

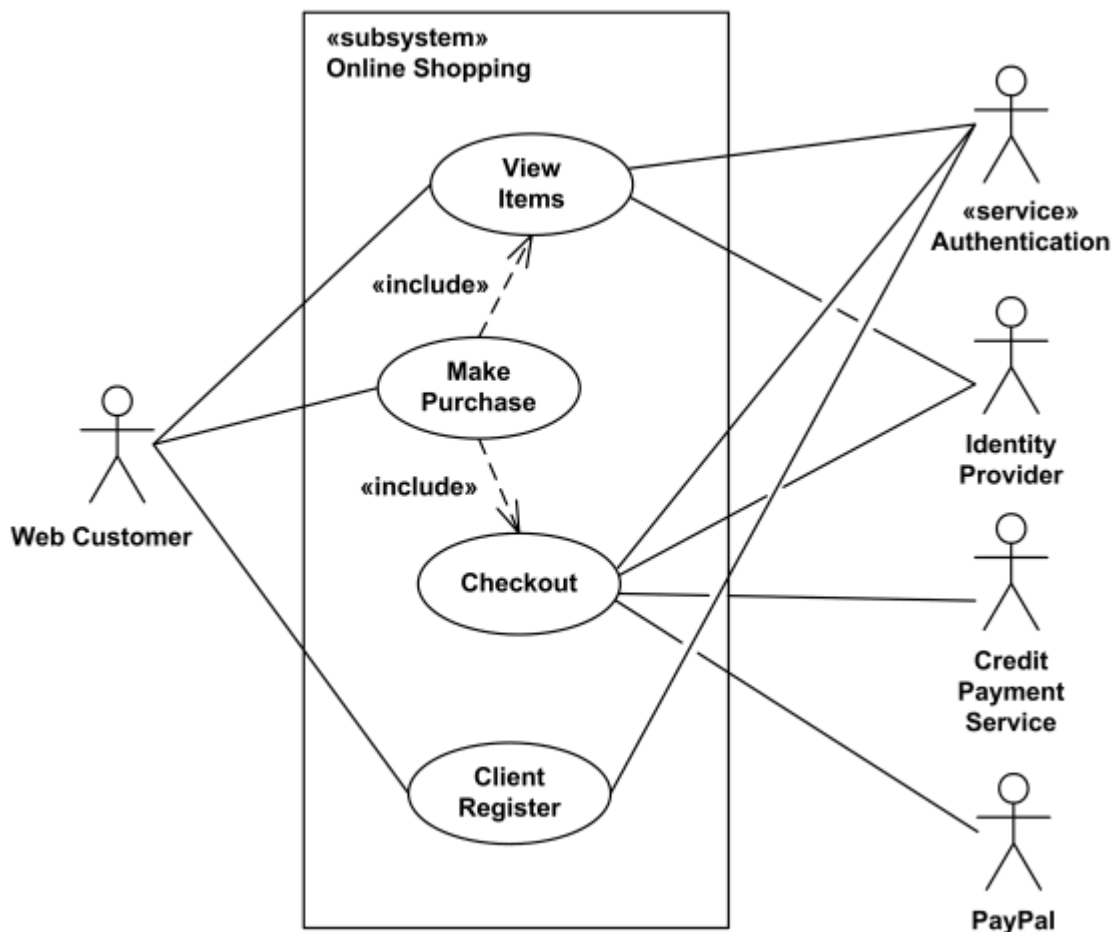
Online Shopping UML Examples

Here we provide several UML diagrams of different types, all part of a simple fabricated **Online Shopping** model.

Use Cases

Web Customer [actor](#) uses some web site to make purchases online. Top level [use cases](#) are **View Items**, **Make Purchase** and **Client Register**. View Items use case could be used by customer as top level use case if customer only wants to find and see some products. This use case could also be used as a part of Make Purchase use case. Client Register use case allows customer to register on the web site, for example to get some coupons or be invited to private sales. Note, that **Checkout** use case is [included use case](#) not available by itself - checkout is part of making purchase.

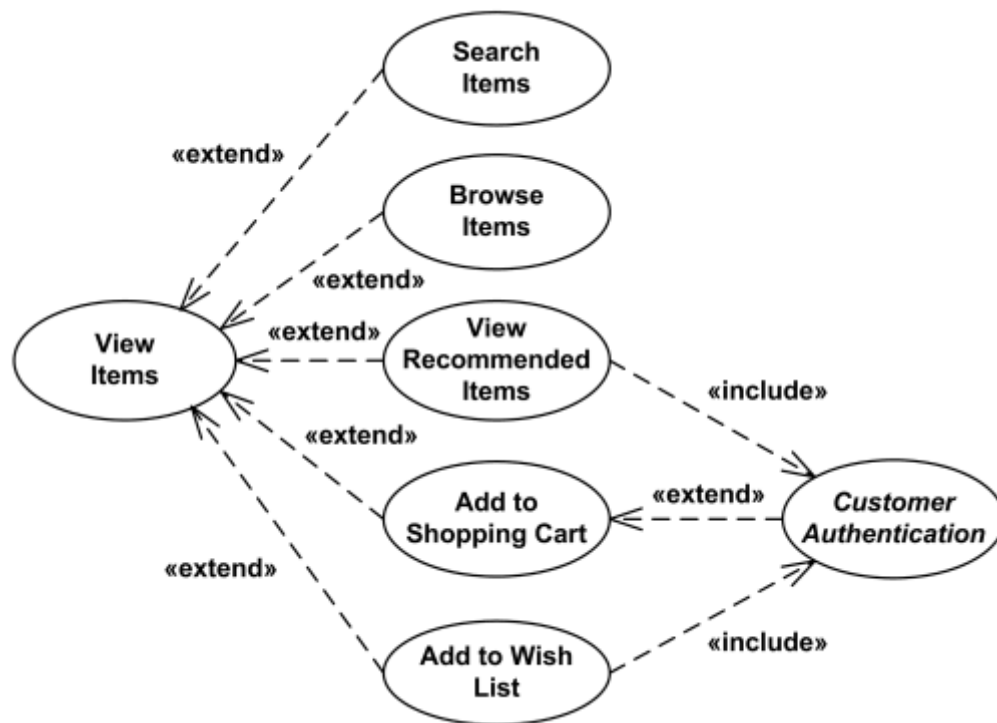
Except for the **Web Customer** actor there are several other actors which will be described below with detailed use cases.



Online Shopping - Top Level Use Cases

View Items use case is [extended](#) by several optional use cases - customer may search for items, browse catalog, view items recommended for him/her, add items to shopping cart or wish list. All these use cases are extending use cases because they provide some optional functions allowing customer to find item.

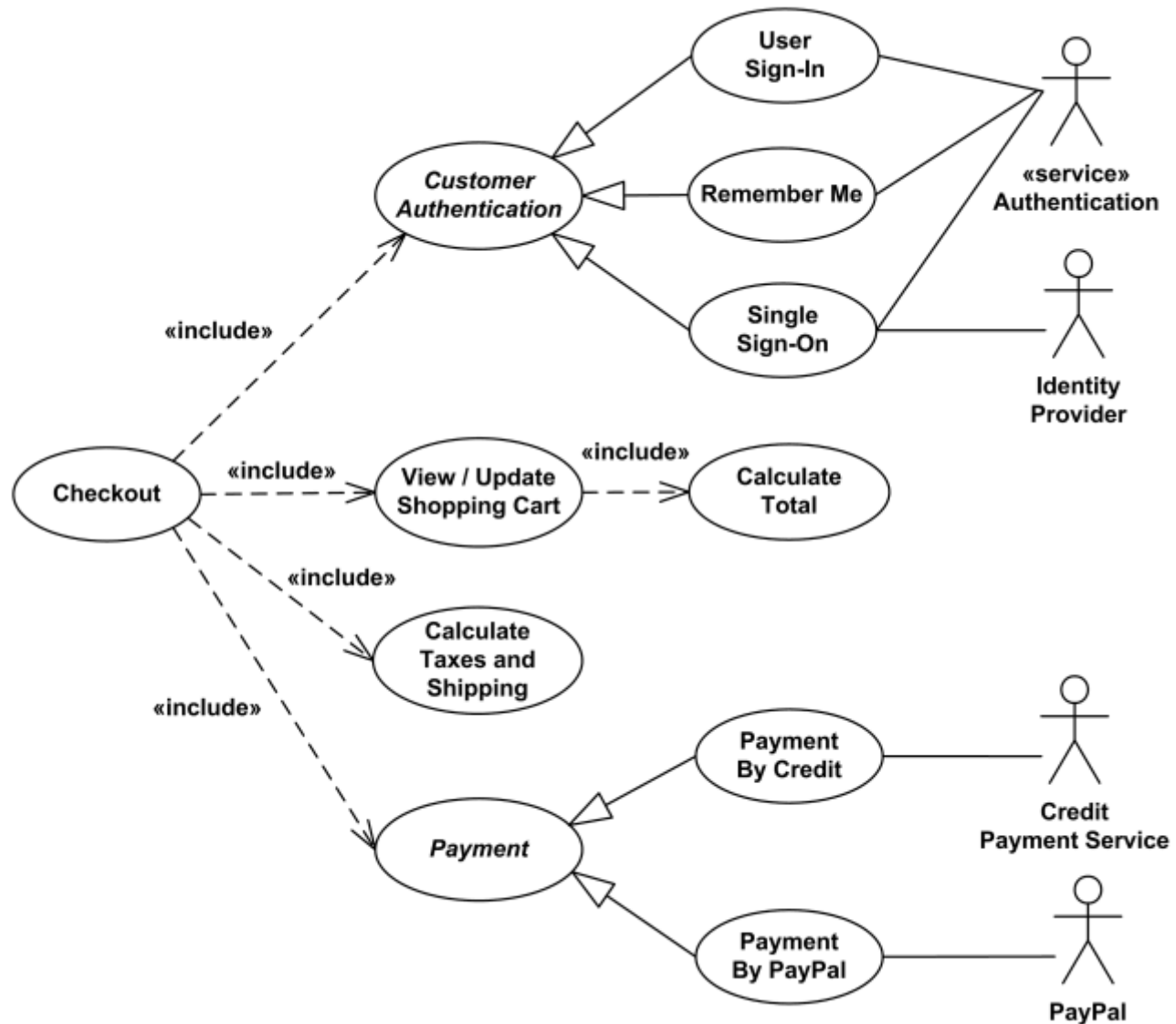
Customer Authentication use case is [included](#) in **View Recommended Items** and **Add to Wish List** because both require customer to be authenticated. At the same time, item could be added to the shopping cart without user authentication.



Online Shopping - View Items Use Case

Checkout use case includes several required uses cases. Web customer should be authenticated. It could be done through user login page, user authentication cookie ("Remember me") or Single Sign-On (SSO). Web site authentication service is used in all these use cases, while SSO also requires participation of external identity provider.

Checkout use case also includes **Payment** use case which could be done either by using credit card and external credit payment service or with PayPal.



Online Shopping - Checkout, Authentication and Payment Use Cases

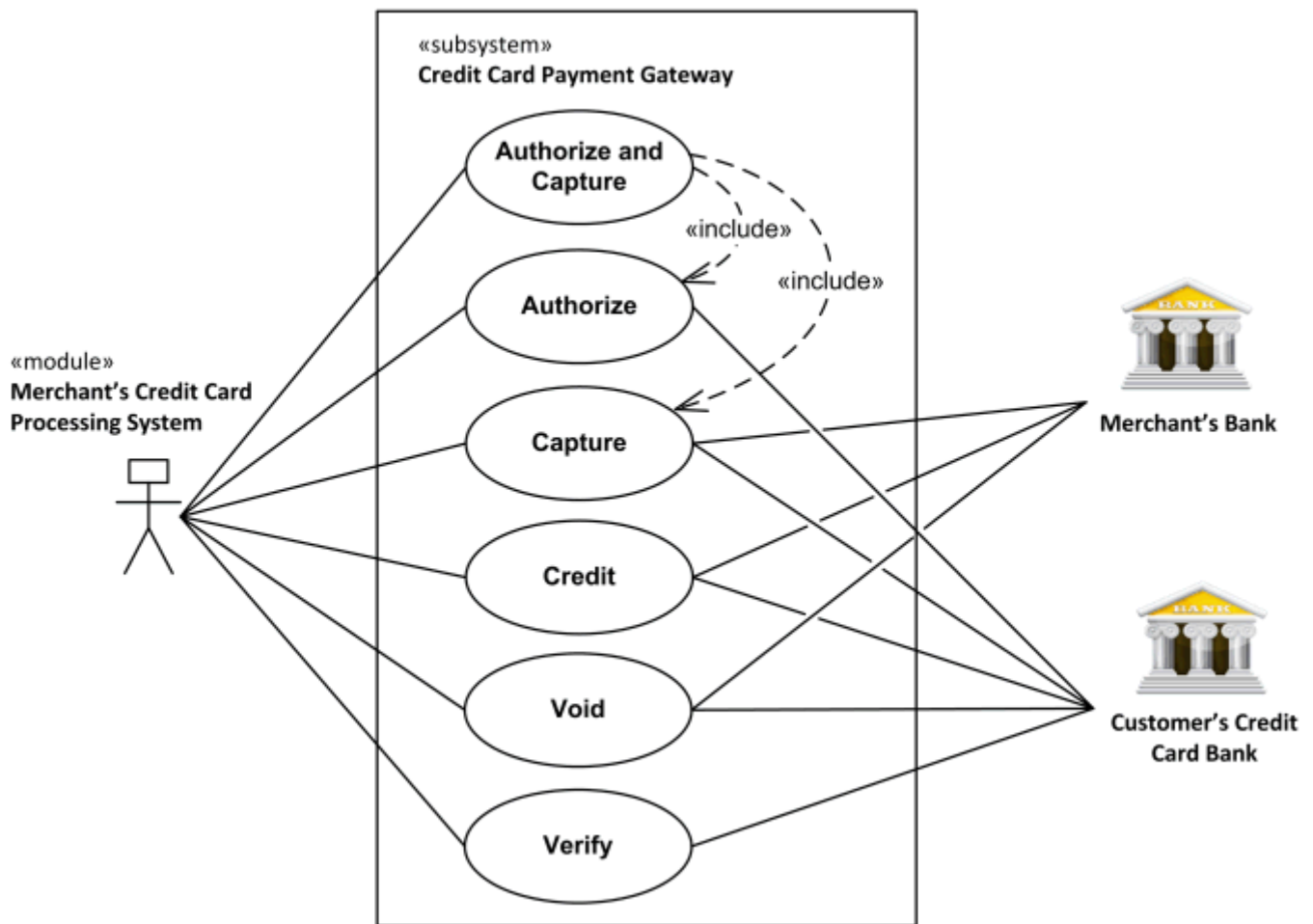
Credit Card Processing System Use Cases

In this use cases example, **Credit Card Processing System** (Credit Card Payment Gateway) is a [subject](#), i.e. system under design or consideration. Primary [actor](#) of the system is the **Merchant's Credit Card Processing System**. The merchant submits a credit card transaction request to the credit card payment gateway on behalf of a customer. Bank which issued customer's credit card is actor which could approve or reject the transaction. If transaction is approved, funds will be transferred to merchant's bank account.

Authorize and Capture use case is the most common type of credit card transaction. The requested amount of money should be first authorized by **Customer's Credit Card Bank**, and if approved, is further submitted for settlement. During the settlement funds approved for the credit card transaction are deposited into the **Merchant's Bank** account.

In some cases, only **authorization** is requested and the transaction will not be sent for settlement. In this case, usually if no further action is taken within some number of days, the authorization expires. Merchants can submit this request if they want to verify the availability of funds on the customer's credit card, if item is not currently in stock, or if merchant wants to review orders before shipping.

Capture (request to capture funds that were previously authorized) use case describes several scenarios when merchant needs to complete some previously authorized transaction - either submitted through the payment gateway or requested without using the system, e.g. using voice authorization.



Credit Card Processing System Use Cases.

Credit use case describes situations when customer should receive a refund for a transaction that was either successfully processed and settled through the system or for some transaction that was not originally submitted through the payment gateway.

Void use case describes cases when it is needed to cancel one or several related transactions that were not yet settled. If possible, the transactions will not be sent for settlement. If the Void transaction fails, the original transaction is likely already settled.

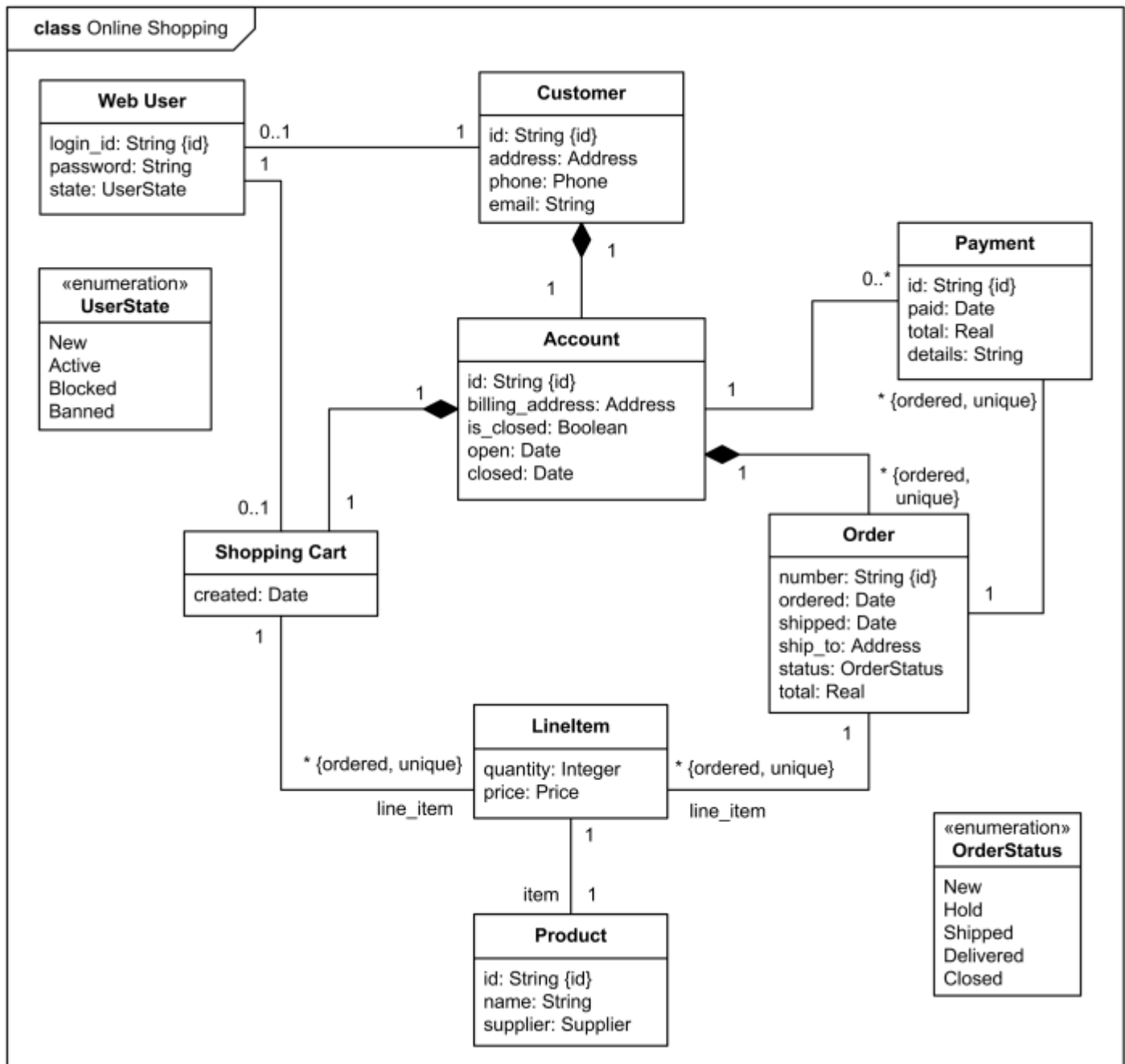
Verify use case describes zero or small amount verification transactions which could also include verification of some client's data such as address.

You can find excellent resources, documentation, white papers, guides, etc. related to the credit card processing at [Authorize.Net - Payment Gateway to Accept Online Payments.](#)

Online Shopping Domain

This diagram is an example of [class diagram](#) which shows some domain model for online shopping. Each **customer** could have some **web user** identity. Web user could be in several states and could be linked to one **shopping cart**.

Each customer has exactly one **account**. Account owns shopping cart and orders. Orders are sorted and unique. Each order is linked to none to several **payments**.

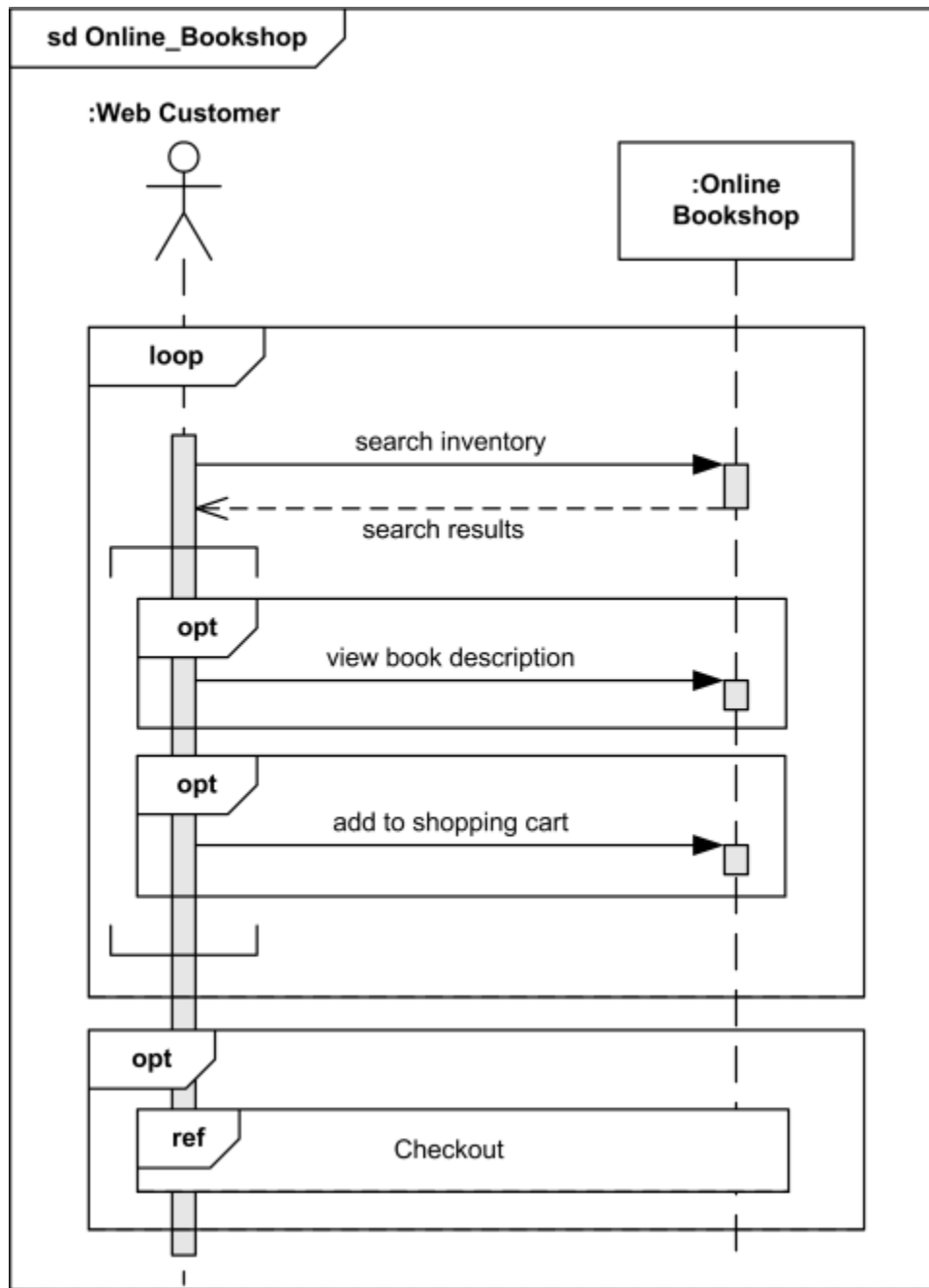


Class diagram example - Online Shopping Domain.

Each order has current order status. Both order and shopping cart have **line items** linked to specific product.

Sequence Diagram

An example of high level **sequence diagram** for **Online Bookshop**. Online customer can search book catalog, view description of a selected book, add book to shopping cart, do checkout.



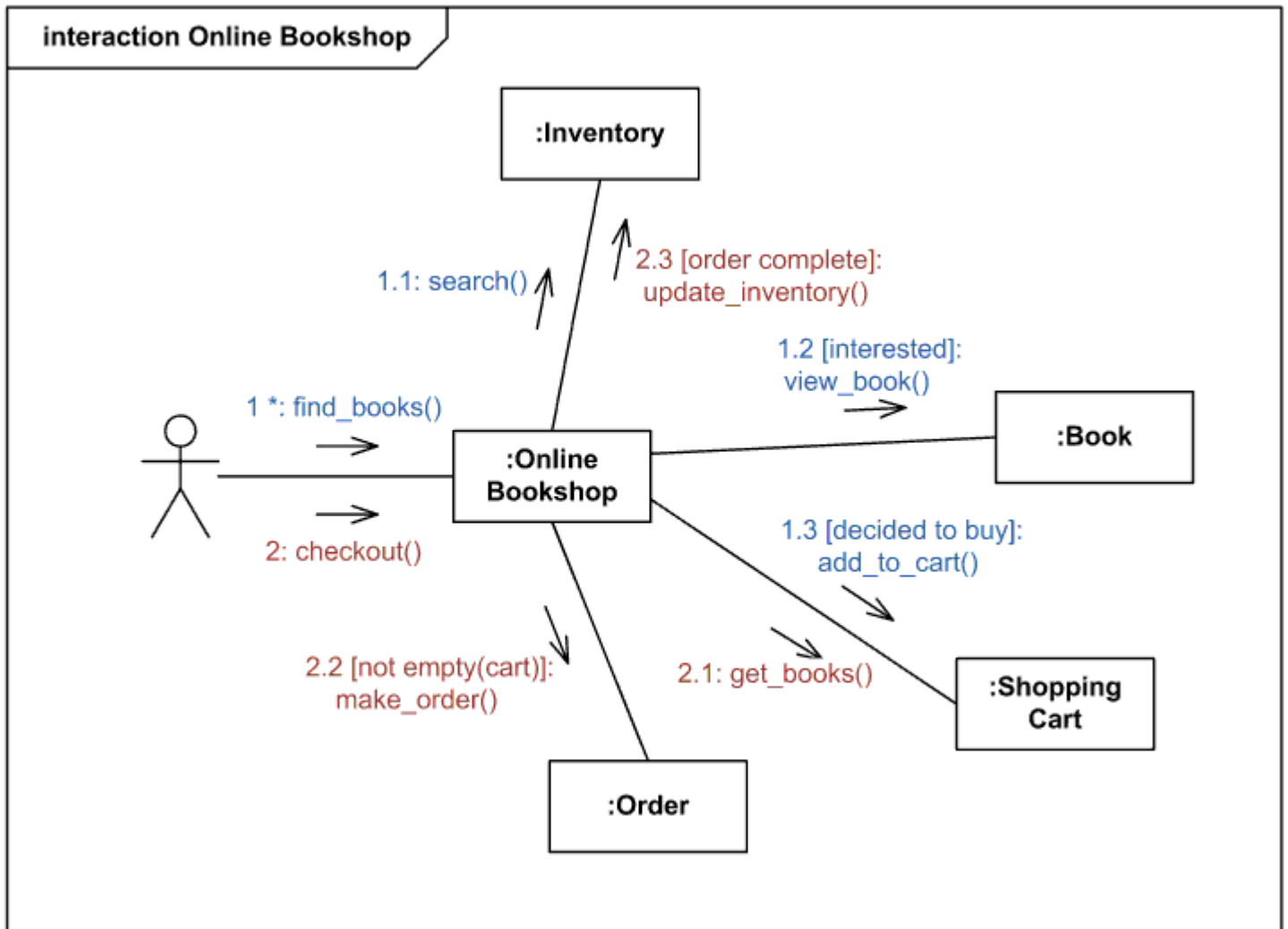
An example of sequence diagram for Online Bookshop.

Communication Diagram

An example of [communication diagram](#) for **Online Bookshop**. **Web customer** (depicted as [actor](#)) can search, view and buy books.

Communication starts with **1 *: find_books()** - iterative message which could be repeated some unspecified number of times. Client searches inventory of books, and if he/she is interested in some book, he/she can view description of the book (**1.2 [interested]: view_book()**). If client decides to buy, he/she can add book to the shopping cart - **1.3 [decided to buy]: add_to_cart()**.

Checkout includes getting list of books from shopping cart, creating order, and updating inventory, if order was completed.

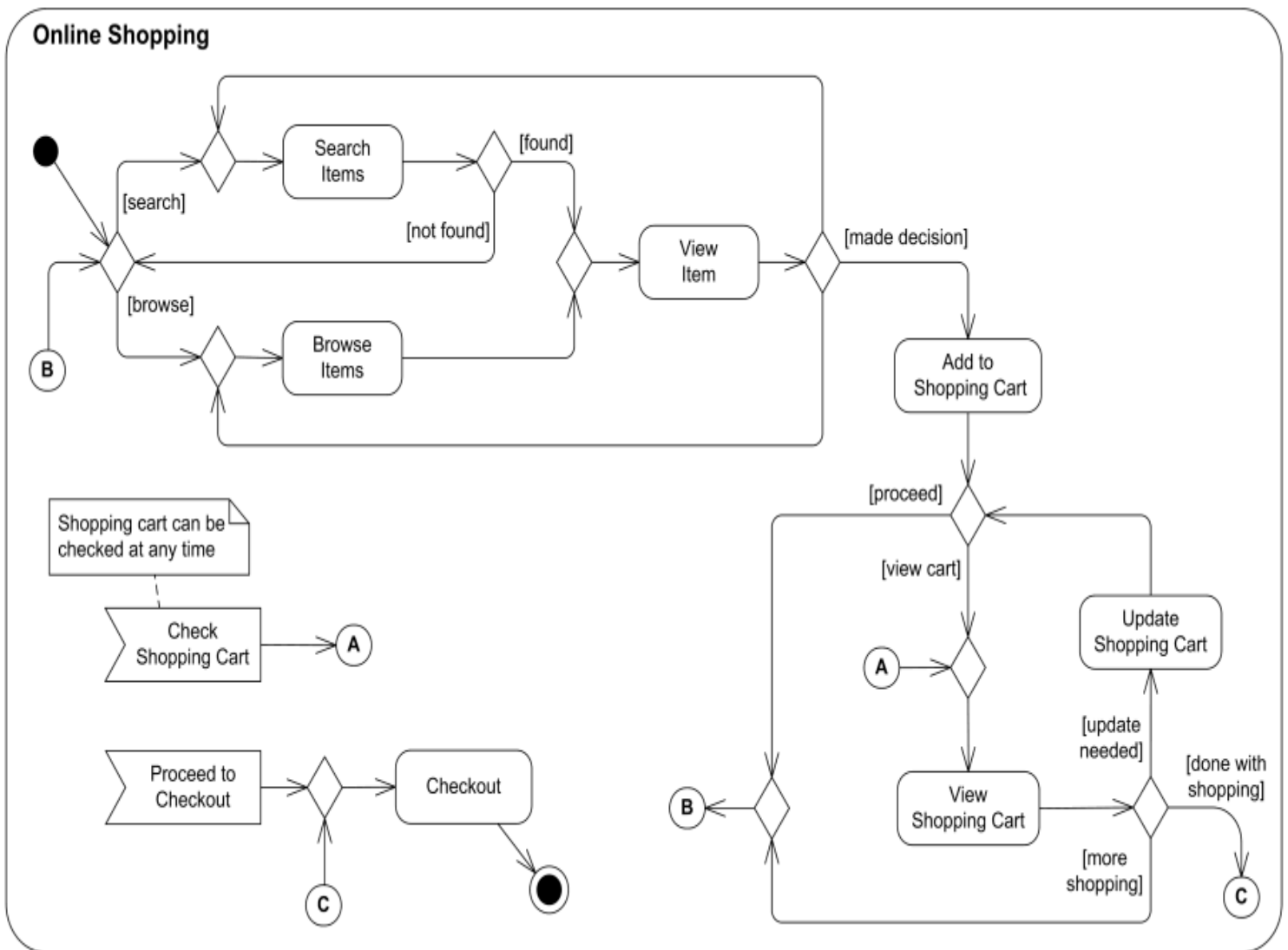


An example of communication diagram for online bookshop

Online Shopping Activity Diagram

An example of [activity diagram](#) for **online shopping**. Online customer can browse or search items, view specific item, add it to shopping cart, view and update shopping cart, checkout. User can view shopping cart at any time. Checkout is assumed to include user registration and login.

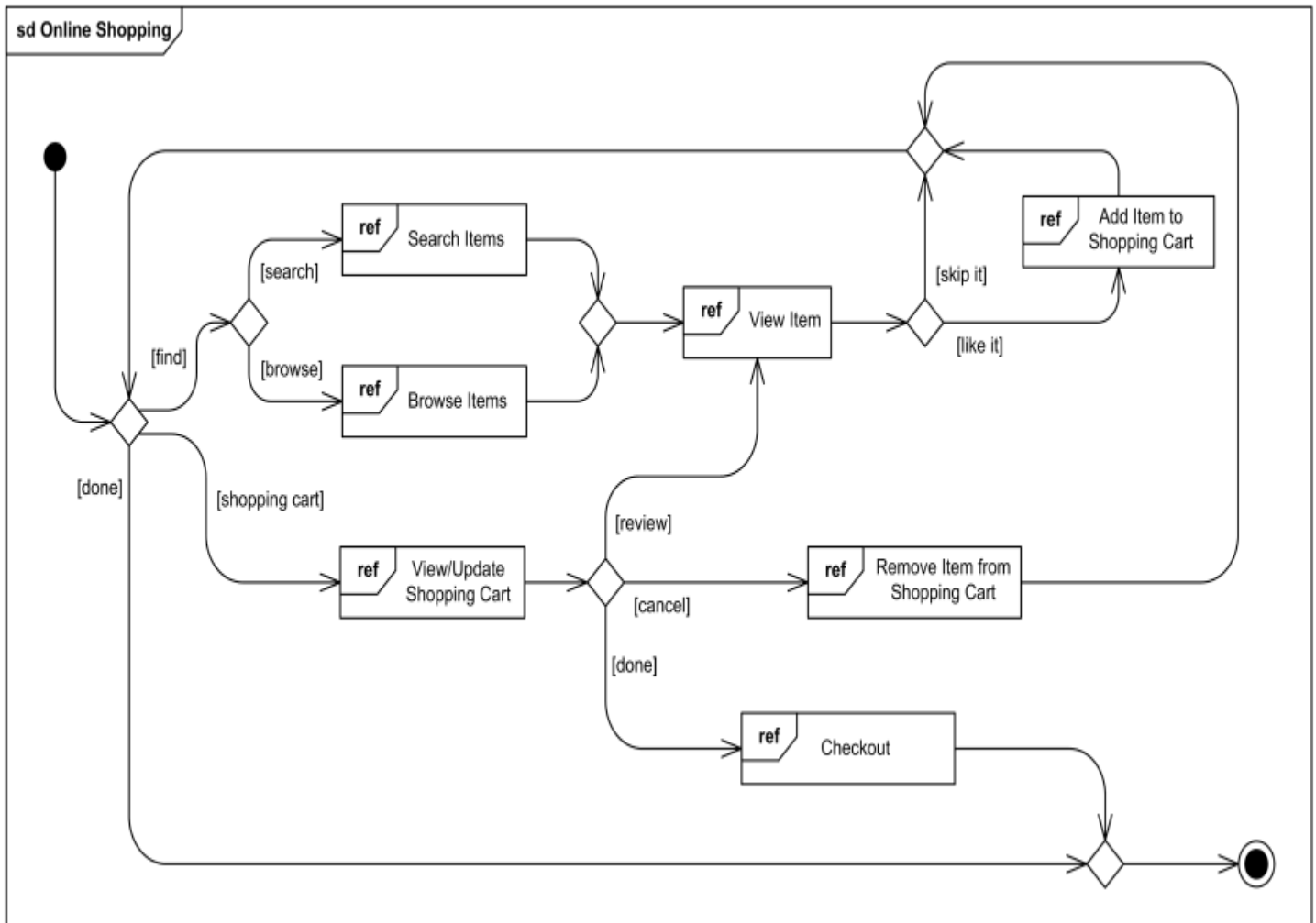
This example is not using partitions, most of the actions are assumed to be fulfilled by online customer.



An example of activity diagram for online shopping.

Interaction Overview Diagram

An example of [interaction overview diagram](#) for online shopping. Customer may search or browse items, add or remove items from shopping cart, do checkout.



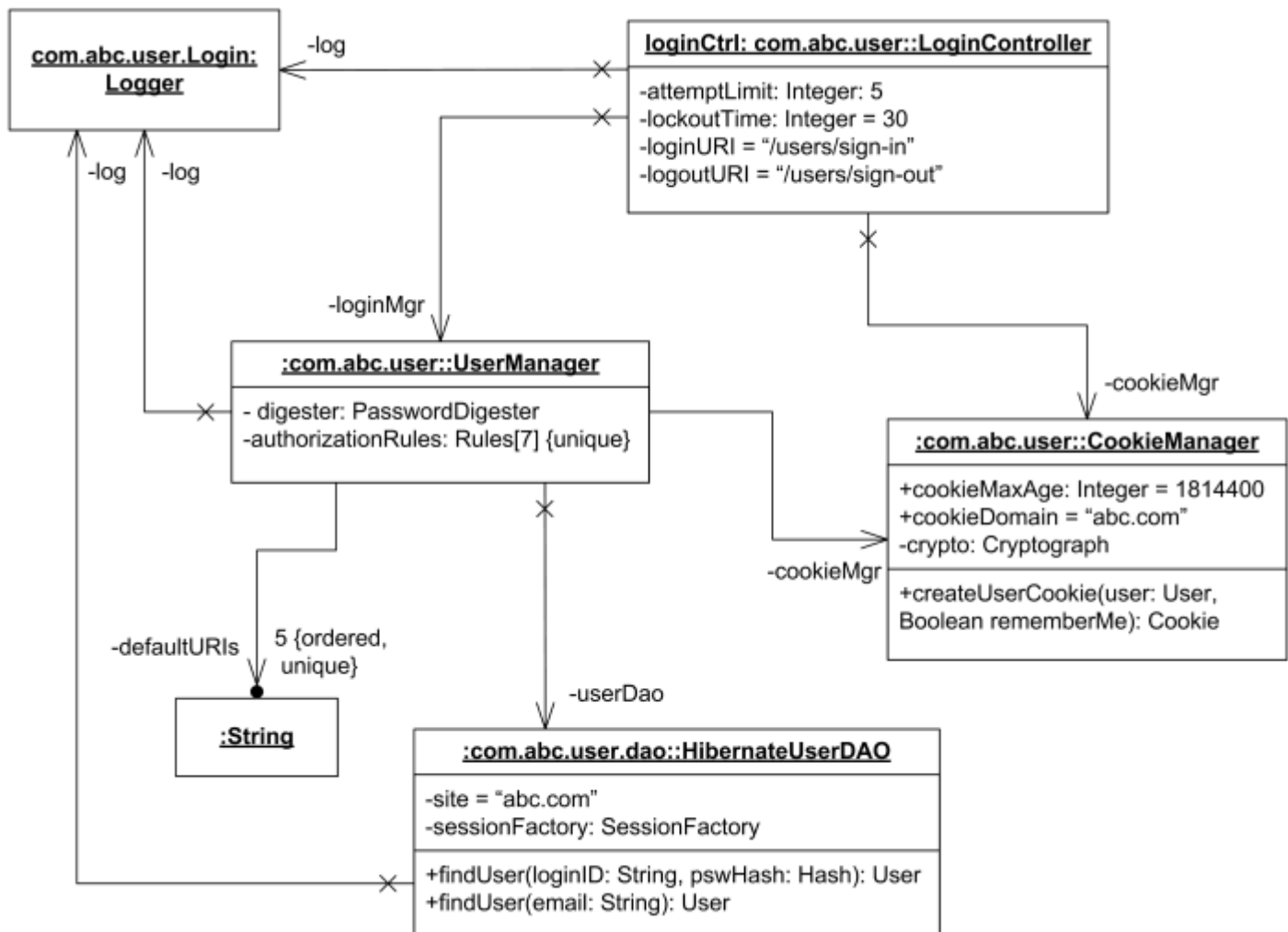
An example of interaction overview diagram for online shopping.

This interaction overview diagram is enclosed by **sd** frame (abbreviated form for all kinds of interaction diagrams). All non-control nodes of the diagram are references to interactions - [interaction uses](#).

Login Controller Object Diagram

This is an example of [object diagram](#) which shows some runtime objects related to web user login process. Class [instance](#) loginCtrl of the Login Controller has several [slots](#) with [structural features](#) of Integer and String types and corresponding value specifications.

The instance of Login Controller is also associated with instances of User Manager, Cookie Manager, and Logger. Login Controller, User Manager, and Hibernate User DAO (Data Access Object) share the single instance of Logger.



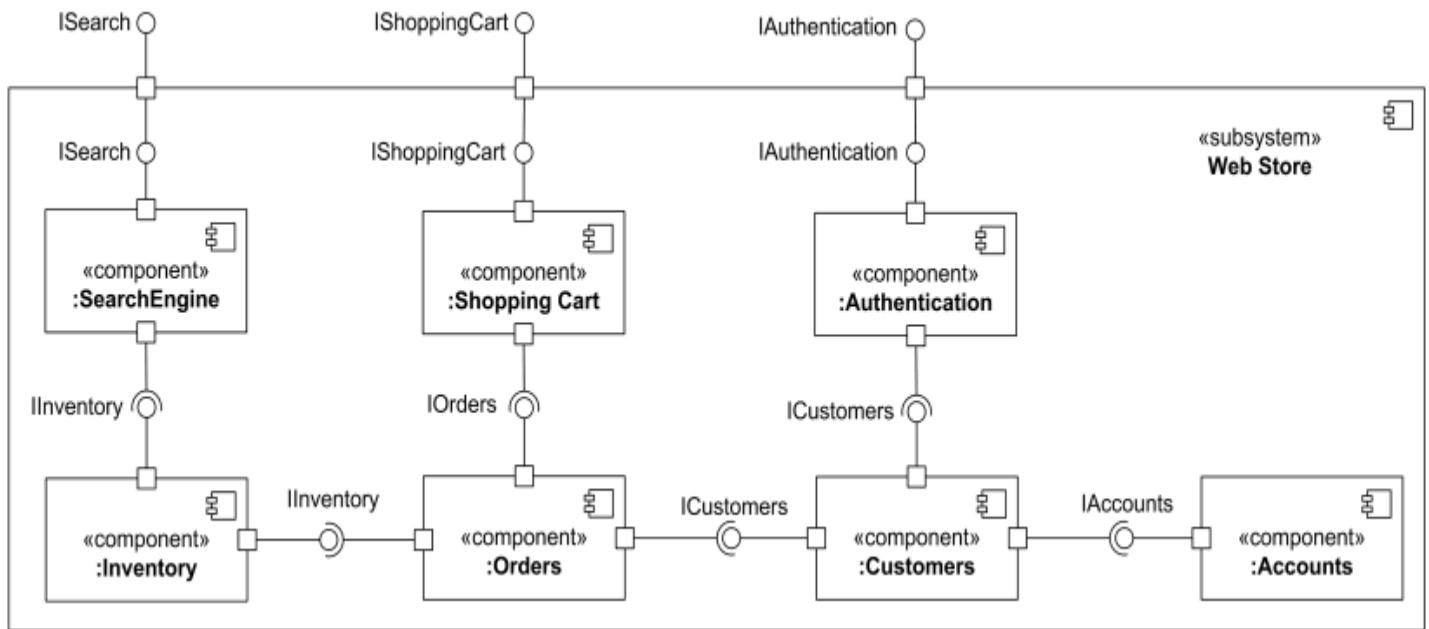
Login Controller object diagram.

User Manager has private attribute defaultURIs which is ordered collection (array) of 5 unique Strings. Instance of the Cookie Manager has two public structural features with specified values. Most [links](#) are non navigable backward.

Online Shopping Component Diagram

An example of [component diagram](#) for a **retail website**.

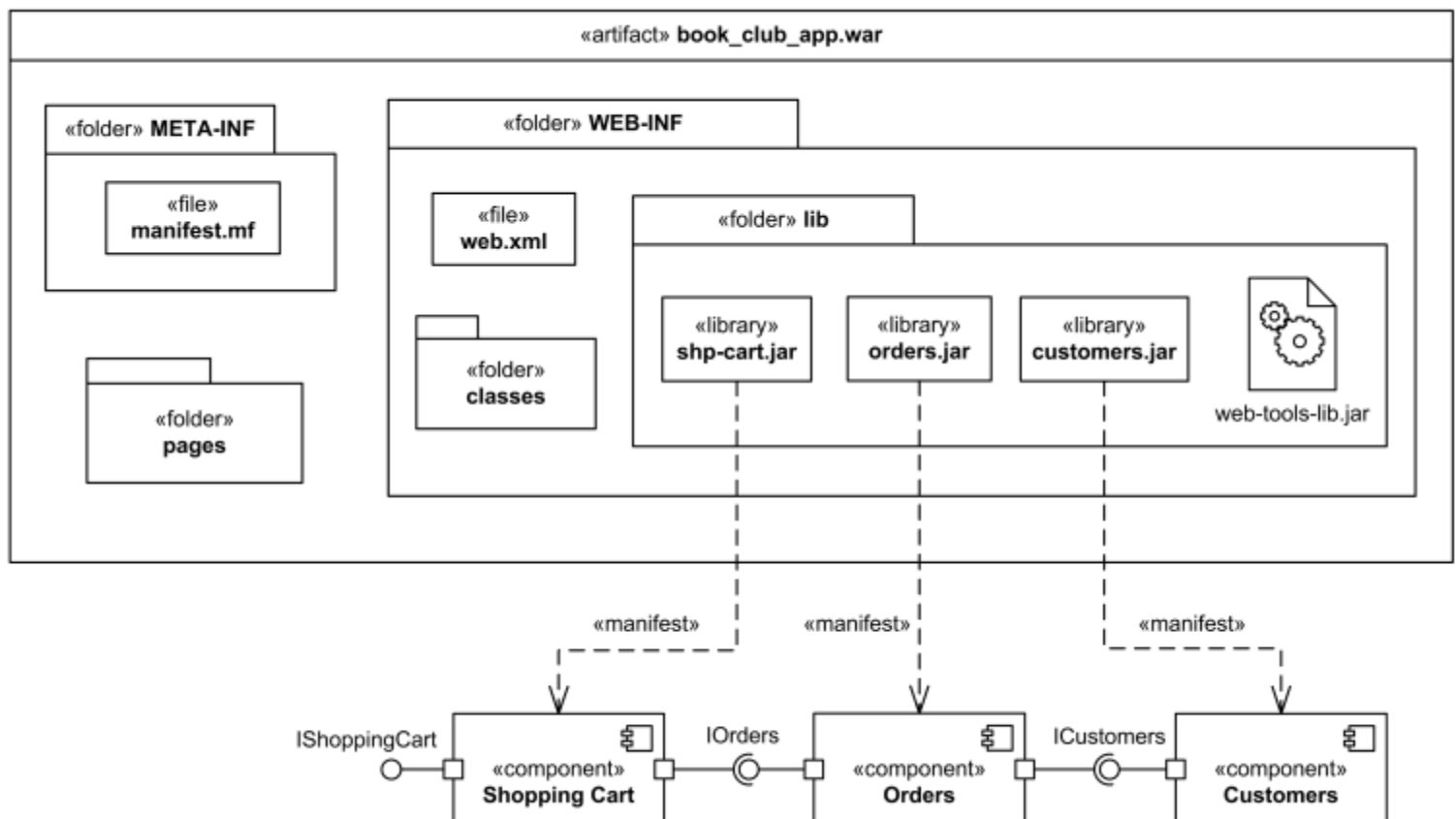
Search Engine [component](#) uses **IInventory** interface to allow customers to search or browse items. **Shopping Cart** component uses **Orders** component during checkout process. **Authentication** component allows customer to login and binds the customer to **Account**.



An example of component diagram for a retail website.

Web Application Manifestation

This is an example of UML [deployment diagram](#) which shows [manifestation](#) of [components](#) by [artifacts](#) and internal structure of artifacts. I would call this kind of deployment diagrams **manifestation diagrams** or **implementation diagrams** as they actually do not show any [deployments](#).



Example of manifestation diagram for web application.

Web application archive artifact `book_club_app.war` contains several files, folders and subfolders. Stereotypes «file» and «library» are standard UML stereotypes applicable to artifacts. This diagram also shows non-standard stereotype «folder» which is rendered as package.

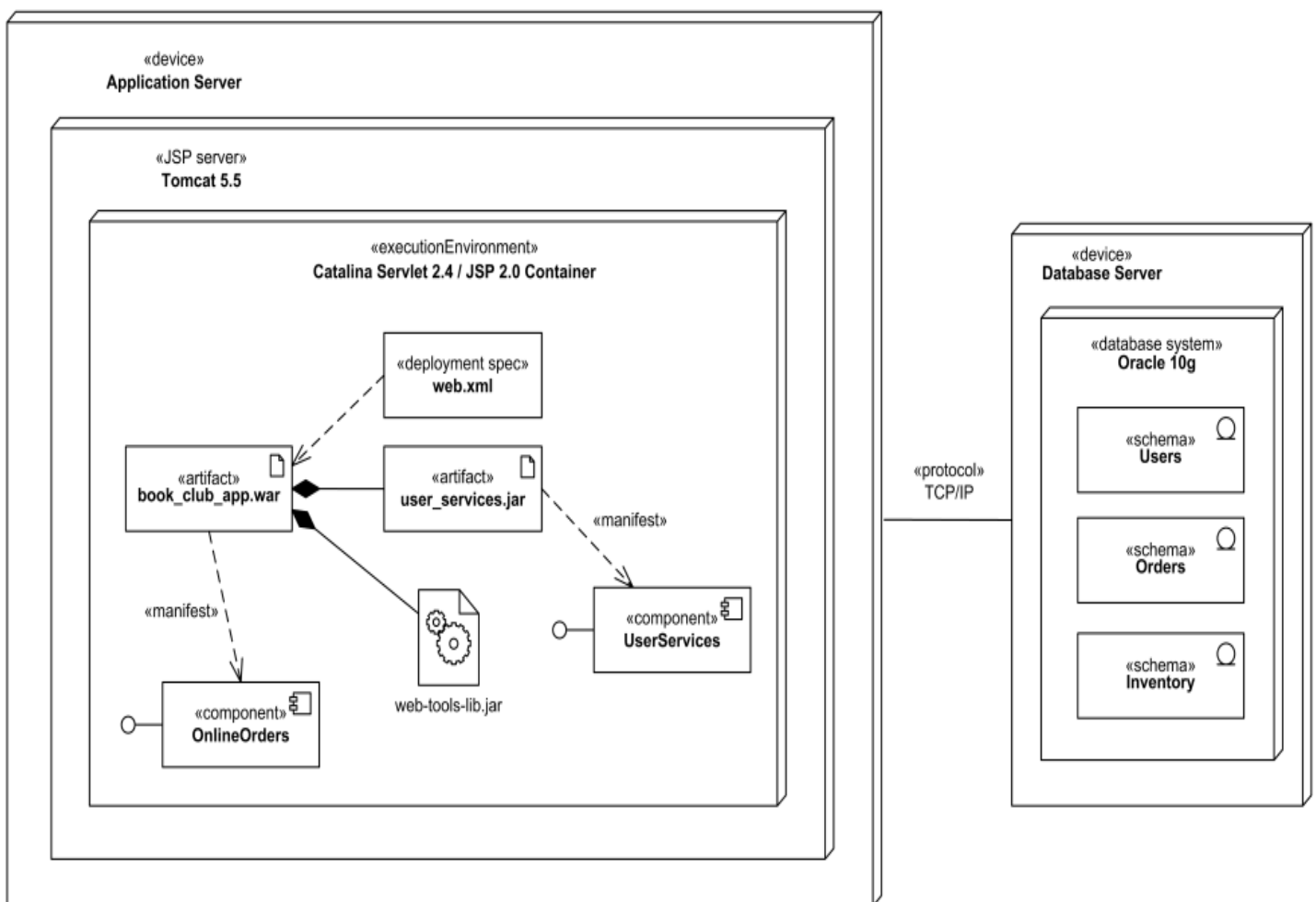
Several components are shown on the diagram manifested (implemented) by jar archive files.

Deployment Diagram

An example of **web application** UML [deployment diagram](#). Book club web application **artifact** `book_club_app.war` is deployed on Catalina Servlet 2.4 / JSP 2.0 Container which is part of Apache Tomcat 5.5 **web server**.

The `book_club_app.war` [artifact manifests](#) (embodies) `OnlineOrders` [component](#). The artifact contains three other artifacts, one of which **manifests** `UserServices` [component](#).

The Application Server «[device](#)» (computer server) has [communication path](#) to Database Server «[device](#)» (another server).



An example of deployment diagram for web application.