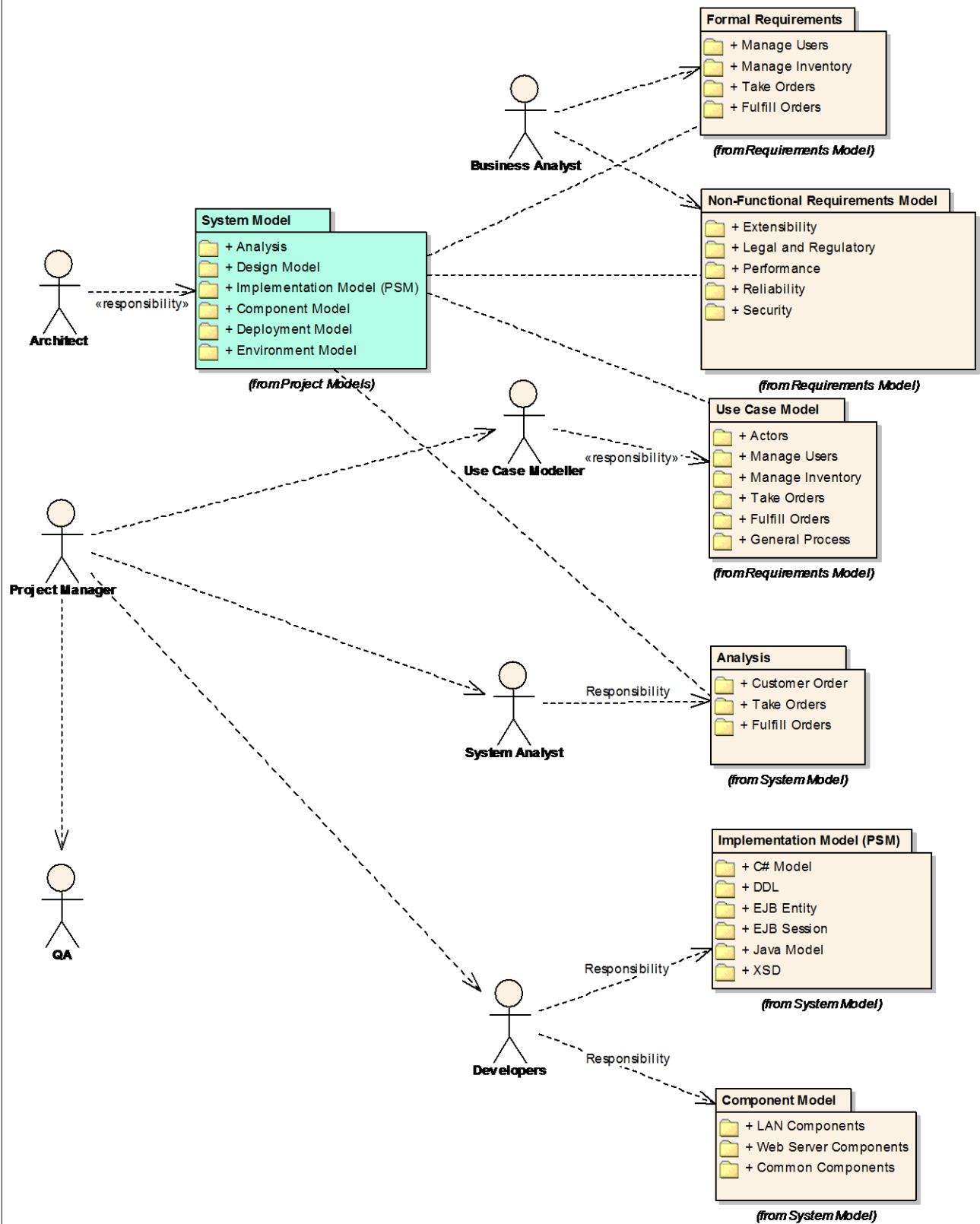


Diagram: Resources Overview

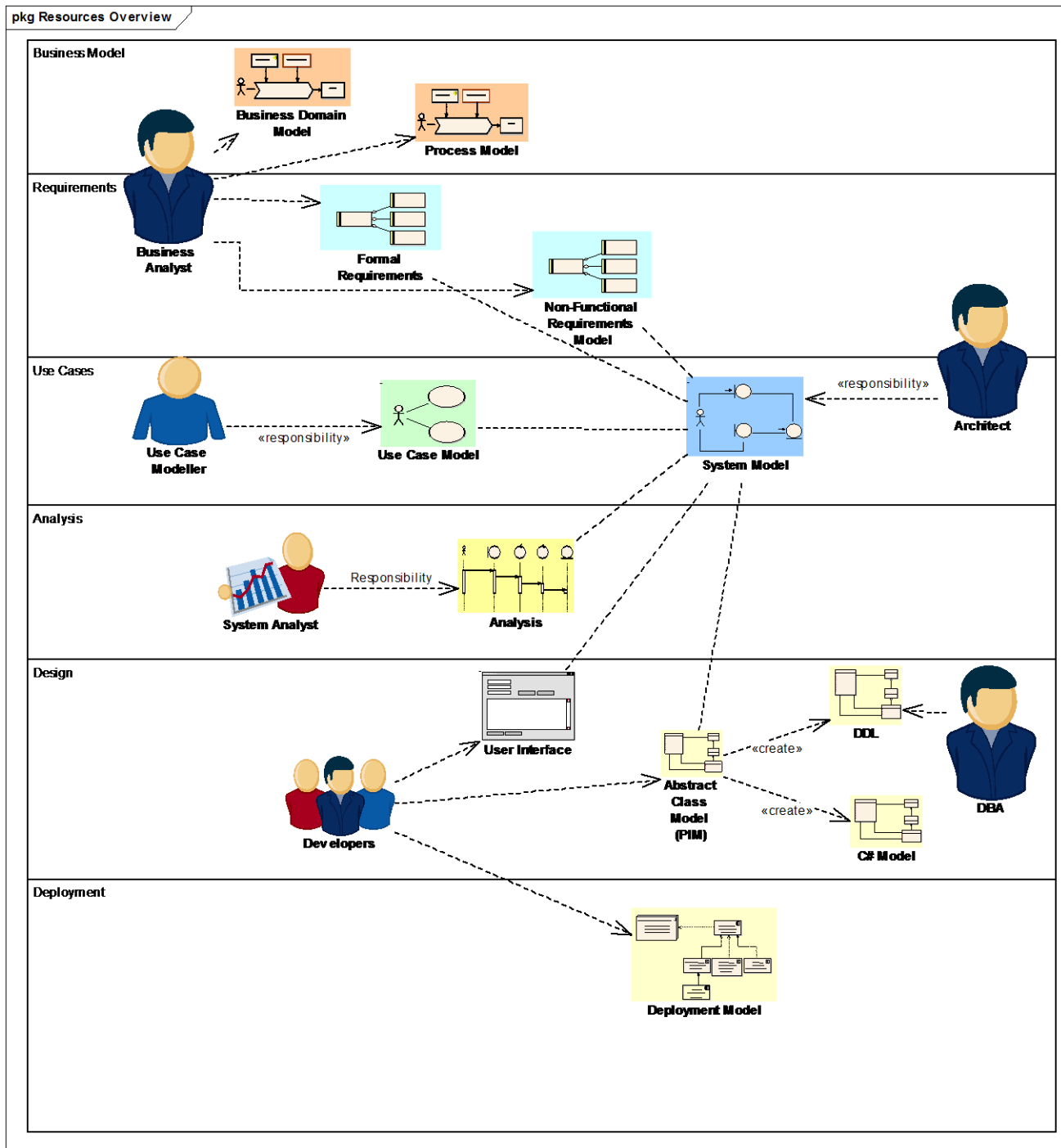


Diagram: Resource Allocation

class Resource Allocation

Resource Allocation

To view the Resource Allocation - select from the main menu: View | Project Management - or use Ctrl+Shift+7. Selecting any of the elements below will then display, in the Project Management window, the resources that have been allocated.

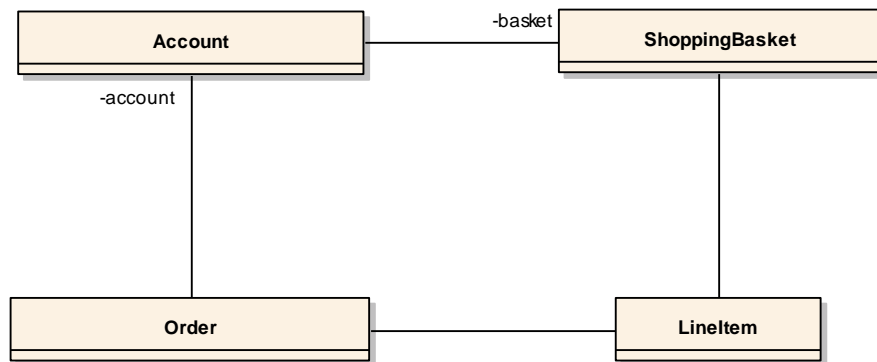


Diagram: Metrics and Estimation

class Metrics and Estimation

Metrics and Estimation

EA supports the definition of Use Case Metrics for elements defined in the Use Case model. For a set of Use Cases with Metrics set - see the following diagram:



Manage Users

To set up the Use Case Metrics and Estimation open the properties for a Use Case element and set value for: Complexity [].

For defining the technical and environmental factors -TCF and ECF open from the main menu: Configuration | Metrics and Estimation Types.

For more details on creating Use Case Metrics see:



<http://www.sparxsystems.com.au/UCMetrics.htm>

For reports on estimating the project size use the option on the main menu:
- Project | Use Case Metrics.

EA also supports the definition of Effort and Metrics for elements defined in the model. To access this use the main menu option: View | Project Management (or Ctrl+Shift+7).

Diagram: Risk

object Risk

Risk

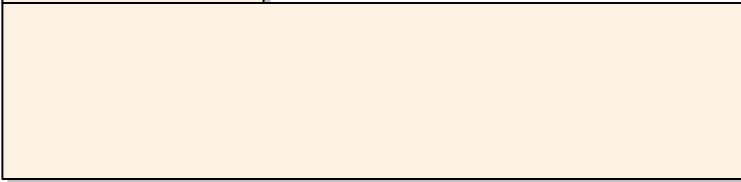
Below are the risks defined in the model. These are grouped in the packages related to the typical risk areas. The elements contained in these are linked elements defined in the source packages.

To view the risk definitions- from the main Menu select:

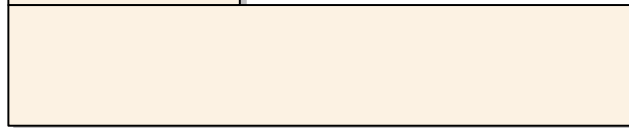
- View | Project Management (Ctrl+Shift+7).
- then select the Risk tab on the base of the window.

- Open a package below (double-click on it) then select an Element from the diagram to view its risk definition.

Uncertain Requirements



Effort Estimate Risks



Feasibility of Design



Diagram: Uncertain Requirements

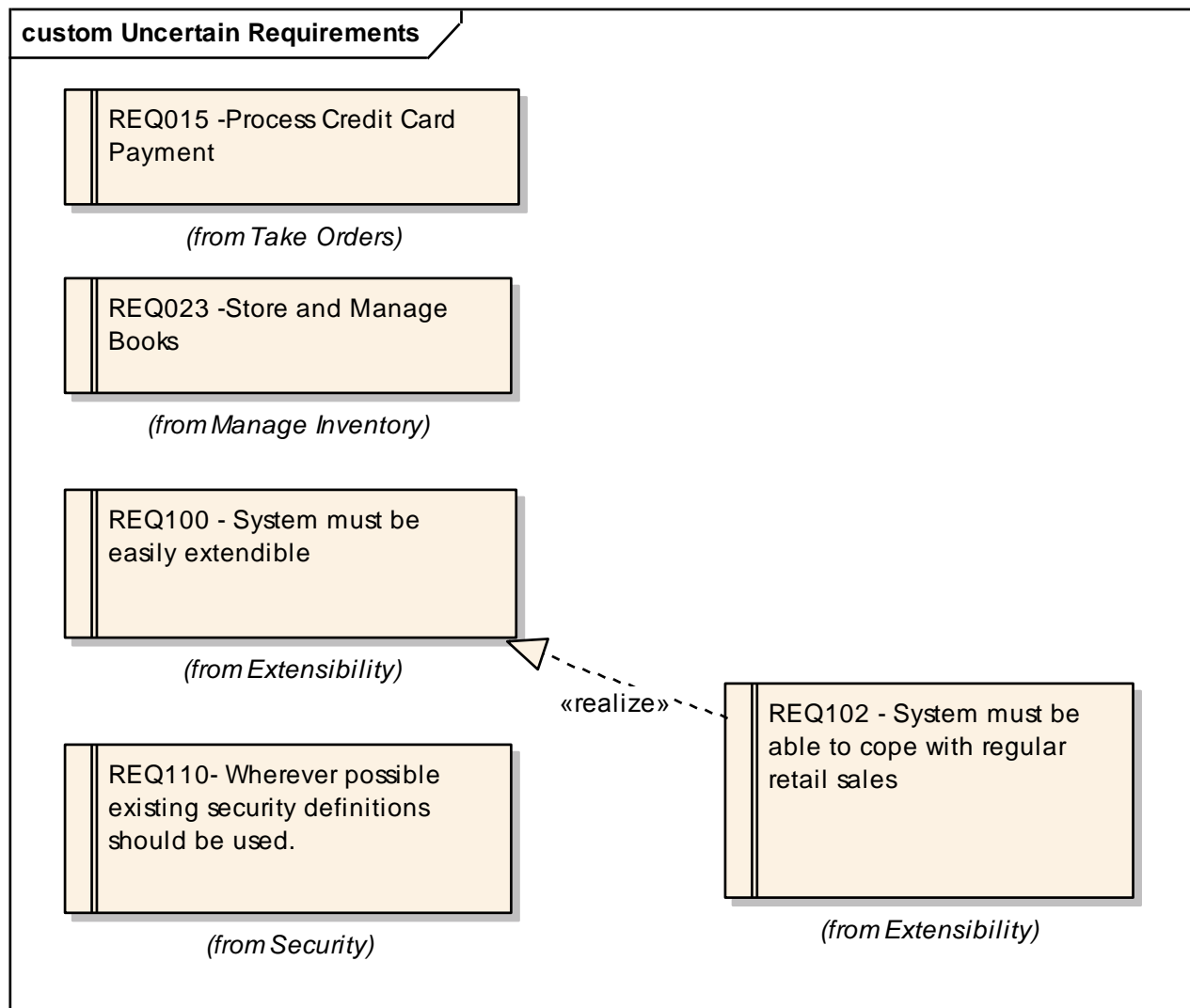


Diagram: Effort Estimate Risks

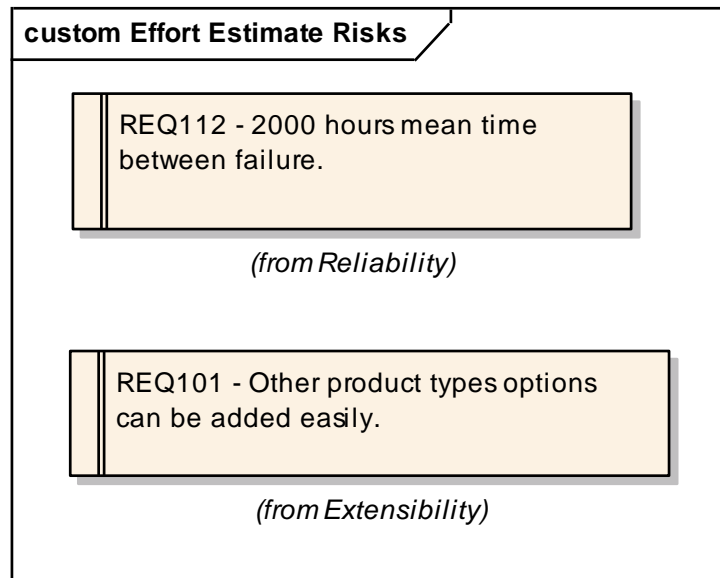


Diagram: Feasabilty of Design

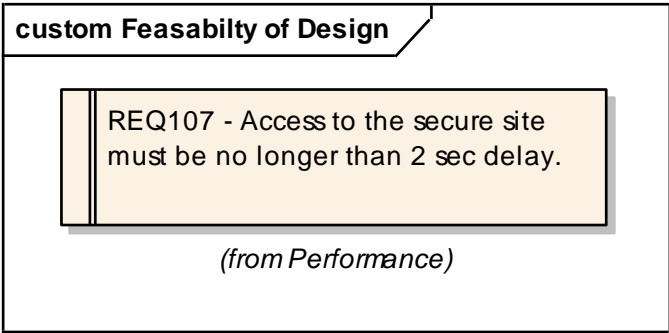


Diagram: Custom Diagrams

class Custom Diagrams

Other Diagrams Types

Enterprise Architect supports numerous variations on using the standard UML 2.0 diagrams type. Some of the key formats are shown below.



Requirements



Testing - QA



Business Modeling



Data Modeling



User Interface Example



Using Diagrams to Create Different Views



Diagrams Using Images



Traceability

Diagram: Diagram using Images

class Diagram using Images

Using Images on Elements

EA allows standard UML elements to be displayed with alternate images. This can be useful for communicating concepts to clients or non-technical staff. Double click the hyperlinks below for example diagrams using alternate images.



[Resource Overview](#)



[Actors](#)



[Deployment Model](#)



[Stakeholders](#)

Diagram: Model Overview

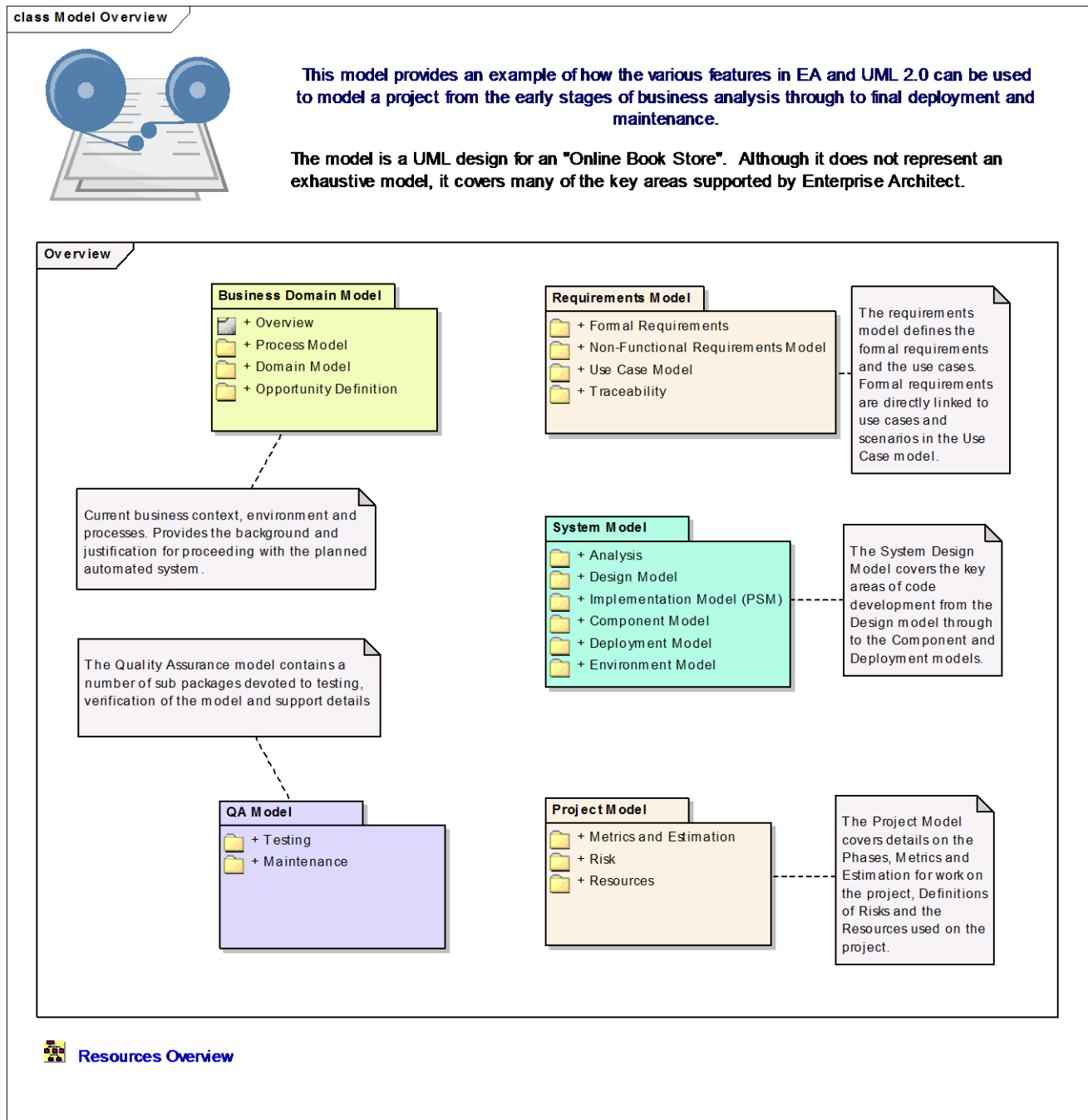


Diagram: UML 2.0 Diagrams

class UML 2.0 Diagrams

UML 2.0 Diagram Types

Below are some examples of the UML 2.0 diagrams types supported in Enterprise Architect. This section gives examples of all 13 diagram types supported.

Structural Diagrams



Package



Class



Object



Composite Structure



Component



Deployment



Custom

Behavioral Diagrams



Use Case



Analysis



Activity



State



Communication



Sequence



Timing



Interaction

Diagram: Welcome

[act Welcome](#)

Enterprise Architect

[Overview](#)
[Business Domain Model](#)
[Formal Requirements](#)
[Use Case Model](#)
[User Interface](#)
[Class Model \(PIM\)](#)

[Implementation Model \(PSM\)](#)
[Data Model](#)
[Deployment Model](#)
[QA Model](#)
[Resources](#)

Internet Links

- [Spax Systems Web site](#)
- [UML Tutorial](#)
- [Resources](#)
- [Purchase EA](#)

Overviews

- [UML 2.0 Diagrams](#)
- [Diagram Variations](#)
- [Diagrams using Images](#)
- [Generating Documents](#)

Help Topics

- [Using Enterprise Architect](#)
- [Modeling with UML](#)
- [Code Engineering](#)
- [Data Modeling](#)
- [Creating Documents](#)

The guide can be used to explore the example model. The links in the top section cover the key areas addressed in the model. The links under Overviews give a summary of key features using diagrams as well as reporting using the document generation.

Diagram: Activity Diagrams

act Activity Diagrams

Activity Diagrams

Below are some Activity Diagrams that use a number of the new features added to activity diagrams in UML 2.0.



Standard Activity Diagram



Activity & Subactivity Diagram



Partition Diagram



InterruptibleActivityRegion

Diagram: Class diagrams

class Class diagrams

Class Diagrams

Below are some examples of Class diagrams used in this model.



[C# Model](#)



[C# Model - No Attributes](#)



[EJB Entity](#)



[Java Model](#)

Note: The above class diagrams were generated using the MDA transform from the Abstract Class Model. See:



[Abstract Class Model \(PIM\) : Class Model](#)



[MDA Transforms](#)

Diagram: Communication diagrams

sd Communication diagrams

Communication Diagrams

Below are some examples of Communication Diagrams as used in the Use Case model.



Create Account



Delete User



View Account details



View History



Login



Close Account

Diagram: Component Diagrams

cmp Component Diagrams

Component Diagrams

Below are some examples of Component Diagrams used in this model.



LAN Components



Web Server Components

Diagram: Custom Diagrams

custom Custom Diagrams

Custom Diagrams

Below are some examples of Custom Diagrams used in this model.



Requirements: Manage Users



Traceability : ManageUsers



Traceability : Manage Inventory

Diagram: Diagram Settings

pkg Diagram Settings

Diagram Settings

Diagrams can be created with a number of different settings for displaying the same information. Below are some examples of this.



Class Diagram with Attributes and Operations Viewable



Class diagram with the Attributes and operations not shown.



Class Diagram only showing the Testing .

To alter these settings bring up the diagram properties using the option from the main menu:

- Diagram | Properties

- or right-click on the diagram and from the context menu select: Properties.

The Diagram Properties window includes - Appearance Options (see below). This displays options for showing/hiding the properties viewed on the current diagram.

Appearance Options			
<input checked="" type="checkbox"/> Use Stereotype Icons	<input checked="" type="checkbox"/> Highlight Foreign Objects	<input type="checkbox"/> Hide Attributes	<div>Visible Class Members</div> <div><input checked="" type="checkbox"/> Public</div> <div><input checked="" type="checkbox"/> Protected</div> <div><input checked="" type="checkbox"/> Private</div> <div><input checked="" type="checkbox"/> Package</div>
<input checked="" type="checkbox"/> Show Page Border	<input checked="" type="checkbox"/> Show Package Contents	<input type="checkbox"/> Hide Operations	
<input checked="" type="checkbox"/> Show Table Owner	<input type="checkbox"/> Show Details on Diagram	<input type="checkbox"/> Show Tags	
<input type="checkbox"/> Use Alias if Available	<input type="checkbox"/> Show Sequence Notes	<input type="checkbox"/> Show Requirements	
<input type="checkbox"/> Hide Property Methods	<input type="checkbox"/> Hide Additional Parents	<input type="checkbox"/> Show Constraints	Show Parameter Detail: Type Only ▾
<input type="checkbox"/> Hide Collaboration Numbers	<input type="checkbox"/> Hide Relationships	<input type="checkbox"/> Show Testing	
<input type="checkbox"/> Hide Element Stereotype	<input type="checkbox"/> Hide Stereotype on Features	<input type="checkbox"/> Show Maintenance	
<input type="checkbox"/> Hide Visibility Indicators			
<input type="checkbox"/> Explicit Assoc. Navigability			
Connector Notation			UML 2.0 ▾

Diagram: Interaction diagrams

sd Interaction diagrams

Interaction Overview Diagrams

The following are some examples of Interaction Overview Diagrams.



[View Orders](#)



[View Account details](#)



[Interactions](#)



[Sale](#)

Diagram: Package Diagrams

pkg Package Diagrams

Package Diagrams

Below are some examples of Package diagrams used in this model:



Business Domain Model



Formal Requirements



EJB Entity

Diagram: Sequence Diagrams

sd Sequence Diagrams

Sequence Diagrams

Below are some examples of Sequence Diagrams:



Create Account



Delete User



View Account details



View History



View Open Orders

Diagram: State Diagrams

pkg State Diagrams

State Chart Diagrams

Below are some examples of Statechart Diagrams used in the model:



Login



Manage Titles State



Search

Diagram: Timing

sd Timing

Timing Diagrams

Enterprise Architect supports the UML 2.0 Timing Diagram. There are two lifelines defined in UML 2.0 - the State Lifeline and the Value Lifeline. The following are examples of these:



State Lifeline



Value Lifeline

These elements can be combined within a single diagram:



State and Value combined

Diagram: Use Case diagrams

uc Use Case diagrams

Use Case Diagrams

Below are a number of examples of Use Case diagrams:



Manage Users



Manage Inventory



Take Orders



Fulfill Orders

Diagram: State Lifeline

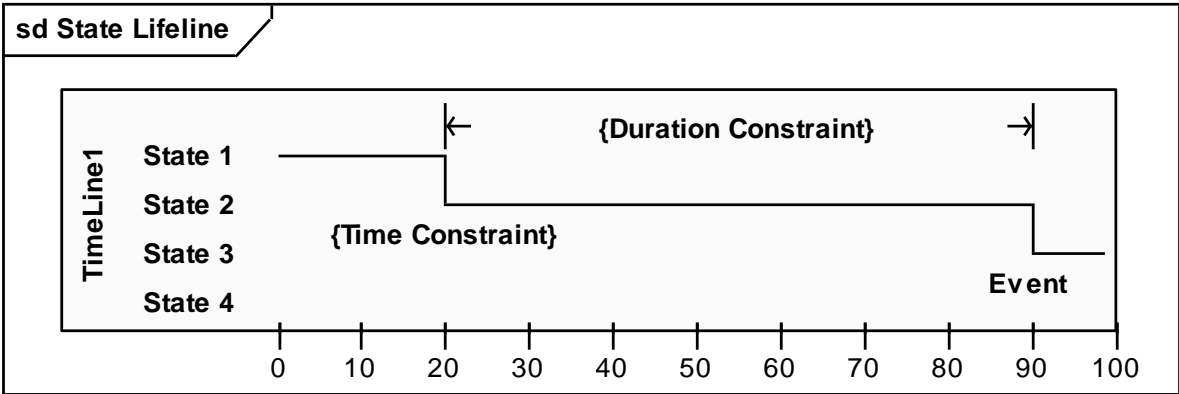


Diagram: Timing Diagram

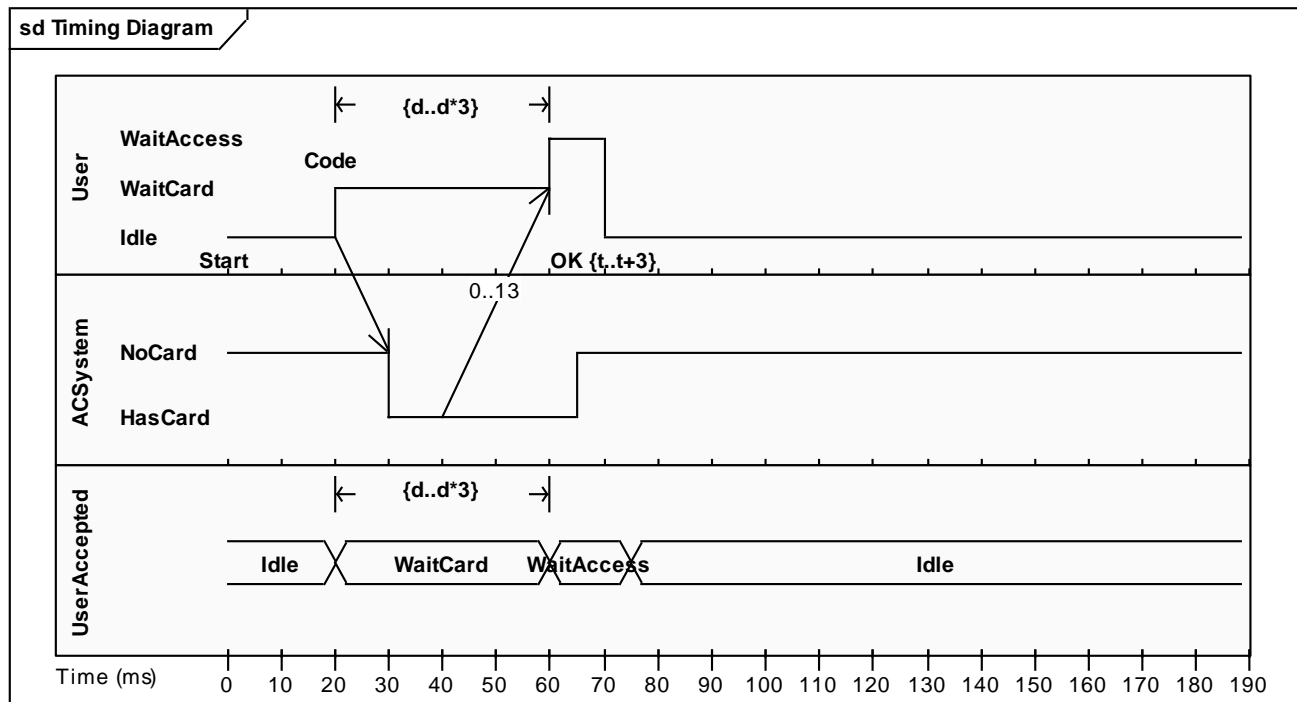


Diagram: Value Lifeline

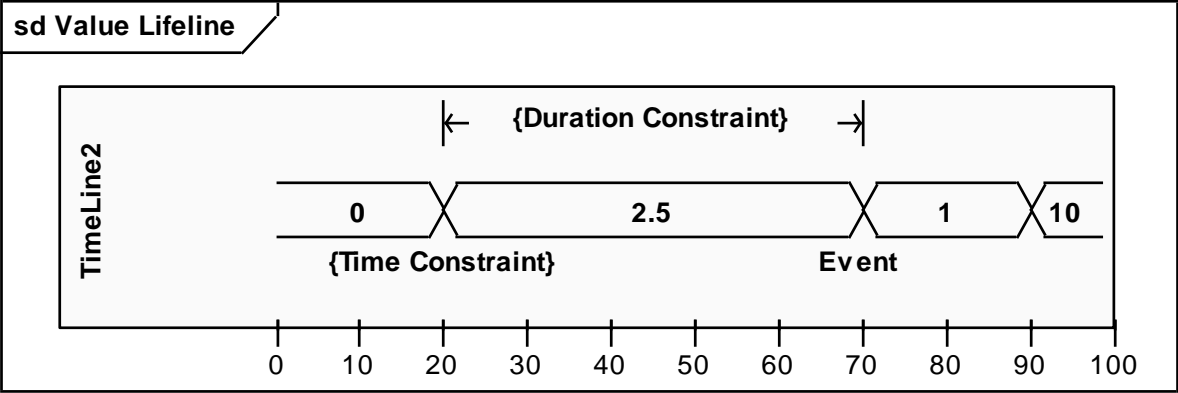


Diagram: CancelSale



Diagram: Checkout

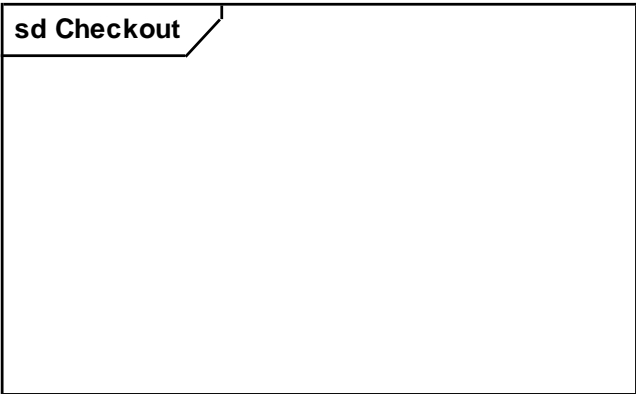


Diagram: CreateRecord

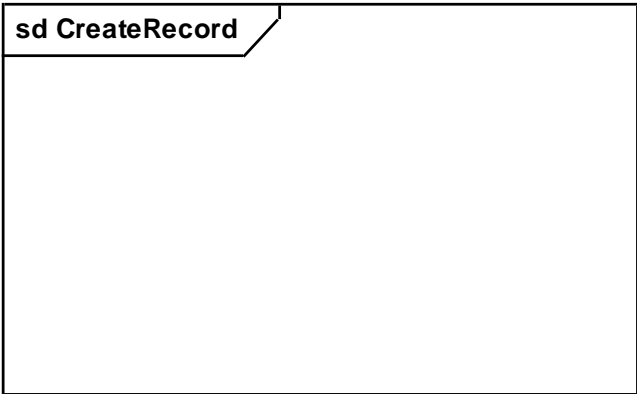


Diagram: Sale

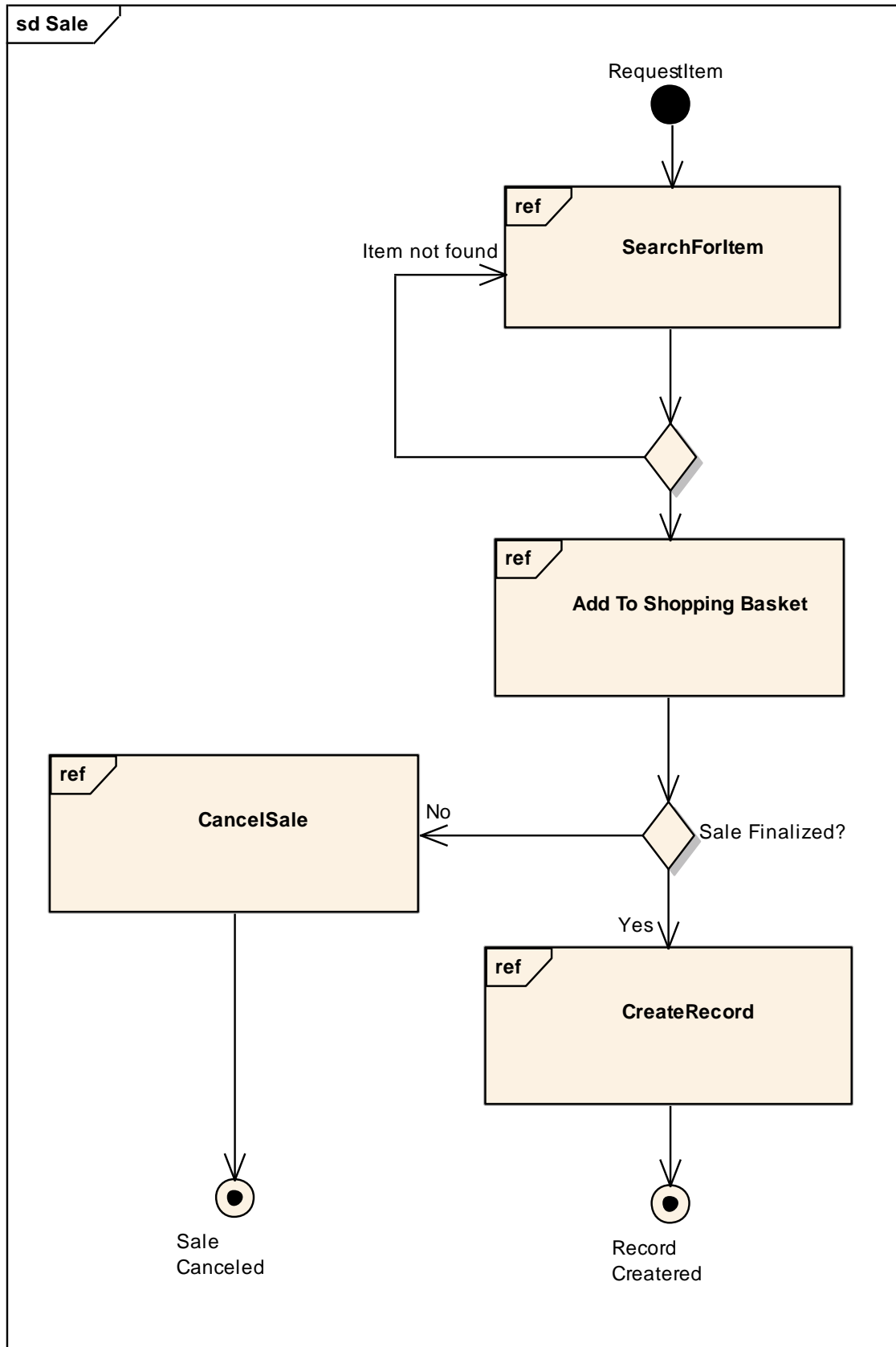


Diagram: Fragment

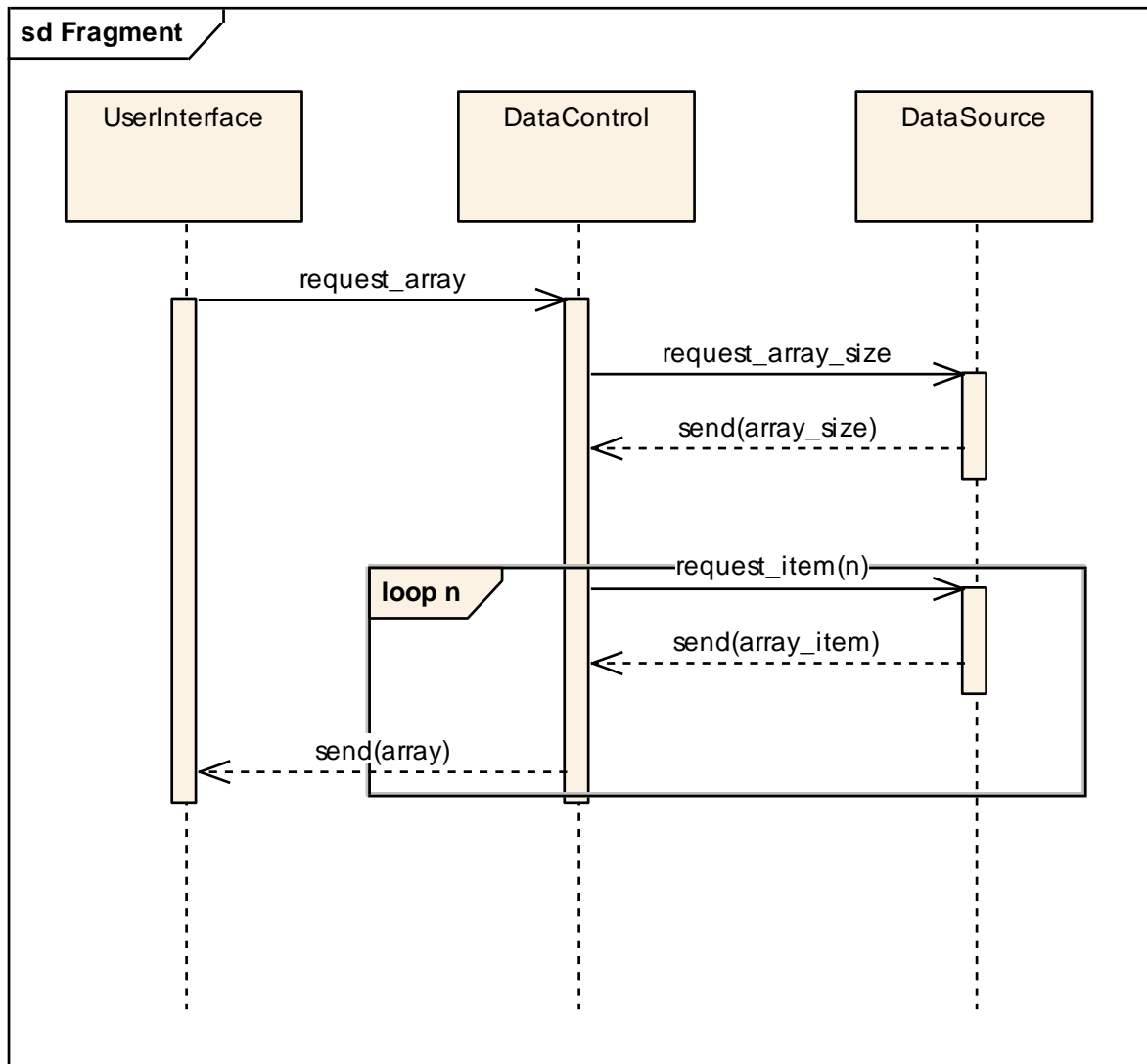


Diagram: Composite Structure

composite structure Composite Structure

Composite Structure Diagrams

Below are two examples of composite structure diagrams, one giving an example of the CollaborationOccurrence element, the other giving an example of Property.



Composite Structure - CollaborationOccurrence



Composite Structure - Properties

Diagram: Composite Structure-Properties

composite structure Composite Structure-Properties

This composite structure diagram represents properties of the Stock class two ways. The first encompasses the properties within the class element, whereas the second uses connectors to reflect the properties.

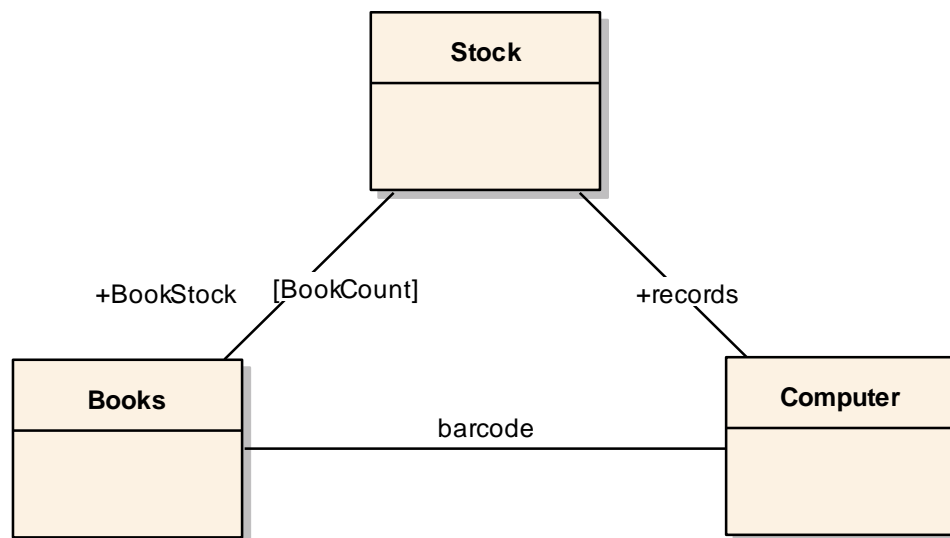
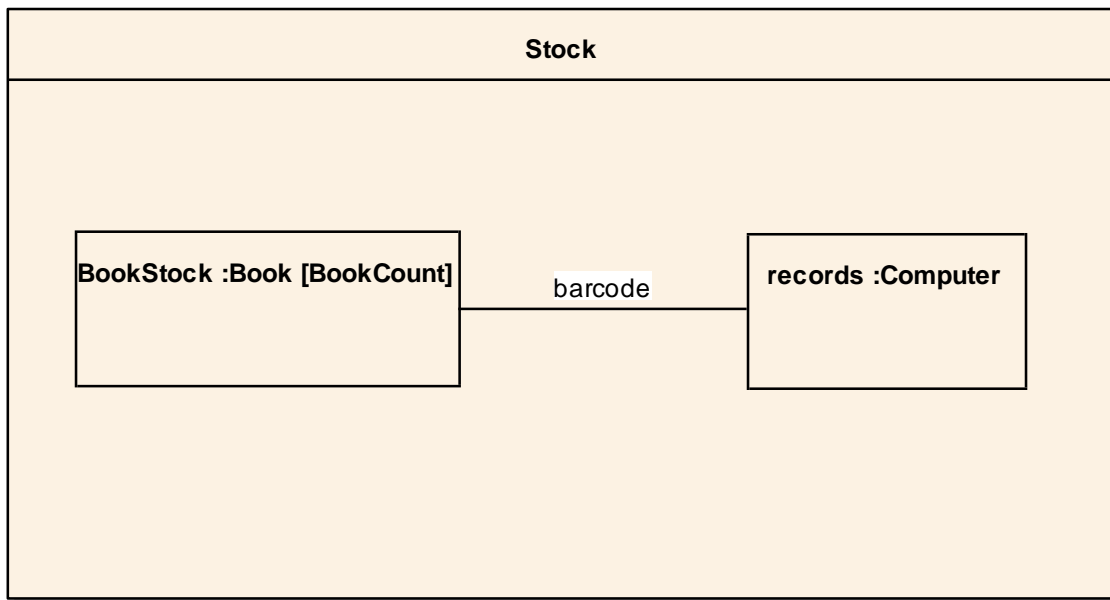
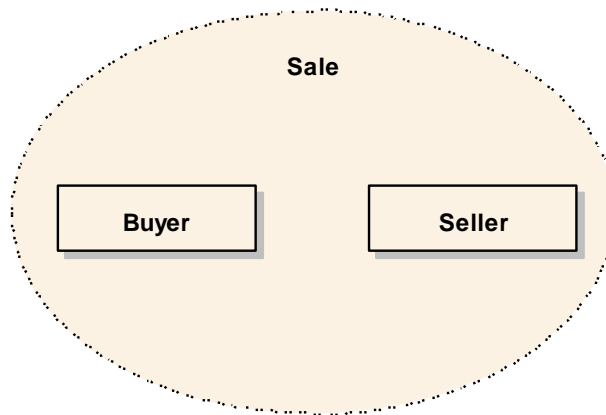


Diagram: Sale

composite structure Sale

This diagram defines a collaboration that is used twice in the BrokeredSales diagram - the two sales being Wholesale and Retail.



This diagram is taken from UML 2.0 Superstructure Specification - page 162

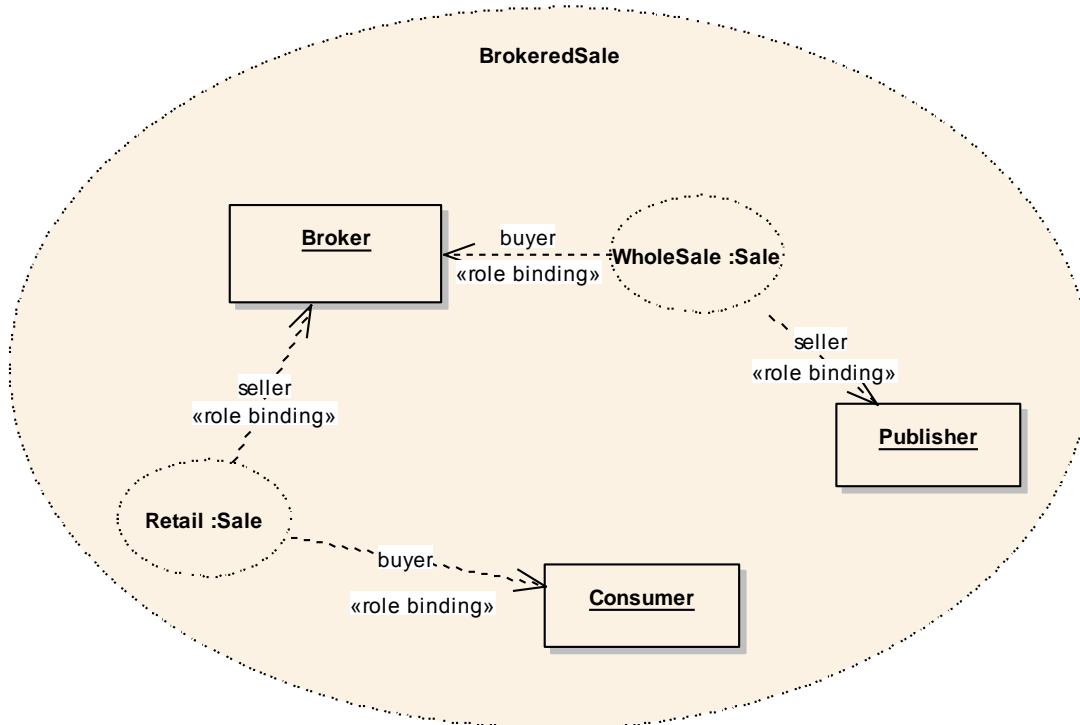
Diagram: brokered

composite structure brokered

This diagram is an example of the use of the CollaborationOccurrence. This collaboration has two occurrences of the Sale Collaboration - Wholesale and Retail. See the attached link for the original definition of the Sale Collaboration.



Sale Collaboration



This diagram is a variation on the example defined in the UML 2.0 Superstructure Specification - page 161.

Diagram: Object

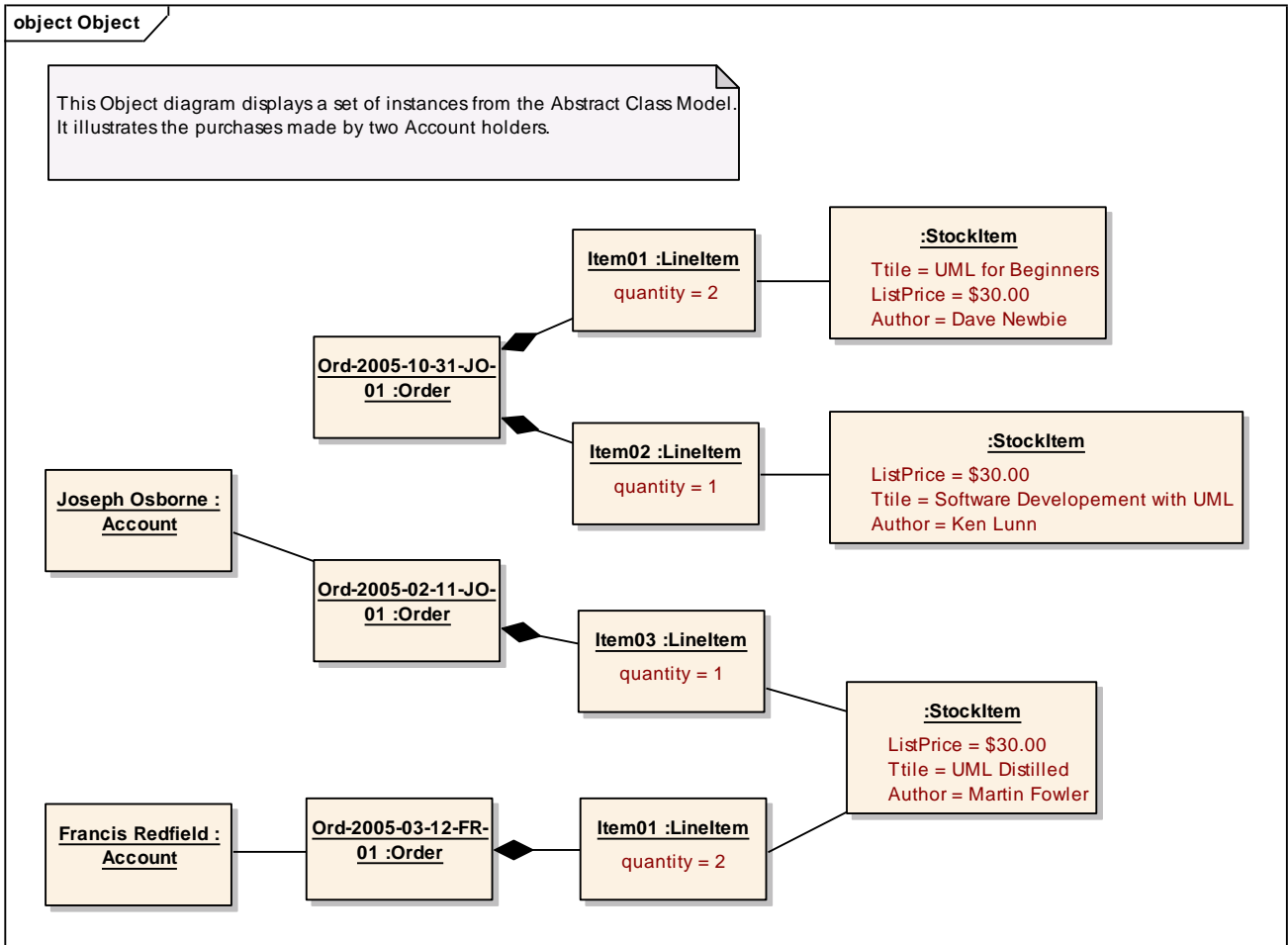


Diagram: Documentation

custom Documentation

Guide to Using EA's Documentation

Enterprise Architect supports the creation of professional and completely customized documentation directly from the application using both RTF report generation and HTML generation.

There are a number of different methods for running reports. Details for these are as follows:

Standard RTF reports:

Standard RTF reports are user definable and generate a report on a package tree:



[Details and Examples of Standard RTF Documents](#)

HTML Reports

Enterprise Architect supports the generation of HTML reports based on a package tree:



[Details on HTML Reports](#)

Virtual Documents

Virtual Documents allow packages, unrelated in the project tree, to be grouped into a single RTF report.



[Details and Examples of Virtual Documents](#)

Word Master Documents

EA supports book-marking Elements and Packages in reports. These can be used with Word to group parts of different reports into a master document.



[Details on Using Word Master Documents](#)

RTF Reports Using Search Results

The results of a search can be used to create an RTF document.



[Details on Creating Reports from Search Results](#)

Diagram: HTML reports

custom HTML reports

Generating HTML Reports

EA has the ability to generate a HTML report of an entire model or a single branch of the model. The HTML report provides a convenient frames-based structure navigator and content explorer with much of the output hyperlinked to related sections within the model.

The HTML report produced is compatible with any standard web server - either on Unix or Windows allowing the output to be placed within the context of your web server.



HTML Reports

The HTML generation is based on internal templates and generated Java script. Alterations can be made to the basic template. For more information on this see:



Web Style Templates

Diagram: Reports from Search Results

custom Reports from Search Results

Reports from Search Results

Enterprise Architect now supports the creation of RTF documents based on the results of a Search (i.e. Ctrl-F). The search facility can be used to create a group of similar elements. A selection from the search results can then be used to generate the report.

Any of the RTF templates can then be used to generate a report from the search results.

For more information on this see - "Element Selection in the View" in:



[RTF Report on Search Results](#)

Diagram: Standard RTF

custom Standard RTF

Standard RTF Documents

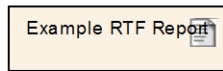
Enterprise Architects RTF document generation allows the user to create or edit a template that defines the structure of a report to be generated. Once the template is created, the report can be run on any package or tree of packages to produce an RTF document.

This generated document can be viewed internally or it can be viewed using the systems default word processor (i.e. the internal RTF editor or Word).

To run a report using an existing report template, select a package in the Project View and either:

- Press (F8) or
- From the main menu select Project | Documentation | Rich Text Format Report

To view a Basic Template document generated from this model - double-click on the Element below:



For more information on creating RTF reports see the following:



[Help on RTF Reports](#)



[Whitepaper on RTF Reports](#)

Diagram: Virtual Documents

custom Virtual Documents

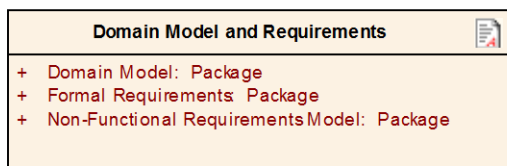
Virtual Documents

Enterprise Architect can generate a report of a group of packages that are not directly related in the Project Tree. These reports are called a "Virtual Document".

A virtual document is defined using a Class with the Stereotype set to "Model Document". With this stereotype set the Class shows a document icon.

To define what packages to report on, simply drag packages from the Project View on to the "Model Document" class. The class can then be selected and used to generate a document from any of the RTF templates.

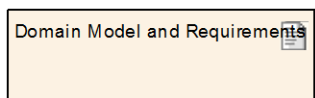
Below is an example of a Model Document class:



To run a report on this package:

- Select the package and press (F8)
- Select a template to generate from (i.e. Basic Template)
- Select [Generate]
- When complete:
 - Select [View] to open the generated document

Below is an example of the document generated using the RTF "Basic Template" - double-click to open it



For more information on setting up "Virtual Documents" see the following section in the help file:



[Help - Virtual Documents](#)

Diagram: Word Master Documents

custom Word Master Documents

Word Master Documents

When Working with RTF documents, MS Word can be used to create a master document that can consist of parts of one document or parts from a series of RTF documents. When an RTF report is generated, each Package and Element is grouped as a book-marked item. A Word document can be used to group whole documents or book-marked sections of any of the RTF reports generated.

For more information on setting up Master Documents using Word see:



[Using Word - Bookmarks](#)