**Workshop Persistent**

University College Nordjylland

Computer Science



**Group 6**

**Kacinskas Aistis Arvydas, Darahi Alexandru, Ramanauskas Karolis**

Table of contents

[Mock-up 2](#_Toc414620226)

[Fully dressed use case 3](#_Toc414620227)

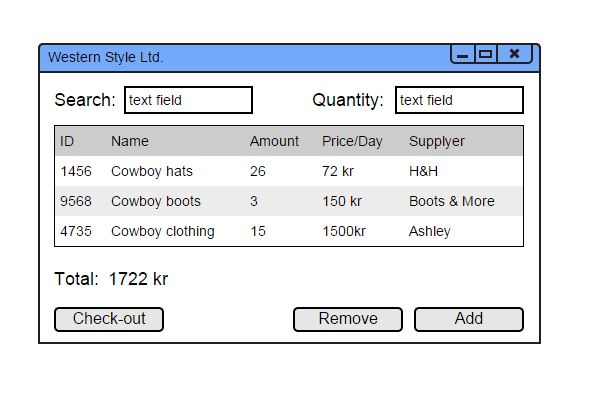
[System sequence diagram 4](#_Toc414620228)

[Interaction diagrams 4](#_Toc414620229)

[Domain Model 6](#_Toc414620230)

[Relational model 6](#_Toc414620231)

Mock-up

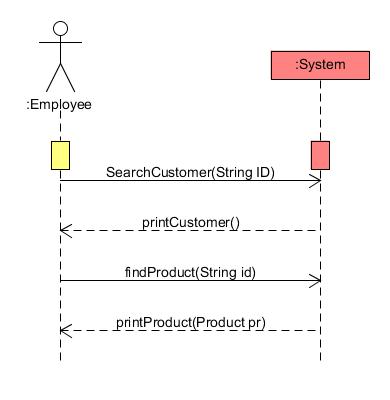
****

This mock-up shows how we imagine the system should look like. This kind of design is user friendly and do not require a lot of learning. Use: in the search field employee writes products name and amount of it then presses “Add” button. Then information of product pops up in the window. After adding everything employee presses “Check-out” button to end order and go to invoice.

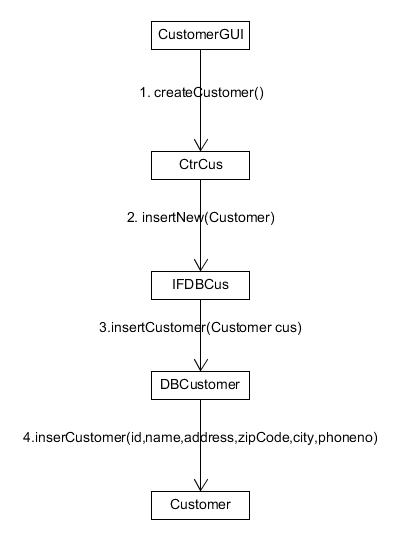
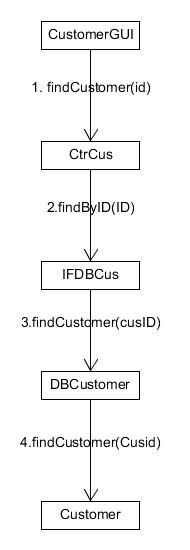
Fully dressed use case

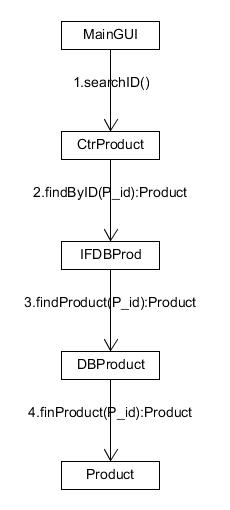
|  |  |  |
| --- | --- | --- |
| Use case name | Order processing | |
| Actors | Employee | |
| Pre-conditions | Customer is registered;  Products exist | |
| Post-conditions | Order accepted | |
| Frequency | Approx. 10 times per day | |
| Main Success scenario  (Flow of events) | **Actor (Action)** | **System (Response)** |
| 1. Employee enters customer ID | 2. Checks whether customer exists, shows customer information and system creates order |
| 3. Enters items ID and their amount | 4. System adds items to the order |
| 5. Employee ends the order | 6. System saves the order, creates an invoice, calculates the total price and shows information |
| Alternative flows | 2.A) Customer does not exist. Employee creates customer. | |
| 3. & 4. Action will be repeated for each item | |

System sequence diagram

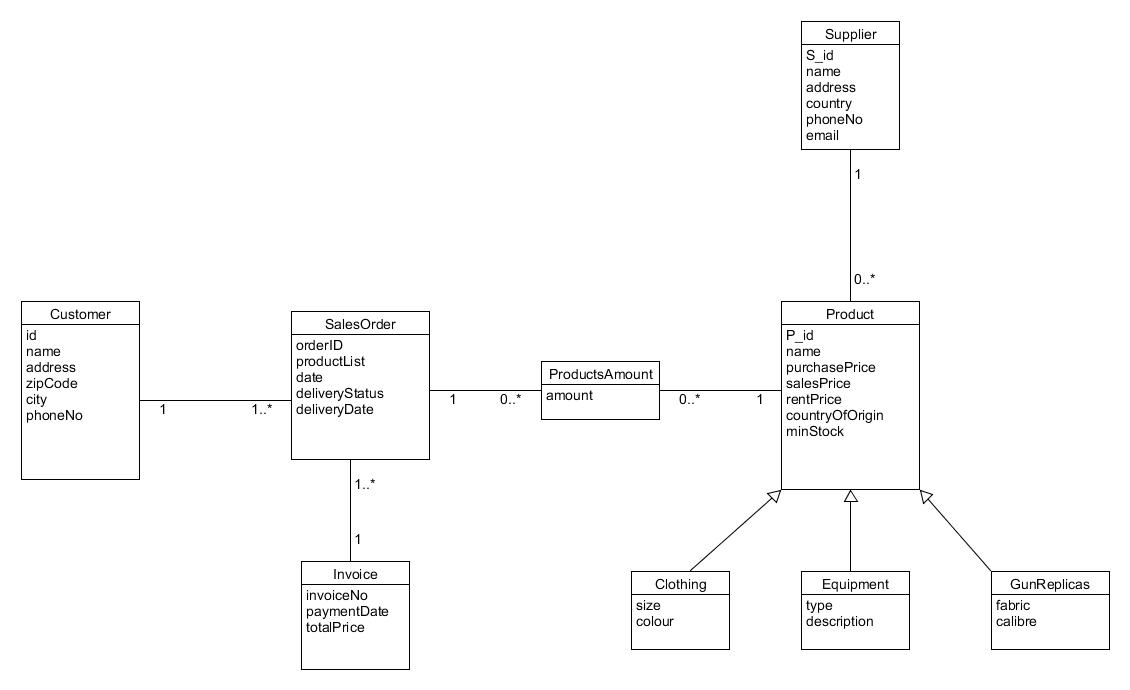
****

Interaction diagrams

****

****

Domain Model



It was decided that one customer could have as many orders as he wants. Also one order could have as many products as needed. However, for creating an order the system should have at least one customer and product. Also there are sub-classes to attach additional information of certain products and these items have their supplier class.

Relational model

**Customer**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **c\_id** | **cname** | **caddress** | **czip\_code** | **ccity** | **cphone\_no** |
| 3579 | Karolis | vesterbro | 9000 | aalborg | 52346422 |
| 1596 | Aistis | vesterbro | 9000 | aalborg | 52365489 |
| 9865 | Alex | vesterbro | 9000 | aalborg | 52624289 |

For customer class this is relational schema. All attributes from domain model is columns. “Char” type was given to “c\_id” attribute. “Varchar” was given to: “cname”, “caddress”, “ccity” and “cphone\_no” attributes and integer to “czip\_code”. While “c\_id” was considered to be primary key and “cphone\_no” unique.

**SalesOrder**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| o\_id | odate | cid | oDelStatus | oDelDate | inNO |
| 1234 | 2015-05-13 | 3579 | null | null | 1 |
| 1235 | 2015-05-14 | 1596 | null | null | 2 |
| 1236 | 2015-05-15 | 9865 | null | null | 3 |

This is relational scheme for SalesOrder class. “Char” type was given to “o\_id, cid, inNo” attributes. “Varchar” was given to: “oDelStatus” and “oDelDate” both attributes can be null. For “odate” attribute we gave “datetime” type. Primary key was considered to be “o\_id”. While first foreign key “cid” references to Customer class “c\_id” and second foreign key “inNO” references to Invoice class ”inNO” attribute.

**Invoice**

|  |  |  |  |
| --- | --- | --- | --- |
| **inNo** | **inPayDay** | **price** | **oid** |
| 3 | null | 200 | 1236 |
| 2 | null | 100 | 1235 |
| 1 | null | 400 | 1234 |

Relational scheme for Invoice Class. We gave “char” type to: “inNO”, “price”, “oid” attributes. “Varchar” was given to “inPayDay” attribute which can be null. While primary key is “inNo”.

**ProductsAmount**

|  |  |  |
| --- | --- | --- |
| **p\_id** | **amount** | **order\_id** |
| 7777 | 3 | 1234 |
| 332 | 50 | 1235 |
| 2222 | 1 | 1236 |

For this relational scheme “char” types was given to: “p\_id” and “order\_id”. Integer was given to the “amount” attribute. Primary keys are: “p\_id” and “order\_id”. While first foreign key “order\_id” references to SalesOrder class “o\_id” attribute and second foreign key “p\_id” references to Product class “pid” attribute.

**Supplier**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **sName** | **sAddress** | **sCountry** | **sPhone** | **sEmail** | **s\_id** |
| Patrick | Hobrovej | Denmark | 123456 | [c@d.dk](mailto:c@d.dk) | 501 |
| Anthony | Nibevej | Germany | 654321 | [a@b.dk](mailto:a@b.dk) | 502 |
| George | Letvadvej | Norway | 987654 | [fg@k.dk](mailto:fg@k.dk) | 503 |

Supplier relational scheme mapped from the domain model. Type of “varchar” was given to: “sName”, “sAddress”, “sCountry”, “sPhone”, “sEmail” attributes. “Char” type is “s\_id” attribute and primary key is also “s\_id”.

**Product**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **pid** | **pName** | **pPP** | **pSP** | **pRP** | **pCountry** | **pMinStock** | **supplier\_id** | **type** |
| 7777 | shirts | 10 | 35 | 5 | USA | 200 | 503 | Clothing |
| 332 | shotgun | 50 | 175 | 20 | India | 100 | 501 | GunReplicas |
| 2222 | flags | 100 | 350 | 40 | China | 50 | 502 | Equipment |
| 8788 | vest | 33 | 110 | 15 | China | 50 | 503 | Clothing |
| 9997 | jacket | 50 | 200 | 20 | India | 25 | 503 | Clothing |
| 2221 | belt | 15 | 40 | 8 | Vietnam | 50 | 502 | Equipment |
| 2233 | box | 66 | 220 | 24 | Korea | 25 | 502 | Equipment |
| 331 | revolver | 150 | 350 | 60 | China | 65 | 501 | GunReplicas |
| 333 | pistol | 120 | 300 | 50 | USA | 100 | 501 | GunReplicas |

This is relational scheme for a product class. “Char” type was given to: “pid” and “supplier\_id” attributes. “Varchar” was given to: “pname”, “pcountry” and “type” attributes. Now we used new type that was not used in previous classes “decimal” which we used for: “pPP (purchase price)”, “pSP (sales price)” and “pRP (rent price)” attributes. Primary key for this class is “pid” attribute. While foreign key “supplier\_id” references to Supplier class “s\_id”.

**Clothing**

|  |  |  |
| --- | --- | --- |
| **size** | **colour** | **product\_id** |
| 55 | red | 7777 |
| 60 | blue | 8788 |
| 45 | green | 9997 |

**Equipment**

|  |  |  |
| --- | --- | --- |
| **type** | **description** | **product\_id** |
| belts | pistol belts | 2221 |
| flags | western flags | 2222 |
| boxes | gift boxes | 2233 |

**GunReplicas**

|  |  |  |
| --- | --- | --- |
| **fabric** | **calibre** | **product\_id** |
| plastic | 12mm | 331 |
| metal | 9mm | 332 |
| plastic | 6mm | 333 |

These all three relational schemes uses only “char” and “varchar” types for all the attributes. Primary key for all are the same “product\_id” and foreign key “product\_id” references to Product class “pid” which is the same also for all three schemes.