

# Group 17 – Online Retailing

- Pavel Petrukhin - 19328624
- Christiana Popoola - 19334280
- Pascal Raos - 19335585
- Oluwatuminu Ogunbadejo - 18328099
- David Olowookere - 19335061

## 1 - Background Research

Initially, we decided to do the things together to learn more about UML, so that later on we could do things in parallel. Our first step was setting up the means of communication, we almost immediately chose Discord, because it is a voice messenger we are all familiar with and it offers a variety of functions including screen sharing, pinning messages in the chat (which we found particularly useful) and etc. Blackboard collaborate space was not considered at all, because its interface is not user-friendly (most of the time) and it does not have an easy to use text messaging system, unlike Discord. As for the software for UML diagrams, there were a lot of options considered, including:

- <https://www.visual-paradigm.com/solution/freeumltool/> - required download, we wanted a web-based solution
- <https://www.gleek.io/> - included some markup language for diagrams, we needed a visual editor
- <https://www.lucidchart.com/pages/landing/uml-diagram-software> - a lot of templates we were unsure about
- <https://www.yworks.com/products/yed> - again, a desktop application
- <https://argouml.en.softonic.com/> - from what we saw last supported version was Windows XP

Finally, we found one service that was suitable for us - <https://creately.com/>. It combines tools for collaboration, easy-to-use templates, and works in the browser. We used it for the whole project.

We did use-case, class diagrams, and ethics canvas together, and activity diagrams were done in parallel in two groups. As for the report, we started writing it in advance, so that we could meet the required deadline.

We began our project by allocating different online retail systems (websites) to each member, this was to divide the work and also give us a wider range of information.

We compiled a list and then each member chose one of the following: Amazon, Donedeal, Aliexpress, or Wildberries. We pooled our research together and from that concluded functionality was common between websites in order to create a generalized use case diagram for an online retail system.

Some research is provided below in the form of textual descriptions of the websites. We also did research on the structures of use case diagrams, cross checking each connection

(lines/associations) made on the diagram with our lecture notes, to better understand and speed up the process of creating one ourselves.

### Donedeal:

- Basic user features like logging in, registering accounts, providing account details were present.
- Donedeal focused heavily on the user to user purchasing interactions, such as listing an item for sale, contacting the seller, and in general was very heavy on communication between the two parties.
- The website offered a wide variety of searching/filtering options for items/sellers
- Donedeal offered multiple different payment options, however was restricted by whichever the seller could access.

### Amazon:

- Amazon being big online retail focused more on customers and sellers. Customers being able to order as registered users or guests.
- Amazon also has different ways of searching for products like scanning the barcode, taking a picture, and even finding the look from your photos.
- The website also makes communication between the customers and the seller better. They provide reviews and ratings from other customers.
- Amazon also offers discount codes and coupons to customers. They also have the Amazon prime which provides users with a lot of benefits including free shipping on most items.

### Aliexpress:

- Aliexpress being an originally Asian founded company turned international focuses on making the site accessible for people across the globe. They achieved this by offering different currency conversion and language translations, as well as size conversion tables.
- It also follows suit and has the basic user features of logging in, registering, wishlist, basket, etc....
- The site itself is more oriented towards buyers rather than sellers (despite having both), this has resulted in better features for the buyers such as filtering by different categories, shipping cost, item price, etc. Which all provide a better buyer experience.
- The website is also very heavily influenced by reviews of previous buyers, this often dictated the popularity of various items.
- Discount code banners/other types of advertising were also extremely prevalent on the site, encouraging the buyers to make immediate purchases rather than saving for later (wishlist/basket).

## eBay

- Based in America. Originally an auction site, it has since added a “Buy it Now” option to auctions that allows buyers to circumvent auctions on items they want, at the cost of a higher price than bidding
- It also supports regular non-auction shops, and allows users to search items by category or by name, with an extensive filtering system
- Users can register account to buy or sell, but can also buy as guests
- eBay allows sellers to set up a Q&A for their listings to facilitate communication between seller and buyer, and both sellers and buyers can leave Feedback on each other

## Wildberries:

- A Russian analog of Amazon founded in 2000, the largest online retailer in the CIS region. Recently it was launched in Slovakia and Poland.
- Sells everything from clothes and groceries to cars and flight tickets
- Has a nice interface, but requires registration for buying products
- The listings of items are impacted by advertisements and rating

## Conclusions from the research:

- Search filters were common between all websites to allow faster and more efficient access to specific products the customer desires, but also as a form of advertisement, showcasing the large variety the website offers. The top results of the searches were also often advertising specific item(s), our discounts/offers.
- Basic Account functionality was offered on all websites, however was not always necessary for the user to buy/view products, this encourages buyers to make quick purchases. In terms of selling a product, however, an account was always needed, which we understand is to regulate items being sold and ensure regulations are being abided by.
- All websites offered a wide selection of payment options to the customer. This meant regardless of the location of the users payment options were available, the majority of the sites offered payment systems such as PayPal, (or equivalent) which helps guarantee legitimacy and security with the payment details.
- Sellers could manage their shop on every website, which included adding, deleting, and changing products. Moreover, there was a clear way of withdrawing their earnings from their account, whether it was Stripe, some internal payment gateway, or Paypal.

- Finally, every platform had a special type of user called administrators/moderators whose primary use-case is updating the system and making sure that the content on the website is appropriate and fair.

## 2 - Ethics canvas

Here is a textual description of our ethics canvas.

### Individuals Affected

Non-locals: people who don't usually like to or do online retailing would be heavily affected by this. This could be buyers who prefer to try out items before committing to a purchase, using the online retailing system for them would mean having to possibly return items unfit for use, or order multiple sizes to see which would fit the best for purpose.

People without strong internet/electricity: people living in slightly more rural areas or areas that don't have consistent electricity or internet access.

### Behaviour

Online shopping removes the social aspect of shopping entirely. Less active communication with others.

Online shopping and other online services all promote a more sedentary lifestyle, increasing potential health risks.

Users may end up becoming too dependent on technology.

### Relations

People would become less sociable because online shopping is another service, which prevents them from meeting with their family and friends.

### What can we do?

We can make the online system more user friendly. Make the shops more easily accessible so that people can carry out their orders easily.

We could highlight the benefits of the online retailing system, explain how it is often more efficient, how items ordered go straight from the factory untouched by other buyers, how it wastes less time compared to in-person shopping (no queuing, can check stock immediately, etc...).

We could offer free returns on items unfit for purpose and provide a more detailed description for sizing (possibly sizing guide) to prevent the wrong orders in the first place.

We could try to optimize server use by dynamically changing the number of computational resources we use depending on the load. This is both beneficial in an environmental and economic sense.

Offer a service such as saving a basket/wishlist, in case people get logged out and need to access items they were previously looking at.

## **Worldviews**

In relation to work, online shopping reduces stress for daily workers. As they don't have to physically go to the shops to purchase items.

It makes consumption easier since the products can be purchased easily.

Some religions or groups of people might have a problem with online shopping. They might not want their information to be on the system.

## **Group Conflicts**

Online shopping, much like many innovative online services, convert low educational requirement jobs such as sales, stock keeping, etc, into an automated system created and run by web designers and software engineers.

Depending on the extent of integration, non-IT related jobs may become redundant.

## **Groups affected**

Elderly: the elderly are affected by the online retailing system because some of them do not know how to use them, which leads to the necessity of going shopping in a traditional way that might be hard/dangerous (e.g. COVID).

Local/physical retailing: online retailing systems might have a bad impact on local businesses because more and more people tend to order things online from eBay or Amazon and etc and hence the local businesses lose their customers.

## **Product or Service Failure**

If the service is unavailable it might be a big problem for people who rely on ordering things online. Perhaps, groceries, essential things, or medication. Moreover, in a B2B model of interaction disruption of the service might result in local businesses not getting the products delivered, which would have an economic impact.

## **Problematic Use of Resources**

Once the number of users of the system increases, it will need more computational resources to function properly. Therefore, we, as developers, would have to increase the number of servers we are using to process and store data. This, in turn, causes more damage to the environment, as using more servers means using more electricity.

## **3 - Use case diagram**

Based on our research above and all of our overall pooled information we created the use case diagram. Our approach was to apply the common functionality between all of our websites of research and use that to produce the use cases. For example, the search filter/box was a functionality found on every website we researched, serving a similar purpose, and thus based on that we included it in the diagram. And so we followed suit with the rest of the use cases present in our diagram. This method seemed to be the most

efficient because it gave each individual an opportunity to figure out the important functions in their area of research but also allowed us as a group to disregard the functionalities that weren't too common or just didn't really add any importance to the overall diagram. This was a little difficult due to the nature of each website placing emphasis on different users (buyer-oriented or seller oriented) or different functionalities, but we think overall we came to a good understanding of the vital functionalities which were well translated into our use case diagram.

After creating the use-case diagram we spent a significant amount of time reviewing it and discussing things over and over again, which led to changing the type of some associations and splitting the diagram into two phases. The logic between splitting was creating two independent tasks for a team of developers so that the project could be delivered faster. Moreover, we thought that in this way we could explain the design to our customers easier.

Finally, we discussed what fields we would like to include in use-case descriptions, we went with extended descriptions to define the system even more. We created thorough descriptions for 8 main use-cases. The research we undertook for the use-case diagram pretty much shaped the rest of our project. We think that the listing of all the functions available to the user is the most essential part of any system design and that is why we showcased our use-case diagram in the video presentation.

The diagram is split into two parts (by phases)

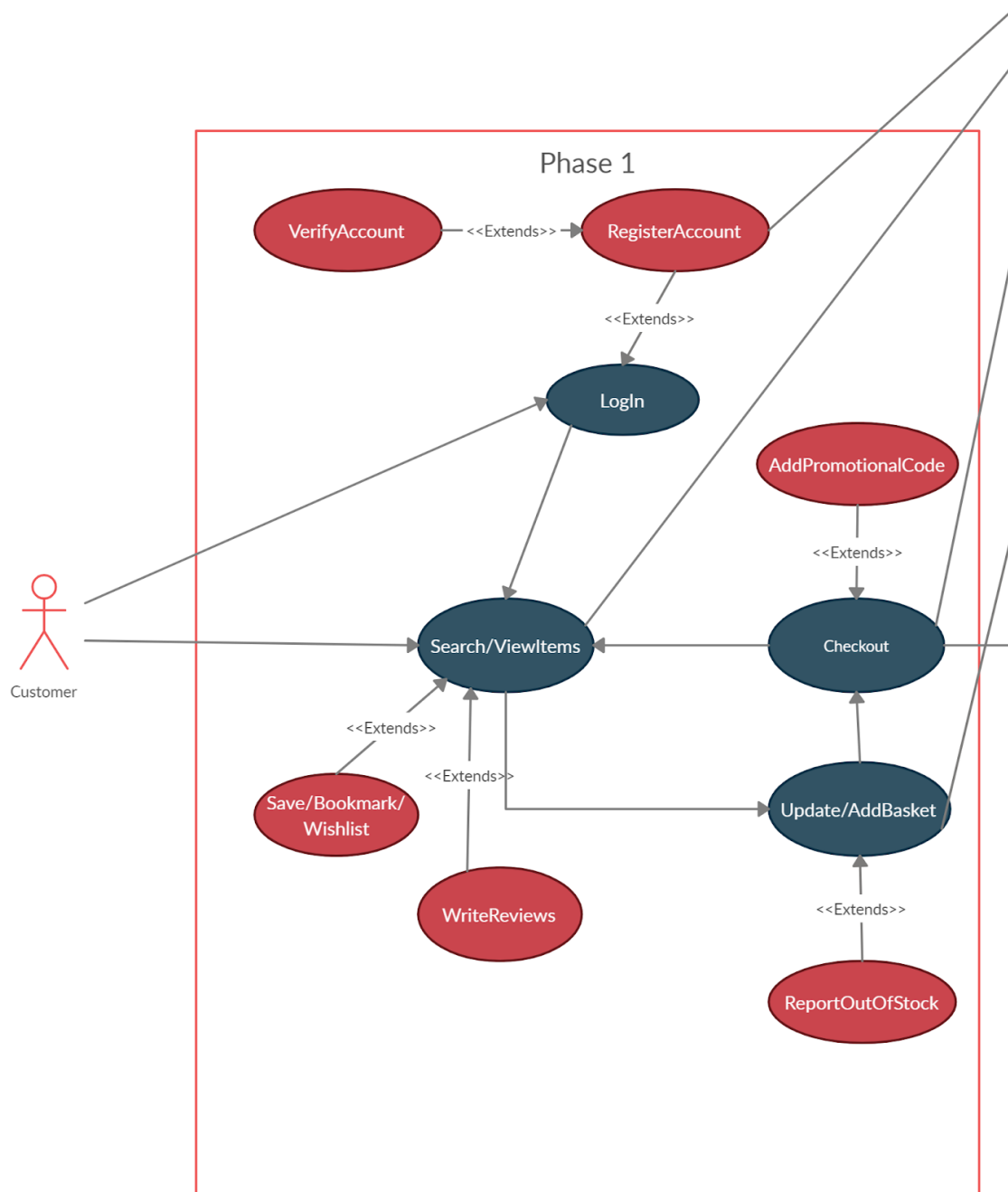
Please also note that we have used color coding for better readability.

Navy: represents our main use cases

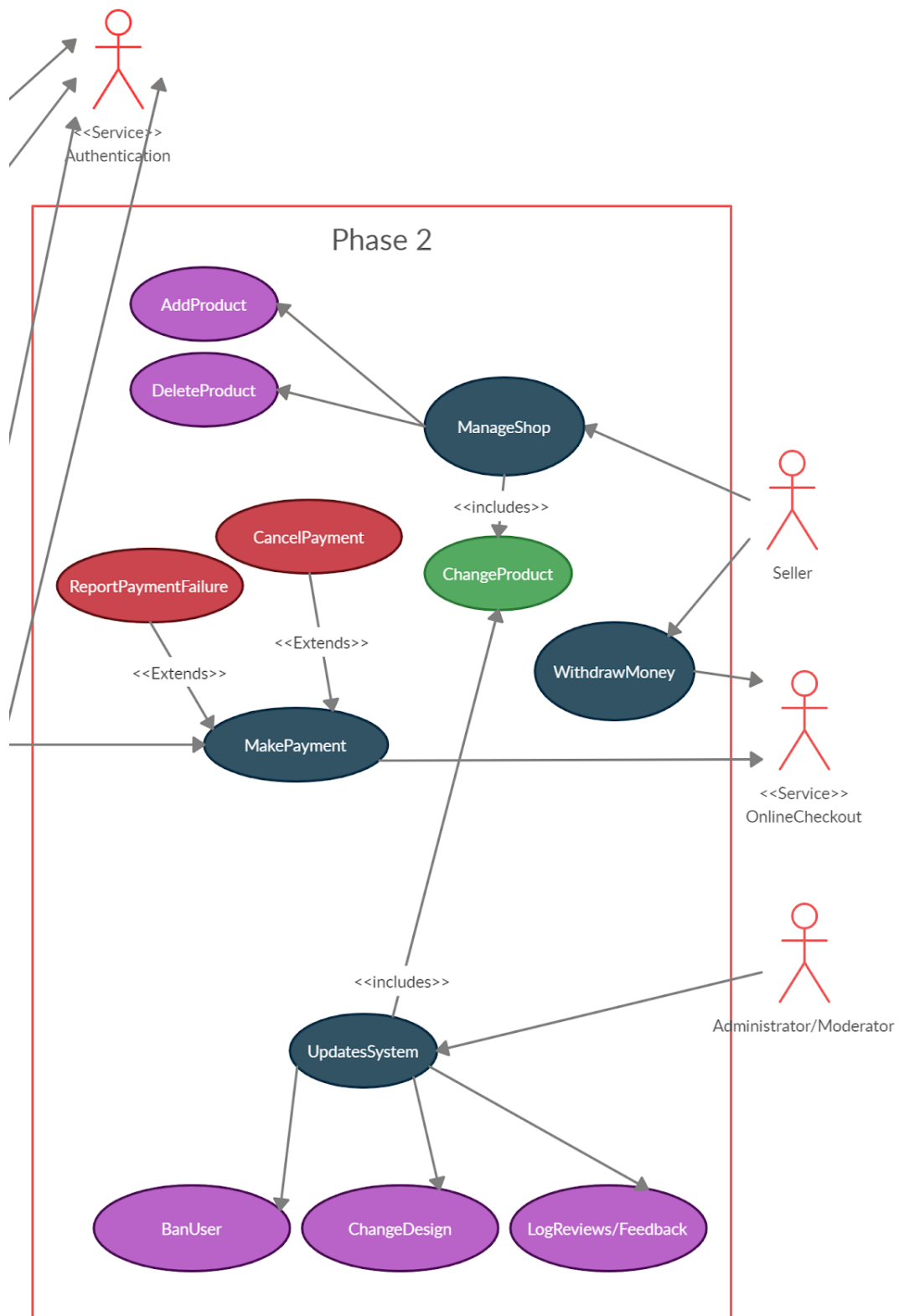
Red: represents a use case that extends from the main use cases

Green: represents a use case that includes 2 or more main use case

Pink: represents use cases that follow as branching choices







## Online retailing system

Author(s): Christiana Popoola

Date: 18.10.20

Version: 1

Use-case name	<b>Search/ViewsItems</b>
Use-case type	Business Requirement
Use-case ID	BR - 01
Priority	High
Source	Search/ViewsItems
Primary Actor	Customer
Other Actors	<<Service>> Authentication
Other Interested Stakeholders	N/A
Description	<p>This use case describes the event of the customer (main actor), searching for/viewing items. Once the item(s) are available on the system, if searched they will be displayed. For any item available the customer may write a review or give feedback on it. For any item available the customer may also save to their wishlist/bookmarks for easy direct access later on.</p> <p>&lt;&lt;Service&gt;&gt; Authentication will authenticate items availability (stock).</p>
Precondition - 1	Item(s) must be available on the system
Trigger - 2	This use case is initiated when the customer enters the website
Typical Course of Events - 3	<p><u>Actor Action:</u>  <b>Step 1:</b> customer enter name of the item available on the system</p> <p><u>System Response:</u>  <b>Step 2:</b> display all items related to the search input</p> <p><u>Actor Action:</u>  <b>Step 3:</b> customer select item(s)  <b>Step 4:</b> customer view item(s)  <b>Step 5:</b> &lt;&lt;service&gt;&gt; authentication verifies stock status of items  <b>Step 6:</b> customer proceeds to checkout</p>
Alternate courses - 4	<p><b>Alt Step 1:</b> the customer enters unrelated input (neither item nor relating items exist on system)</p> <p><b>Alt Step 1:</b> customer searches through categories of items</p>

	<b>Alt Step 2:</b> system fails to find items available relating to search input and displays error <b>Alt Step 5:</b> <<service>> authentication fails to verify stock status (item(s) is discontinued) <b>Alt Step 6:</b> customer adds an item(s) to basket <b>Alt Step 6:</b> customer saves item(s) to wishlist/bookmarks <b>Alt Step 6:</b> customer writes a review of item(s)
Conclusion - 5	The use case concludes that the customer proceeds to checkout or adds an item(s) to the basket
Postcondition - 6	Item(s) searched for are found
Business rules - 7	The system will only display item(s) available
Implementation constraints and specifications - 8	<ol style="list-style-type: none"> <li>1. Customer will be able to input searched word into the search bar in the upper center of the page</li> <li>2. Customer will be able to search through categories of items on the left-hand side</li> </ol>
Assumptions - 9	The customer wants to make a purchase.
Open Issues - 10	None currently

### Online retailing system

Author(s): Christiana Popoola

Date: 18.10.20

Version: 1

Use-case name	<b>UpdatesSystem</b>
Use-case type	System Requirement
Use-case ID	SR-01
Priority	High
Source	N/A
Primary Actor	Administrator/Moderator
Other Actors	N/A
Other Interested Stakeholders	N/A
Description	<p>This use case describes the event of the administrator/moderator (main actor), updating the system. Anytime necessary administrator/moderator will perform maintenance updates on the system to ensure functionality. For any reviews/feedback given by customers, the</p>

	administrator/moderator will log and apply them as seen fit. When deemed fit, the administrator/moderator may ban a customer or seller from the system (due to breaking rules/regulations).
Precondition - 1	N/A
Trigger - 2	This use case is initiated when the administrator/moderator enters the system in admin mode.
Typical Course of Events - 3	<u>Actor Action:</u> <b>Step 1:</b> administrator/moderator enters the system using their login (admin mode) <b>Step 2:</b> administrator/moderator reviews and makes note of all relevant reviews and feedback <b>Step 3:</b> administrator/moderator performs maintenance reviews <u>System Action:</u> <b>Step 4:</b> system is found to be updated (perfect condition) <u>Actor Action:</u> <b>Step 5:</b> administrator/moderator reviews reported cases for banning users (customer or seller) and bans users
Alternate courses - 4	<b>Alt Step 1:</b> administrator/moderator enters the system without their login, as customer <b>Alt Step 2:</b> deletes irrelevant review with unfounded claims <b>Alt Step 4:</b> errors and faults found in the system <b>Alt Step 5:</b> case viewed as unnecessary and the user is not banned
Conclusion - 5	The use case concludes that the system is updated
Postcondition - 6	The system is in the best condition, updated
Business rules - 7	Only the administrator/moderator can access these activities through the admin-specific login
Implementation constraints and specifications - 8	1. administrator/moderator has admin-specific login used to access the system in editor mode
Assumptions - 9	Administrator/moderator is knowledgeable on the maintenance of the system and has the ability to fix issues
Open Issues - 10	None currently

## Online retailing system

Author(s): Pavel Petrukhin

Date: 18/10/2020

Version: 1

Use-case name	<b>Update/AddBasket</b>
Use-case type	Business Requirements
Use-case ID	BR - 2
Priority	High
Source	Update/AddBasket
Primary Actor	Customer
Other Actors	Authentication server - check that the target product is available now
Other Interested Stakeholders	Sellers are concerned about the popularity of their product, e.g. they want to know how many times people bought their product with regard to the number of times it was added to the cart.
Description	This use-case describes the event when a user decides to change their cart. After the user chooses which products to select/remove the items are staged to be added/deleted from the cart respectively. The external server is used to check whether a particular product is currently in-stock. On completion, the products are added to the cart/removed from the cart.
Precondition - 1	User has to choose which products to add/remove
Trigger - 2	This use-case is initiated when the user updates the cart
Typical Course of Events - 3	<ol style="list-style-type: none"><li>1. The user chooses which items to add and/or remove.</li><li>2. User clicks update cart</li><li>3. The availability of added items is checked via the external server</li><li>4. The server returns that the items are in-stock, the cart is updated</li></ol>
Alternate courses - 4	<ol style="list-style-type: none"><li>1. The user chooses which items to add and/or remove.</li><li>2. User clicks update cart</li><li>3. The availability of added items is checked via the</li></ol>

	<p>external server</p> <p>4. The server returns that some of the items are not in-stock, the cart is updated partially and the user is asked if they want to browse again or go with the existing cart.</p>
Conclusion - 5	The use-case concludes when the user is transferred to the updated cart page
Postcondition - 6	The cart is updated, all the items added have been checked to be in-stock.
Business rules - 7	<p>1. Users under the age of 18 are not allowed to add age-restricted items to the cart e.g, alcohol or tobacco</p>
Implementation constraints and specifications - 8	<p>1. A cart button in the footer. Whenever the user clicks it he is transferred to an updated cart (might be also told that some of the items are not available)</p>
Assumptions - 9	<p>1. Users have to specify their age at the registration</p> <p>2. The server for checking the availability is reachable</p>
Open Issues - 10	None currently

## Online retailing system

Author(s): Oluwatumininu Ogunbadejo

Date:19/10/2020

Version: 1

Use-case name	<b>ManageShop</b>
Use-case type	Business Requirements
Use-case ID	BR - 3
Priority	High
Source	ManageShop
Primary Actor	Seller
Other Actors	N/A
Other Interested Stakeholders	Administrator/Moderator might want to check the items/products being added by the seller.
Description	This use case describes the event of a seller updating their online shop. They could either change, delete, or add a product. After the seller makes some changes, the products are then made available for the public to see.
Precondition - 1	Only sellers can make changes to their own shop
Trigger - 2	This use-case is initiated when the seller clicks edit product.
Typical Course of Events - 3	<ol style="list-style-type: none"><li>1. The seller clicks on "edit product".</li><li>2. The seller then chooses whether to add, delete, or change a product.</li><li>3. The seller clicks confirm changes.</li><li>4. The seller's shop updates to the modified version.</li></ol>
Alternate courses - 4	<ol style="list-style-type: none"><li>1. At stage 2 of the Typical Course of Events, the seller could decide not to make any changes.</li></ol>
Conclusion - 5	This use-case concludes when the seller gets a confirmation email.
Postcondition - 6	The shop is updated, and the customers can see the latest products added to the shop.
Business rules - 7	<ol style="list-style-type: none"><li>1. Seller's products must meet the requirements of the terms and conditions of the company.</li></ol>
Implementation	

constraints and specifications - 8	N/A
Assumptions - 9	The seller has updated the shop.
Open Issues - 10	N/A



## Online retailing system

Author(s): Pavel Petrukhin

Date: 18/10/2020

Version: 1

Use-case name	<b>MakePayment</b>
Use-case type	Business Requirements
Use-case ID	BR - 4
Priority	High
Source	MakePayment
Primary Actor	Customer
Other Actors	Online Checkout
Other Interested Stakeholders	Seller is obviously interested in getting profit
Description	After the user chooses the payment option in the checkout, they are transferred to a particular online checkout. The user enters their credentials and the payment is made. The payment confirmation is sent to the customer's email along with the confirmation of the order.
Precondition - 1	A particular payment method must be chosen
Trigger - 2	The use-case is initiated when the customer clicks 'Pay'
Typical Course of Events - 3	<ol style="list-style-type: none"><li>1. Customer clicks 'Pay' on the checkout page</li><li>2. The customer enters their credentials</li><li>3. Customer clicks confirm payment and the online checkout system sends a query to their bank/online wallet to proceed with the payment</li><li>4. The user has enough money to pay for products, so the transaction is successful</li></ol>
Alternate courses - 4	<ol style="list-style-type: none"><li>1. At stages 1 and 2 of the 'Typical course of the Events' user can cancel the payment if they decided not to buy the products in the end</li><li>2. At stage 3 transaction might be automatically canceled if the user does not have enough money/bank API or the online wallet's API is inaccessible</li></ol>
Conclusion - 5	The use-case concludes when the user is notified of the success/failure of the transaction

Postcondition - 6	If the payment is successful, the order is confirmed. Otherwise, the user is notified that there was a problem with the payment method they chose.
Business rules - 7	<ol style="list-style-type: none"> <li>1. The location of the user may affect the amount of tax applied</li> <li>2. Payments are made before the order is sent</li> <li>3. Payments in cash are not acceptable</li> <li>4. Payments via direct bank transfers are also not acceptable</li> </ol>
Implementation constraints and specifications - 8	<ol style="list-style-type: none"> <li>1. Stripe, an online payment system, should be connected</li> <li>2. Paypal payment option should also be available.</li> </ol>
Assumptions - 9	The user has chosen the products to buy
Open Issues - 10	None currently

## Online retailing system

Author(s): Pascal Raos

Date: 18/10/2020

Version:1

Use-case name	<b>Checkout</b>
Use-case type	Business Requirements
Use-case ID	BR - 5
Priority	High
Source	Checkout
Primary Actor	Customer
Other Actors	<<Service>> Authentication
Other Interested Stakeholders	Seller?
Description	A use case describes the act of listing out products in a basket, and calculates VAT, shipping cost, and allows the addition of promotional codes to lower net price
Precondition - 1	Customer must have items in the basket
Trigger - 2	Initiated when the user clicks on the basket/view basket button
Typical Course of Events - 3	<p>Step 1: User clicks on the view basket button</p> <p>Step 2: The system responds by checking to see if all products are within the stock, calculates discounts, and shipping costs</p> <p>Step 3: The user continues to payment</p>

Alternate courses - 4	<p>Step 1: The user decides to cancel their order</p> <p>Step 2: The user decides to use a promotional code, augmenting the final price</p>
Conclusion - 5	The use case decides the final cost of the combined products and shipping
Postcondition - 6	Allows the user to proceed towards payment
Business rules - 7	<p>1. Some promotional codes are limited-time use or only apply to specific products</p> <p>2. A free delivery may only apply to specific users, and delivery cost may vary depending on the customer's region</p>
Implementation constraints and specifications - 8	Choose a method of payment, must integrate specific methods into the website, (see MakePayment use case for examples)
Assumptions - 9	The user has items in their cart
Open Issues - 10	None currently

Author(s): Pascal Raos

Date: 18/10/2020  
Version: 1

Use-case name	<b>Login</b>
Use-case type	System Requirements
Use-case ID	SR-2
Priority	High
Source	Login
Primary Actor	Customer
Other Actors	<<Service>> Authentication

Other Interested Stakeholders	N/A
Description	A use case describes the event in which a customer attempts to log in onto the website. Check by Authentication is run to see whether the account exists
Precondition - 1	The customer must already have an account
Trigger - 2	Initiated when the user attempts to log in
Typical Course of Events - 3	<p>Step 1: User clicks Log In and provides email/username accompanied by a password</p> <p>Step 2: The system responds by verifying that the user has an account within the system and that the user credentials are correct</p> <p>Step 3: The user signs into his account</p>
Alternate courses - 4	<p>Step 1: The user has provided inadequate credentials, preventing them from logging into their account.</p> <p>Step 2: User is asked if they would like to make an account if they do not already have one, which if they do make a new one, will be asked to verify their newly created account</p>
Conclusion - 5	The use case concludes when the user has logged into their account
Postcondition - 6	Allows the user to access website functionality that other users can not
Business rules - 7	A user's created password must meet an acceptable standard in order to ensure account security

Implementation constraints and specifications - 8	User must have a valid means of making an account, i.e an email
Assumptions - 9	The user has an account
Open Issues - 10	None currently

Author(s): David Olowookere

Date: 19/10/2020

Version: 1.0.0

Use-case name	<b>WithdrawMoney</b>
Use-case type	Business Requirements
Use-case ID	BR - 6
Priority	High
Source	WithdrawMoney
Primary Actor	Seller
Other Actors	OnlineCheckout
Other Interested Stakeholders	The site owners, for keeping track of finances.
Description	This use case describes the event of a seller deciding to withdraw the money they've earned from their shop.
Precondition - 1	The seller must have money to withdraw from their shop.
Trigger - 2	This use case is initiated when a seller decides to withdraw money from their shop.
Typical Course of Events - 3	<p><b>Step 1:</b> The seller clicks "Withdraw" on their account page.</p> <p><b>Step 2:</b> The seller enters how much they want to withdraw and their bank details.</p> <p><b>Step 3:</b> The system verifies that the bank details are correct.</p> <p><b>Step 4:</b> The online checkout system verifies that the seller has at least the amount they want to withdraw in their shop account.</p> <p><b>Step 5:</b> The system sends a query to the seller's bank/online wallet to transfer the money.</p>

	<b>Step 6:</b> The money is transferred, and the system updates the amount in the seller's shop account and sends the seller a receipt.
Alternate courses - 4	<p><b>Alt-Step 2:</b> The seller can choose to cancel the withdrawal. Terminate use case.</p> <p><b>Alt-Step 3:</b> If the bank details are incorrect, the system informs the seller and asks them to resubmit or terminate the use case.</p> <p><b>Alt-Step 4:</b> If the seller does not have enough money to withdraw what they entered, the system informs the seller and asks them to change the amount or terminate the use case.</p> <p><b>Alt-Step 5:</b> If there are issues communicating with the seller's bank, inform the user. Terminate use case to guard against any issues.</p>
Conclusion - 5	The use case concludes when the money has been transferred.
Postcondition - 6	The money is in the seller's bank account, and their shop account accurately reflects how much money they have.
Business rules - 7	<ul style="list-style-type: none"> <li>• The seller cannot request cash or cheques for payment.</li> </ul>
Implementation constraints and specifications - 8	Online payment methods need to be incorporated into the site.
Assumptions - 9	The seller has already logged in (separate use case).
Open Issues - 10	None currently

## 4 - Class diagram

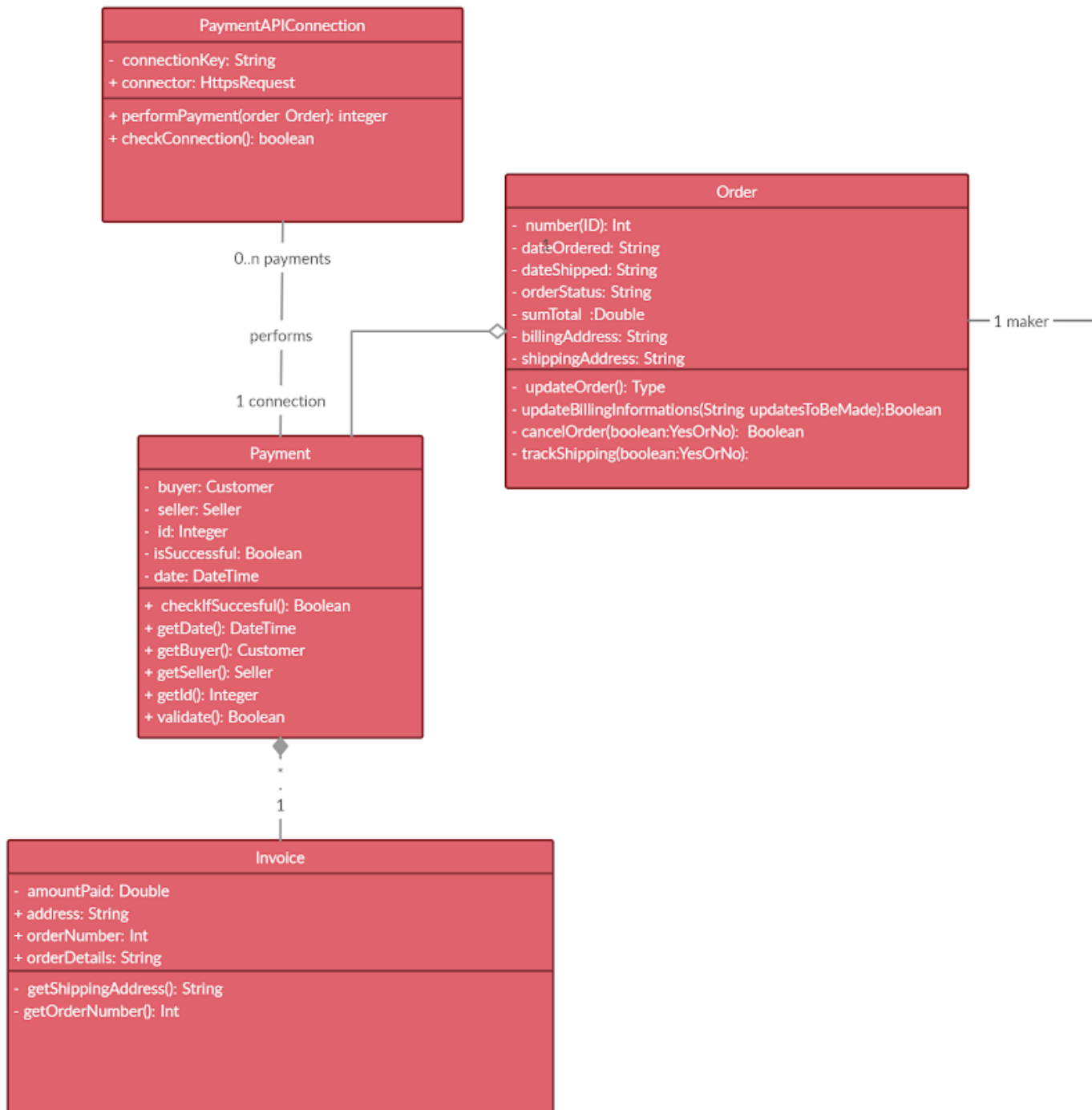
Color coding: Red - payment-related classes, Green - authentication-related classes, Blue - selling related classes.

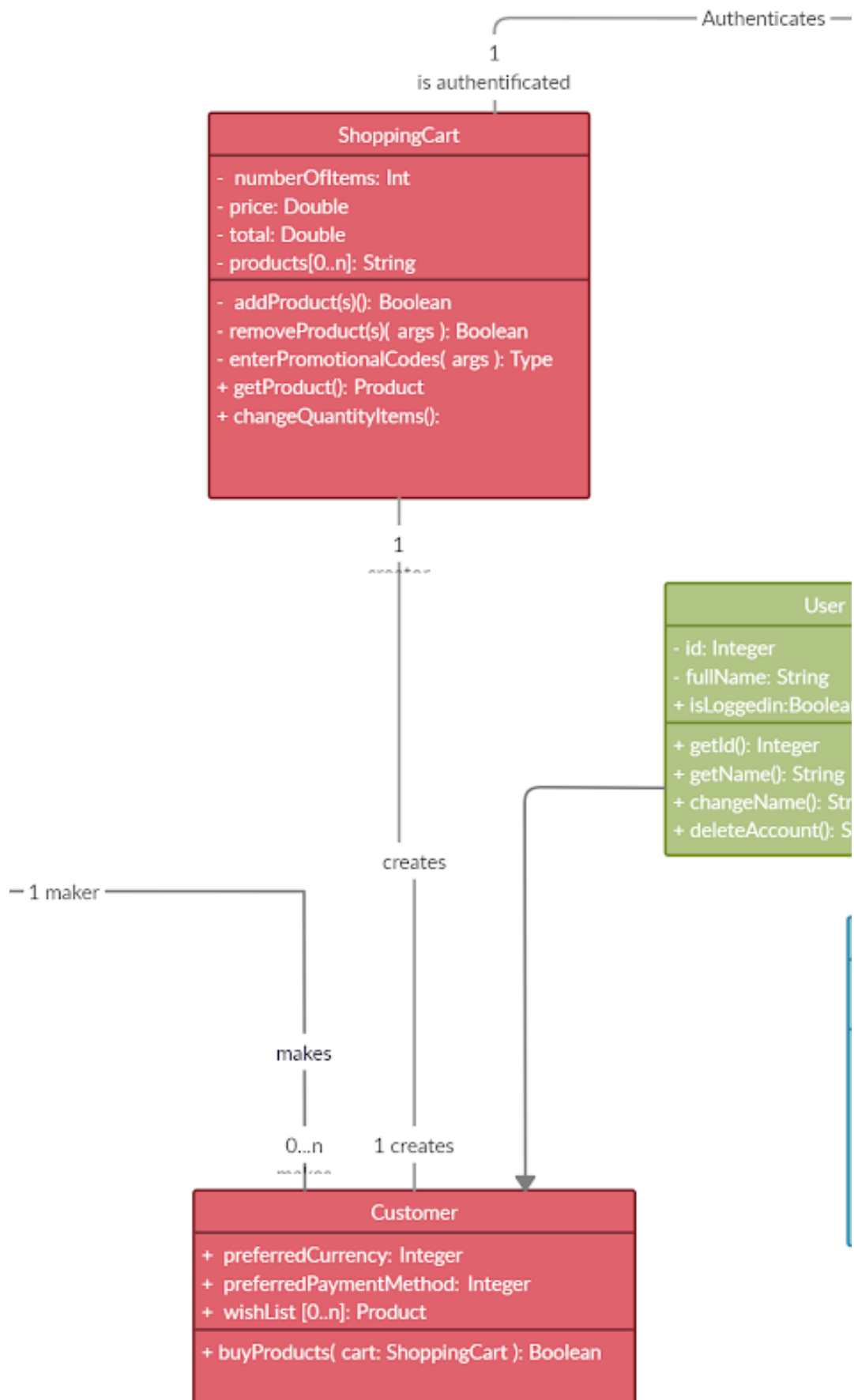
When coming up with classes, we applied the technique suggested in class. We discussed our use-case diagram and system in general and watched for the nouns. With that we came up with quite a few classes, however we wanted to keep the diagram understandable, so we had to agree on which classes to include. For some of the cases, we saw that the some classes we suggest were actually arrays of already existing more basic classes. For example, Wishlist, which would be an array of Products or Reviews page which would be an array of Comments. Moreover, there were situations when the classes were not relevant or crucial for the system, such classes included Sort (already implemented in standard libraries in most of the programming languages), PayedPromotion (which could be added to the system later on with more people using it) and AutomaticDiscount (which, again is not essential for the system to run).

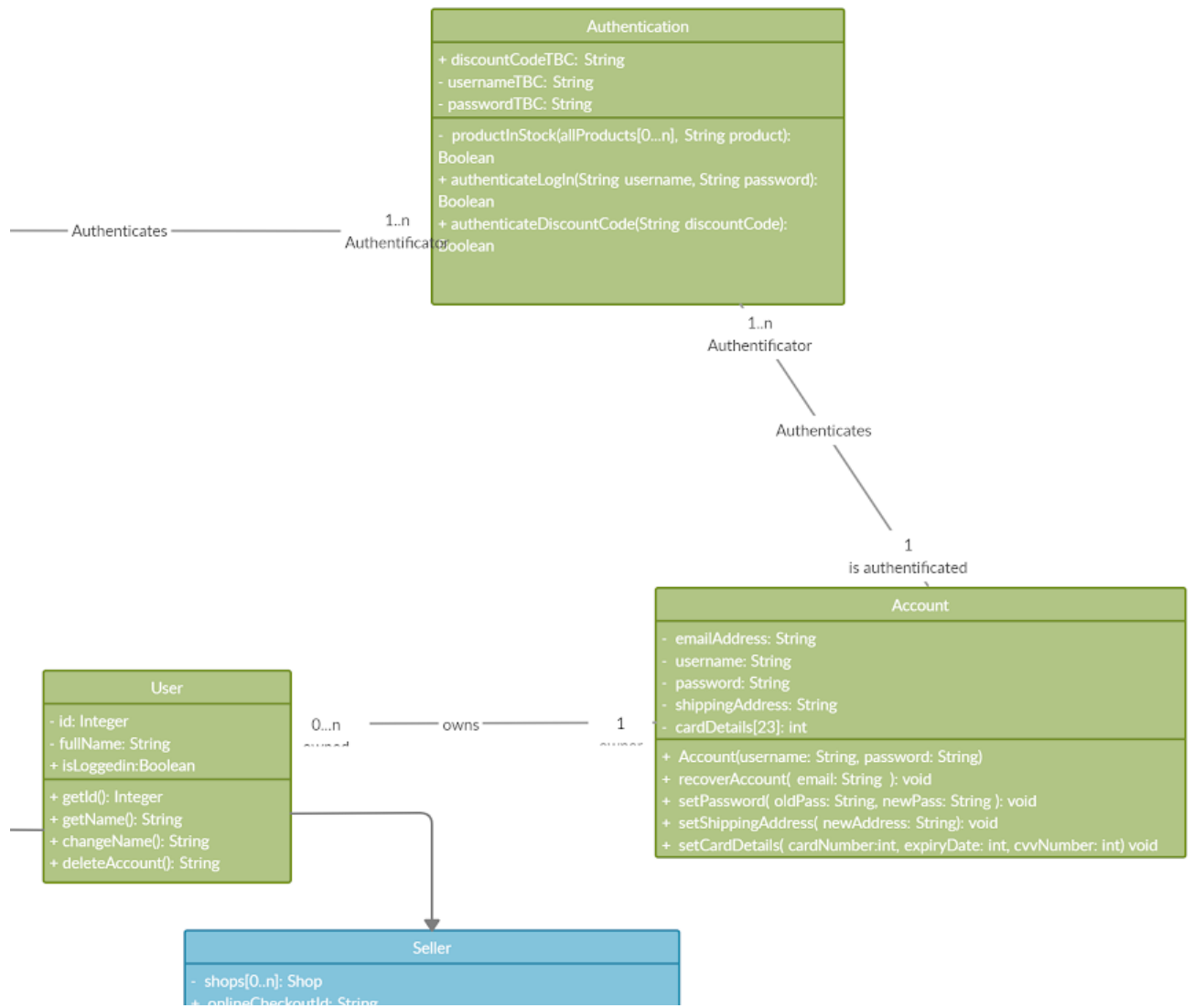
We reviewed the diagram multiple times and discussed the accessibility of each field/method in each class. After looking at the finished diagram, we understood that adding a bit of color coding would make it more readable, so we have split the diagram in three parts as stated above.

Moreover, we carefully reviewed every association to make sure that it reflected something meaningful, because initially we wanted to interconnect pretty much every class with the other (it is always possible to find a reason, because the classes are anyway a part of one integral system, so all of them are somehow connected, although the connections might be indirect). We filtered out only the essential connection and left them in the final version of the diagram.









```

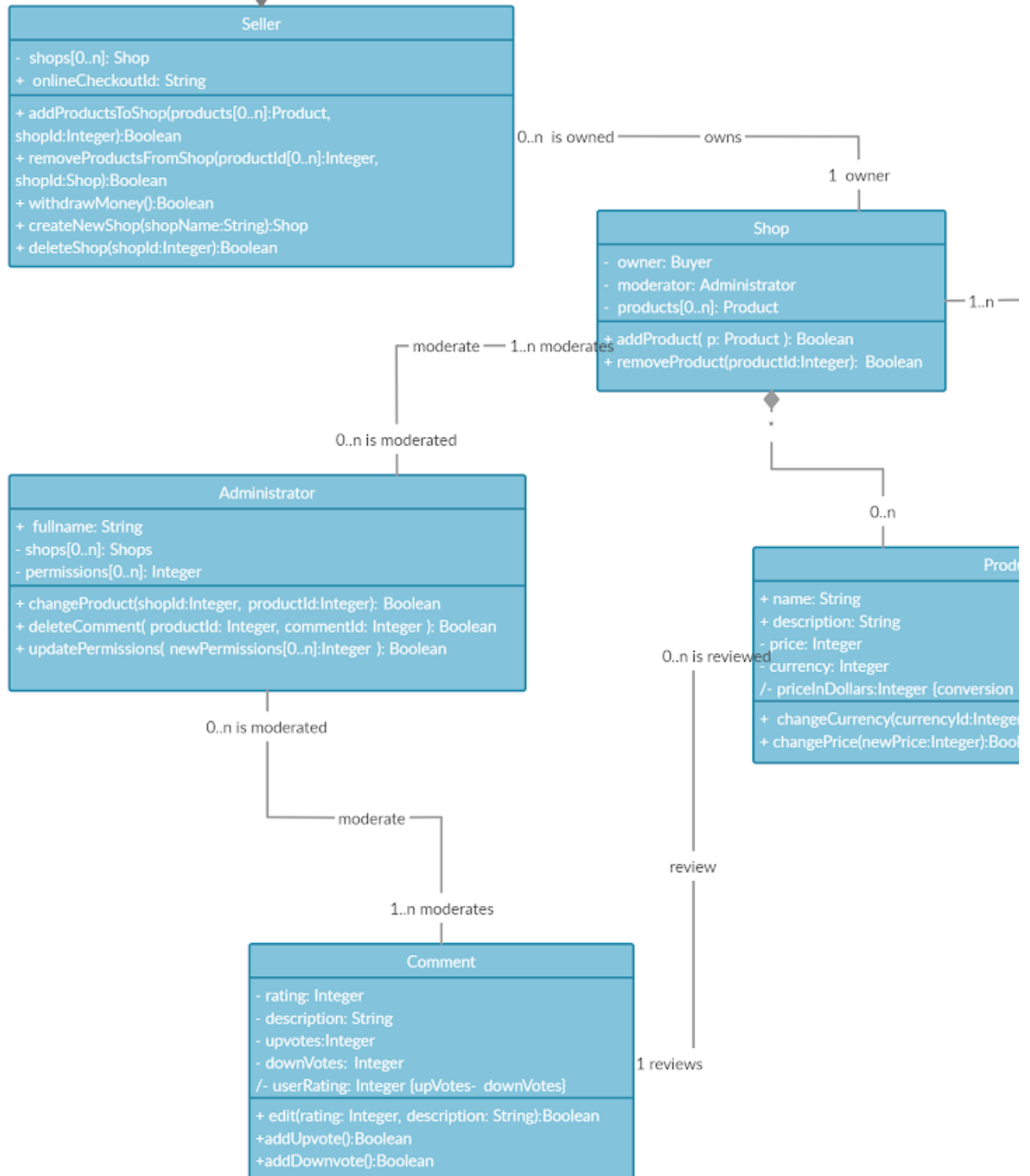
+ String
+ e(): String
+ nt(): String

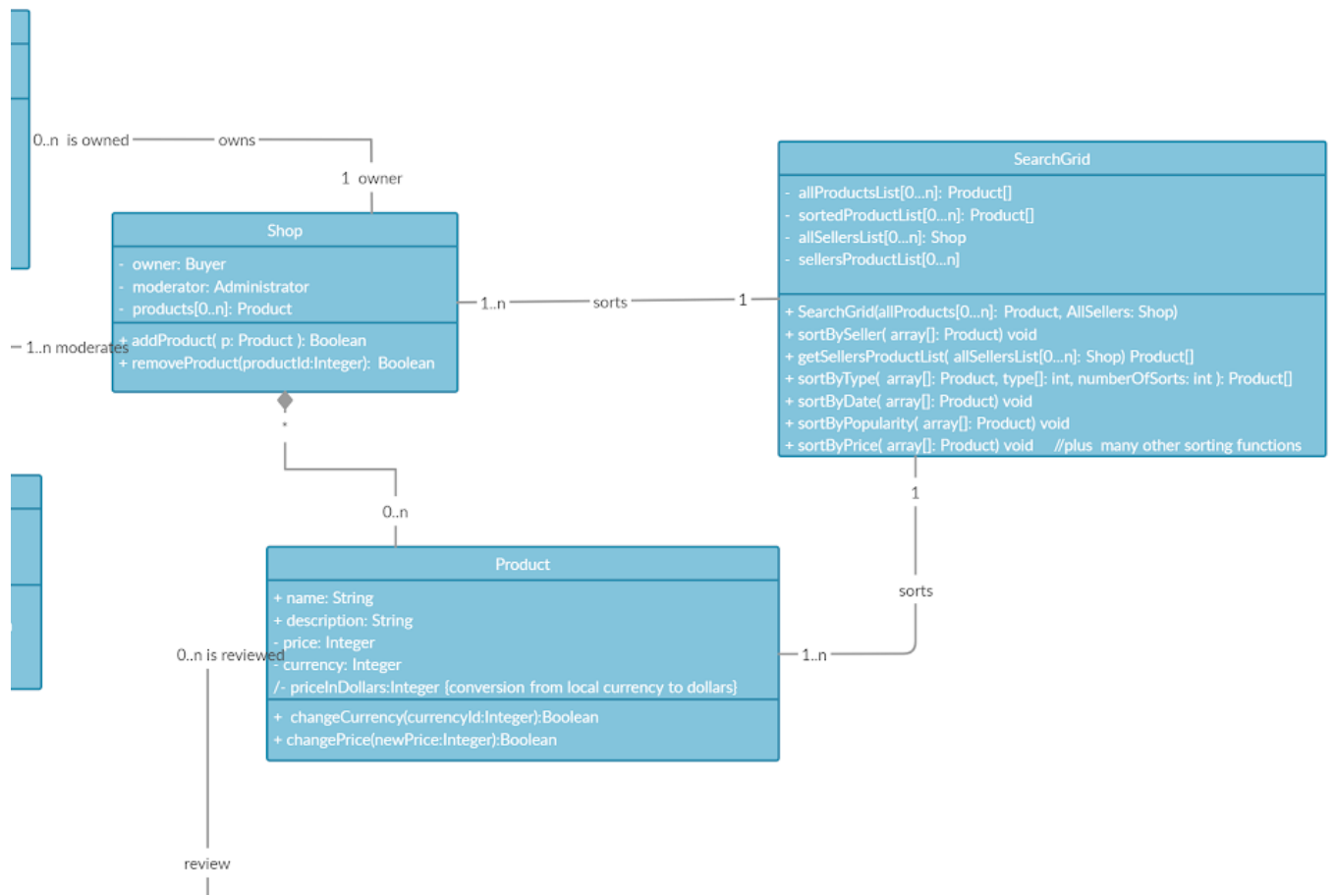
```

```

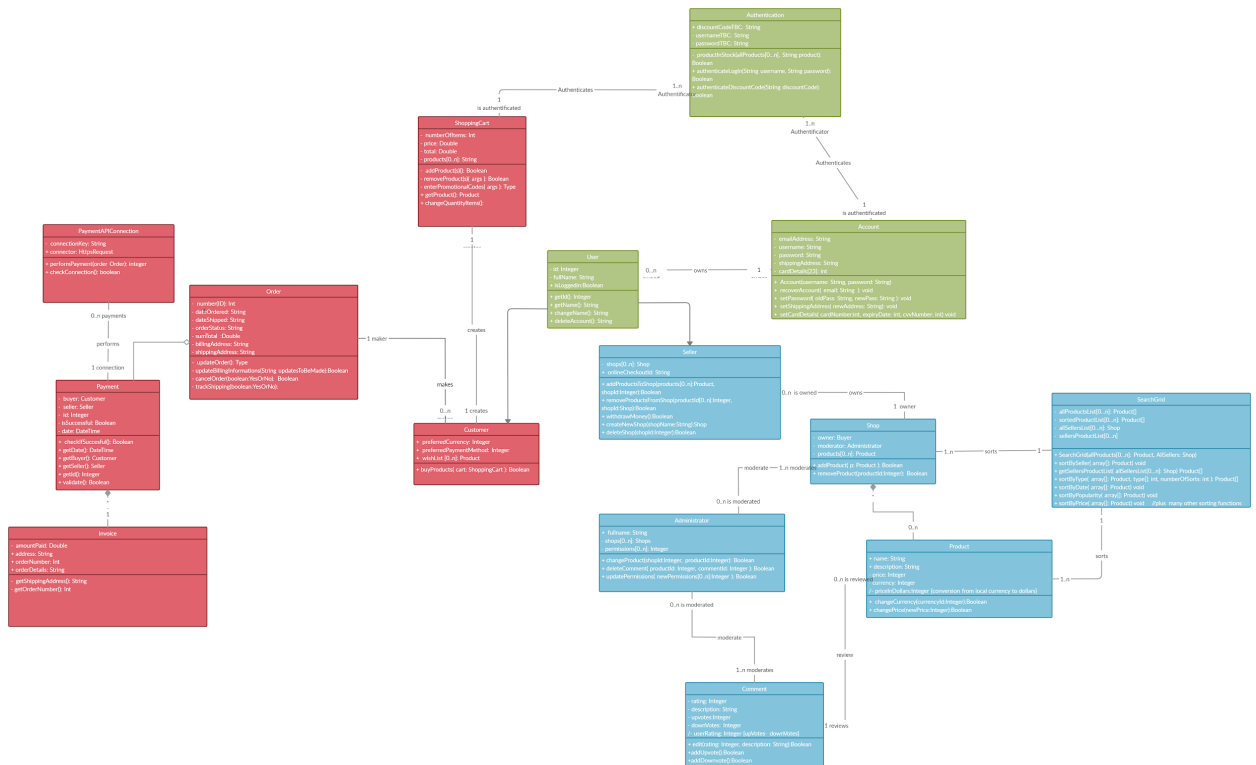
+ setShippingAddress( newAddress: String): void
+ setCardDetails( cardNumber:int, expiryDate: int, cvvNumber: int) void

```





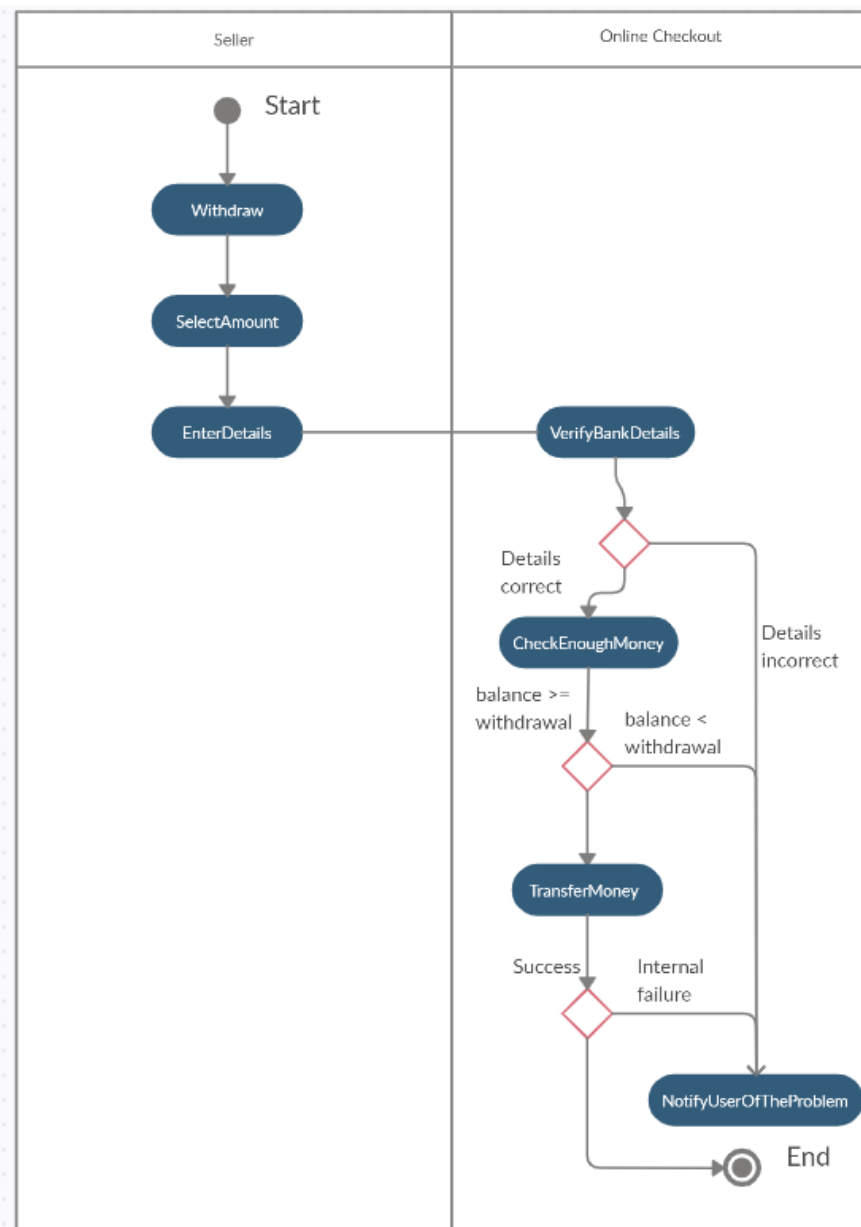
## Full view



## 5 - Activity diagram

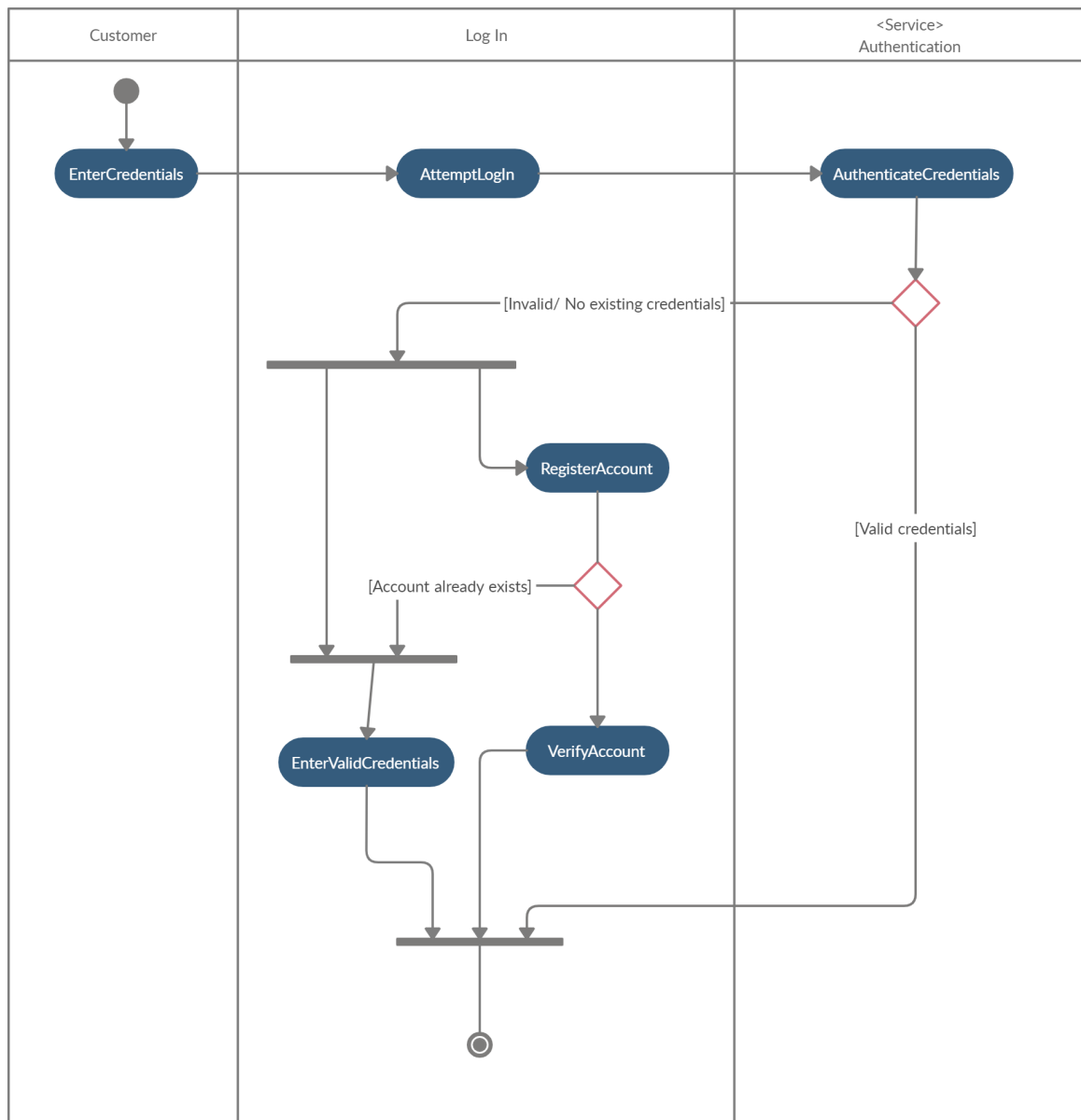
### Activity diagram 1:

The first activity diagram focuses on the interaction between the seller actor and the checkout use case. The seller selects the option to withdraw funds from their respective account (possibly Stripe or PayPal), selects the amount they would like to withdraw and then proceeds to enter their details for verification (password, pin, etc....). The checkout system then checks to verify the details are correct, else the transaction is void and the user gets notified. Once the details have been verified, the system then checks if there is sufficient funds for the withdrawal, once again if there is not the transaction will fail and the user gets notified. Finally once sufficient funds are verified to be available in the account, the system proceeds to transfer the requested amount and return to the seller. Just in case of internal failure due to a system error the system will cancel the transaction and notify the user. So the activity diagram ends with either a successful withdrawal or a notice sent to the user explaining why the transaction failed.



## Activity diagram 2:

The second activity diagram focuses on the interaction between the customer actor and the LogIn useCase. The authentication service is used to verify credentials inputted by the user and allows the user to log in if they are correct, otherwise the user must input valid credentials, or make an account if they do not already have some. The activity diagram ends with a successful login.



## 6 - Contributions

### **Pavel:**

- Researched wildberries.ru
- Worked on the use-case diagram and actively participated in the discussion and reviewing of the diagram
- Wrote two descriptions for the use-cases.
- Created a draft of class diagram and extended it together with the team.
- Created a draft of WithdrawMoney activity diagram and extended it together with the team.
- Recorded a video for the presentation and slightly changed the script written by the team
- Planned work and set approximate deadlines so that project is delivered on time
- Wrote a few points in the Ethics Canvas and Strengths/Weaknesses
- Formatted report (text, pictures and etc.) and added a few paragraphs about our working process (what we did, how we planned, how we reviewed/discussed)

### **Pascal:**

- Researched DoneDeal.ie
- Participated in group discussions regarding all aspects of the project
- Helped with the creation of the use-case diagram
- Wrote two use-case descriptions
- Helped with the creation of the final class diagram
- Created a draft of the Login activity diagram, and extended it with the help of the team
- Participated in the script writing process of the presentation
- Wrote an initial draft for the beginning of the research section of the report

### **Christiana:**

- Researched Aliexpress (Asian online retailing system)
- Actively participated in group discussions, sharing of information and deciding of general organisation
- Contributed to the use case diagram and wrote 2 of the use case descriptions
- Alongside the team worked on class diagram, making 3 classes and contributing to others, also coloured-coded the class diagram
- Contributed to the in the script for the video presentation
- Wrote description of the first activity diagram



**Tumi:**

- Researched Amazon.uk
- Participated in group discussions about the use-case diagrams and the class diagrams.
- Wrote one of the use-case descriptions.
- Worked with the rest of the group to make two classes.
- Wrote a few points on the Ethics Canvas.

**David:**

- Researched Ebay
- Participated in group discussions
- Worked with the rest of the group to create the use case diagram
- Wrote a use case description for Withdraw Money
- Added a number of ideas to Ethics canvas and actively participated in reviewing the class diagram.

## 7 - Strengths and weaknesses

As for the strengths, we have managed to create a visual representation of a complex system, which is fairly readable. Moreover, we/other developers could use the diagram again in the future to understand how the system works or to extend it, because the diagrams are standardized. Finally, the biggest advantage is that we are able to plan out our work, before actually starting coding, which saves us a lot of time.

As for the weaknesses, some of our diagrams have become overwhelming, particularly our class diagram is quite large and a bit messy. We even had to divide it into multiple parts to put in the report.

While the final design looks good, we should not forget the amount of time we had to spend to come up with it. For us it seems that UML diagrams place too much attention to the design of the system, which consumes a lot of time that could be spent on actually developing and testing software. From our experience in the Programming Project module, sometimes features arise during the development process rather than at the planning stage. Therefore, UML diagrams are only applicable for the most essential and basic features of the system.

In view of the above mentioned, it is unclear who actually benefits from the diagrams. While it was stated that the diagrams allow us to explain the design of the system to our customer easier, in most of the tech companies it is the duty of management to work with the customers. Hence the diagrams inarguably simplify the explanation and make it more understandable, but it seems that developers do not really benefit much from it.