

Univerzitet u Beogradu
Fakultet organizacionih nauka
Laboratorija za softversko inženjerstvo

Seminarski rad iz predmeta Napredne softverske tehnologije

Tema: Razvoj softverskog sistema za online kupovinu
primenom .NET Core tehnologija

Mentor:
prof dr. Saša D. Lazarević

Student:
Katarina Simić 2020/3701

Beograd, 2020.

1. Prikupljanje zahteva

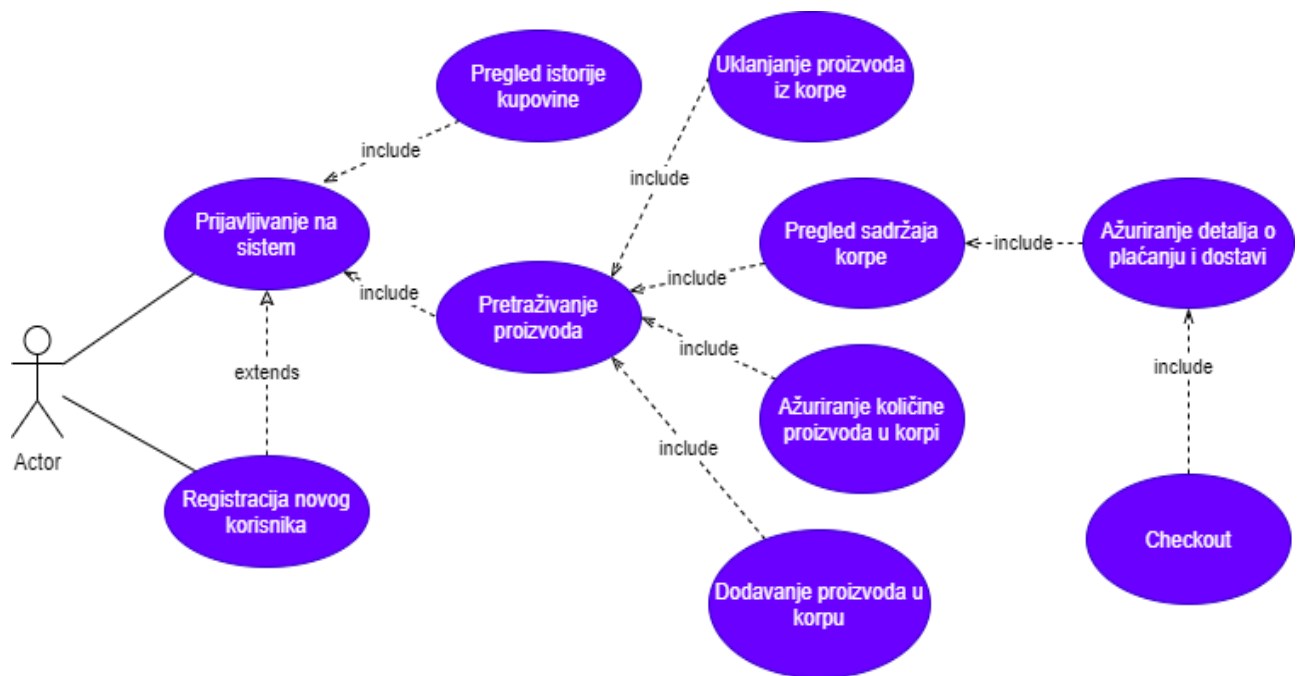
Verbalni opis

Softverski sistem za online kupovinu predstavlja primer veb aplikacije koja omogućava korisnicima kupovinu vina različitih proizvođača. Podaci o proizvodima nalaze se u katalogu vina koji se prikazuje korisnicima. Korisnik se prijavljuje na sistem koristeći svoje kredencijale, a ukoliko nema svoj profil, potrebno je da se registruje. Korisnik može pretraživati katalog vina i dodavati proizvode u svoju korpu. Potrebno je omogućiti korisniku pregledanje sadržaja korpe, uklanjanje proizvoda iz nje i ažuriranje količine proizvoda koji se nalaze u korpi. Kada se odluči za kupovinu, potrebno je da korisnik popuni formu sa neophodnim podacima o plaćanju i dostavi i najzad da klikom na dugme potvrdi kupovinu. Za određene korisnike postoje kuponi pomoću kojih korisnik ostvaruje popust na kupovinu. Korisnik u svakom trenutku može videti istoriju naručivanja.

Model slučajeva korišćenja

Na osnovu verbalnog modela identifikovani su neki od slučajeva korišćenja:

- Registracija korisnika
- Pretraga kataloga
- Prijava korisnika na sistem
- Dodavanje proizvoda u korpu
- Brisanje proizvoda iz korpe
- Ažuriranje proizvoda iz korpe
- Pregledanje sadržaja korpe
- Ažuriranje detalja o naplati i dostavi
- Naručivanje proizvoda (checkout)
- Pregled istorije kupovine

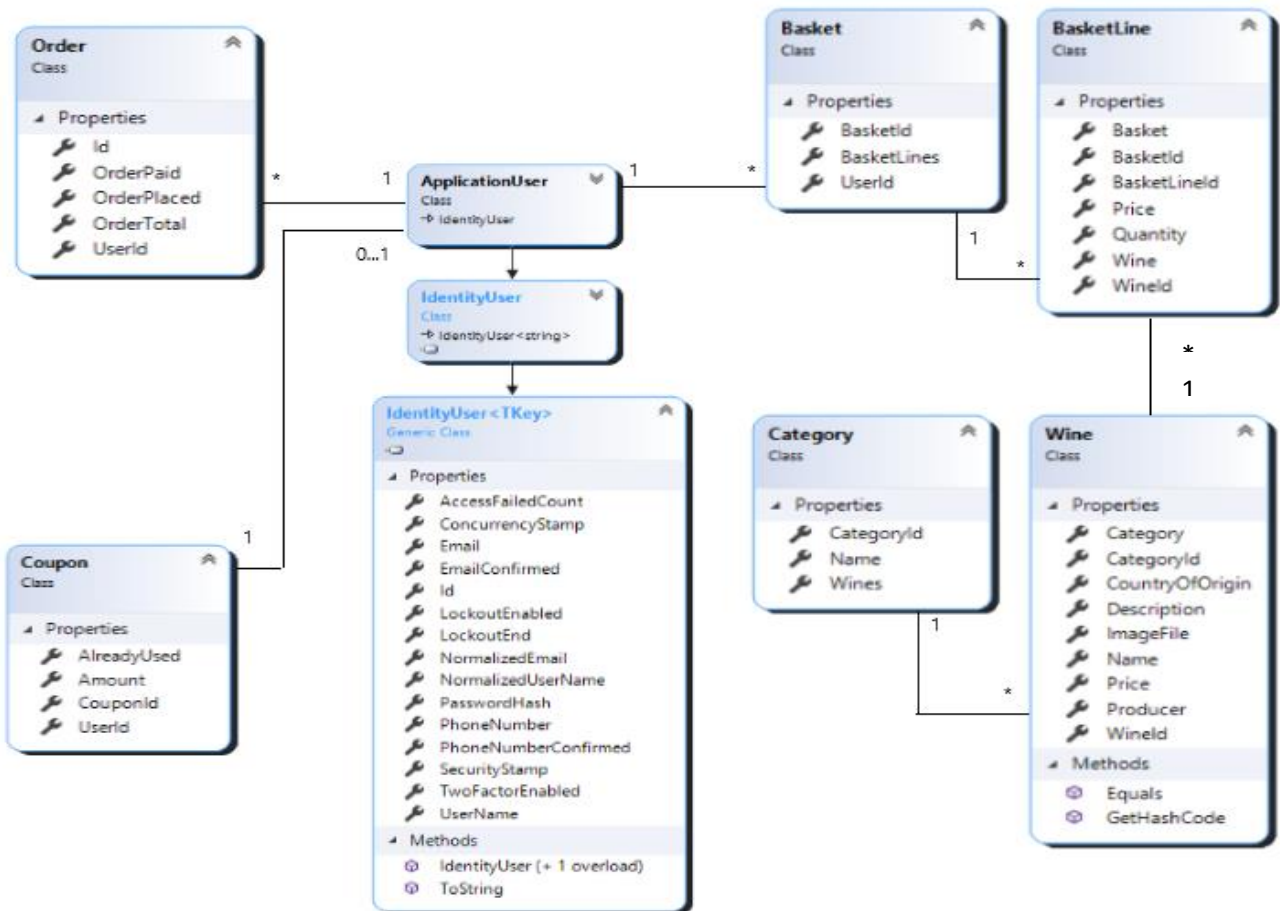


Slika 1 – Dijagram slučajeva korišćenja

2. Analiza

Dijagram klasa

Na sledećoj slici prikazan je konceptualni model sistema.



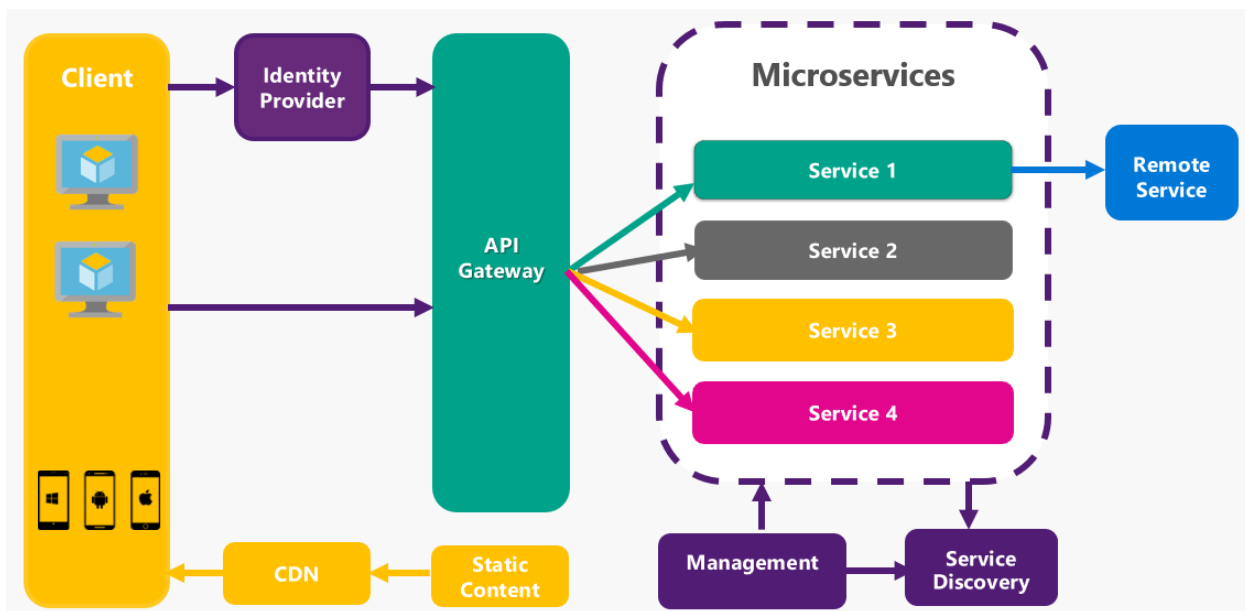
Slika 2 – Konceptualni model sistema

3. Projektovanje softverskog sistema

Arhitektura ovog softverskog sistema predstavlja arhitekturu zasnovu na mikroservisima. Projektovanje arhitekture obuhvata projektovanje svakog mikroservisa zasebno. Nivo korisničkog interfejsa smešten je na strani klijenta, dok svaki mikroservis ima svoju aplikacionu logiku i koristi određenu bazu za skladištenje podataka na strani servera.

4.1 Arhitektura softverskog sistema

Korisnički interfejs je smešten je u prezentacionom sloju jednog od mikroservisa u okviru MVC paterna. Svaki mikroservis poseduje svoju aplikacionu logiku gde je smeštena poslovna logika sistema kao sloj pristupa podacima ostvaren preko Entity Framework Core ORM alata. Svaki mikroservis komunicira sa svojom bazom podataka koja predstavlja skladište podataka.



Slika 3 – Prikaz mikroservisne arhitekture https://miro.medium.com/max/2266/1*m4Q4ojNKiqKREv4DKiqJmg.png

2. Korisnički interfejs

Kako bi koristio funkcionalnosti sistema potrebno je da se korisnik prijavi na sistem. Korisniku se prilikom pokretanja sistema otvara forma za prijavljivanje. Ukoliko nema profil, potrebno je da klikne na dugme za registraciju, čime će korisnik biti preusmeren na stranicu na kojoj se nalazi forma za registraciju.

Register

Email

Username

Password

Confirm password

Register Cancel

Nakon toga, korisnik će biti preusmeren na stranicu na kojoj se nalazi forma za prijavu na sistem. Za procesiranje ovih akcija zadužen je *Identity* mikroservis.

Local Account

Username

Username

Password

Password

☐ Remember My Login

Login


Cancel


Don't have an account yet?

Register

Nakon toga korisniku se prikazuje stranica na kojoj može pretraživati proizvode. Izgled korisničkog interfejsa i kontroler korisničkog interfejsa implementirani su u okviru *WineShopMVC* mikroservisa.

My orders Logout (Alice)



 0 PRODUCTS

All

All


Sparkling wines

White wines


Red wines

Rose wines

FILTER

| | COUNTRY OF ORIGIN | NAME | PRODUCER | PRICE | |
|---|-------------------|------------------------|------------------------------|--------|--------------------|
|  | Serbia | Rose Radovanović 0.75l | Vinarija Radovanović Krnjevo | \$8.50 | <div>DETAILS</div> |

Klikom na *Details* korisniku se otvara stanica na kojoj može videti više detalja o izabranom proizvodu. Svi podaci o proizvodima dobijaju se putem komunikacije sa *WineCatalog* mikroservisom. Tada korisnik može dodati proizvod u svoju korpu. Za akcije koje se vrše nad korpom korisnika zadužen je *ShoppingBasket* mikroservis.



WHITE WINES

Toplički Vinogradi Epigenia Chardonnay 0.75l

Vinarija Toplički Vinogradi

Serbia

White wine made from the Chardonnay variety of the Toplicki vinogradi winery. Made from exceptional fruit, this wine has a yellow color with green tones. Nice and pleasant smell.

\$15.80 PER BOTTLE

QUANTITY:

1

2

3

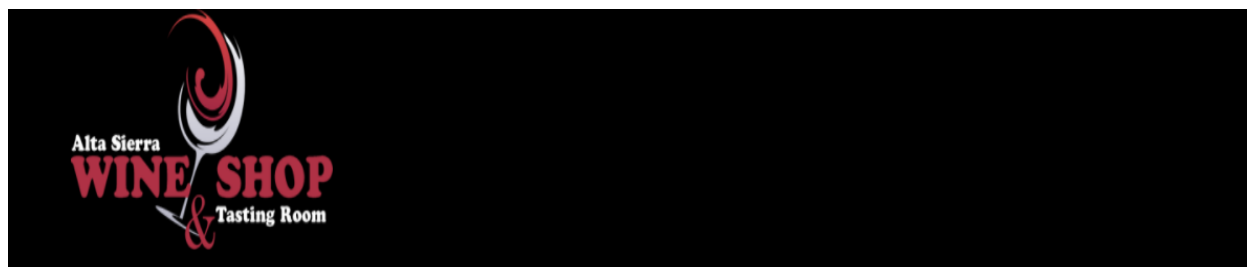
4

5

1

ADD TO BASKET

Korisnik u svakom trenutku ima uvid u sadržaj svoje korpe. Takođe, ukoliko je korisniku omogućen kupon, korisnik ostvaruje popust na porudžbinu. U ovom odeljku korisnik može ažurirati količinu odabranog proizvoda ili može ukloniti proizvod iz korpe. Za akcije koje se vrše nad kuponom korisnika zadužen je *Discounts* mikroservis.



Alice, this is your shopping cart!

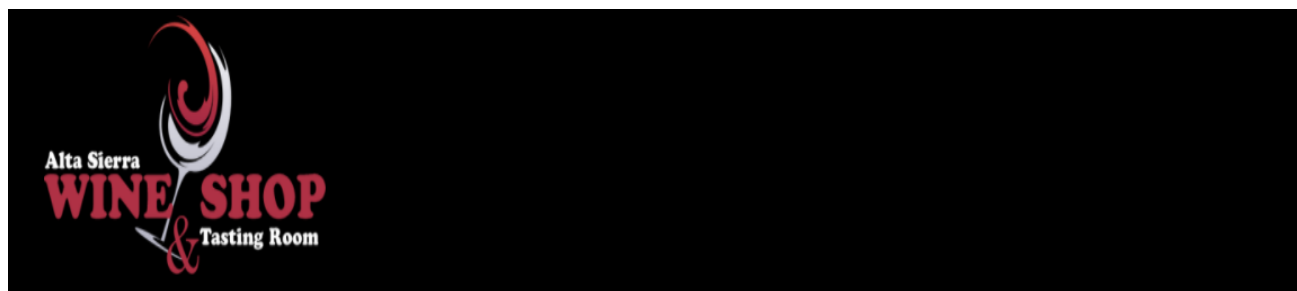
| PRODUCT NAME | PRICE PER BOTTLE | QUANTITY | TOTAL |
|--|------------------|------------------------------------|--|
| Toplički Vinogradi Epigenia Chardonnay 0.75l | \$15.80 | 3 <input type="button" value="v"/> | <input type="button" value="Update"/> \$47.40 <input type="button" value="x"/> |
| | | | Discount: \$10.00 |
| | | | Total: \$37.40 |

You have a coupon!

| | |
|----------|---------|
| DISCOUNT | \$10.00 |
|----------|---------|

[Back to catalog](#)

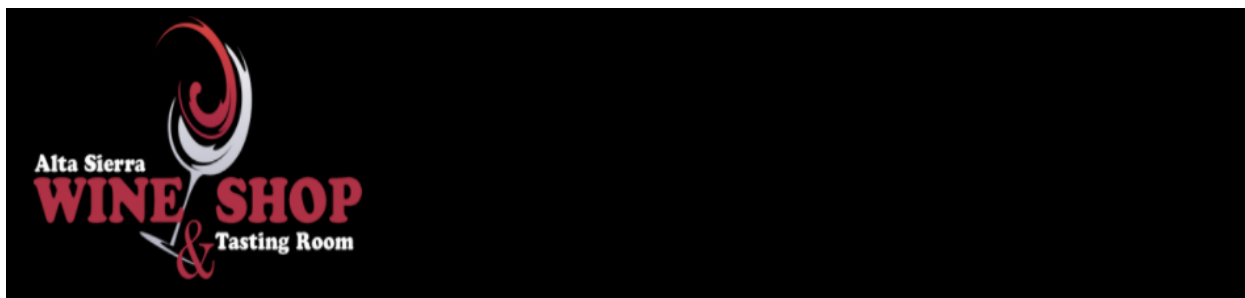
Klikom na dugme *Check out now!* korisniku se otvara forma gde je potrebno da korisnik unese podatke o plaćanju i dostavi.



You're just one step away from placing your order.

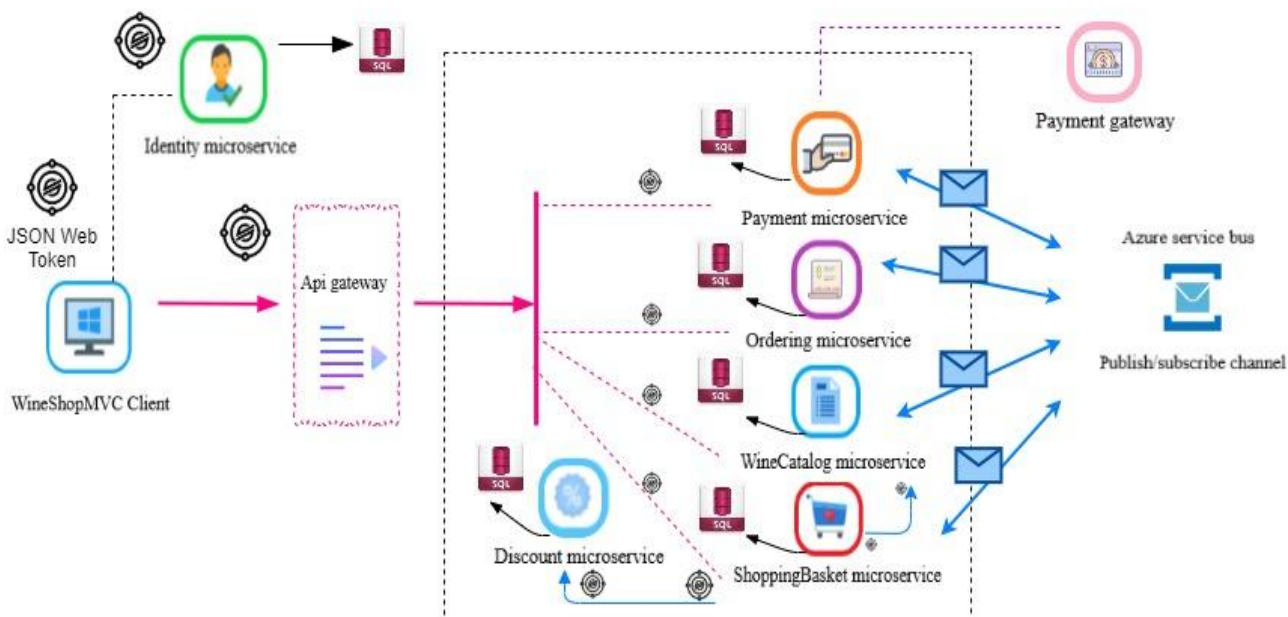
| | | | |
|--|---|----------------------------------|--|
| Last name <input type="text"/> | First name <input type="text"/> | Address <input type="text"/> | Zip code <input type="text"/> |
| City <input type="text"/> | Country <input type="text"/> | Email <input type="text"/> | Credit card name <input type="text"/> |
| Credit card number <input type="text"/> | Expiration date <input type="text"/> | CVV code <input type="text"/> | |
| <input type="button" value="Place order"/> | | | |

Nakon uspešno procesuirane porudžbine korisniku se prikazuje poruka:



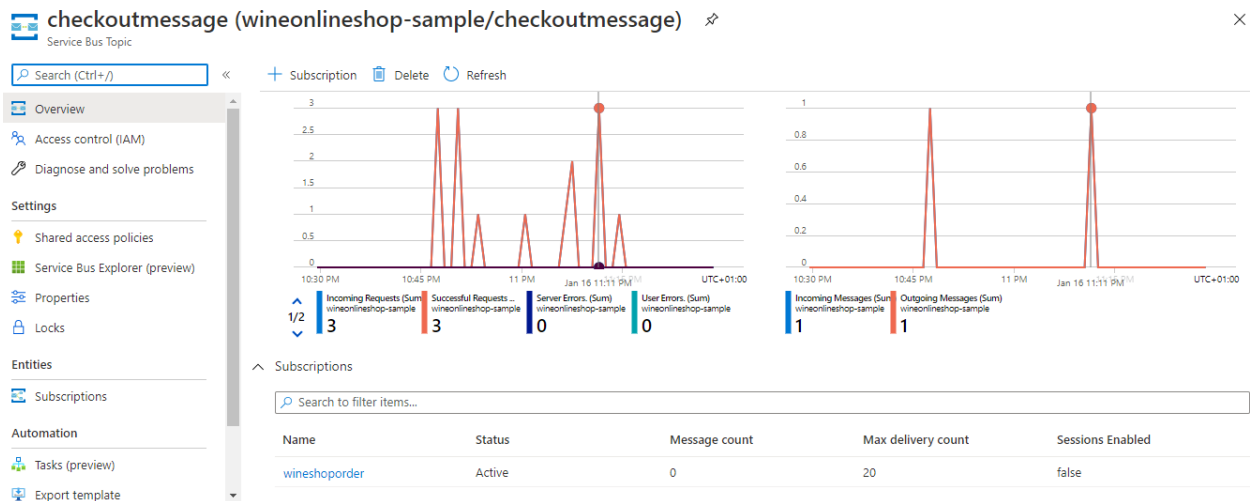
Your order is now submitted. Check the [orders page](#) for the status of your order

Konačan izgled arhitekture



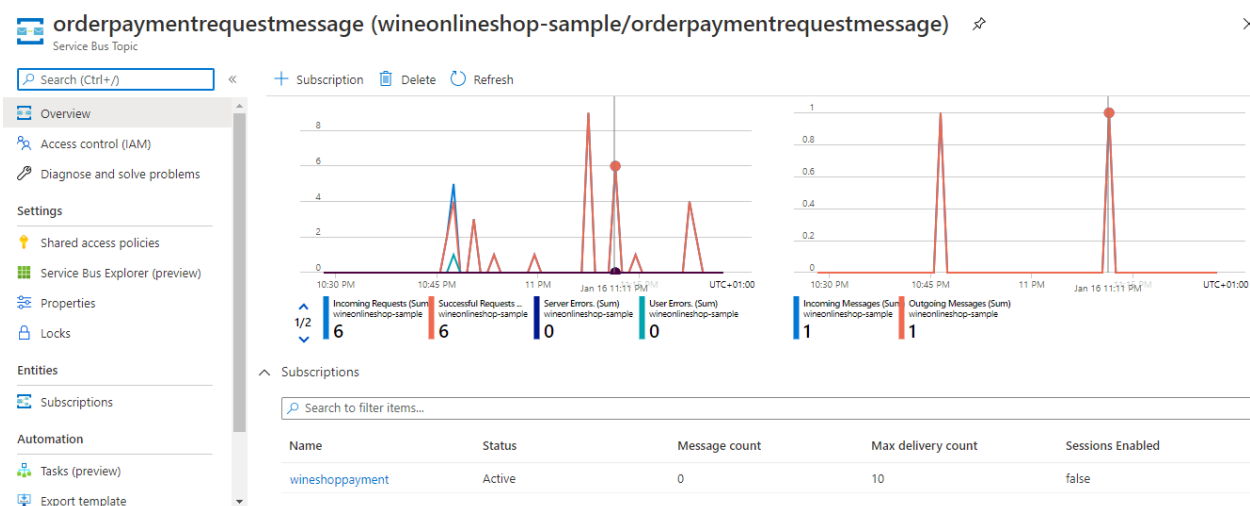
4. Implementacija

Komunikacija između mikroservisa odvija se sinhrono ili asinhrono, u zavisnosti od potrebe. Kada korisnik klikne na dugme *Place order* čime potvrđuje kupovinu, mikroservis *ShoppingBasket* šalje poruku na udaljeni *Azure Service Bus*. U ovom scenariju *ShoppingBasket* mikroservis je *publisher*. *Ordering* mikroservis ima ulogu *consumer*-a, odnosno osluškuje konekciju i kada je poruka poslata na *Bus*, ovaj servis prima poruku, odnosno objekat *ShoppingBasket* u *JSON* formatu. Na Azure portalu može se videti komunikacija između ova dva mikroservisa.



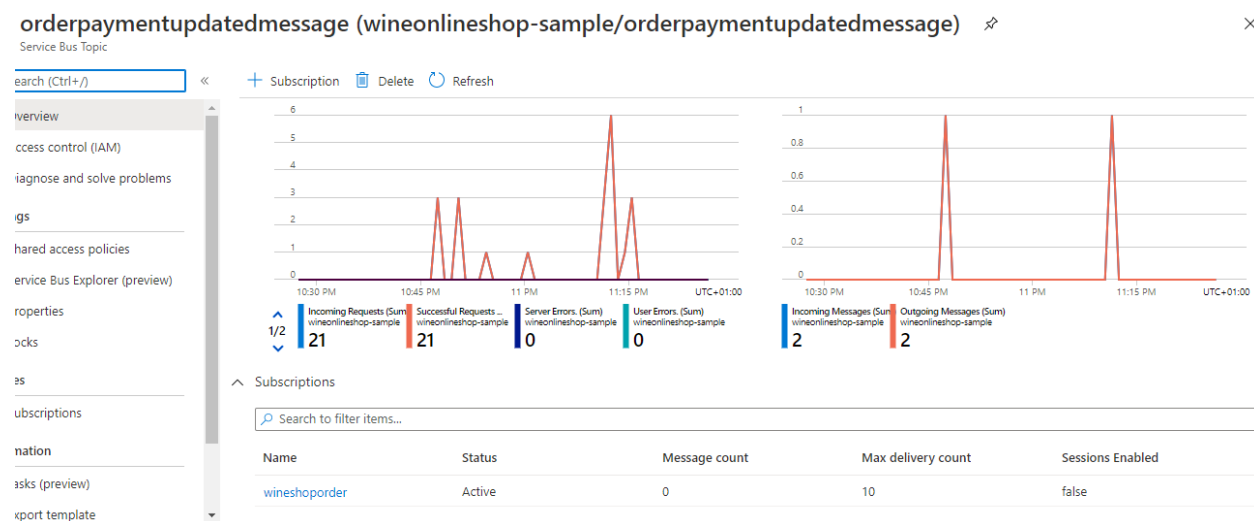
U odeljku *subscriptions* vidi se da je polje *Message count* postavljeno na 0. To znači da trenutno nema poruka u redu, odnosno da je primljena jedna poruka od mikroservisa i da je ista konzumirana od strane drugog mikroservisa. Ova razmena poruka desila u 11:11 što je prikazano na dijagramu iznad (onda kada je korisnik kliknuo na dugme kojim potvrđuje kupovinu).

Nakon što je *Order* mikroservis primio poruku o porudžbini i uspešno sačuvao istu u svojoj bazi podataka, ovaj mikroservis komunicira putem *Azure Service Bus* - a sa *Payment* mikroservisom kako bi se naplatila porudžbina za proizvode. *Order* mikroservis šalje poruku, odnosno objekat u *JSON* formatu na udaljeni *Azure Service Bus*. Ova poruka sadrži informacije o identifikatoru porudžbine, ukupnom iznosu, podatke o kreditnoj kartici i njenom vlasniku. U ulazi *consumer* - a sada se javlja *Payment* mikroservis koji osluškuje komunikaciju i konzumira poruku koja je poslata od strane *Order* mikroservisa.



Nakon toga, potrebno je da mikroservis *Payment* obavi svoj zadatak, odnosno da komunicira sa odgovarajućim *Payment Gateway* servisom (Za potrebe ove aplikacije kreiran je *FakePaymentGateway*

koji potvrđuje da je transakcija izvršena). Nakon što od ovog servisa stigne potvrda je transakcija izvršena i porudžbina plaćena, Payment mikroservis šalje poruku na *Bus* kojom obaveštava odgovarajuće *subscriber*-e da je porudžbina plaćena. U ovom scenariju, *Payment* mikroservis je *publisher*, a *Order* mikroservis je *consumer*. Order mikroservis nakon primljene poruke ažurira polje *isPaid* na *true* za odgovarajući *OrderId*.



Za ovakav tip komunikacija korišćen je *Azure Service Bus*. Mikroservisi implementirani su kao *.NET Core API* projekti, dok je klijent implementiran kao *ASP .NET CORE MVC* projekat. Svaki mikroservis koristi svoju bazu podataka. Baza podataka implementirana je pomoću *SQLServer*-a. Struktura projekta prikazana je na slici.

