TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technologies

Aleksandr Lerko 202027IADB

Online Shopping System

Homework for ICD0021

Supervisor: Andres Käver

Tallinn 2021

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Aleksandr Lerko

17.02.2022

1. Introduction

The main goal of this project is to create an online shopping system application where different users (authorized or unauthorized) can scroll through the webpage or use a searching system to find themselves needed products.

The idea is to create a fully working shopping system where users could theoretically “buy” products, choose different payment (Apple Pay, Google Pay etc.) and delivery (Omniva, SmartPost etc.) methods, get an invoice from the transaction and see who is the seller of this product. Users should have an opportunity to see which local stores have this exact product in stock. User has to be able to send feedback

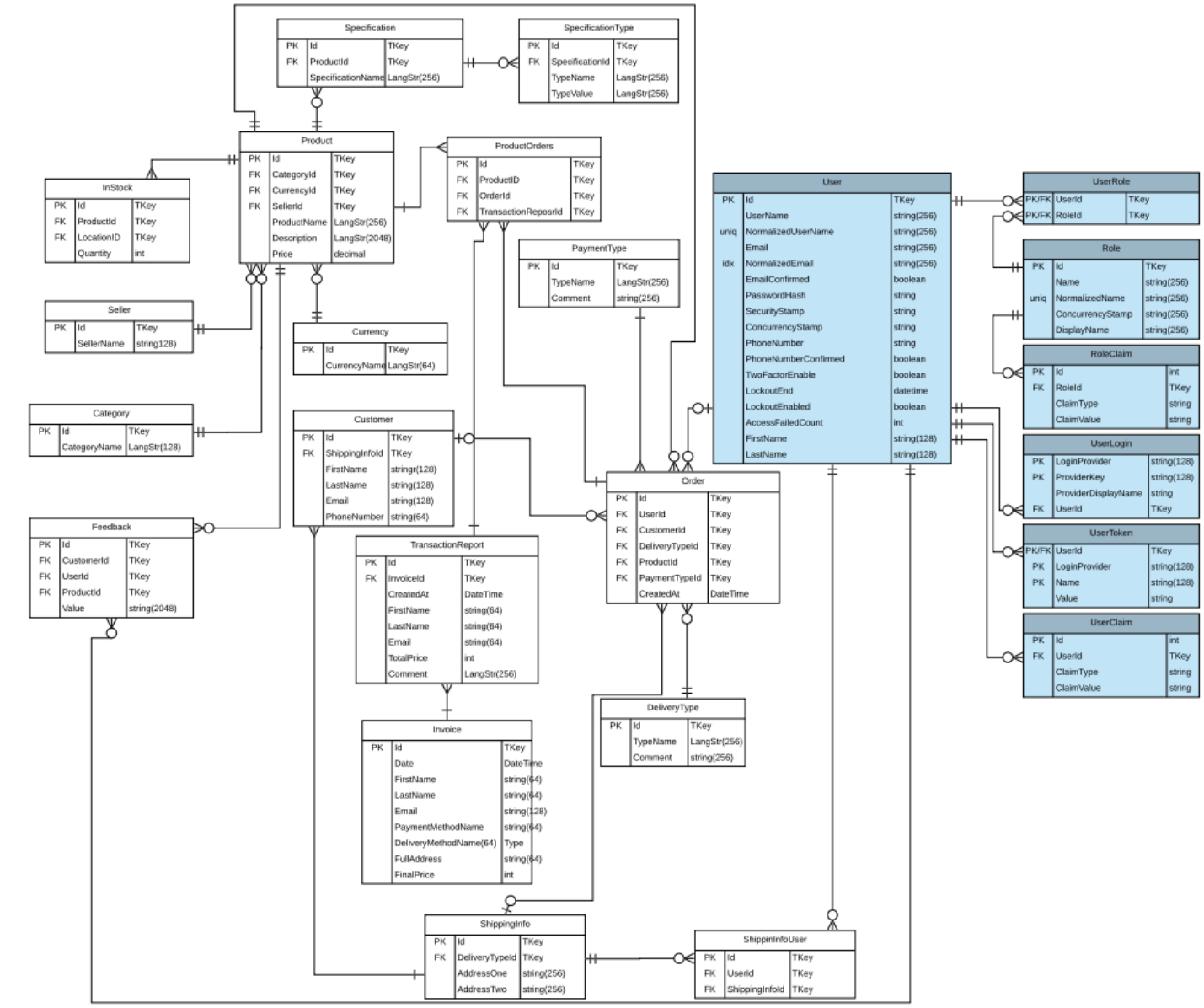
If a user is from a different country or region with a different currency, then the user should have an option to switch currency on the webpage or also if needed language. If the user is authorized so that he can save his payment or delivery methods to use them later.

If a user clicks on a product, then the user should see product info (small product review), technical specifications, price, availability and other users feedback.

Users who have a seller manager role have an option to add, edit or delete products, all the products information etc. Other users without roles can only search for products, watch or buy them.

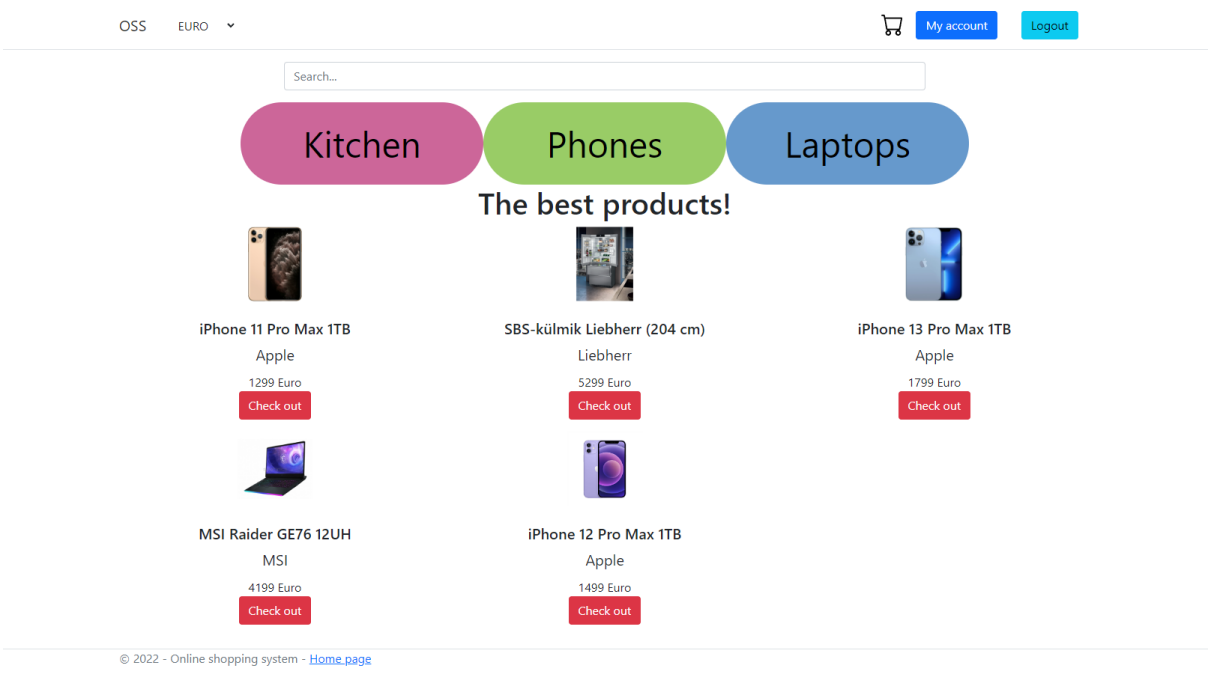
Application supports different languages and cultures. If a user is from another country even with a different language type (e.g., russian), so users from Russia should have an opportunity to switch between languages and see all the information in their native language. Even if both users speaks one language (e.g., english), but they are from different countries (e.g., USA and UK), so that they have to see their culture differences.

2. ERD schema

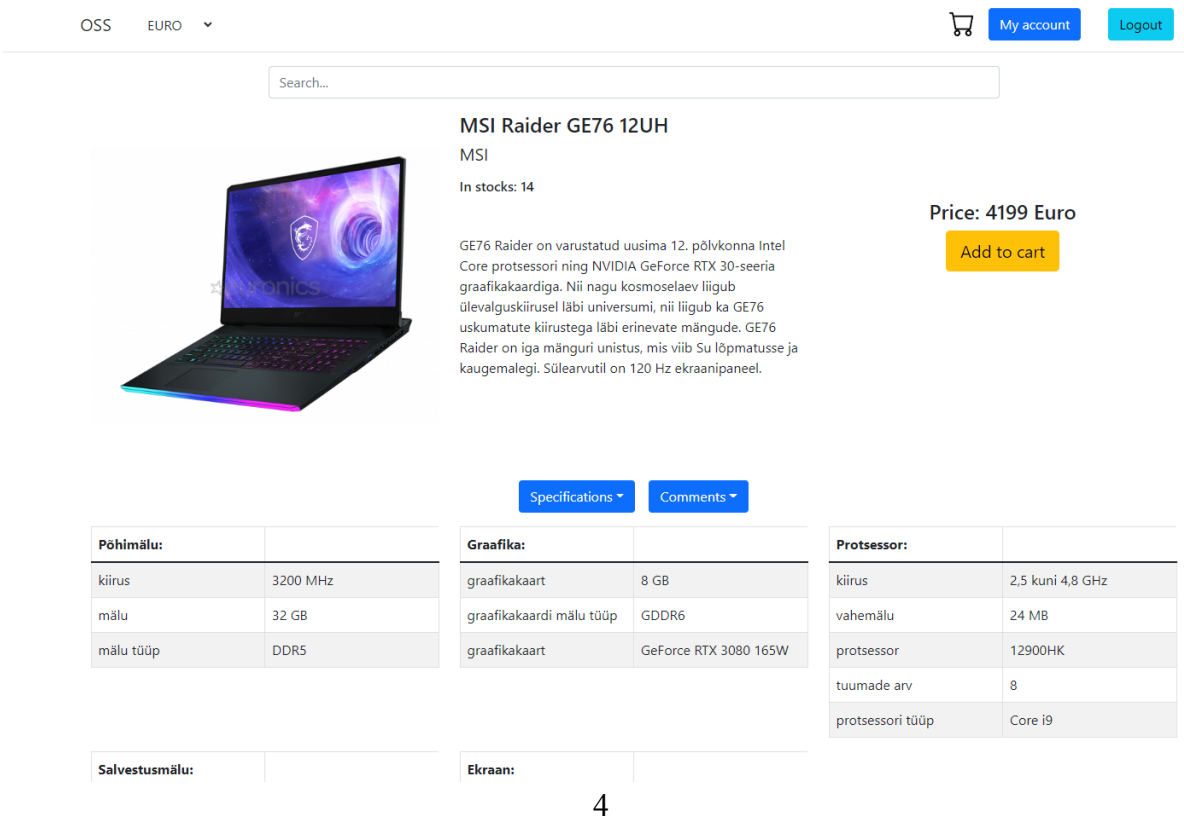

Specification
PK Id TKey
FK ProductId TKey
SpecificationName LangStr(256)
SpecificationType
PK Id TKey
FK SpecificationId TKey
TypeName LangStr(256)
TypeValue LangStr(256)
Product
PK Id TKey
FK CategoryId TKey
FK CurrencyId TKey
FK SellerId TKey
ProductName LangStr(256)
Description LangStr(2048)
Price decimal
ProductOrders
PK Id TKey
FK ProductID TKey
FK OrderId TKey
FK TransactionReposrId TKey
InStock
PK Id TKey
FK ProductId TKey
FK LocationID TKey
Quantity int
PaymentType
PK Id TKey
TypeName LangStr(256)
Comment string(256)
Seller
PK Id TKey
SellerName string128)
Currency
PK Id TKey
CurrencyName LangStr(64)
Category
PK Id TKey
CategoryName LangStr(128)
Customer
PK Id TKey
FK ShippingInfold TKey
FirstName stringr(128)
LastName string(128)
Email string(128)
PhoneNumber string(64)
Order
PK Id TKey
FK UserId TKey
FK CustomerId TKey
FK DeliveryTypeId TKey
FK ProductId TKey
FK PaymentTypeId TKey
CreatedAt DateTime
Feedback
PK Id TKey
FK CustomerId TKey
FK UserId TKey
FK ProductId TKey
Value string(2048)
TransactionReport
PK Id TKey
FK InvoiceId TKey
CreatedAt DateTime
FirstName string(64)
LastName string(64)
Email string(64)
TotalPrice int
Comment LangStr(256)
DeliveryType
PK Id TKey
TypeName LangStr(256)
Comment string(256)
Invoice
PK Id TKey
Date DateTime
FirstName string(64)
LastName string(64)
Email string(128)
PaymentMethodName string(64)
DeliveryMethodName(64) Type
FullAddress string(64)
FinalPrice int
ShippingInfo
PK Id TKey
FK DeliveryTypeId TKey
AddressOne string(256)
AddressTwo string(256)
ShippinInfoUser
PK Id TKey
FK UserId TKey
FK ShippingInfold TKey
User
PK Id TKey
UserName string(256)
uniq NormalizedUserName string(256)
Email string(256)
idx NormalizedEmail string(256)
EmailConfirmed boolean
PasswordHash string
SecurityStamp string
ConcurrencyStamp string
PhoneNumber string
PhoneNumberConfirmed boolean
TwoFactorEnable boolean
LockoutEnd datetime
LockoutEnabled boolean
AccessFailedCount int
FirstName string(128)
LastName string(128)
UserRole
PK/FK UserId TKey
PK/FK RoleId TKey
Role
PK Id TKey
Name string(256)
uniq NormalizedName string(256)
ConcurrencyStamp string(256)
DisplayName string(256)
RoleClaim
PK Id int
FK RoleId TKey
ClaimType string
ClaimValue string
UserLogin
PK LoginProvider string(128)
PK ProviderKey string(128)
ProviderDisplayName string
FK UserId TKey
UserToken
PK/FK UserId TKey
PK LoginProvider string(128)
PK Name string(128)
Value string
UserClaim
PK Id int
FK UserId TKey
ClaimType string
ClaimValue string

3. Screens

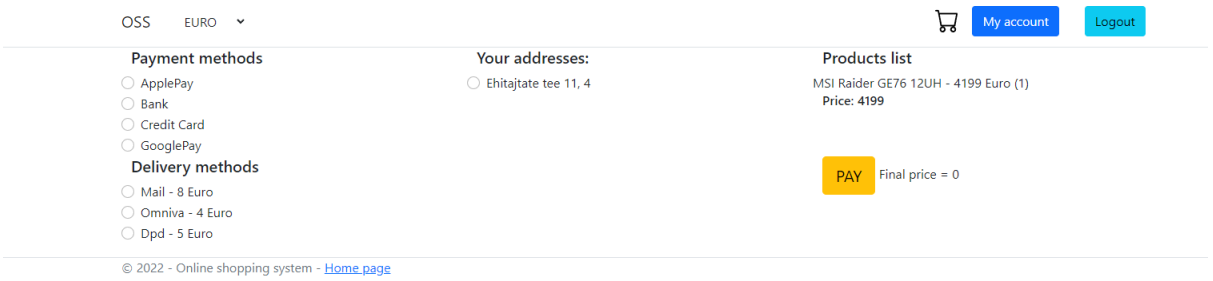
Front page



Product page



Payments page



Analyze

Application was created to provide a possibility for users to buy a product from an online shopping store.

In the rest backend was created a decent system with repositories, services and dto's for every layer. Security was integrated to the system as well, so that every user will see only it's own data (if this option is needed) and didn't have opportunity to execute some methods if he is unauthorized.

For the front end development it was decided to use JavaScript and React framework. React provides enough for the goals of this project and has given quite well tools to state all the data.

Images of the front and backend were created in docker container and the deployed to Microsoft Azure.

Architecture

A way of the user input data via the rest api controllers to the database was separated using repositories, unitofwork, services and mappers between all the layers.

UnitOfWork

Provides all the base options to save changes to db and connect repositories with needed mappers.

Repositories

Provides basic CRUD operations with other custom methods to successfully map and get/set data to/from user.

Services

Layer which is located between rest apis and repositories which provides additional business logic of the application.