The Figure 1 below depicts the typical interaction between the View, Controller and Model. This is the design patter that was adhered to in the CD project.

The views are jsp pages that interact with the Model through calls to the controllers. Each jsp page will call a specific controller in order to display the desired information. The table 1 outlines which jsp pages call which servlet.

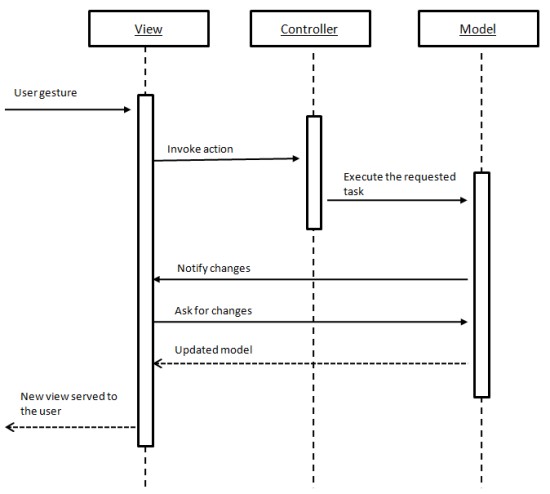


Figure 1 Generic sequence diagram for a typical MVC application

TODO – add table to map jsp to controller.

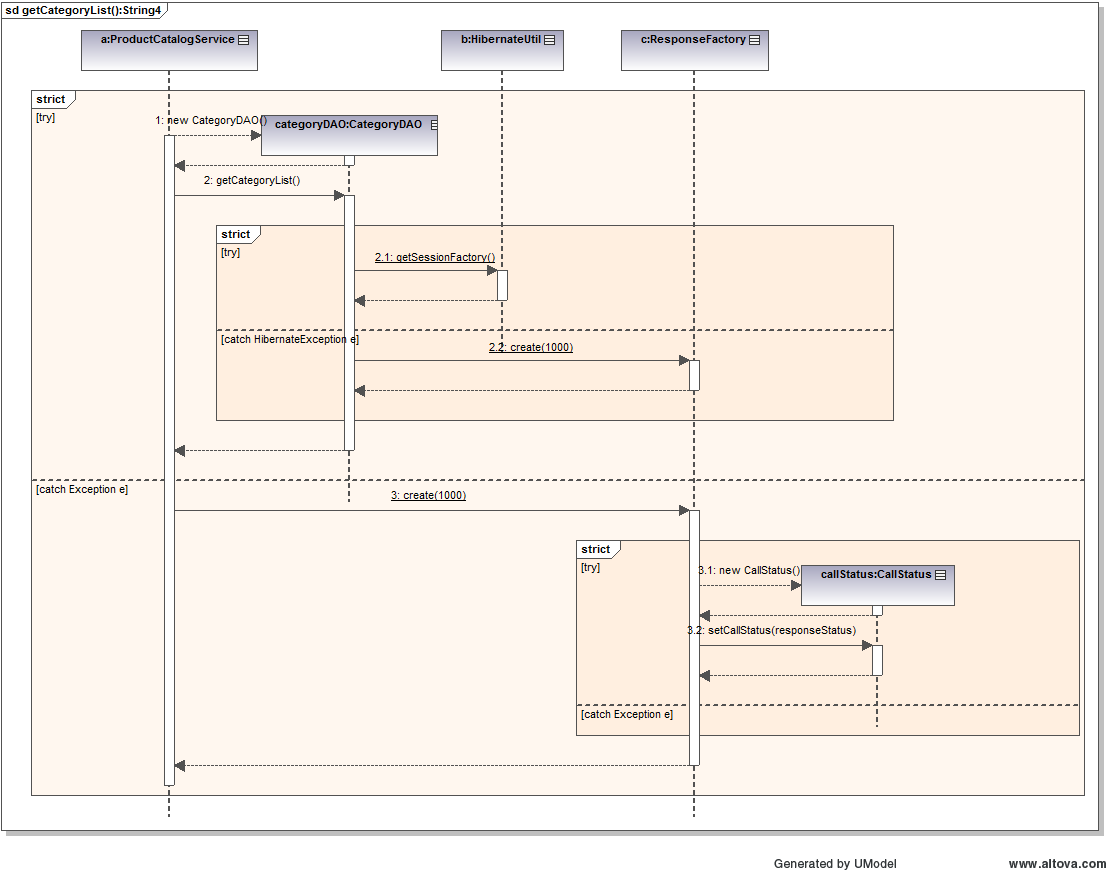
|  |  |
| --- | --- |
| JSP name | Controller Name |
| cart | addtoCartAction |
| category | categoryAction |
| finish | logoutAction |
| index |  |
| info | ProductDetailAction |
| payment |  |
| register |  |
| shipping |  |
| signin | loginAction |

Table 1 Mapping of Jsp calls to Servlets

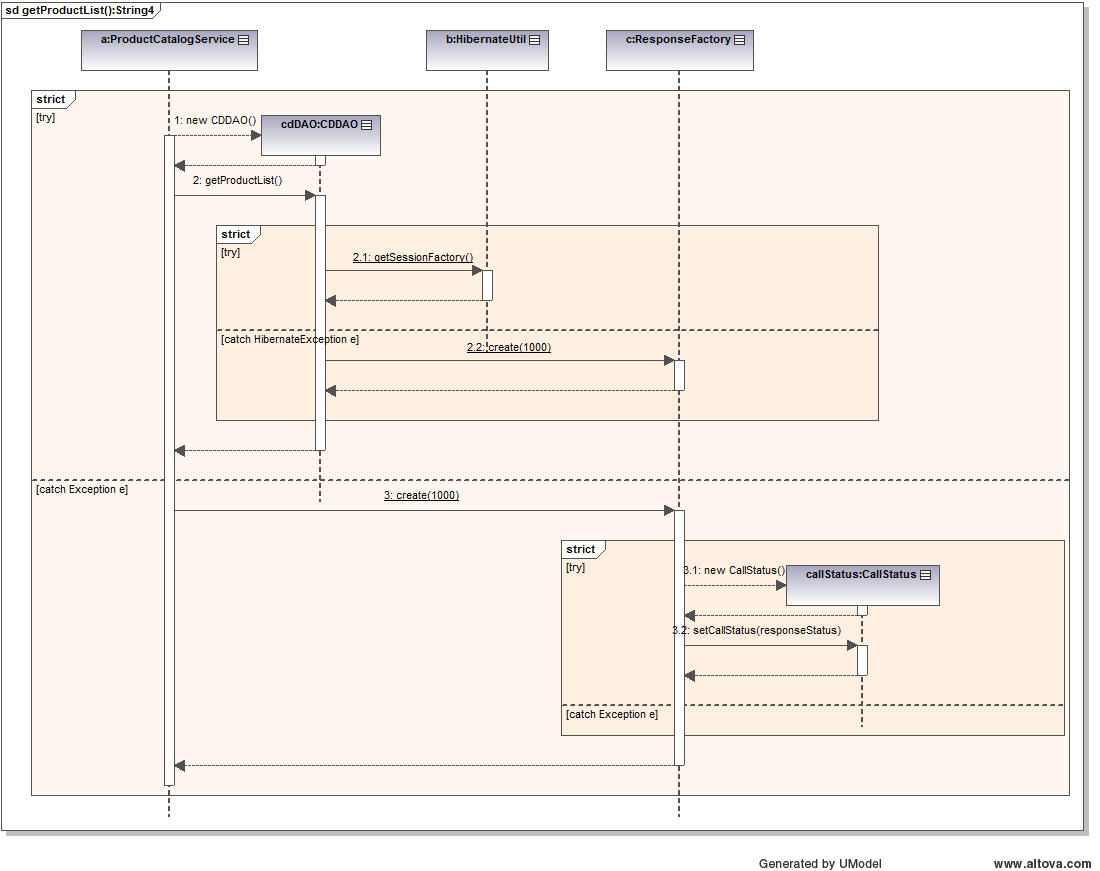
The controllers are responsible for interacting with the Model. All Model-related functionality is exposed through two webservices. Client stubs were created so that the controllers could use them to call the webservices as if the webservices resided locally. The following sequence diagrams depict the flow of each method call contained within the two webservices. The calls to the database are handled through hibernate calls so the specifics of the interactions between the webservices and the databases are abstracted out. They are not explicitly depicted since the hibernate framework does the work. No programming was required, only configuration.

ProductCatalogService Webservice

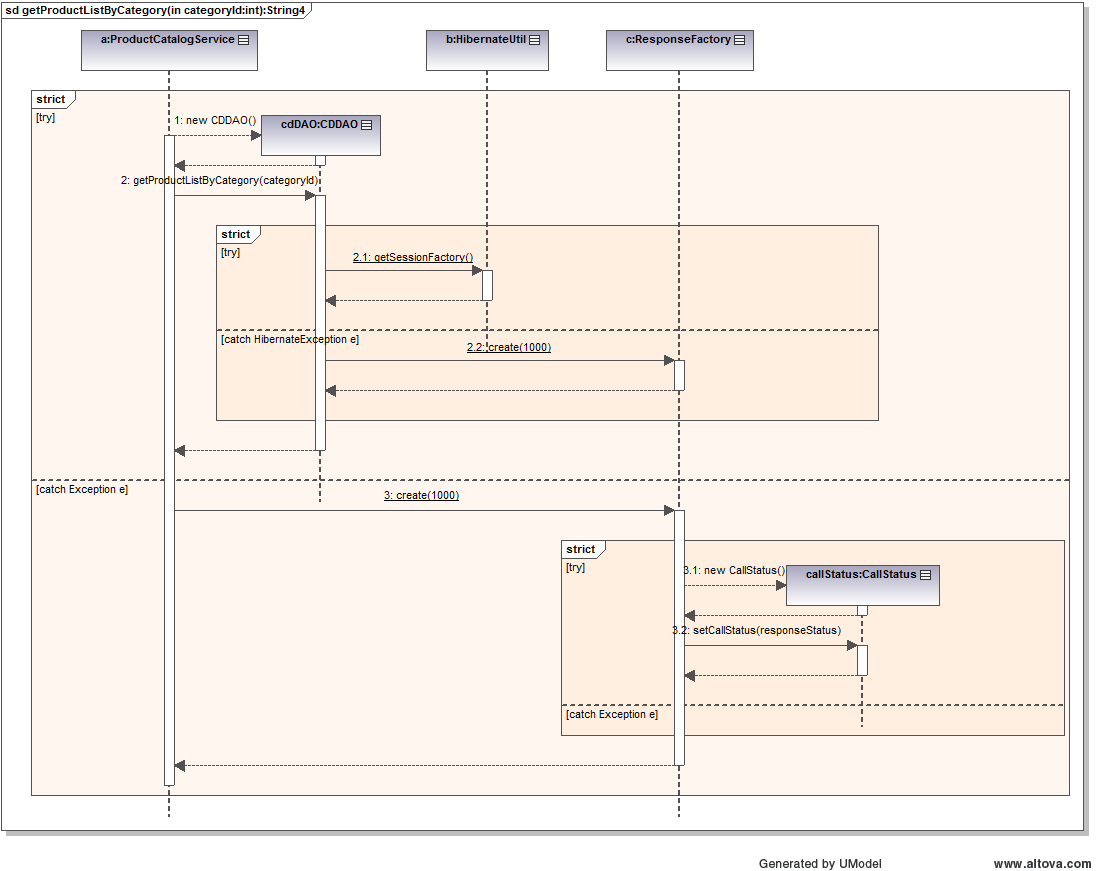
getCategoryList Sequence Diagram



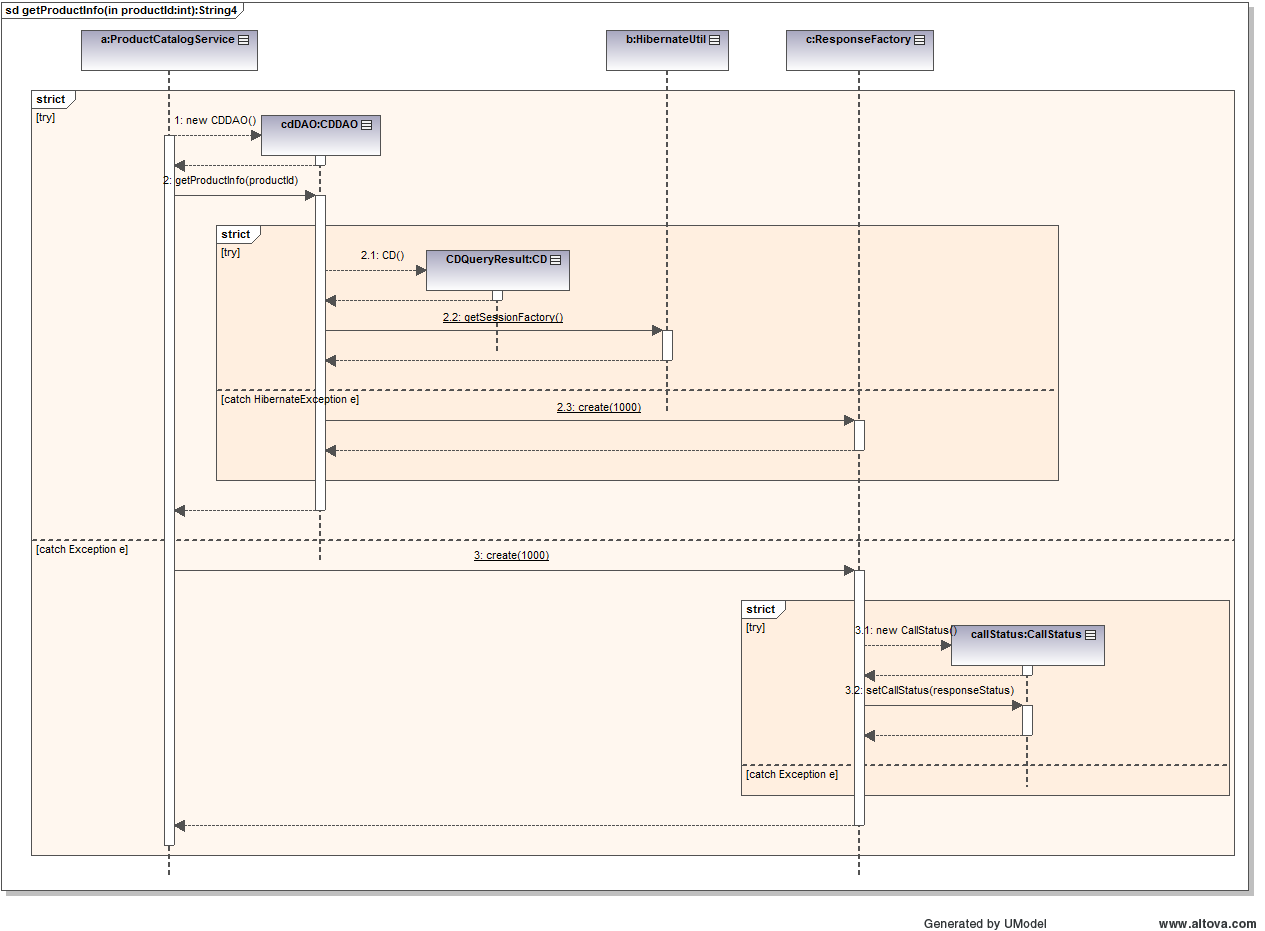
getProductList Sequence Diagram



getProductListByCategory Sequence Diagram

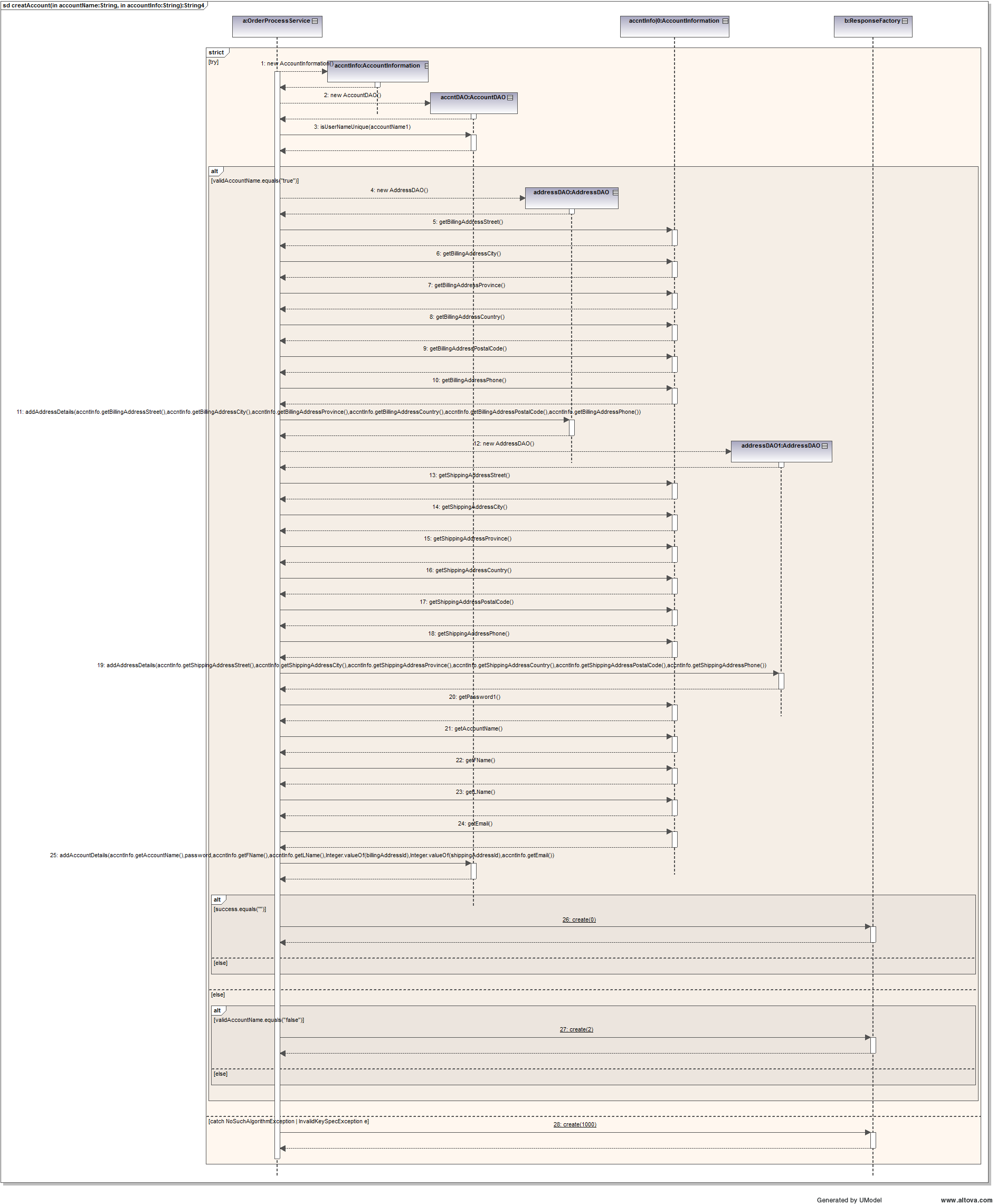


getProductInfo Sequence Diagram

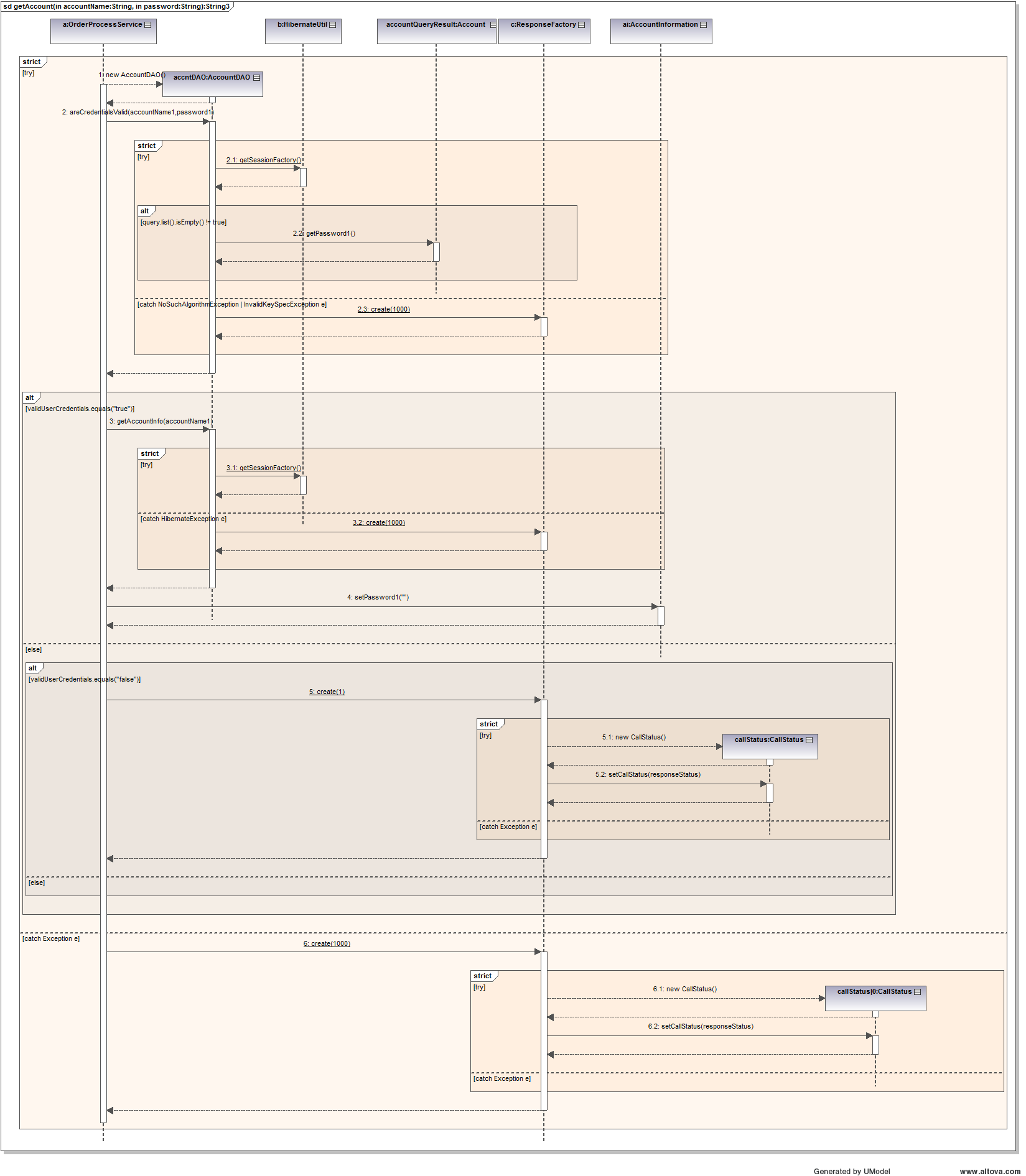


ProcessOrderService WebService

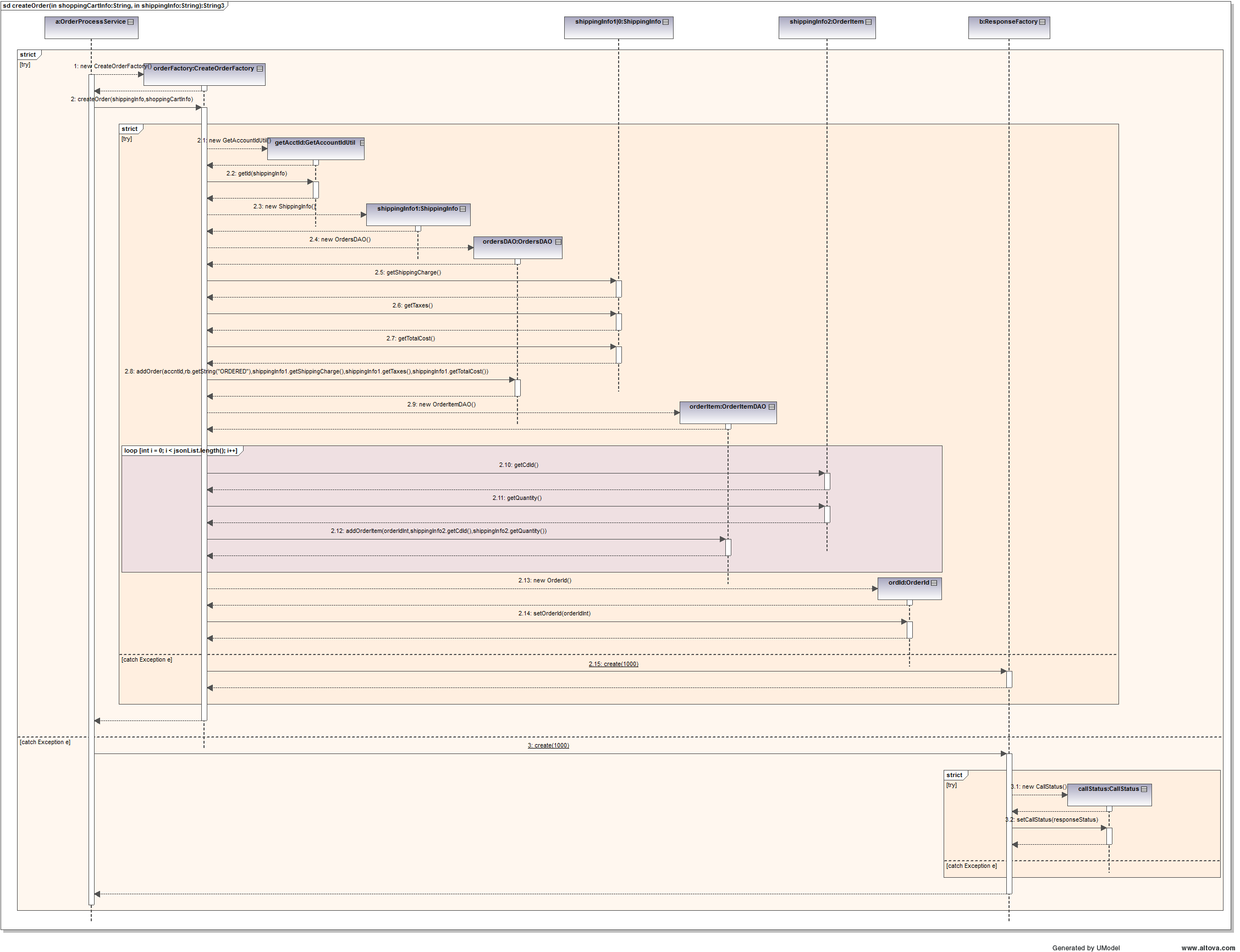
createAccount Sequence Diagram



getAccount Sequence Diagram



createOrder Sequence Diagram



confirmOrder Sequence Diagram

