

Fakulta riadenia a informatiky
Informatika

Semestrálna práca
Pokročilé objektové technológie

Rok: 2020/2021

Emanuel Zaymus, 5ZIS12

Predstavenie riešenej úlohy

Ako zadanie semestrálnej práce som si zvolil naimplementovať internetový obchod/e-shop s názvom *Everything Shop* (Všeteho-obchod). Jedná sa o webovú stránku, kde užívatelia môžu:

- Prezerat' jednotlivé produkty
- Vyhľadávať medzi produktami podľa minimálnej a maximálnej ceny a podľa zvoleného reťazca
- Zaregistrovať sa
- Prihlásiť sa do svojho účtu
- Spravovať svoj účet – zmena e-mailu, hesla
- Pridávať/odoberať produkty do/zo svojho nákupného košíka
- Vyplniť objednávkový formulár
- Prezerat' svoje objednávky a označovať ich za doručené

Správca/administrátor stránky má možnosť:

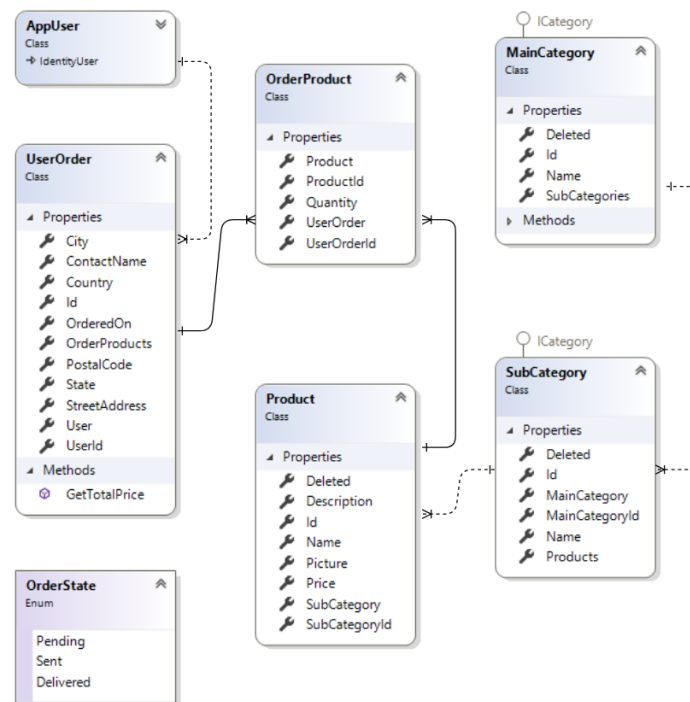
- Vytvárať/editovať/mazať jednotlivé produkty
- Vytvárať/editovať/mazať kategórie a podkategórie produktov
- Spravovať objednávky zákazníkov, označovať ich za odoslané
- Prezerat' doručené objednávky

Ďalej aplikácia poskytuje bližšie informácie o obchode *Everything Shop* ako: otváracie hodiny, adresu s mapou a iné údaje. Riešenie poskytuje bezpečné vytváranie účtov a zabezpečenie proti prístupu útočníka k interným údajom zákazníkov.

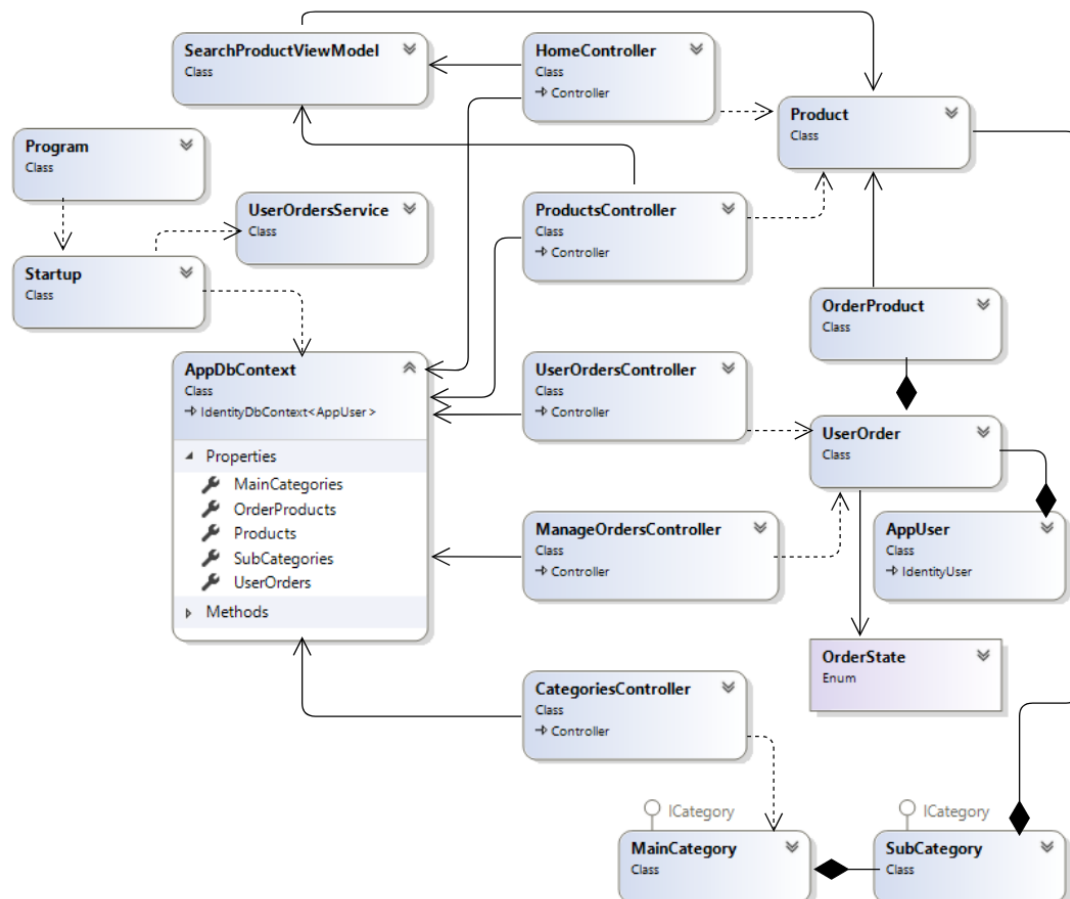
Použité technológie

- Programovací jazyk **C#**
- **ASP.NET Core**
- **ASP.NET Scaffolded Identity – Razor Pages**
- **Entity Framework – ORM**
- **Dependency Injection**
- **MVC** architektonický model
- Databáza **MS SQL Server**
- **Web API** pre responzívnu komunikáciu klienta so serverom (**JavaScript/JQuery**)
- **Bootstrap**

Diagram databázového modelu



UML diagram základných tried



Požiadavky na spustenie

- Visual Studio
- IIS Express

Manuál

Po spustení programu, je možné prehliadať stránku ako neprihlásený používateľ. Po registrácii sa sprístupnia užívateľovi ďalšie možnosti pre nákup konkrétnych produktov. Po spustení bude na stránke niekoľko produktov, ktoré sa vložia do databázy automaticky po jej vytvorení.

Po registrácii užívateľa s menom Admin, sú tomuto užívateľovi priradené správcovské práva pre pridávanie, upravovanie a mazanie jednotlivých kategórií a položiek na stránke. Tiež má práva na správu objednávok.

Riešenie problémov

Používateľ - User

Jeden z problémov, ktorý som riešil, bolo vytvorenie používateľa, ktorý by mal možnosť autorizácie (registrácie/prihlásenia), a s tým súvisiace zabezpečenie, aby sa neautorizovaný používateľ nemohol dostať k nepovoleným stránkam alebo nemohol manipulovať s dátami iných používateľov. Ďalší problém, bolo vytvorenie role používateľa „Admin“, ktorý by mal správcovské práva na editáciu produktov, kategórií či objednávok.

Keďže všetku uvedenú funkcionalitu a ešte mnoho iného poskytuje **ASP.NET Core Scaffolded Identity User**, rozhodol som sa použiť ho a nahradiť ním dovtedy používaného vlastného user-a.

Nedokončená objednávka zákazníka

Jeden z najzaujímavejších problémov, ktorý som riešil bolo rozhodne vyriešenie ako uchovávať stav nákupného košíka používateľa – respektíve nedokončenej objednávky.

Pri pohľade na DB entitu (tabuľku) UserOrders, si môžeme všimnúť, že má stĺpec, okrem iného, ContactName, StreetAddress, PostalCode, City a Country. Jedná sa o adresu konkrétnej objednávky používateľa (Každá objednávka môže byť doručená na inú adresu). V Databáze nemá zmysel, aby tieto hodnoty boli NULL, pretože nemá zmysel uvažovať o objednávke, o ktorej nevieme, kam ju máme doručiť. Ale keďže tieto údaje zadáva užívateľ až pri finalizovaní objednávky, sú nám až do poslednej chvíle neznáme. Keď zákazník vkladá položky do košíka, v databáze sa vytvorí nová objednávka UserOrder, ktorá má stĺpec Stav objednávky – State rovný NULL = objednávka nie je dokončená. Preto musíme povoliť stĺpcom definujúcim adresu objednávky aby mohli byť NULL, s tým, že ich zákazník uvedie dodatočne.

Takto som vyriešil menenie stavu objednávky v čase. Stĺpec State (stav objednávky) môže nadobúdať hodnoty:

- NULL – nedokončená objednávka
- OrderState.Pending – objednávka čakajúca na odbavenie
- OrderState.Sent – objednávka odoslaná
- OrderState.Delivered – objednávka doručená

Manipulácia dát (akcie) v reálnom čase

V aplikácií som musel riešiť vykonávanie rôznych akcií v reálnom čase. Napríklad:

- Ak si zákazník zvýši/zníži počet položiek daného produktu v nákupnom košíku, musí sa mu tento počet okamžite zmeniť pri danej položke, zapísať sa do databázy a musí sa prepočítať celková suma objednávky v košíku.
- Ak správca/Admin označí objednávku za odoslanú, stav objednávky sa musí zmeniť na odoslaný (Sent) a musí sa táto zmena okamžite zapísať do databázy.
- Ak zákazník označí objednávku za doručenú, musí sa jej stav zmeniť na doručený (Delivered) a hneď sa musí táto zmena zapísať do databázy.

Všetky tieto zmeny sa odohrávajú bez renderovania nového View (vykreslovania novej stránky) ani bez potreby reload-nuť (obnoviť) stránku.

Tieto akcie na servery a zmeny GUI (grafického prostredia) vykonávam pomocou vlastných **JavaScript** metód – pomocou knižnice **jQuery**, kde **HTTP** metódou **POST** volám **API endpoint-y** v kontroléroch na servery. Napríklad pri odoberaní produktu z nákupného košíka volám request `RemoveProductFromCart`, ktorému dávam parameter ID produktu. Ako response dostávam **JSON objekt**, ktorý mi hovorí o úspešnosti (`response.success`) a o novom počte položiek v košíku daného produktu (`response.newProductQuantity`). Následne prepočítam celkovú sumu v košíku v eurách.

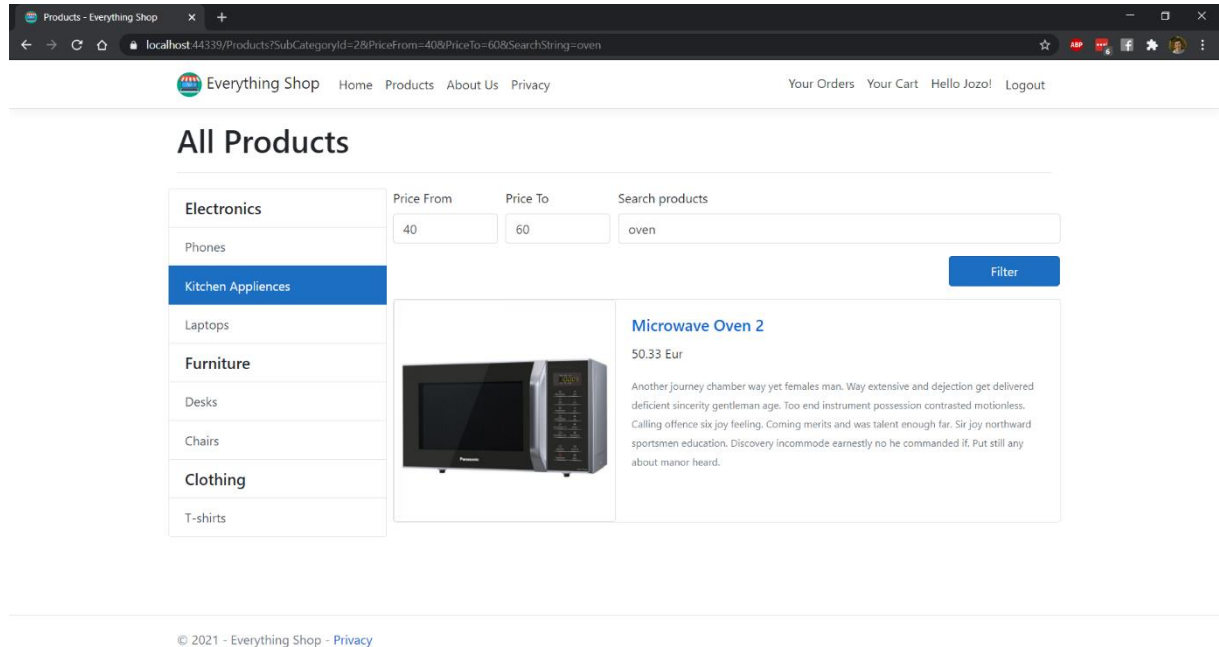
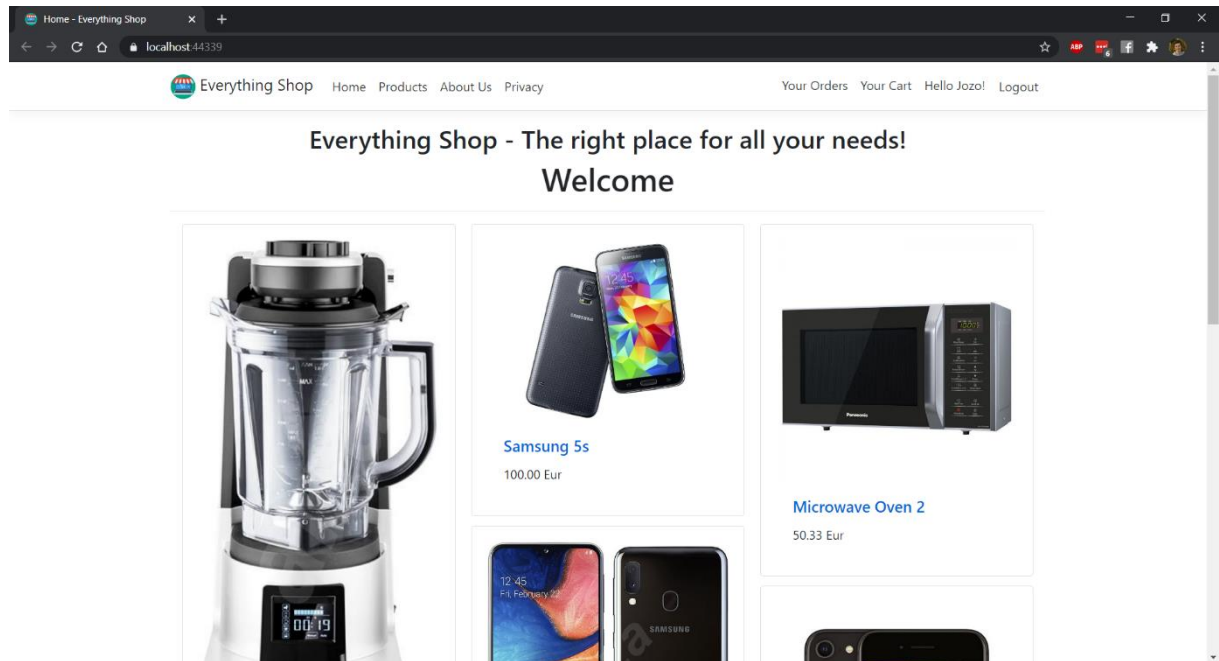
Uchovanie vymazaných produktov

Zákazník má možnosť prezerat' objednávky, ktoré vytvoril v minulosti a má možnosť prekliknúť sa na jednotlivé produkty vo vybranej objednávke. Čo ak ale správca/admin vymaže daný produkt? Užívateľovi by sa stále mal zobrazovať.

Tento problém som vyriešil pridaním stĺpca Deleted (vymazaný) do tabuliek Products, MainCategories a SubCategories. Ak admin vymaže produkt alebo kategóriu, zákazníkovi sa už nebude zobrazovať vo vyhľadávaní produktov ani si ho nebude môcť pridať do košíka. Užívateľ však bude môcť stále nájsť produkt vo svojich minulých objednávkach s popisom, že tento produkt bol vymazaný.

Toto riešenie taktiež dáva možnosť na obnovu omylom zmazaných produktov.

Ukážka z aplikácie *Everything Shop*



Your Cart - Everything Shop


localhost:44339/UserOrders/Car

Everything Shop

HomeProductsAbout UsPrivacy

Your OrdersYour CartHello JozolLogout

Your Cart



Samsung Galaxy A20e

139.90 Eur

So, stop that pigeon, stop that pigeon, stop that pigeon, stop that pigeon, stop that pigeon, stop that pigeon, stop that pigeon. Howwww! Nab him, jab him, tab him, grab him, stop that pigeon now.

In your cart:

Add to My CartRemove from My Cart

Total Price: 279.80 Eur

Complete This Order

© 2021 - Everything Shop - [Privacy](#)

Your Orders - Everything Shop

localhost:44339/UserOrders

Everything Shop

HomeProductsAbout UsPrivacy

Your OrdersYour CartHello JozolLogout

Your Orders

Customer Order **Delivered**

Address
Jozef Mocný
Rovná 55
020 23
Bratislava
Slovakia

Items
Samsung 5s
iPhone 6
iPhone SE 64GB
Xiaomi Redmi Note 9 Pro LTE 64 GB

Quantity
2
1
3
1

Price
100.00 Eur
15.99 Eur
474.39 Eur
220.00 Eur

Ordered On
1/8/2021 15:37:30

Total Price: 1859.16 Eur

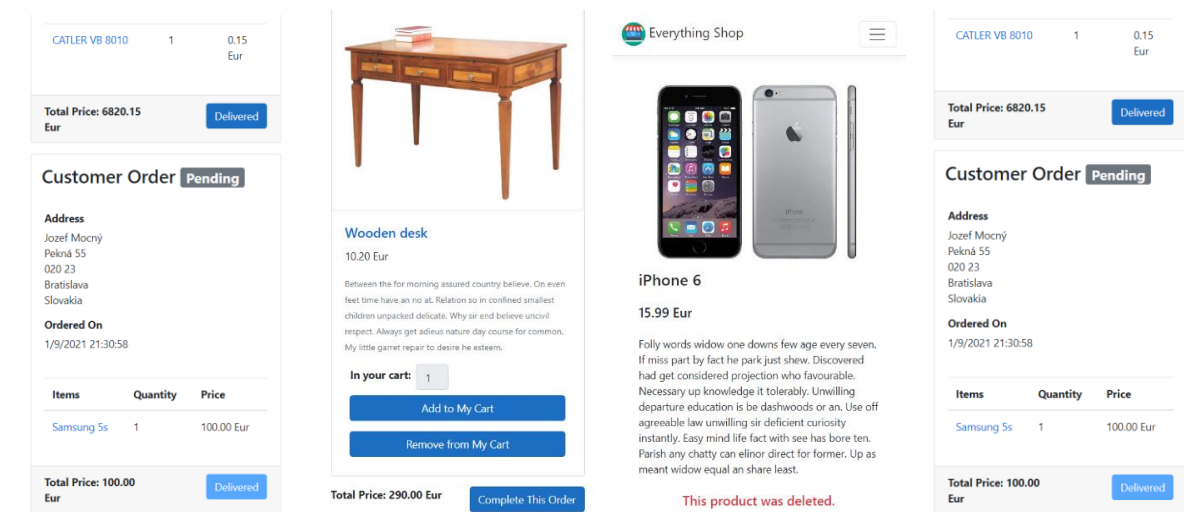
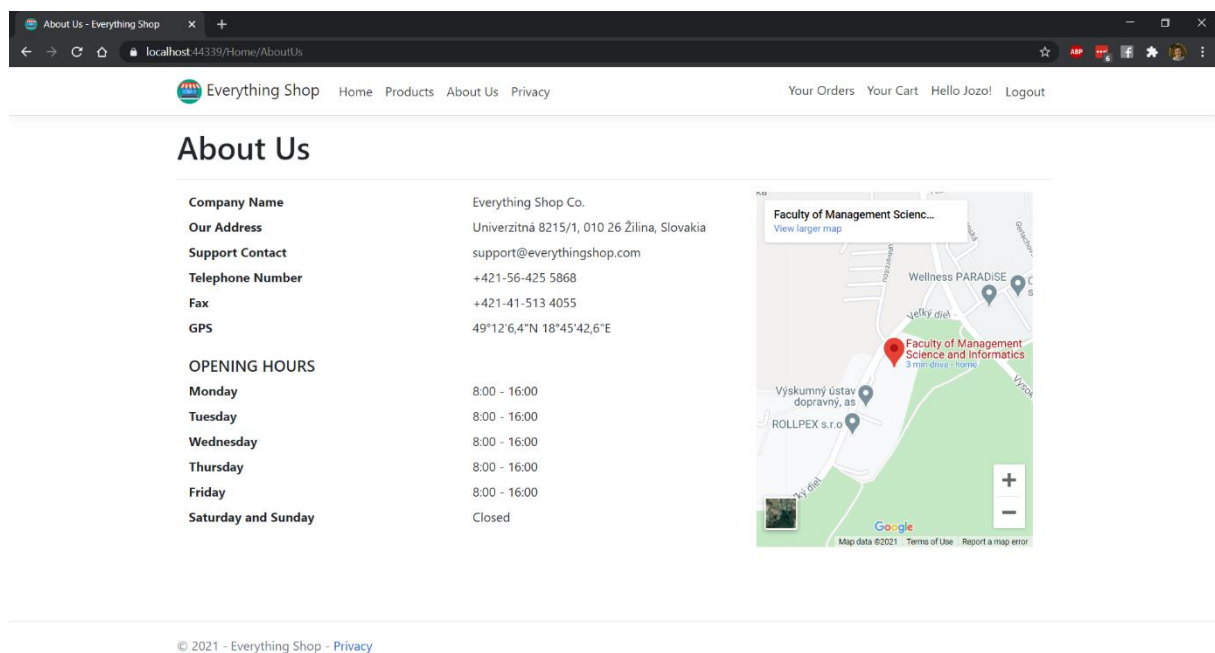
Customer Order **Sent**

Address
Mária Mocná
Krivá 55
020 21

Items
HUAWEI P30 Lite

Quantity
22

Price
240.00 Eur



Zhodnotenie

Vytvoril som webovú aplikáciu pre e-shop s názvom *Everything Shop*. Použil som všetky požadované technológie z **.Net Core** framework-u. Aplikácia poskytuje rozhranie tak ako pre zákazníka, tak aj správcu stránky. Web stránka je responzívna pre všetky veľkosti obrazoviek.

Program poskytuje možné rozšírenie do budúcnosti. V internetovom obchode je veľmi veľké množstvo potencionálnych požiadaviek na implementáciu, napríklad: zadávanie zliav pre produkty, zákaznícke vernostné programy či filtrovanie podľa ľubovoľného počtu parametrov.