Sign up

st-tu-dresden / videoshop Public

Code Issues 6 Pull requests Actions Projects Security Ins

main

videoshop / src / main / asciidoc / Pflichtenheft.adoc

HamannM #147 - Update requirements specification (Pflichtenheft). ... History

4 contributors

1003 lines (799 sloc) | 45.5 KB

| Version | Processing Date | Author |
|---|---|---|
| 1.0 | October 9th, 2017 | Marc Kandler |
| 2.0 | June 21st, 2019 | Daniel Schoenicke |
| 2.5 | October 5th, 2021 | Markus Hamann |

# Pflichtenheft *Videoshop*

## 1. Purpose of this Document

This document represents the Software Requirements Specification (SRS, German: close to a "Pflichtenheft") of the project **Videoshop**. It aims to provide an overview of the software product to be built and functions as a basis for communication between the stakeholders of the project, mainly the client and the development team. It is desired to have this document as the foundation for a contract between the client and the contractor, and should therefore be validated and checked for consistency. The SRS describes *what* the desired system has to fulfill and *partially how* the contractor intends

to implement the solution.

In general, the SRS should be **correct, complete, and consistent** (CCC). As it is used at the end of the project to validate whether or not the defined software has been delivered, this goal should be **verifiable**. It is going to be used and consulted throughout the whole project which is why the contents should be **traceable** throughout all artifacts that are created. In conjunction with the above, it is desired to be simple to change and evolve, even though changes should be kept to a minimum after the stakeholders once agreed on the content. However, as requirements and circumstances constantly change during a project, adjustments are to be expected and have to be documented.

We aim to provide an example of a SRS, which can be used as a reference during the course "Softwaretechnologie-Projekt" at Technische Universität Dresden (TU Dresden). This SRS is not complete and in larger projects many more aspects can and should be considered. The provided document is only to be used for educational purposes. Contributions are welcome.

*Note: We included several remarks in this document, just like the one you are reading right now, to provide you some additional information. These remarks would obviously not be in a real SRS, but are used here for educational purposes. Said remarks are labeled with the word "Note" and are written in italic.*

*Note: You do not necessarily have to use the same table structure or formatting we used. Regardless of the presentation of the information, you should aim to provide all necessary details.*

*Note: You can also split the chapters into single files and merge them (`include::<filename>.adoc[]`) into a single file to allow for a better distributed workflow.*
*Please mind that there is a problem with this approach. GitHub is currently not supporting the `include` statement and will not render the file on the GitHub website. Please consult with your tutor before using this feature.*

## 2. Task Definition

*Note: This task aims to provide an example for the Videoshop. It is written from the perspective of the client of this project (Chair of Software Technology) and therefore can be seen as a requirements specification. You usually cannot expect such a document to be complete or even consistent, which is why you should always ask in case of uncertainty. Wherever information about the domain is provided, we used italic to show the representation in the domain model.*

The times when people went to buy their movies physically in the store are mostly over. As we, the Chair of Software Technology, have always had our little secondary business

of selling movies to students, this change affects us as well. Therefore, we finally want to take the leap to move our business into an online shop. We need a software, which can support all of the core aspects of our current shop and automatize processes wherever possible.

Our Shop (*Videoshop*) can have any number of users (*User*) which may interact with it differently. Every visitor of our shop may access the catalog (*VideoCatalog*) and its whole functionality. The catalog contains every article (*Disc*) we offer and distinguishes between DVDs (*Dvd*) and Blu-Rays (*BluRay*). The Discs are stored in an inventory (*Inventory*), where they are represented by items (*InventoryItem*). An item saves the current stock (*quantity*) of the Disc. Whenever something is sold, the quantity of the item has to be reduced (*decreaseQuantity*) accordingly to represent the stock correctly.

Besides normal users in our system (*Customer*), we also want administrative access (*Boss*) to manage our shop. Whenever a customer likes something from our catalog, he may add it (*addDisc*) to his virtual basket (*Cart*, *CartItem*) in any quantity (*quantity*) . The contrary is obviously desired as well, to allow our customers to change their mind (*removeDisc*). During the whole process, the customer shall obviously be able to view his selection and see total price of it (*getPrice*). We are especially interested in the automation of the ordering process, which is why we require support for directly buying the content of the cart (*buy*).

After deciding to buy something, an order (*Order*) with the current time (*dateCreated*) is created. It contains each of the chosen items (*OrderLine*) with their quantity (*quantity*) and price (*price*). If the chosen item is not available in the sufficient quantity (*hasSufficientQuantity*), an error should be shown to the customer. As long as the customer did not pay, the order is registered in the system and assigned a status (*OrderStatus*), but not yet processed (*OPEN*). After receiving the money from the customer (*payOrder*), which he may provide through different methods (*paymentMethod*), the order may be processed further (*PAID*). As the order is shipped to the customer, it should be archived (*completeOrder*, *COMPLETED*), as returns or refunds are ruled out :). Should any unforeseen circumstances occur, we obviously do not want an order to be stuck in the system (*CANCELLED*).

Our shop should obviously provide the means for a visitor to register (*register*). As we do only want registered users to have access to some functionality, a security system is required. We do trust the state-of-the-art authentication mechanism with e-mail (*email*) and a password (*password*). However, as we do not want to force visitors to register, they shall be able to leave a comment (*addComment*) with their opinion (*text*) and a rating (*rating*) for every disc in the catalog.

All in all, we want a nice, fast and secure system, which allows us to administrate all of our customers and the stock. It should support our ordering process and allow us to manage everything related to it. The user experience should be awesome, with a

beautiful user interface and a layout, which boosts our sales.

*Note: Especially the last paragraph contains close to no information. In such cases, you have to get more and concrete information regarding requirements from the client. The paragraph as it stands there could pretty much mean anything and force you into critical situations at the end of the project, if the regarding concerns are not addressed appropriately in the rest of the SRS.*

## 3. Product Usage

This section is going to give an overview of how the product is intended to be used upon completion and under which circumstances.

The system is going to be used as a web shop by the Chair of Software Technology to sell movies (discs) to students. The software is supposed to run on a server and be available through the internet (via a browser) to interested customers 24/7.

The system shall be accessible and visually optimized for the following browsers:

- Mozilla Firefox, version 92.0.1+
- Google Chrome, version 94.0.4606+

The primary users of the software are students (customers), who supposedly know typical website navigation schemas, as well as administrators (Boss), who do not necessarily have a technical background.

The system shall not need technical maintenance, as the staff of the Chair of Software Technology already has its hands full. Any data shall be stored persistently in a database and be accessible through the application (e.g. no SQL knowledge should be required for a boss).

## 4. Stakeholders

Here is every group or individual (real or juristical) listed, which/who has an impact on the requirements of the system. In the below table, these stakeholders are listed, a priority is assigned (in case requirements should clash this allows for easier decisions) and their high-level goals are described.

The assigned priorities range from 1 (lowest priority) to 5 (highest priority).

| Name | Priority (1..5) | Description | Goals |
| --- | --- | --- | --- |
| Chair of Software Technology | 5 | The primary client of this project. | • Sell more movies<br>• Automate processes<br>• Have an example application for students<br>• Prevent inventory differences |
| Customers (Students) | 4 | Primary user of the application, supposed to generate income | • Good user experience<br>• Easily browsable catalog<br>• Fast order processing |
| Administrators | 2 | Users who administer the application (e.g. overview all orders) | • Possibility to overview all the data in the system<br>• Manage processes |
| | | | • Easily extendable |

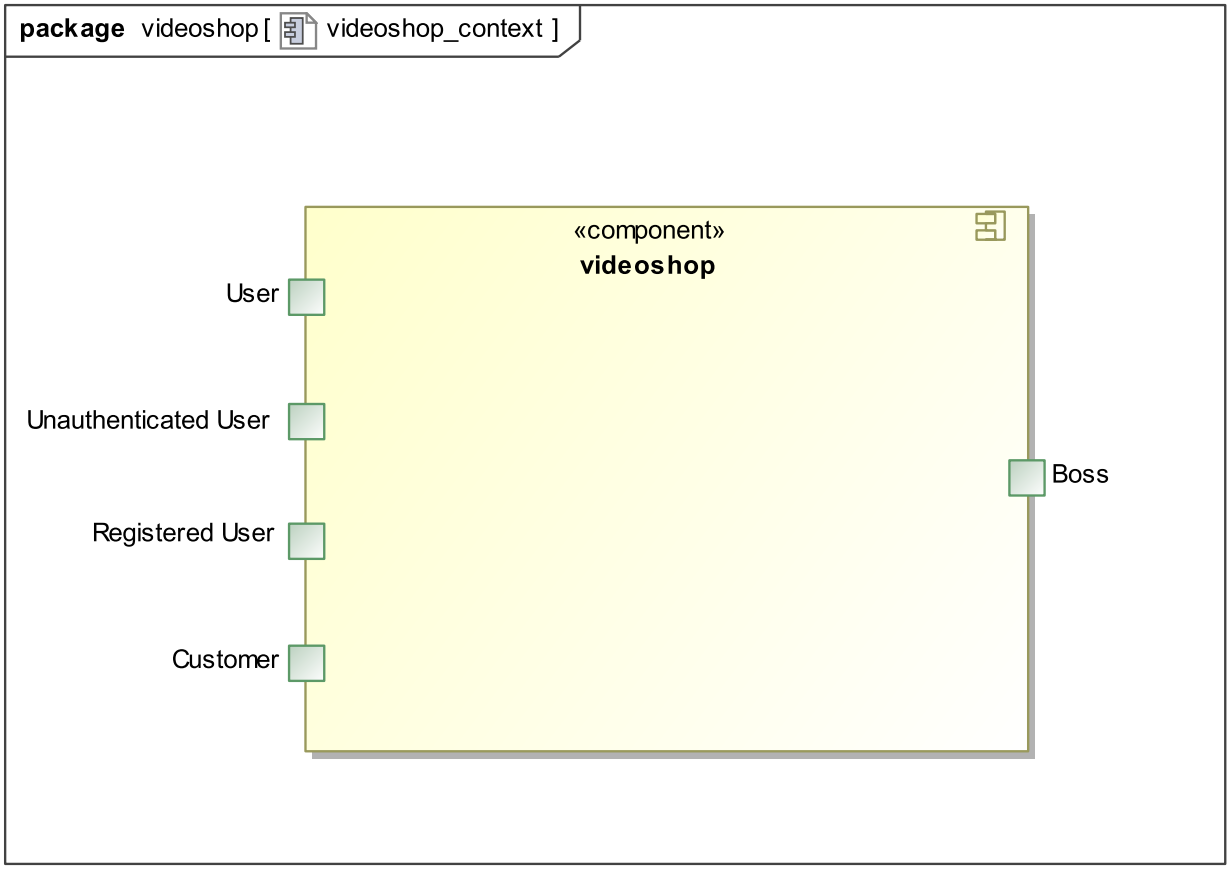
| Name | Priority (1..5) | Description | Goals |
| --- | --- | --- | --- |
| Developers | 3 | People who are either implementing the application or are responsible for maintenance later on. | application<br>• Low maintenance effort<br>• Good debugging mechanisms |

# 5. System Boundaries and Component Structure

## 5.1. System Context Diagram

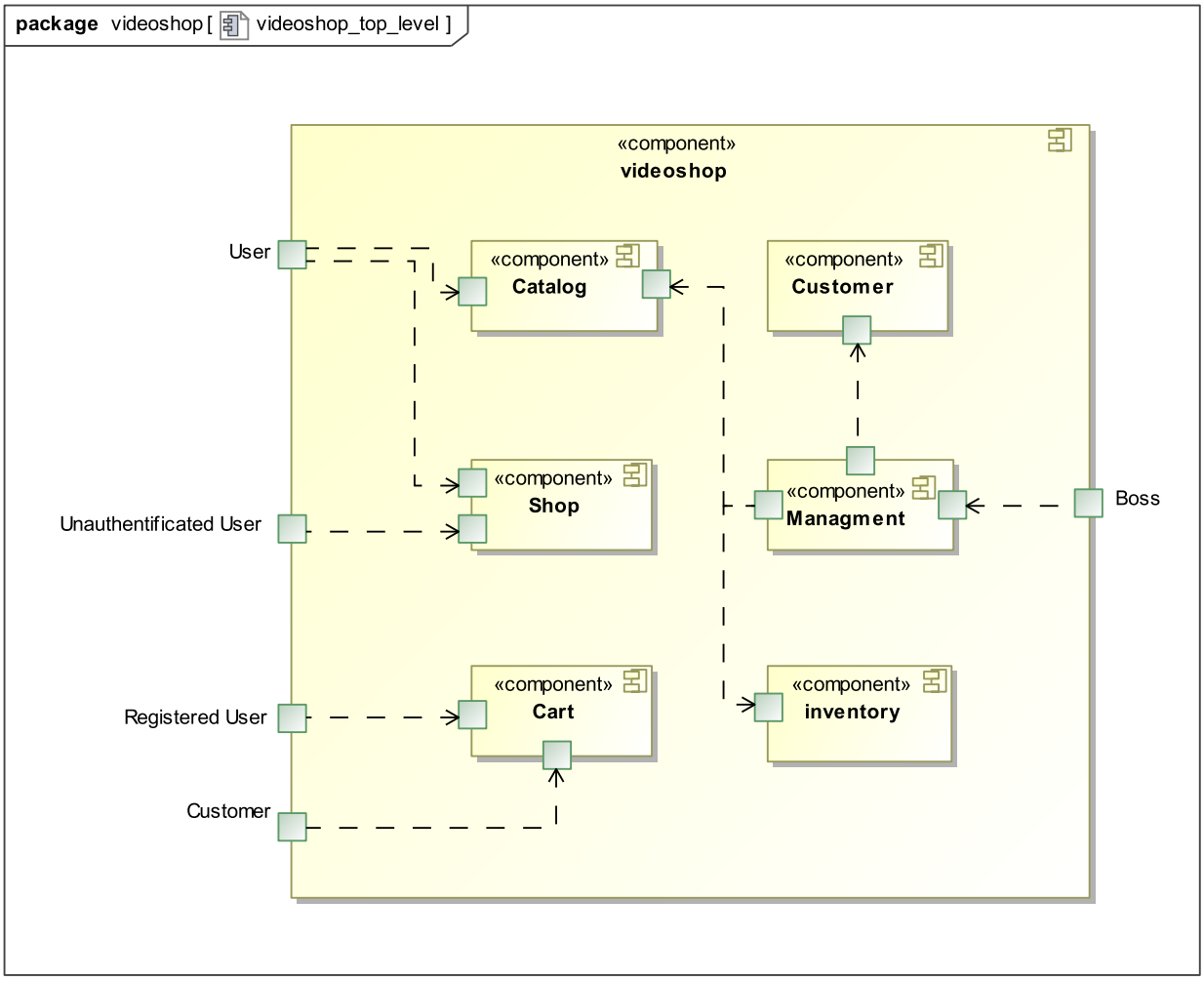
The system context diagram shows the planned system in its environment. This includes all user types, their ways to access the system, as well as third-party systems, which access our system or are accessed by it (not the case here).

*Note: Informal graphics are usable as well (e.g. created with Visio).*



## 5.2. Top-level architecture

Top-Level view of the system.




# 6. Use-Cases

This section will give an overview of the use cases the system has to support. These use cases describe what functionality the system has to provide (mostly) from the client's point of view and which actors are involved.

## 6.1. Actors

Actors are users of the system or neighboring systems who/which access it. The following table summarizes all actors of the system and provides a description of the actor. Abstract actors (i.e. an actor which groups other actors, written in *italic*) are used to generalize and group.

| Name | Description |
| --- | --- |
| *User* | Representative for every person, who interacts who interacts with the system, regardless if authenticated or not. |
| *Registered User / Authenticated User* | Representative for every person, who does have an account, is authenticated and interacts with the system. |
| Unauthenticated User | Representative for unauthenticated access (i.e. unauthenticated visitors) |
| Boss | Any registered (and authenticated) user, who has the Role "BOSS". Is responsible for administration of the application. |
| Customer | Any registered (and authenticated) user, who has the Role "CUSTOMER". Only role in the system, which is allowed to buy the content of the cart. |

## 6.2. Use-Case Diagram

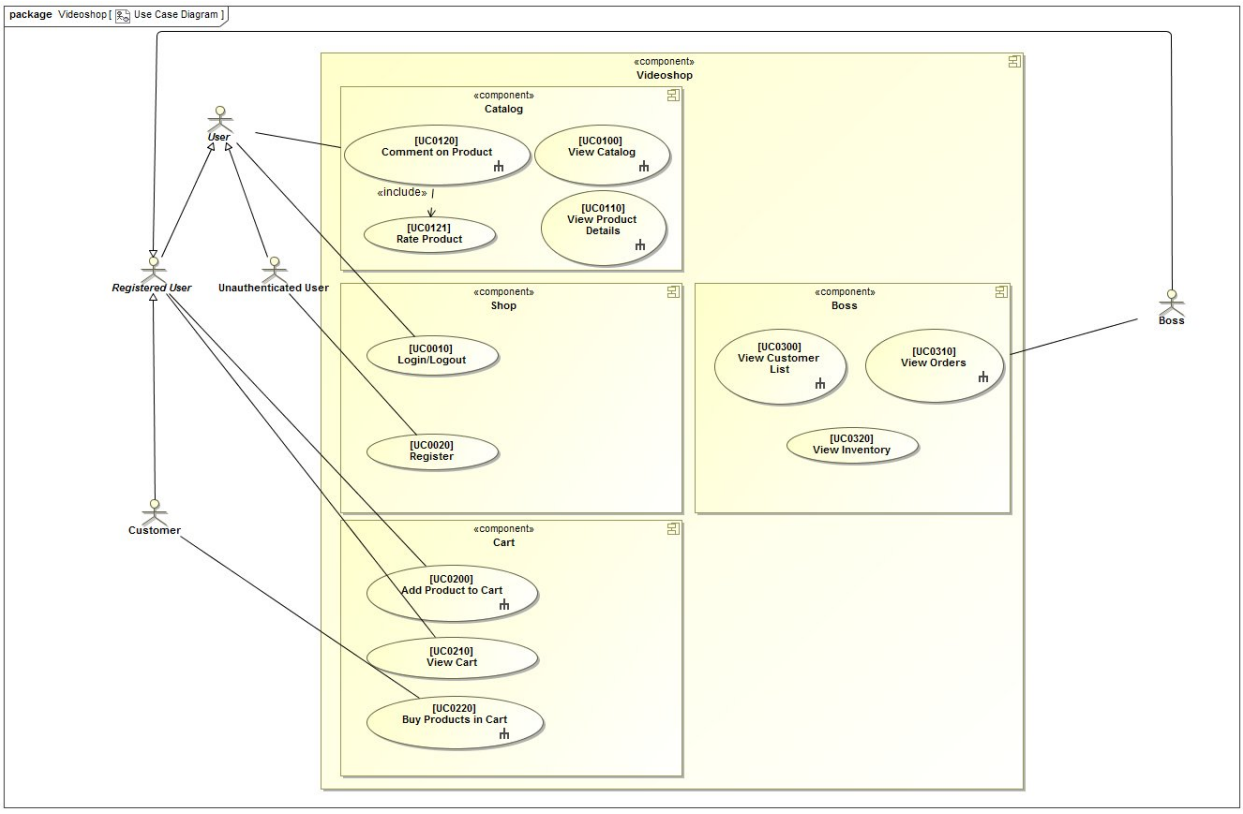



Figure 1. Use case diagram of Videoshop

## 6.3. Use-Case Descriptions

This section describes the use cases shown in the use case diagram in detail.

*Note: It is not yet necessary to fully include all special cases and variants (scenarios) of the use case (e.g. what happens if the stock is not sufficient), but the general purpose of the system should become visible. Typical CRUD (create, read, update, delete) use cases can be condensed into one use case.*

*Note: We did not provide a sequence diagram for every use case. In general, especially complex use cases should be shown in detail with a sequence diagram. Simple use cases should be described in the text only.*

*Note: See the following Link for examples of use case descriptions: https://www.sophist.de/fileadmin/user_upload/Bilder_zu_Seiten/Publikationen/UML2_glasklar/4._Auflage/12-1_Schablone_fuer__Use-Case-Beschreibung.pdf*

| | |
|---|---|
| **ID** | [UC0010] |
| **Name** | Login/Logout |
| **Description** | A user shall be able to login (authenticate) with the system to access further functionality. This process shall be reversible by logging out. |
| **Actors** | User |
| **Trigger** | *Login*: User wants to access "hidden" functionality by logging in.<br><br>*Logout*: User wants to leave the shop. |
| **Precondition(s)** | *Login*: User is not authenticated yet<br><br>*Logout*: User is authenticated |
| **Essential Steps** | *Login*:<br><br>1. User accesses "Einloggen" in the navigation bar<br><br>2. User enters his credentials<br><br>3. User hits "Log in" button<br><br>*Logout*: |

| | |
|---|---|
| | 1. User hits "Ausloggen" in the navigation bar<br>2. User is unauthenticated and is shown the home screen |
| **Extensions** | - |
| **Functional Requirements** | [F0010] |

| | |
|---|---|
| **ID** | **[UC0020]** |
| **Name** | Register |
| **Description** | An unauthenticated user shall be able to create an account for himself. |
| **Actors** | Unauthenticated User |
| **Trigger** | Unauthenticated user wants to create an account for himself by pressing "Registrieren" |
| **Precondition(s)** | Actor is not logged in (authenticated) yet |
| **Essential Steps** | 1. Unauthenticated user presses "Registrieren"<br>2. He enters his desired username, password, and delivery address<br>3. System checks username uniqueness<br>    1. If Unique: An account is created with the provided data<br>    2. Otherwise: An error message is shown |
| **Extensions** | - |
| **Functional Requirements** | [F0020], [F0021] |

| | |
|---|---|
| **ID** | [UC0100] |
| **Name** | **View Catalog** |
| **Description** | Every visitor of the Videoshop (i.e. **User**) shall be able to access the Catalog, which displays all the offered discs. The Catalog must provide the possibility to distinguish between different types of Discs (Dvd, Blu-Ray). |
| **Actors** | User |
| **Trigger** | Accessing the navigation element, which is responsible for displaying the Catalog. |
| **Precondition(s)** | None |
| **Essential Steps** | 1. User clicks on the navigation element named "DVD Katalog" or "BluRay Katalog".<br>2. User is shown all Discs of the selected category. |
| **Extensions** | None |
| **Functional Requirements** | [F0100], [F0110], [F0111], [F0112] |

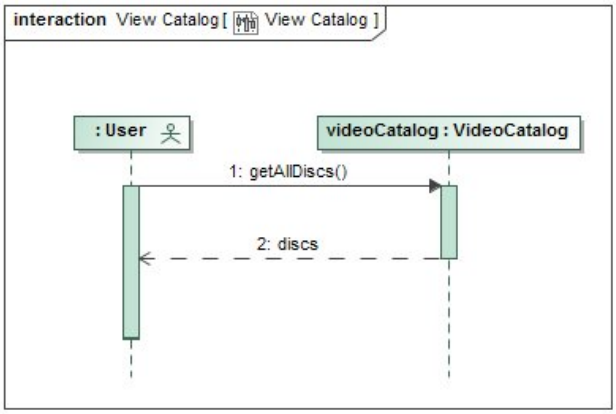



Figure 2. Sequence diagram: View Catalog

| | |
|---|---|
| **ID** | [UC0110] |

| | |
|---|---|
| **Name** | View Product Details |
| **Description** | A user shall be able to view the details of a disc on an extra page. |
| **Actors** | User |
| **Trigger** | User views the catalog and presses on an entry to view the details of the disc. |
| **Precondition(s)** | User is viewing the catalog. |
| **Essential Steps** | 1. User presses on a displayed entry of the catalog (disc)<br>2. User is shown the details of the selected disc. |
| **Extensions** | - |
| **Functional Requirements** | [F0120] |

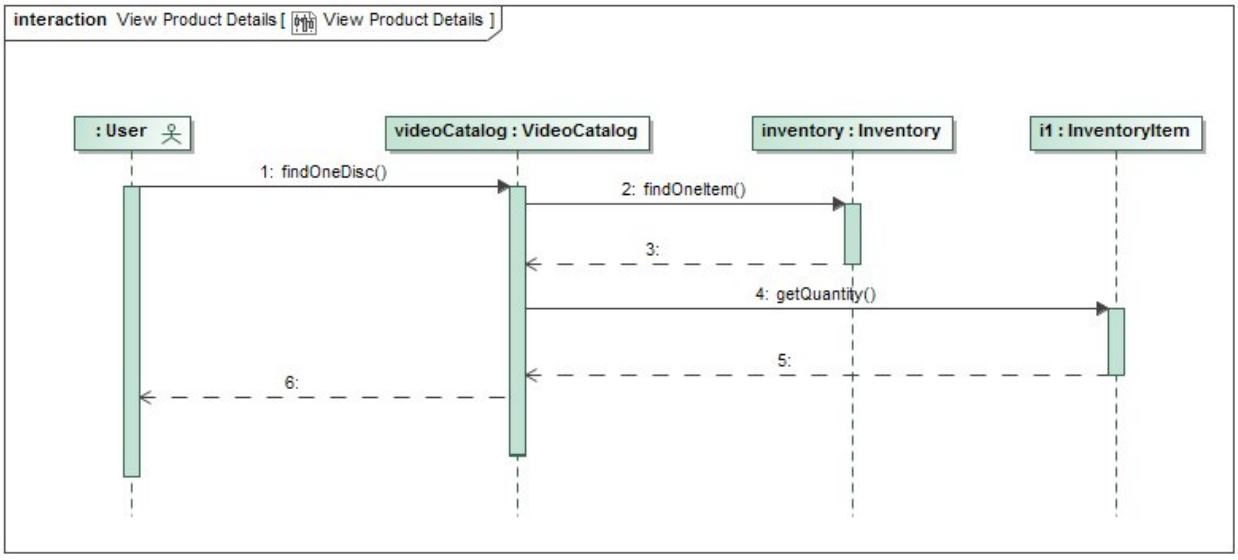



Figure 3. Sequence diagram: View Product Details

| | |
|---|---|
| **ID** | **[UC0120]** |
| **Name** | Comment on Product |

| | |
|---|---|
| **Description** | A user shall be able to leave his opinion about a disc, visible to all other users. |
| **Actors** | User |
| **Trigger** | User wants to comment on a disc |
| **Precondition(s)** | User views the details page ([UC0110]) of a disc. |
| **Essential Steps** | 1. User enters his textual comment on the details page of a disc<br>2. User presses "Senden" to persist his comment<br>3. Persisted comment is listed on the details page of the disc |
| **Extensions** | • Only authenticated users shall be able to leave a comment<br>• Only authenticated users, who bought this disc, shall be able to comment it |
| **Functional Requirements** | [F0121] |

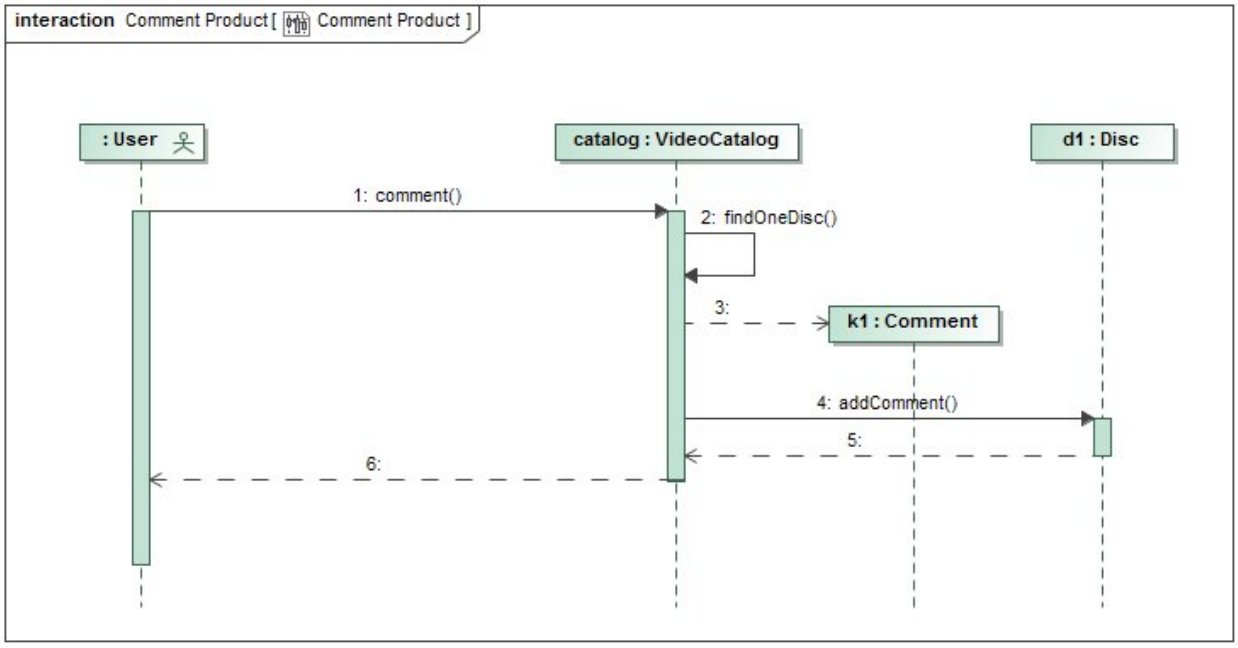


Figure 4. Sequence diagram: Comment on Product

| ID | [UC0121] |
|---|---|
| Name | Rate Product |
| Description | A user shall be able to support his comment with a rating.<br><br>*Please Note: As it is implemented, this is not a use case in itself, as the rating is part of use case [UC0120]. We decided to model it this way to show an example of the "include" in a use case diagram. The meaning would be: During the process of [UC0120], [UC0121] is executed mandatorily (if you decide to leave a comment, it is also necessary to leave a rating).* |
| Actors | User |
| Trigger | User is about to comment on a disc |
| Precondition(s) | User views the details page ([UC0110]) of a disc and is about to leave a comment ([UC0120]) |
| Essential Steps | Actor enters a numerical rating besides the comment |
| Extensions | - |
| Functional Requirements | [F0121] |

| ID | [UC0200] |
|---|---|
| Name | Add Product to Cart |
| Description | A registered user shall be able to add a disc of a chosen quantity to his cart. |
| Actors | Registered User |
| Trigger | A registered user views the details page of a disc and wants to enter it to his cart. |

| | |
|---|---|
| **Precondition(s)** | • Actor has authenticated with the system (i.e. is a registered user)<br>• Actor views the details page of a disc |
| **Essential Steps** | 1. Actor enters a desired quantity for the selected disc (1..5)<br>2. Actor presses "zum Warenkorb hinzufügen"<br>3. Disc is added to his cart with the selected quantity |
| **Extensions** | - |
| **Functional Requirements** | [F0200], [F0201] |

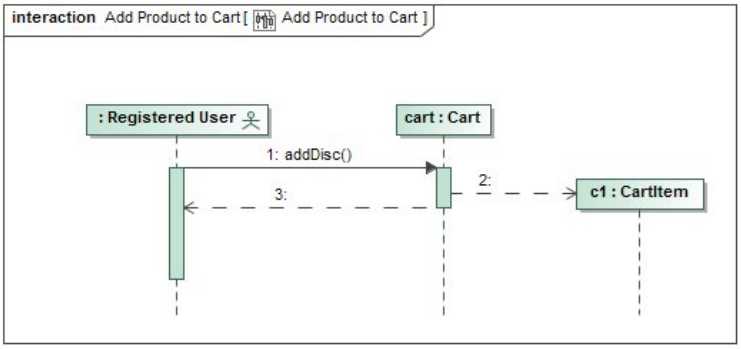



Figure 5. Sequence diagram: Add Product to Cart

| | |
|---|---|
| **ID** | **[UC0210]** |
| **Name** | View Cart |
| **Description** | A registered user shall be able to view the contents of his cart and the total price of his choice. |
| **Actors** | Registered User |
| **Trigger** | Actor presses "Warenkorb" in the navigation bar |
| **Precondition(s)** | • Actor has authenticated with the system (i.e. is a registered user) |

| | |
|---|---|
| **Essential Steps** | 1. Actor presses "Warenkorb" in the navigation bar<br>2. Actor is shown the content of his cart as well as he total price of it |
| **Extensions** | - |
| **Functional Requirements** | [F0210] |

| | |
|---|---|
| **ID** | **[UC0220]** |
| **Name** | Buy Products in Cart |
| **Description** | A customer shall be able to buy the content of his cart. |
| **Actors** | Customer |
| **Trigger** | Customer |
| **Precondition(s)** | • Actor is authenticated and has the role "CUSTOMER" in the system<br>• Cart is not empty |
| **Essential Steps** | 1. (Customer has put at least one item into his cart ([UC0200]))<br>2. Customer presses "Buy"<br>3. Order is checked against stock<br>4. Order is paid automatically<br>5. Discs are removed from the inventory in the chosen quantity<br>6. Order is archived |

| | |
|---|---|
| **Extensions** | |
| **Functional Requirements** | [F0101], [F0220], [F0230], [F0240], [F0241], [F0242], [F0243] |

*Note: Such a complex use case as UC0220 does definitely need to be shown in detail with a sequence diagram. We opted out of showing more diagrams to reduce the size of this document.*

| | |
|---|---|
| **ID** | **[UC0300]** |
| **Name** | View Customer List |
| **Description** | A Boss should be able to view the whole list of customers of the application. |
| **Actors** | Boss |
| **Trigger** | Boss selects "Kunden" in the navigation bar |
| **Precondition(s)** | User is authenticated and has role "Boss" |
| **Essential Steps** | 1. Boss selects "Kunden" in the navigation bar<br>2. Complete list of all registered users with the role "customer" is shown |
| **Extensions** | - |
| **Functional Requirements** | [F0300] |

| | |
|---|---|
| **ID** | **[UC0310]** |
| **Name** | View Orders |
| **Description** | A boss shall be able to view a list of completed orders. |

| | |
|---|---|
| **Actors** | Boss |
| **Trigger** | Boss selects "Bestellungen" in the navigation bar |
| **Precondition(s)** | User is authenticated and has role "Boss" |
| **Essential Steps** | 1. Boss selects "Bestellungen" in the navigation bar<br>2. Complete list of all completed orders is shown |
| **Extensions** | - |
| **Functional Requirements** | [F0310] |

| | |
|---|---|
| **ID** | **[UC0320]** |
| **Name** | View Inventory |
| **Description** | A boss shall be able to view the inventory including the current stock. |
| **Actors** | Boss |
| **Trigger** | Boss selects "Lager" in the navigation bar |
| **Precondition(s)** | User is authenticated and has role "Boss" |
| **Essential Steps** | 1. Boss selects "Lager" in the navigation bar<br>2. Complete list of all items of the inventory and the current stock is shown |
| **Extensions** | - |
| **Functional Requirements** | [F0100], [F0320] |

## 7. Functional Requirements

This section gives an overview of the functional requirements of the system.

The table contains:

- A unique identifier of the requirement (ID), which can be used for referencing throughout the project
- The current version of the requirement, as changes to a requirement can happen throughout the project
- A short name of the requirement
- The description of the requirement

*Note: A functional requirement defines a function of the system, which shall be implemented to satisfy the customer needs (e.g. as shown through use cases). Ideally, it contains a set of inputs for the functionality in question, the intended behavior, and the result of it.*

*Note: Functional requirements are used to depict what exactly has to be implemented (from the developer's point of view). As use cases are mostly relatively close to the domain and mostly non-technical (can even be written by a non-techie client), it is necessary to specify and organize the information provided by the client.*

*See (German): https://www.sophist.de/fileadmin/user_upload/Bilder_zu_Seiten/Publikationen/Wissen_for_free/MASTeR_Broschuere_3-Auflage_interaktiv.pdf*

| ID | Version | Name | Description |
|---|---|---|---|
| [F0010] | v0.1 | Authentication | The system shall be able to be separated into publicly accessible parts, and parts which require authentication to be accessed. If a User is existent in the system (registered user), he or she shall be able to authenticate by providing the following information:<br>• Username<br>• Password |
| | | | The system shall provide an |

| ID | Version | Name | Description |
| --- | --- | --- | --- |
| [F0020] | v0.1 | Registration | Unauthenticated User ([F0010]) the ability to register after accessing the navigation element named "Registrieren".<br><br>The following information has to be provided:<br><br>• Username (unique)<br>• Password<br>• Shipping address<br><br>The system shall validate the provided data ([F0021]). The user shall be registered in the system as customer and he shall be able to authenticate ([F0010]) after successful validation. |
| [F0021] | v0.1 | Validate Registration | The system shall be able to validate the provided data of an unregistered user.<br><br>The uniqueness of the username has to be guaranteed. The user shall be informed of any constraint violations. |
| [F0100] | v0.1 | Inventory | The system shall be able to persistently store data about Discs in an Inventory. |
| [F0101] | v0.1 | Reduce Quantity | The system shall be able to reduce the stock of a product in the inventory. |
| [F0110] | v0.1 | Catalog | The system shall be able to provide read-only access on existing Discs ([F0100]) through a Catalog. |
| [F0111] | v0.1 | View Catalog | The system shall provide a User the ability view the contents of the Catalog. |
| [F0112] | v0.1 | Filter catalog | The system shall provide a user the ability to view discs in the catalog filtered by a |

| ID | Version | Name | Description |
| --- | --- | --- | --- |
| | | | chosen category (i.e. Dvd or BluRay) |
| [F0120] | v0.1 | View Product Details | The system shall provide a user the ability to view the details of a Disc after clicking on it.<br>The following details have to be displayed: - Title of the disc - Price of the disc - Genre of the disc - Current stock - Cover image of the disc - Submitted comments |
| [F0121] | v0.1 | Comment on Product | The system shall provide a user the ability to submit a comment for a product.<br><br>A comment consists of: - A textual opinion regarding the disc - A numerical rating for the disc (low = bad rating, high = good rating)<br><br>*Note: As we have explained in the respective use case, the comment functionality essentially includes the rating. While the client might have described these functions as two potentially different use cases, further domain analysis has led to the conclusion, that we can combine them, as happened with this functional requirement* |
| [F0200] | v0.1 | Cart | The system shall provide every registered and authenticated user with a cart, in which he can temporarily store selected products.<br><br>The cart shall be transiently persistent and be unique to every user. |
| | | | The system shall provide a registered and authenticated user to add a product to his cart in the desired quantity. |

| ID | Version | Name | Description |
| --- | --- | --- | --- |
| [F0201] | v0.1 | Add Product to Cart | Upon adding a product, an entry shall be created in the cart of the authenticated user.<br><br>Unauthenticated users shall be prompted to authenticate to view their cart. |
| [F0210] | v0.1 | View Cart | The system shall provide an authenticated user the ability to access his cart. The cart shall list the following:<br><br>• Movie title<br>• Selected Quantity<br>• Total price for each movie (movie price x movie quantity)<br>• Total price of the cart |
| [F0220] | v0.1 | Buy Products in Cart | The system shall provide an authenticated user the ability to buy the content of his cart.<br><br>Upon attempting to buy the content of the cart, the potential order has to be validated ([F0230]). An order shall be created, if the stock is sufficient ([F0241]). |
| [F0230] | v0.1 | Validate Sufficient Stock | The system shall be able to validate if the current stock of a product matches at least a desired quantity. |
| [F0240] | v0.1 | Orders | The system shall be able to persistently store orders. |
| [F0241] | v0.1 | Create Order | The system shall be able to create an order from the contents of a cart.<br><br>An order shall be initialized with the status "OPEN". |

| ID | Version | Name | Description |
| --- | --- | --- | --- |
| [F0242] | v0.1 | Pay Order | The system shall provide the functionality to pay an existing "OPEN" order with different payment methods.<br><br>After the order was paid, its status shall be set to "PAID". |
| [F0243] | v0.1 | Archive Order | The system shall be able to archive an order.<br><br>An order is archived by setting its status to "COMPLETED". |
| [F0300] | v0.1 | View Customer List | The system shall provide a boss the functionality to view all customers who are registered in the system. |
| [F0310] | v0.1 | View Orders | The system shall provide a boss the functionality to view all orders with the status "COMPLETED".<br><br>The following information shall be shown for each order: - Timestamp of creation - Customer who issued the order - Total paid price of the order |
| [F0320] | v0.1 | View Inventory | The system shall provide a boss the functionality to view the inventory and the current stock.<br><br>The following information shall be shown for each product:<br><br>• Name of the disc<br><br>• Current stock (quantity) |

## 8. Non-Functional Requirements

This section is going to give an overview of non-functional (NF) requirements of the

project Videoshop. These requirements describe how the system works and within which boundaries it is supposed to perform.

*Note: We only picked two small examples of requirements to show which aspects could be considered in this chapter.*

## 8.1. Quality Demands

The following table shows what quality demands have to be fulfilled to which extent. The first column lists the quality demands, while in the following columns an "x" is used to mark the priority. The assigned priority has to be considered in the formulation of the concrete non-functional requirements.

*Note: This is only an abstract example which is derived from the current version of the Videoshop. The priority may vary drastically depending on the project, and even many more aspects could be considered. Additionally, you should provide explanations for the demands, as to avoid any misunderstandings.*

1 = Not Important .. 5 = Very Important

| Quality Demand | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| **Maintainability** | | | | x | |
| **Usability** | | | x | | |
| **Security** | | | | x | |

*Note: It might be necessary to provide a description of the above quality demands, as they are mostly ambiguous or the meaning is unclear.*

## 8.2. Concrete NF Requirements

| ID | Version | Name | Description |
|---|---|---|---|
| [NF0010] | v0.1 | Availability - uptime | The system shall achieve at least **99,5%** uptime. |
| [NF0020] | v0.1 | Security - Password storage | Passwords of Users shall only be stored as hash-values to prevent theft. |

# 9. GUI Prototype

The following pictures show what the GUI of the system could look like.

*Note: The prototype is supposed to give the client an understanding of how the contractor intends to implement and design the solution. The more details you can already finalize, the better, but generally a more abstract design is sufficient at this point (depending on the client and project, even a dialog roadmap is sufficient). A better structure than in this example can also be benefitial in case the GUI or the navigation is more complex. It it not necessary to include every single desired page in the prototype, just the crucial functionalities/pages, as discussed with the client.*

*Note: For a simple prototype, you can use sketching or wireframing. The first few prototype images are using a simple design to show the client the structure and the general feeling of the GUI.*


Figure 6. Landing page of Videoshop

Figure 7. Registration page of Videoshop


Figure 8. Landing page for an authenticated customer of Videoshop

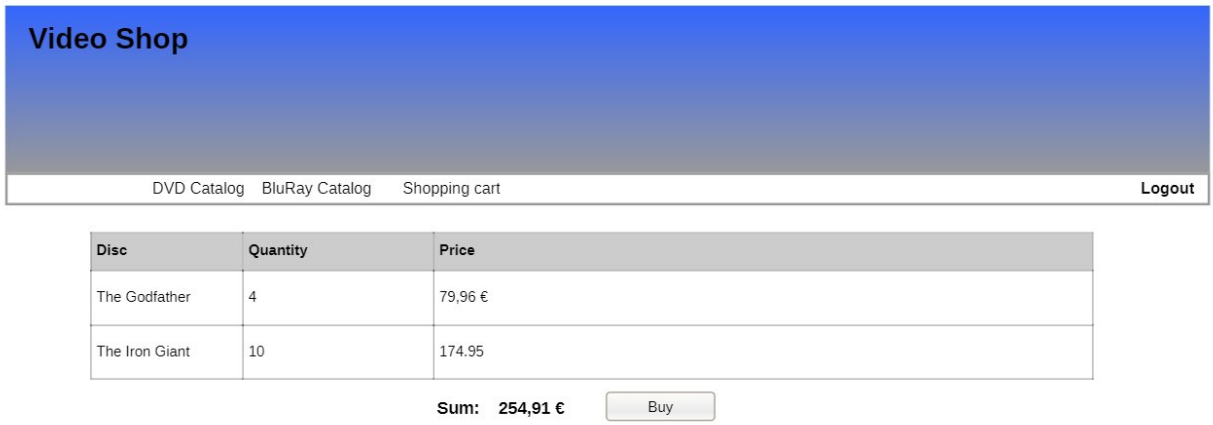
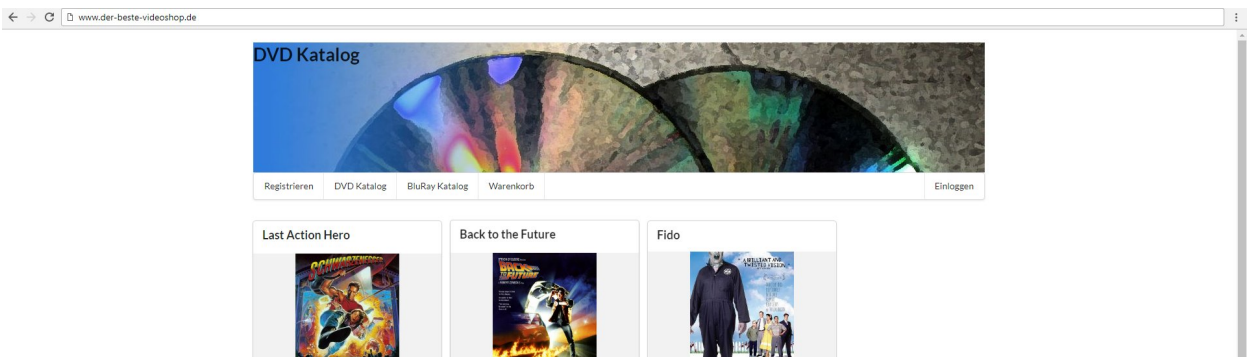



Figure 9. Cart overview page for a customer of Videoshop

*Note: If you want to reuse your prototype later in the project, you can also create an HTML Prototyp. This type of prototype is used in the following images. Please be aware that this type of prototype will initially take more time than simple sketching or wireframing.*

*Note: It is pretty astonishing how close the prototype in this example already is to the final design, isn't it? ;)*



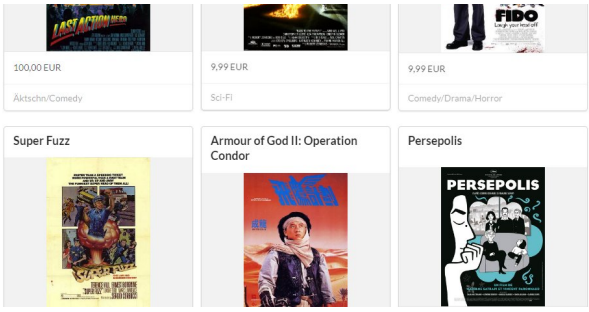



Figure 10. DVD catalog of Videoshop

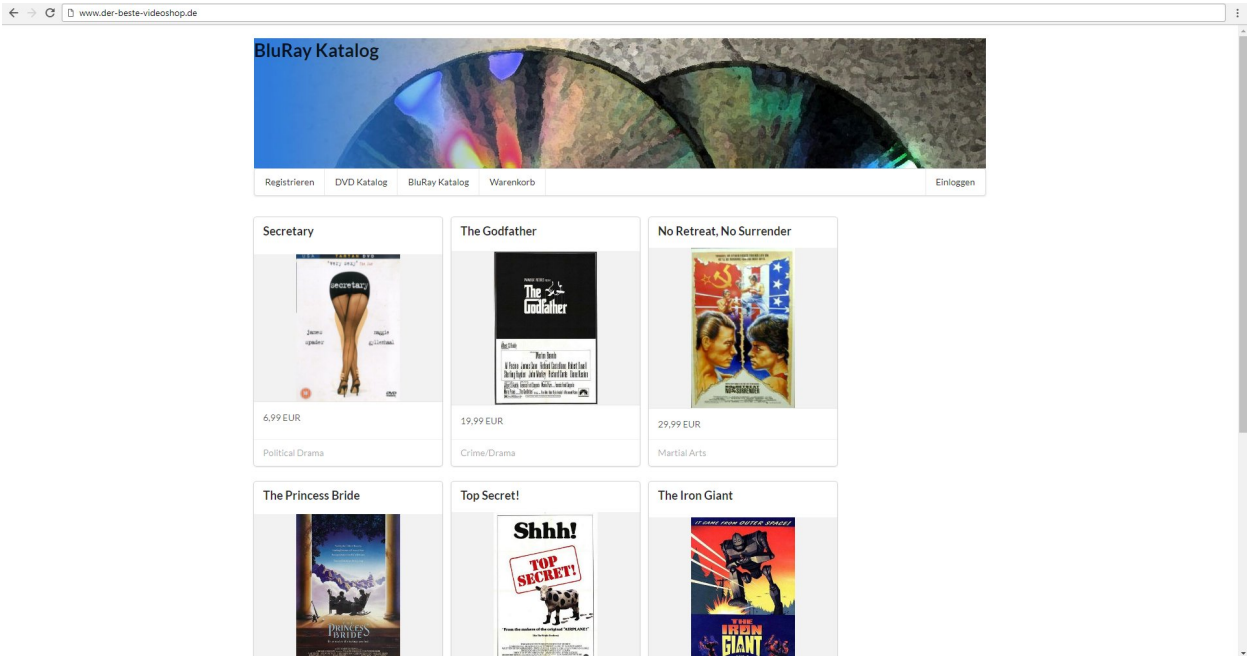



Figure 11. Blu Ray catalog of Videoshop

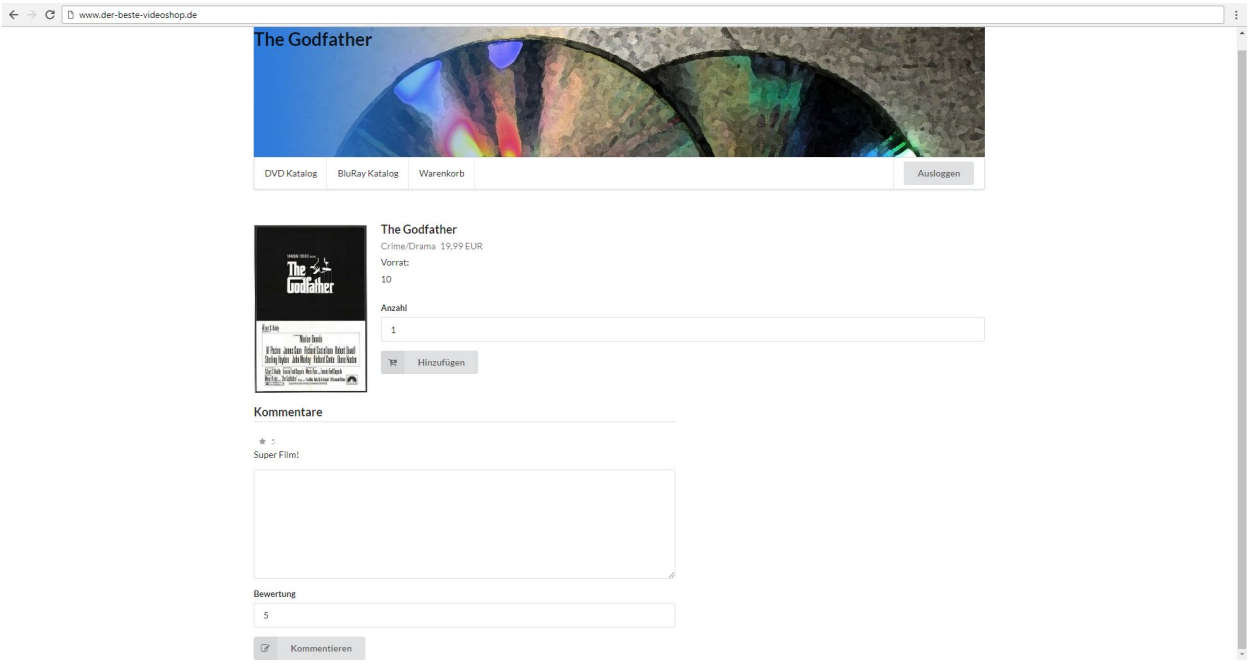
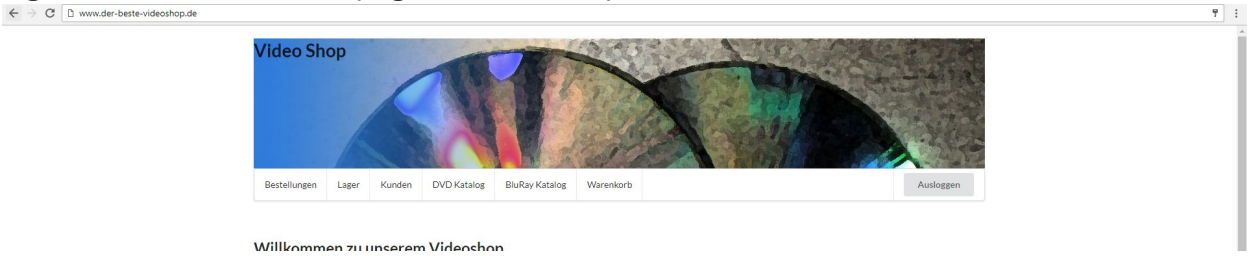



Figure 12. Product detail page of Videoshop



Hallo boss

Figure 13. Landing page for an authenticated boss of Videoshop

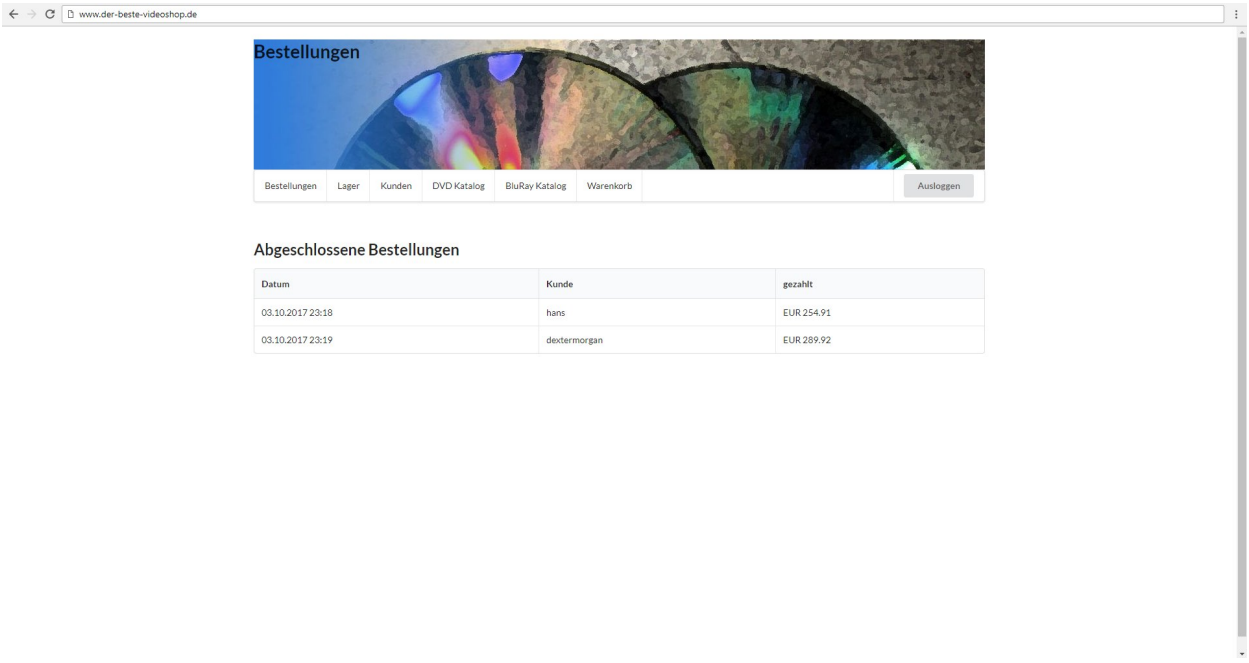



Figure 14. Overview page of all completed orders of Videoshop

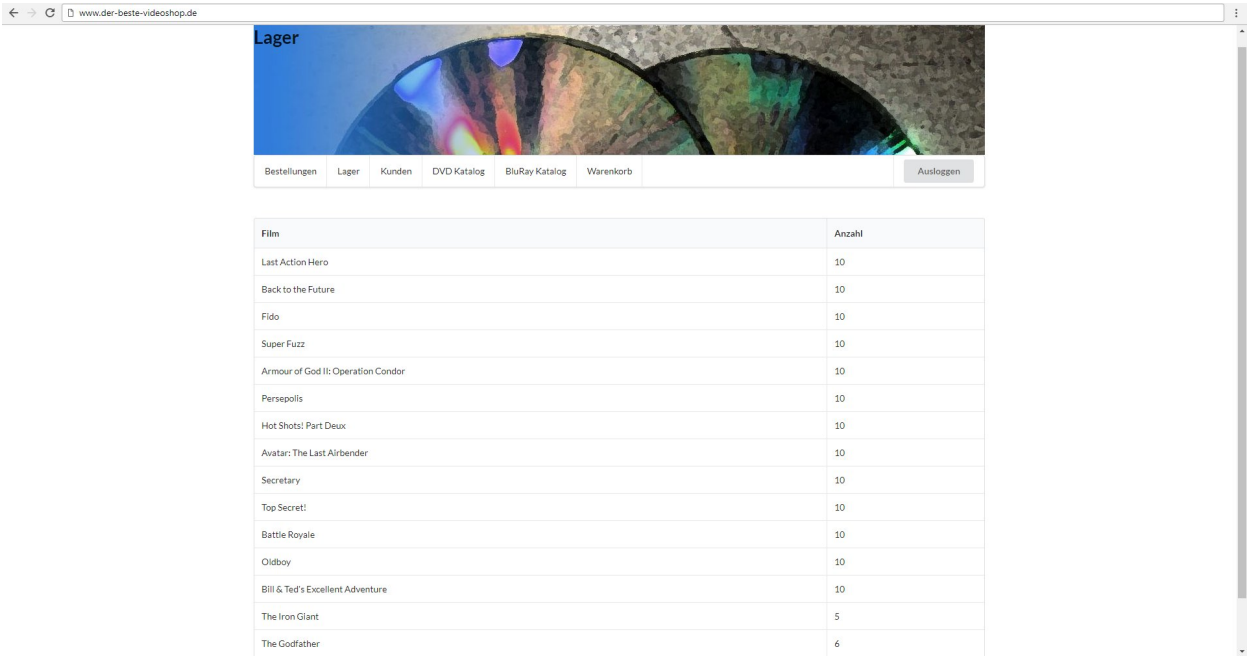
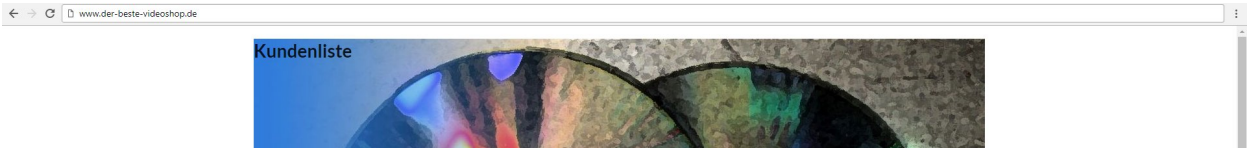



Figure 15. Inventory overview page of Videoshop



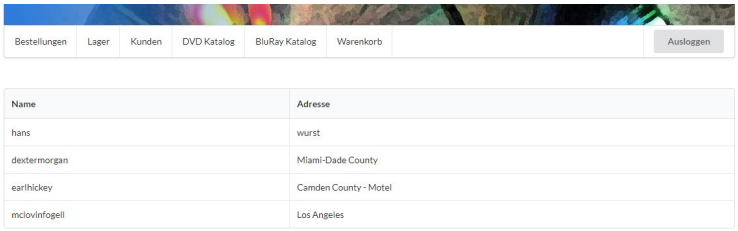



Figure 16. Customer overview page of Videoshop

# 10. Data Model

## 10.1. Class Diagram

The (analysis) class diagram is supposed to give an overview of the domain in the context of the system, which shall be developed in the scope of this project.

*Note: The domain model is supposed to explain the concepts and terms of the domain and their relationships. Please, try to avoid technical terms or implementation knowledge.*

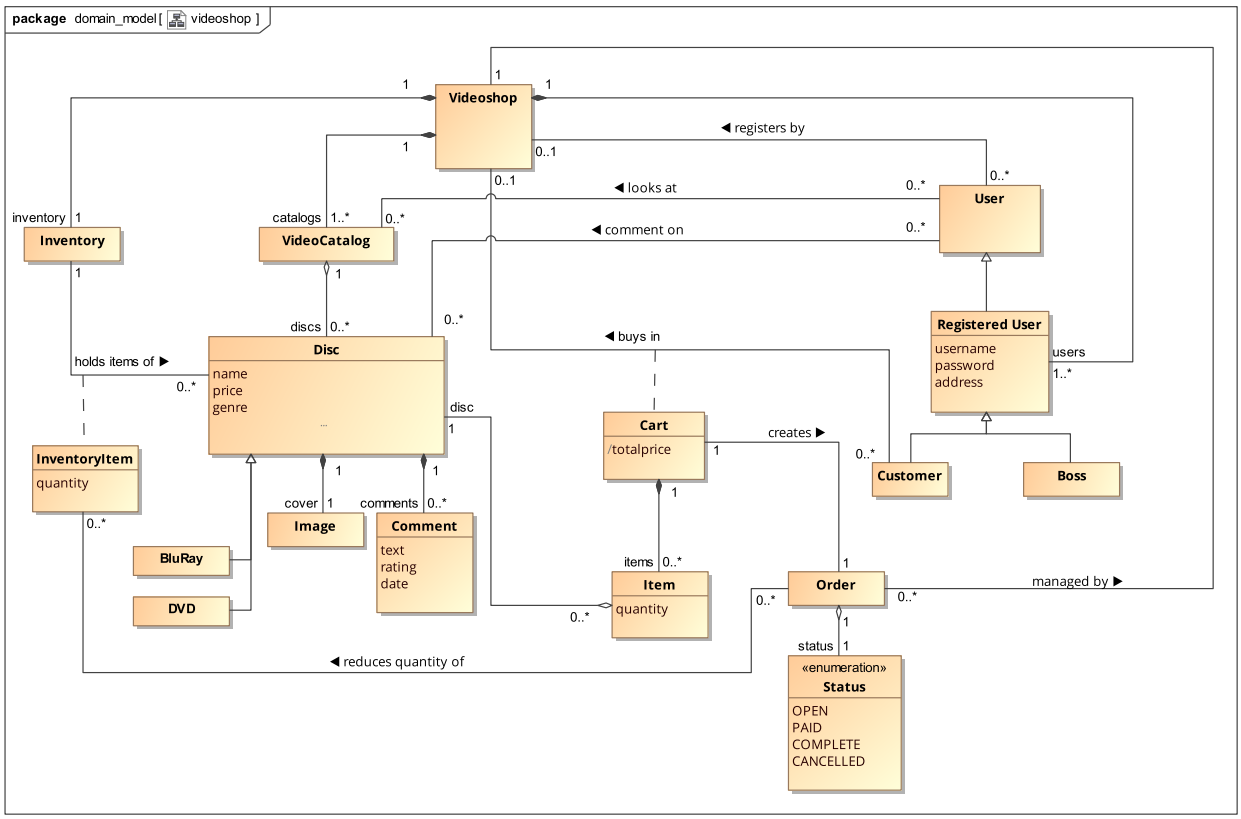



Figure 17. Domain model of Videoshop

## 10.2. Classes and Enumerations

The following table gives an overview of the classes/enumerations used in the domain model. Therefore, this section is a subset of the glossary and shall be used to provide every stakeholder a common understanding of central terms and concepts of the domain of the system.

| Class/Enumeration | Description |
|---|---|
| **Videoshop** | Central class of the system representing the videoshop itself. |
| **User** | General representation of a real person. |
| **Registered User** | General representation of a real person, which has a representation in the system. This representation is only created if a user registers with the system, and only used if he or she authenticates. |
| **Customer** | A user that is registered as a customer of the videoshop. |
| **Boss** | A user that is registered as an administrator/boss of the videoshop. |
| **VideoCatalog** | A video catalog is a group of discs with a common feature. |
| **Disc** | General product of the Videoshop. |
| **BluRay** | A BluRay is a possible type of a disc, used to group discs in the catalog. |
| **DVD** | A DVD is a possible type of a disc, used to group discs in the catalog. |
| **Comment** | A Comment is a textual remark/opinion of a user regarding a disc and includes a rating (see amazon comments/ratings). |
| **Image** | A image is a digital picture that visually represent a disc. |
| **Cart** | A Cart is a temporary storage for discs a customer intends to buy. If a customer decides to buy his selected discs, an order |

| Class/Enumeration | Description |
| --- | --- |
| | is created. |
| **Item** | Represents the quantity of a disc in the cart. |
| **Order** | An order is used to represent what a customer of the shop intends to buy from the shop. An Order can have a varying status to symbolize the current step at which the order is processed. |
| **Status** | This status represents the current processing step of the order it belongs to.<br>*OPEN*: The Order has been created by the system, but not yet processed.<br>*PAID*: The customer has paid his order, the disc quantity still has to be reduced in the inventory to finalize the order.<br>*COMPLETE*: The order was been paid and shipped to the customer.<br>*CANCELLED*: Fallback to allow to mark failed orders or other problems. |
| **Inventory** | An Inventory represents a storage for the videoshop. Can be seen like a warehouse in this project. |
| **InventoryItem** | Represents the quantity of a disc in the inventory. |

## 11. Acceptance Testing

Acceptance tests are used to determine, whether or not the delivered software system fulfills the requirements of the client during the actual usage. The following table shows which acceptance tests the software system does have to pass at the end of the project in order to satisfy the client and complete the contract (regarding the requirements).

*Note: Acceptance tests can be derived from the use cases and the respective sequence diagrams, but also from other parts of the SRS. Each sequence diagram represents one scenario of a use case (e.g. successful order completion). However, another scenario of the same use case (e.g. failed order because of insufficient stock) would require an own sequence diagram as well as at least an own acceptance test. It is also highly necessary to design the test cases in a measurable manner to be able to determine if the acceptance test has passed or not..*

*Note: There are multiple different types of acceptance tests. In this course, we mainly focus on documenting test cases, which show that the functional requirements are fulfilled from the perspective of the user (UAT).*

| | |
|---|---|
| **ID** | [AT0010] |
| **Use Case** | [UC0010] |
| **Precondition(s)** | The system has existing users. |
| **Event** | An unauthenticated user accesses the login screen, enters the credentials of an existing user of the system (hans, 123) and presses "Login" |
| **Expected Result** | • The user is now authenticated as "hans"<br>• The user is redirected to a welcome screen, which displays a personalized welcome message<br>• The user has now access to every functionality, which are accessible to users with the role "Customer" |

| | |
|---|---|
| **ID** | [AT0011] |
| **Use Case** | [UC0010] |
| **Precondition(s)** | An authenticated user is using the system |
| **Event** | The authenticated user presses "Ausloggen" |
| **Expected Result** | • He becomes unauthenticated<br>• He loses all access to functionality only open to authenticated users or certain roles |

| | |
|---|---|
| **ID** | [AT0020] |

| | |
|---|---|
| **Use Case** | [UC0020] |
| **Precondition(s)** | An unauthenticated user is using the system |
| **Event** | The unauthenticated user presses "Registrieren" in the navigation bar and enters the following information:<br>• *Name:* TestCustomer<br>• *Passwort:* 123<br>• *Adresse:* Nöthnitzer Straße 46<br>Finally, he presses "Registrieren" to send the information. |
| **Expected Result** | • An new Customer with the provided data is created<br>• It is possible to authenticate with the credentials of the created customer<br>• The unauthenticated user is still unauthenticated and redirected to the landing page of the Videoshop |

| | |
|---|---|
| **ID** | [AT0021] |
| **Use Case** | [UC0020] |
| **Precondition(s)** | An unauthenticated user is using the system |
| **Event** | The unauthenticated user presses "Registrieren" in the navigation bar and enters the following information:<br>• *Name:* hans<br>• *Passwort:* 123<br>• *Adresse:* Nöthnitzer Straße 46<br>Finally, he presses "Registrieren" to send the information. |
| **Expected Result** | • An error message is shown to inform the user about the problem (user already exists) |

| | |
|---|---|
| **ID** | [AT0100] |
| **Use Case** | [UC0100] |
| **Precondition(s)** | A user is using the system |
| **Event** | The user presses "DVD Katalog" in the navigation bar |
| **Expected Result** | The user is shown an overview of all existing discs that are DVDs (8 different discs) |

| | |
|---|---|
| **ID** | [AT0101] |
| **Use Case** | [UC0100] |
| **Precondition(s)** | A user is using the system |
| **Event** | The user presses "BluRay Katalog" in the navigation bar |
| **Expected Result** | The user is shown an overview of all existing discs that are BluRays (9 different discs) |

| | |
|---|---|
| **ID** | [AT0110] |
| **Use Case** | [UC0110] |
| **Precondition(s)** | A user is using the system and is either viewing the DVD catalog ([AT0100]) or the BluRay catalog ([AT0101]). |
| **Event** | The user presses on one of the shown discs of the catalog. |
| **Expected Result** | The user is shown (on a new page) the details about the disc he selected as specified in [F0120].<br><br>*Note: You could arguably describe the process in more detail, with concrete values (e.g. user selects disc named "Secretary" from the* |

| | |
|---|---|
| | *BluRay catalog, ...)* |

*Note: This list of acceptance tests does obviously not cover every use case. The process is mostly the same for every acceptance test case, which is why we provide only some examples to show you the ropes.*

*Note: It is often also necessary to create test cases for non-functional requirements in order to prove that the requirement has been fulfilled by the finished system.*

## 12. Glossary

The glossary contains a list of all words and phrases used in this project, which require an description to avoid misunderstandings between stakeholders. Please also consult the list of actors, the list of stakeholders and the domain model for further definitions of terms.

*Note: Some terms can be used regularily during a project, while all involved stakeholders think that the meaning is obvious. This not necessarily the case though, as different domains of expertise can mean different levels of knowledge or simply a different understanding of a term.*
*An example from a previous year of this course:*
*Imagine a shift schedule, where every shift is occupied by 3 different kinds of staff. The manager responsible for the schedule would use the term "shift" to describe the whole timeslot with all 3 involved staff members (e.g. "shift X is gonna be hard for you guys, prepare yourselves"). One of the staff members occupying one of these slots would use the term "shift" to describe his one slot (of the three) in one timeslot of the day (e.g. "My shift this time puts me in touch with the customers, while the other two can relax in the warehouse").*
*While this is common sense and does not really affect communication in the real world, it becomes an issue if you have to design a system which represents such a shift schedule. You could - in this case - use "shift" as in the understanding of the manager and use "slot" or "cell" to model what the staff member meant. In such cases, you have to force all stakeholders to use this common wording in order to avoid misunderstandings.*

| Term | Description |
|---|---|
| **Administrator** | Synonym for a Boss |
| **BluRay** | See domain overview |
| **Cart** | See domain overview |

| Term | Description |
| --- | --- |
| **Client** | Synonym for the customer of this project (Chair of Software Technology) |
| **Comment** | See domain overview |
| **Contractor** | Company responsible for implementing the software |
| **DVD** | See domain overview |
| **Inventory** | See domain overview |
| **InventoryItem** | See domain overview |
| **Login** | Successful authentication after entering the correct (i.e. existing) credentials of a user |
| **Order** | See domain overview |
| **OrderStatus** | Synonym for Status, See domain overview |
| **Product** | Abstraction of a disc. Every disc is a product. |
| **Register/Registration** | Process of creating a new account in the system (i.e. a new user representation) |
| **ROLE/Role** | Role of a User (Customer or Boss), See domain overview |
| **Stock** | Amount of discs of one type that are available |
| **System** | General term for the software system that has to be implemented during this project. |
| **User** | See domain overview |
| **Videoshop** | Central class of the system representing the videoshop itself. |

© 2021 GitHub, Inc. Terms Privacy Security Status Docs
Contact GitHub Pricing API Training Blog About