På domænemodellen for Western Style Ltd. ses det at Customer og Employee klasserne, begge er specialiseringer af klassen Person. Dette gør man når to eller flere fælles attributter, for ikke at have ens kode i 2 klasser. Relationen fra Customer samt Employee og til SalesOrder er ’en nul 0’ til mange relation. Dog er relationen fra SalesOrder til Customer og Employee en 1. Dette skyldes at en SalesOrder skal have en Customer og en Employee. Men en Customer og en Employee behøver nødvendigvis ikke at have en SalesOrder for at eksistere.

SalesOrder fungere lidt som samlepunkt for vores klasser i domænemodellen. Den er for eksempel også relateret til Invoice, hvor den har en 1 relation. Dette er fordi en SalesOrder altid skal have 1 Invoice, men den kan heller ikke have flere end 1, for den enkelte SalesOrder. Invoice kan derimod godt have flere SalesOrder, fordi flere SalesOrder godt kan have en Invoice hver. Dette giver Invoice en ’1 til mange’ relation.

SalesOrder har derudover også en abstract klasse tilhørende. Dette er OrderCondition, som er aggregeret til SalesOrder. Dette gøres fordi SalesOrder kan være i flereforskellige stadier feks. Offer, Order eller delivered. SalesOrder har derfor en *OrderCondition* som kan være forskellige stadier, og derfor er den abstract. *OrderCondition* har en ’0 til 1’relation til SalesOrder, og denne har en ’1’ relatation til OrderCondition. Offer, Order og Delivered klasserne er alle specialiseringer af *OrderCondition* klassen, da de deler det at være en ’OrderCondition’.

SalesOrderLine er aggregeret til SalesOrder, da SalesOrder består af ’1 til mange’ SalesOrderLine. SalesOrderLine har en ’1’ relation til Product, da en SalesOrderLine kun består af 1 Product. Derimod kan product godt være på flere SalesOrderLine, og har derved en ’1 til mange’ relation til SalesOrderLine.  
Her er Clothing, Equipment og GunReplica specialiseringer af Product, da de deler mange attributter, og herved ’arver’ de attributter som de har til fælles med de andre klasser.