

Technische Dokumentation

Projektname: Pricely

Teilnehmer: Kao, Chin Wen

Melachio Tsayem, Loic

Binti Abdul Rahman, Nur Syafiqah

Gliederung

1. Einführung

2. Funktionen

 2.1. Funktionale Schicht/ User Schicht

 2.2. Funktionen Detailschicht

 2.2.1. Beschreibung des Use Case Search a Product

 2.2.2. Beschreibung des Use Case View Product List

 2.2.3. Beschreibung des Use Case Create an Account

 2.2.4. Beschreibung des Use Case Login

 2.2.5. Beschreibung des Use Case Store a Product in Wishlist

 2.2.6. Beschreibung des Use Case Rate a Product

3. Komponenten & Kommunikation

 3.1. Struktur-Schicht

 3.1.1. Projektstruktur

 3.1.2. Package-Diagram

 3.1.3. Sequenz-Diagramm User Register

 3.1.4. Sequenz-Diagramm Product Search

 3.2. Logische-Schicht

 3.2.1. Klassendiagramm-Übersicht

 3.2.2. Klassendiagramm-Category/Product_type/Product

 3.2.3. Klassendiagramm-UserController/Service/Repository Einführung

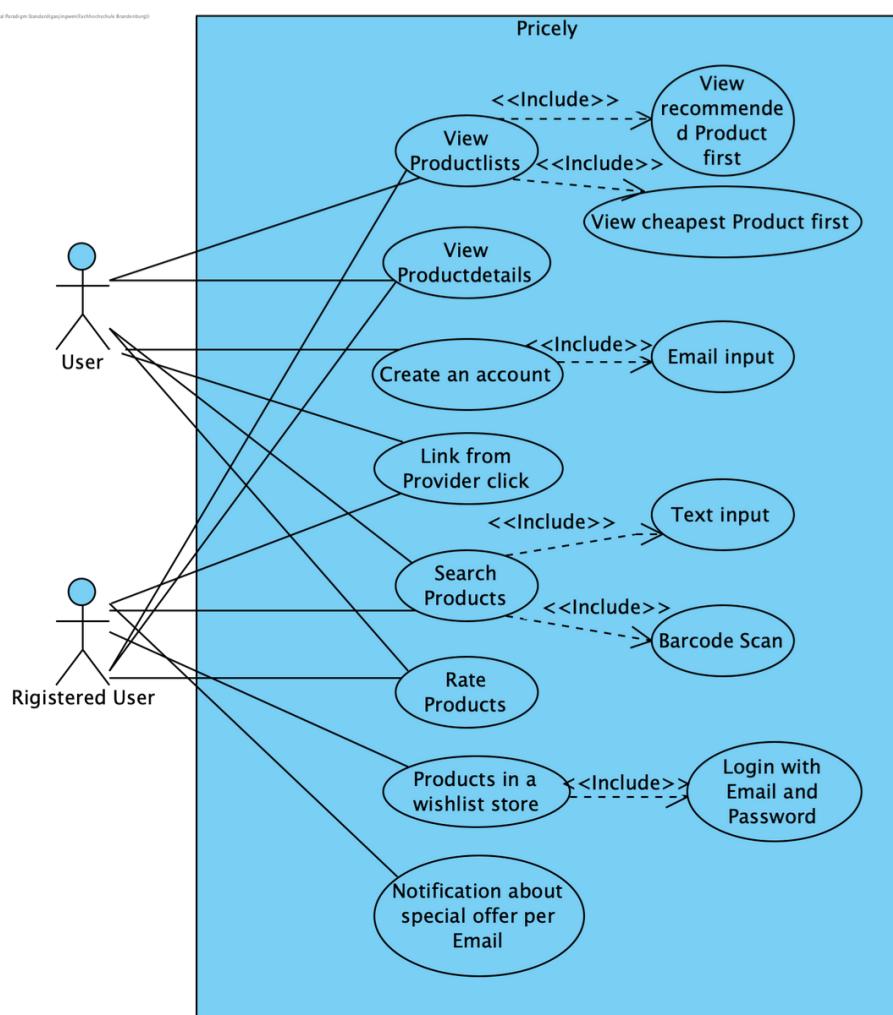
 3.2.4. Klassendiagramm-User/Wishlist Item/Product

1. Einführung

Einen Webservice mit Suchen-Funktion für Wunsch-Produkte, der nach Preisen sortierende Produktliste und Produktinformationen zeigt zu implementieren. Sowohl ein leistungsstarker Marketing- als auch ein effektiver Vertriebskanal für Online-Shops.

2. Funktionen

2.1. Funktionale Schicht/ User Schicht



2.2. Funktionen Detailschicht

2.2.1. Beschreibung des Use Case Search a Product

Name	Search a Product.							
Description	The user has entered a search query and pressed return/search, the system loops through the vendor list and processes the data received, then compile a list of suppliers with the user requested item.							
Actors	User/Registered User.							
Trigger	User enters search query.							
Precondition	A Input text in search field and search button pressed.							
Postcondition	Return a product list with correspondent or relevant Product.							
Main Flow	<p>The user has a valid product as valid text input. And there is this product in the database, it will return a list of these products from different providers.</p> <table border="1"> <tr> <td>Actor</td> <td>System</td> </tr> <tr> <td>1. Input text in search field is valid.</td> <td>2.1 System receives search query 2.2 System looks through vendor list 2.3 System request information from each vendor 2.4 System checks information from vendors and make a list. 2.5 Item list sorted and returned.</td> </tr> <tr> <td>2. Search button click.</td> <td></td> </tr> </table>		Actor	System	1. Input text in search field is valid.	2.1 System receives search query 2.2 System looks through vendor list 2.3 System request information from each vendor 2.4 System checks information from vendors and make a list. 2.5 Item list sorted and returned.	2. Search button click.	
Actor	System							
1. Input text in search field is valid.	2.1 System receives search query 2.2 System looks through vendor list 2.3 System request information from each vendor 2.4 System checks information from vendors and make a list. 2.5 Item list sorted and returned.							
2. Search button click.								
Alternative Flows	<p>If there is no correspondent product in the database, but there is relevant or similar product, it will return a list of them from different providers.</p> <table border="1"> <tr> <td>Actor</td> <td>System</td> </tr> <tr> <td>1. Input text in search field is valid.</td> <td></td> </tr> <tr> <td>2. Search button click.</td> <td>2.1 Return a list of similar products.</td> </tr> </table>		Actor	System	1. Input text in search field is valid.		2. Search button click.	2.1 Return a list of similar products.
Actor	System							
1. Input text in search field is valid.								
2. Search button click.	2.1 Return a list of similar products.							
Exception Flow	<p>The user has nothing or a invalid text input in a search field. It will return text „The field should not be empty.“ / „Sorry, there is no result.“</p> <table border="1"> <tr> <td>Actor</td> <td>System</td> </tr> <tr> <td>1. Nothing/invalid text in search field input.</td> <td></td> </tr> <tr> <td>2. Search button click.</td> <td>2.1 Return text „The field should not be empty.“ / „Sorry, there is no result.“</td> </tr> </table>		Actor	System	1. Nothing/invalid text in search field input.		2. Search button click.	2.1 Return text „The field should not be empty.“ / „Sorry, there is no result.“
Actor	System							
1. Nothing/invalid text in search field input.								
2. Search button click.	2.1 Return text „The field should not be empty.“ / „Sorry, there is no result.“							

2.2.2. Beschreibung des Use Case View Product List

Name	View Product List							
Description	User can view product list according to category							
Actor	User/Registered User							
Trigger	User clicks on 'View all products' button							
Precondition								
Postcondition	User can view product list with different sorting function							
Main Path	<p>The user choose to sort by price or sort by rating and Pricely returns a sorted product list based on the chosen function</p> <table border="1"> <tr> <td>Actor Action</td> <td>System Responses</td> </tr> <tr> <td>Choose 'Sort by price'</td> <td>Return product list sorted by price from low to high</td> </tr> <tr> <td>Choose 'Sort by Rating'</td> <td>Return product list sorted by price from high to low</td> </tr> </table>		Actor Action	System Responses	Choose 'Sort by price'	Return product list sorted by price from low to high	Choose 'Sort by Rating'	Return product list sorted by price from high to low
Actor Action	System Responses							
Choose 'Sort by price'	Return product list sorted by price from low to high							
Choose 'Sort by Rating'	Return product list sorted by price from high to low							

2.2.3. Beschreibung des Use Case Create an Account

Name	Create user account.									
Description	Online User creates customer account by entering basic information and then following up with one or more addresses and a debit or credit card.									
Actor	Customer/User.									
Trigger	The new customer visit The site and wants to set up account online.									
Precondition	User account subsystem must be available. Credit/debit authorization services must be available.									
Related use cases	Might be invoked by the Check out shopping cart use case.									
Postcondition	<p>1. The user must be created and saved. 2. One or more addresses must be created and saved. 3. Account must be created and saved. 4. Address and account must be associated with customer.</p>									
Main Flow	<table border="1"> <tr> <td>Actor</td> <td>System</td> </tr> <tr> <td>1. User indicate desire to create user account and enters basic customer information.</td> <td>1.1 System creates a new customer. 1.2 System prompts for customer addresses.</td> </tr> <tr> <td>2. User enters one or more addresses.</td> <td>2.1 System creates addresses. 2.2 System prompts for credit/debit card.</td> </tr> <tr> <td>3. User enter credit or debit card information.</td> <td>3.1 System verifies authorization for credit/debit card 3.2 System associates user, address, and account. 3.3 System return valid user account details.</td> </tr> </table>		Actor	System	1. User indicate desire to create user account and enters basic customer information.	1.1 System creates a new customer. 1.2 System prompts for customer addresses.	2. User enters one or more addresses.	2.1 System creates addresses. 2.2 System prompts for credit/debit card.	3. User enter credit or debit card information.	3.1 System verifies authorization for credit/debit card 3.2 System associates user, address, and account. 3.3 System return valid user account details.
Actor	System									
1. User indicate desire to create user account and enters basic customer information.	1.1 System creates a new customer. 1.2 System prompts for customer addresses.									
2. User enters one or more addresses.	2.1 System creates addresses. 2.2 System prompts for credit/debit card.									
3. User enter credit or debit card information.	3.1 System verifies authorization for credit/debit card 3.2 System associates user, address, and account. 3.3 System return valid user account details.									
Alternative Flows	<p>1.1 Basic user data are incomplete. 2.1 The address is not valid. 3.2 Credit/debit card is not valid.</p>									

2.2.4. Beschreibung des Use Case Login

Name	Login
Description	This use case allows user to login into the system to access the relevant functions according to the user's role. To login to the system, all users have to enter their Email username and password. The users have a maximum of 3 attempts to login after which their account are locked and they will have to contact the system administrator to unlock their account upon successful login the system will display the relevant user's home page.
Actor	User
Trigger	The customer activates Order or login
Precondition	User hast to have a valid account
Postcondition	The system displays the relevant homepage
Main Flow	<ol style="list-style-type: none"> 1. The user enters the username and password 2. The user submits the username and password 3. The system validates the username and password 4. The system verifies the username and password 5. The system displays the user's homepage 6. The use case ends
Alternative Flows	<ol style="list-style-type: none"> 3a. Missing username and/or password <ol style="list-style-type: none"> 1. The system prompts for username and password 2. Use case resumes at main flow step 1 3b. Maximum 3 attempts exceeded <ol style="list-style-type: none"> 1. The system displays "Maximum attempts exceeded, contact administrator" message 2. The system locks user account 3. The use case ends 4a Invalid username and/or password <ol style="list-style-type: none"> 1. The system displays "Invalid username and/or password" message 2. The system prompts for staffID and password 3. The use case resumes at main flow step 1

2.2.5. Beschreibung des Use Case Store a Product in Wishlist

Name	Store a product in wish list					
Description	User can add the favorite products into wish list to receive notifications about offers and changes in price. Storing product in wish list can also give easy access for the user to search products					
Actor	User/Registered User					
Trigger	User clicks on 'Add to wish list' button					
Precondition	User has an account and logged in					
Postcondition	The user can add the product in his/her wish list.					
Main Path	The user has an account, logged in and click the 'add to wish list' button <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Actor Action</th> <th style="text-align: left; padding: 2px;">System Responses</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;"> 1. Log in successfully with email and password 2. Click 'add to wish list' button next to a product </td> <td style="padding: 2px;"> 1. Show the username beside the account icon 2. Show "The product has been successfully added in you wish list" </td> </tr> </tbody> </table>		Actor Action	System Responses	1. Log in successfully with email and password 2. Click 'add to wish list' button next to a product	1. Show the username beside the account icon 2. Show "The product has been successfully added in you wish list"
Actor Action	System Responses					
1. Log in successfully with email and password 2. Click 'add to wish list' button next to a product	1. Show the username beside the account icon 2. Show "The product has been successfully added in you wish list"					
Alternative Path	The user is not logged in and click the 'add to wish list' button <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Actor Action</th> <th style="text-align: left; padding: 2px;">System Responses</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;"> 1. Click 'add to wish list' button next to a product 2. Log in successfully with email and password 3. Click 'add to wish list' button next to a product </td> <td style="padding: 2px;"> 1. Show "You need to sign in or sign up first" 2. Show the username beside the account icon 3. Show "The product has been successfully added in you wish list" </td> </tr> </tbody> </table>		Actor Action	System Responses	1. Click 'add to wish list' button next to a product 2. Log in successfully with email and password 3. Click 'add to wish list' button next to a product	1. Show "You need to sign in or sign up first" 2. Show the username beside the account icon 3. Show "The product has been successfully added in you wish list"
Actor Action	System Responses					
1. Click 'add to wish list' button next to a product 2. Log in successfully with email and password 3. Click 'add to wish list' button next to a product	1. Show "You need to sign in or sign up first" 2. Show the username beside the account icon 3. Show "The product has been successfully added in you wish list"					
Exception Path	The user does not have an account. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Actor Action</th> <th style="text-align: left; padding: 2px;">System Responses</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;"> 1. Click 'add to wish list' button next to a product 2. Sign up with email and password 3. Click 'add to wish list' button next to a product </td> <td style="padding: 2px;"> 1. Show "You need to sign in or sign up first" 2. Show the username beside the account icon 3. Show "The product has been successfully added in you wish list" </td> </tr> </tbody> </table>		Actor Action	System Responses	1. Click 'add to wish list' button next to a product 2. Sign up with email and password 3. Click 'add to wish list' button next to a product	1. Show "You need to sign in or sign up first" 2. Show the username beside the account icon 3. Show "The product has been successfully added in you wish list"
Actor Action	System Responses					
1. Click 'add to wish list' button next to a product 2. Sign up with email and password 3. Click 'add to wish list' button next to a product	1. Show "You need to sign in or sign up first" 2. Show the username beside the account icon 3. Show "The product has been successfully added in you wish list"					

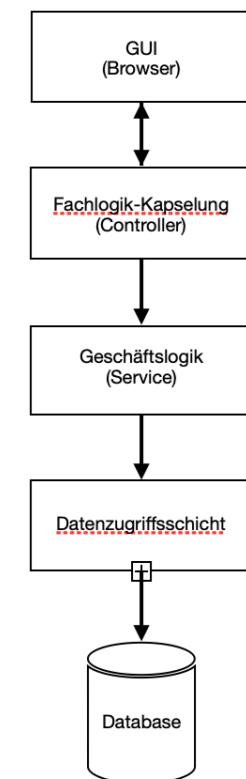
2.2.6. Beschreibung des Use Case Rate a Product

Name	Rate a product	
Description	User can rate a product based on 1 to 5 star rating. The overall rating will be shown next to the products.	
Actor	User/Registered User	
Trigger	User clicks on 'Rate this product' button on the product information page	
Precondition	User has to be registered and signed in	
Postcondition	The user can add the product in his/her wish list	
Related Use Cases	Creating an account, Log in	
Main Path	The user has an account, logged in and click the 'Rate this product' button	
	Actor Action	System Responses
	<ol style="list-style-type: none"> 1. Log in with email and password 2. Click 'Rate this product' button next to a product 	<ol style="list-style-type: none"> 1. Display the username beside the account icon 2. Display "Your rating has been saved! Thank you for the information."
Alternative Path	The user is not logged in and click the 'add to wish list' button	
	Actor Action	System Responses
	<ol style="list-style-type: none"> 1. Click 'Rate this product' button next to a product 2. Log in with email and password 3. Click 'Rate this product' button next to a product 	<ol style="list-style-type: none"> 1. Display "You need to sign in or sign up first." 2. Display the username beside the account icon 3. Display "Your rating has been saved! Thank you for the information."
Exception Path	The user does not have an account.	
	Actor Action	System Responses
	<ol style="list-style-type: none"> 1. Click 'Rate this product' button next to a product 2. Sign up with email and password 3. Click 'Rate this product' button next to a product 	<ol style="list-style-type: none"> 1. Display "You need to sign in or sign up first" 2. Display the username beside the account icon 3. Display Your rating has been saved! Thank you for the information."

3. Komponenten & Kommunikation

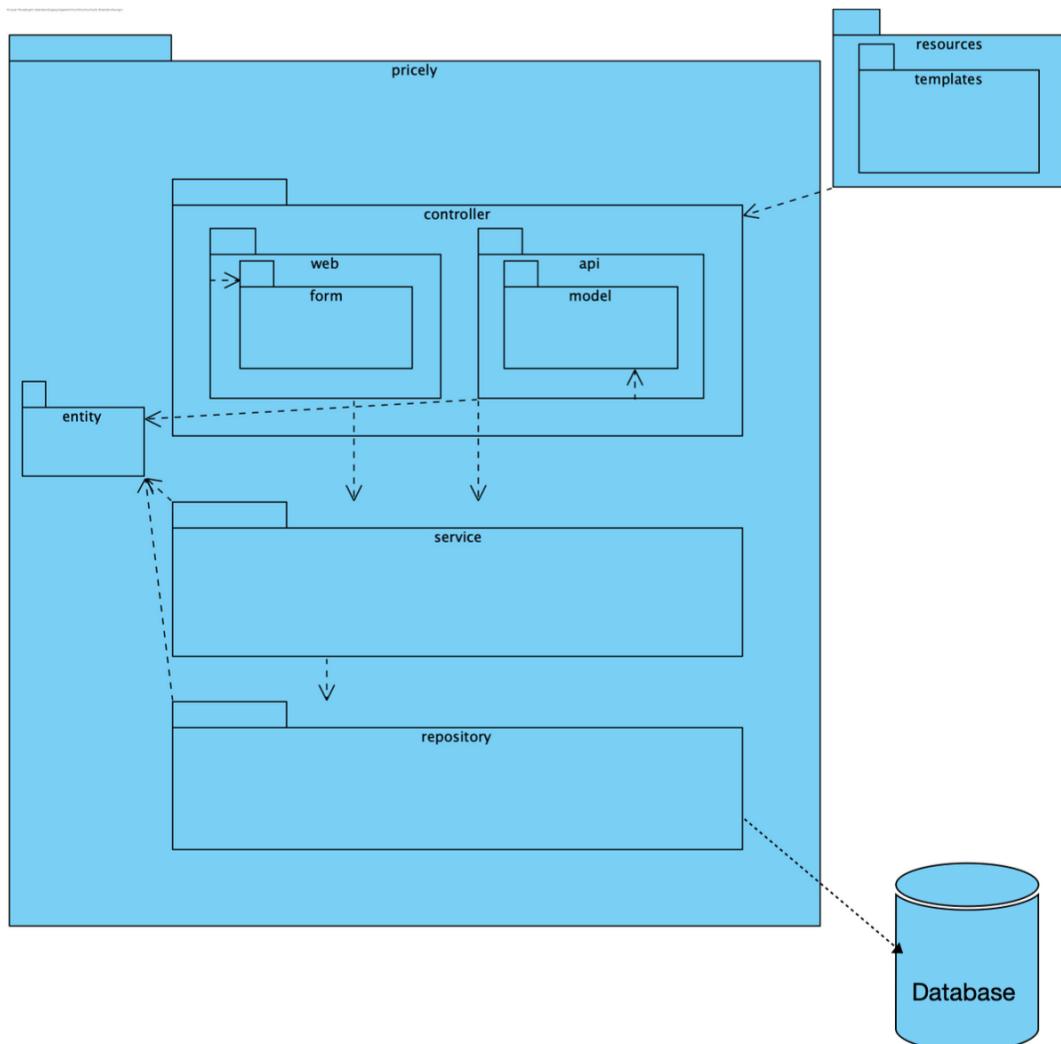
3.1. Struktur-Schicht

3.1.1. Projektstruktur

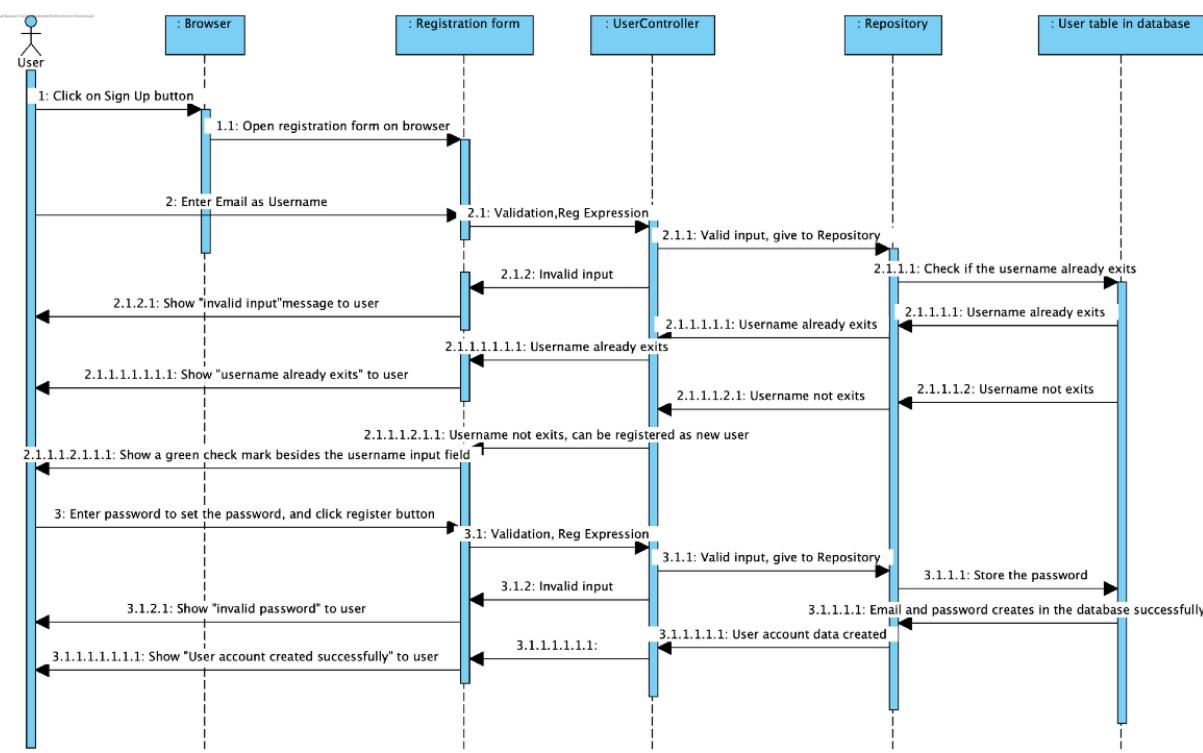


Präsentation-Schicht	<ul style="list-style-type: none"> -Thymeleaf HTML template engine -Benutzungsoberfläche, Daten auf Web Browser zu zeigen. -Datenaustausch mit Controller, z.B. Client request from Browser zu Controller leiten.
Fachkonzepte-Zugriffsschicht	<ul style="list-style-type: none"> -Springframework web bind annotation + URL -Aufbereitung/Filterung der Daten für Präsentation-Schicht -Leiten die Request/Aktionen zu Service, dabei die Controllers benutzen die Methode von Service.
Fachkonzept-Schicht	<ul style="list-style-type: none"> -Springframework service annotation -Daten(Entity)Bearbeitung nach obengegebener Fachlogik -CRUD, dabei die Services benutzen vorgeschriebene Methode be Repository(Man kann auch selbst-definierte Methode rufen.)
Persistent-Zugriffsschicht	<ul style="list-style-type: none"> -Springframework data repository/ data jpa annotation -Datenzugriff auf der Datenbank durch Repository Interface, dabei Java zu SQL binden per framework/ annotations.
Datenhaltungsschicht	<ul style="list-style-type: none"> -phpMyAdmin mySQL Datenbank Server -Datenhaltung mit Tables

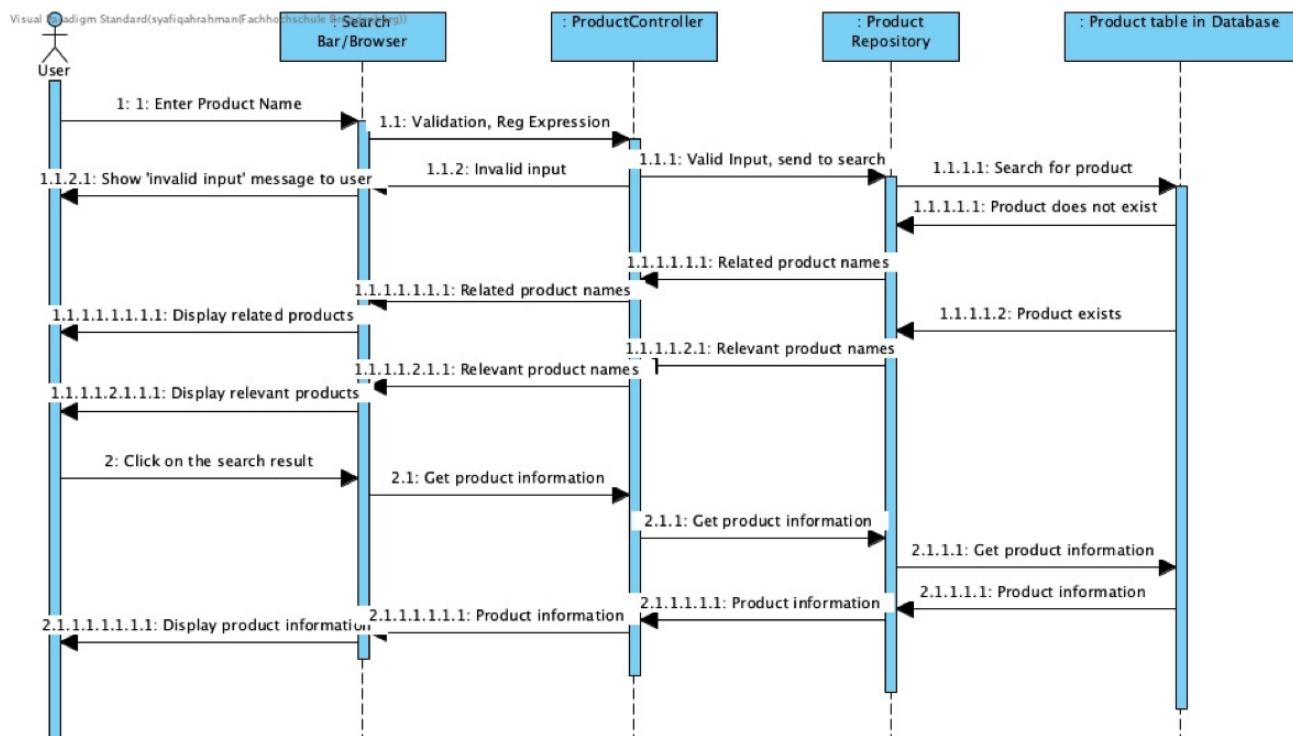
3.1.2.Package-Diagramm



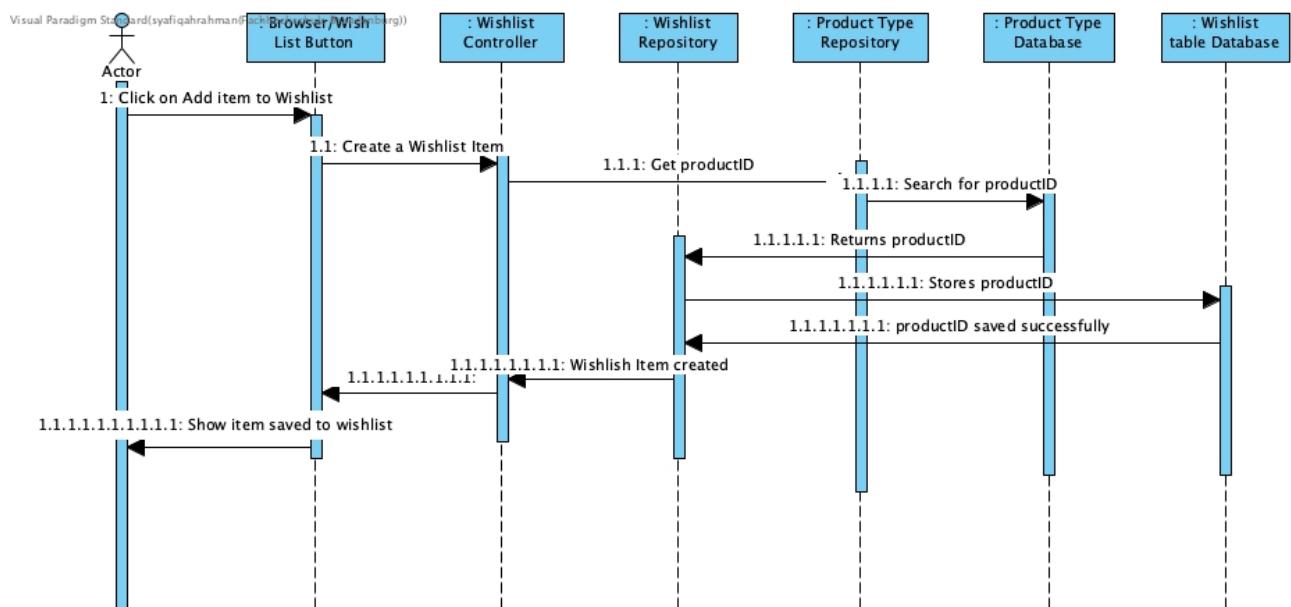
3.1.3. Sequenz-Diagram User Register



3.1.4. Sequenz-Diagram Product Search



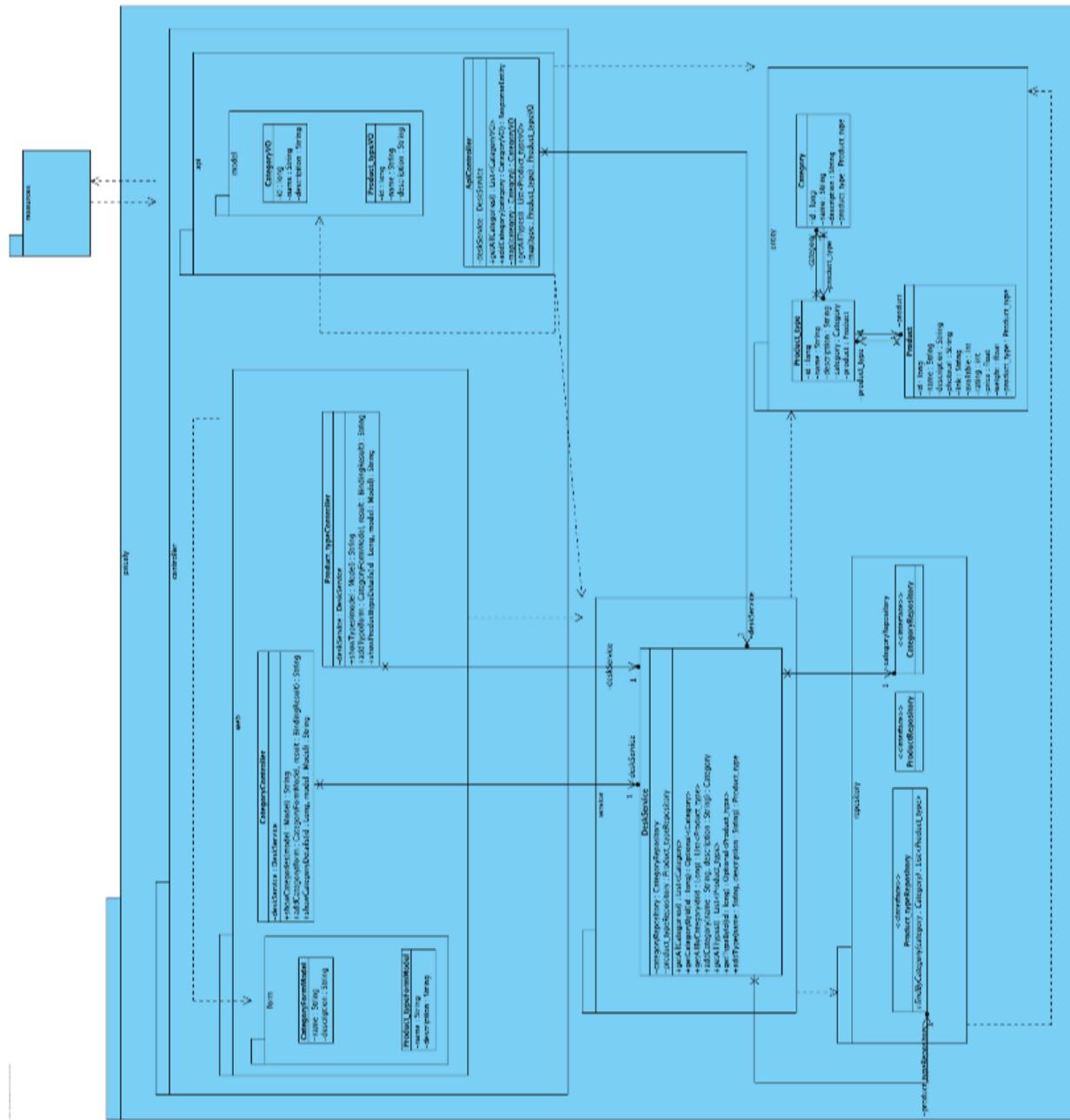
3.1.5. Sequenz-Diagramm Add Items to Wishlist



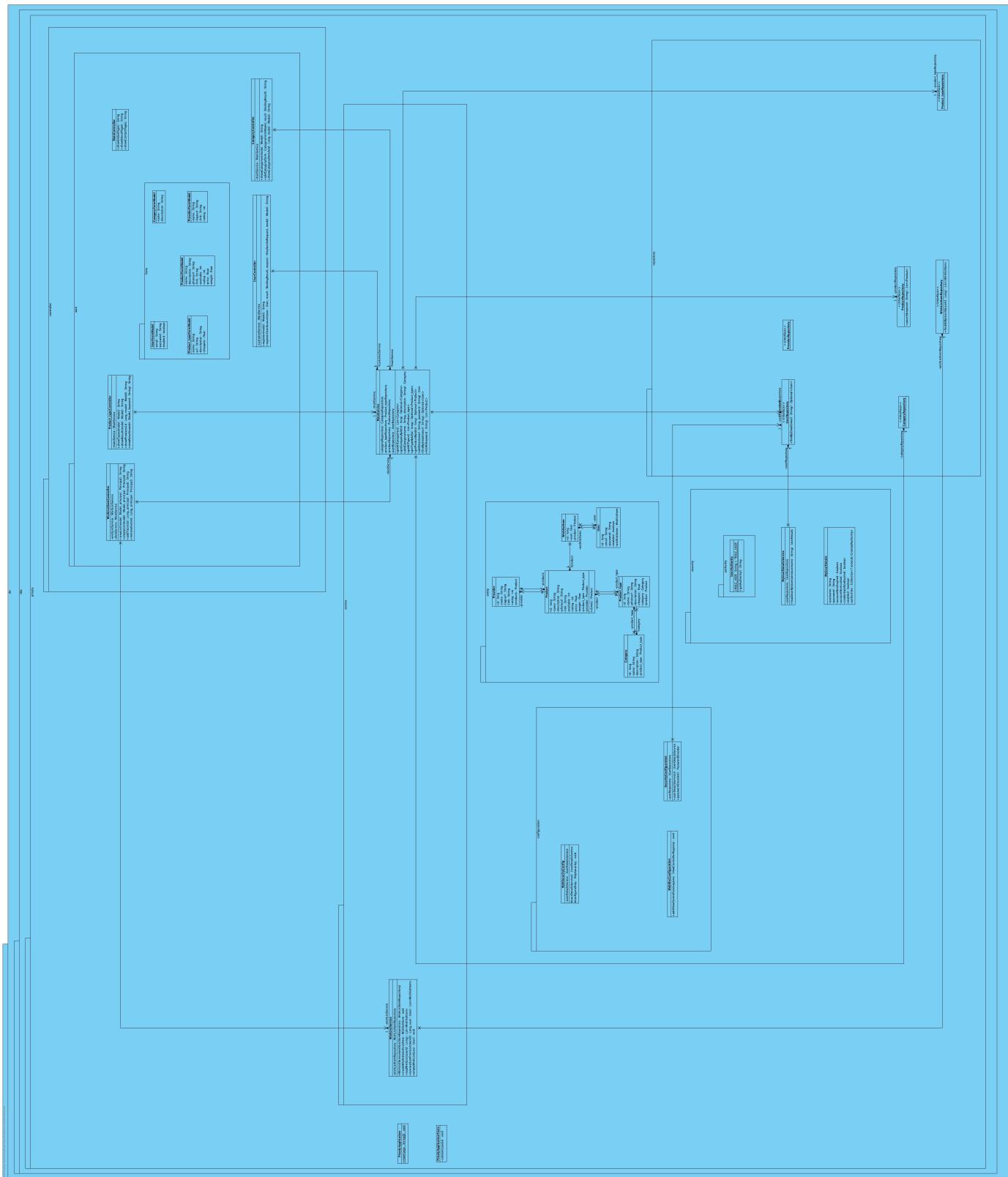
3.2. Logische-Schicht

3.2.1. Klassendiagramm-Übersicht

Version 1



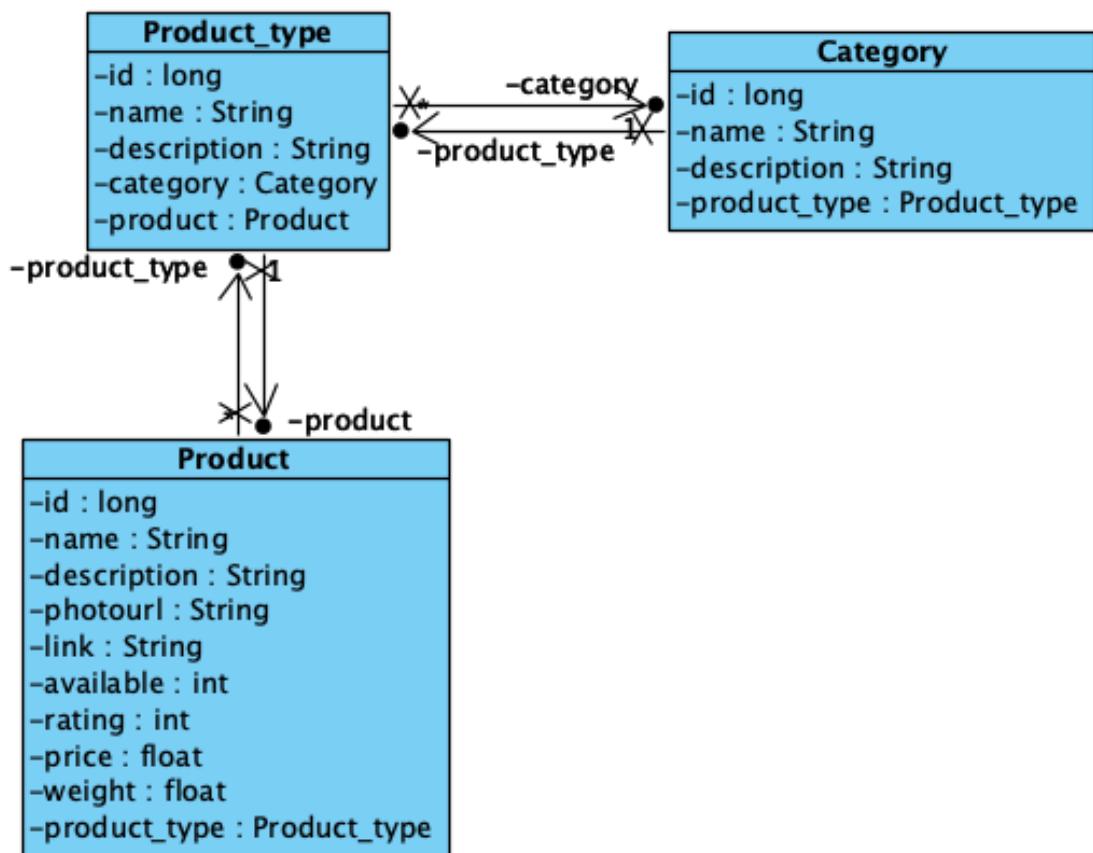
Version 2 (aktuelle)



3.2.2.Klassendiagramm-Category/Product_type/Product

Category und Product_type sind one-to-many , und Product_type und Product sind auch one-to-many. Denn es sollte bei einen Category viele unterschiedliche Product_type geben, und eine Product_type sollte noch viele von unterschiedlichen Providers Angeboten geben.

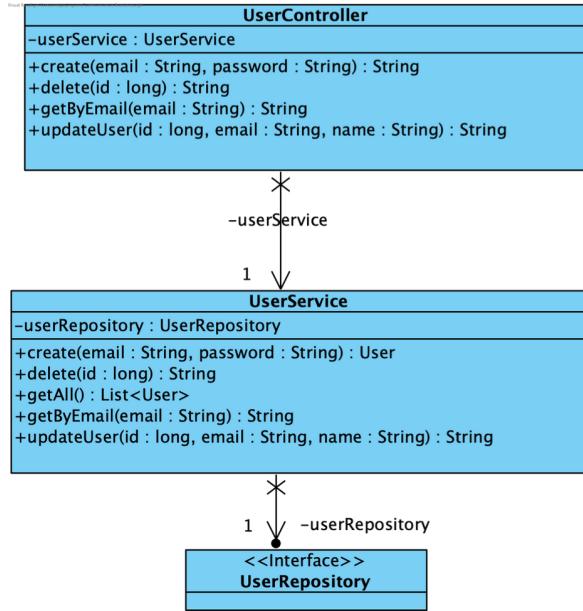
Visual Paradigm Standard(gaojingwen(Fachhochschule Brandenburg))



3.2.3.Klassendiagramm-UserController/Service/Repository

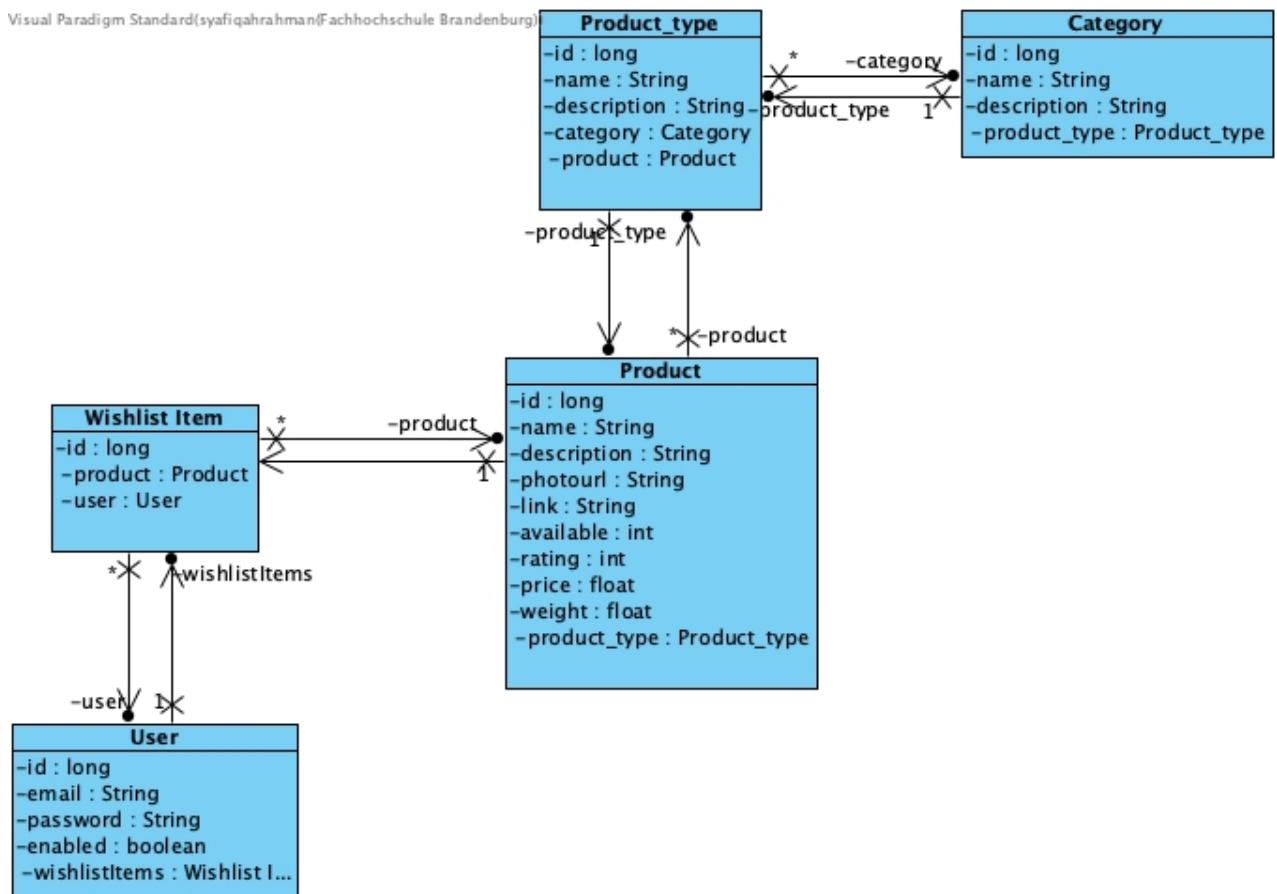
UserController sollte für User Register zur Verfügung sein: -create: erst überprüft, ob die Eingabe gültig sind, und dann überprüft, ob diese Email schon mal registriert hat.Dabei wird entsprechende Rückmeldung für User zu zeigen.

-updateUser: die Users sollten auch die User Informationen bearbeiten und erneuern.



3.2.4. Klassendiagramm-User/Wishlist Item/Product

User und WishlistItem sind one-to-many , und WishlistItem und Product sind auch one-to-many. Ein Benutzer kann viele Artikel auf der Wunschliste haben. Ein Produkt kann auf mehreren Wunschlisten stehen.



Dokument Historie

Version	Datum	Author	Erläuterung
1	30.11.21	Chin-Wen, Kao	Gliederung
1	30.11.21	Chin-Wen, Kao	Use-Case Diagramm
1	30.11.21	Chin-Wen, Kao	Use Case Search a Product
1	30.11.21	Chin-Wen, Kao	Use Case View a Product
1	04.12.21	Chin-Wen, Kao	Projektstruktur
1	04.12.21	Chin-Wen, Kao	Package-Diagram
1	05.12.21	Chin-Wen, Kao	Sequenz-Diagram User Register
1	05.12.21	Chin-Wen, Kao	Klassendiagram Übersicht
1	05.12.21	Chin-Wen, Kao	Klassendiagram- Category/ Product_type/Product
1	05.12.21	Chin-Wen, Kao	Klassendiagram- User Controller/ Service/Repository
1	05.12.21	Chin-Wen, Kao	Use Case Add Product in Wishlist
2	05.12.21	Loic Melachio	Einführung
1	05.12.21	Loic Melachio	Use Case Create an Account
2	05.12.21	Loic Melachio	Use Case Search a Product
1	05.12.21	Syafiqah Rahman	Sequenz-Diagram Product Search
2	05.12.21	Syafiqah Rahman	Use Case Add Product in Wishlist
2	05.12.21	Syafiqah Rahman	Use Case View Product List
2	05.12.21	Syafiqah Rahman	Use Case Rate a Product
2	05.12.21	Syafiqah Rahman	Gliederung
1	19.1.22	Syafiqah Rahman	Sequenz-Diagram Add Item in Wishlist
1	19.12.22	Syafiqah Rahman	Klassen-Diagram User/Wislist/ Product
2	20.12.22	Syafiqah Rahman	Klassendiagram Übersicht